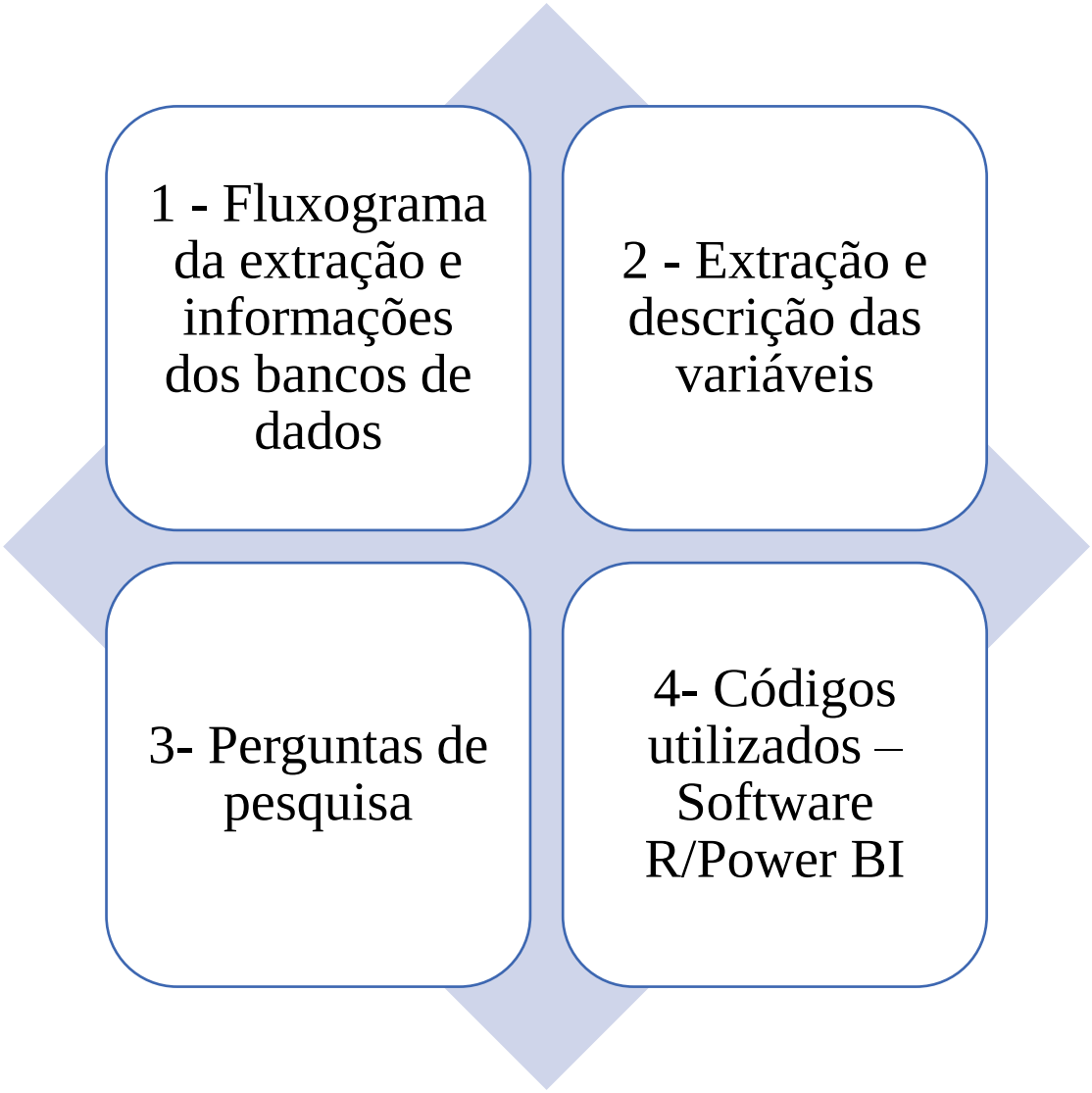
Teste – Analista de Dados

Candidata: Débora Borges dos Santos Pereira

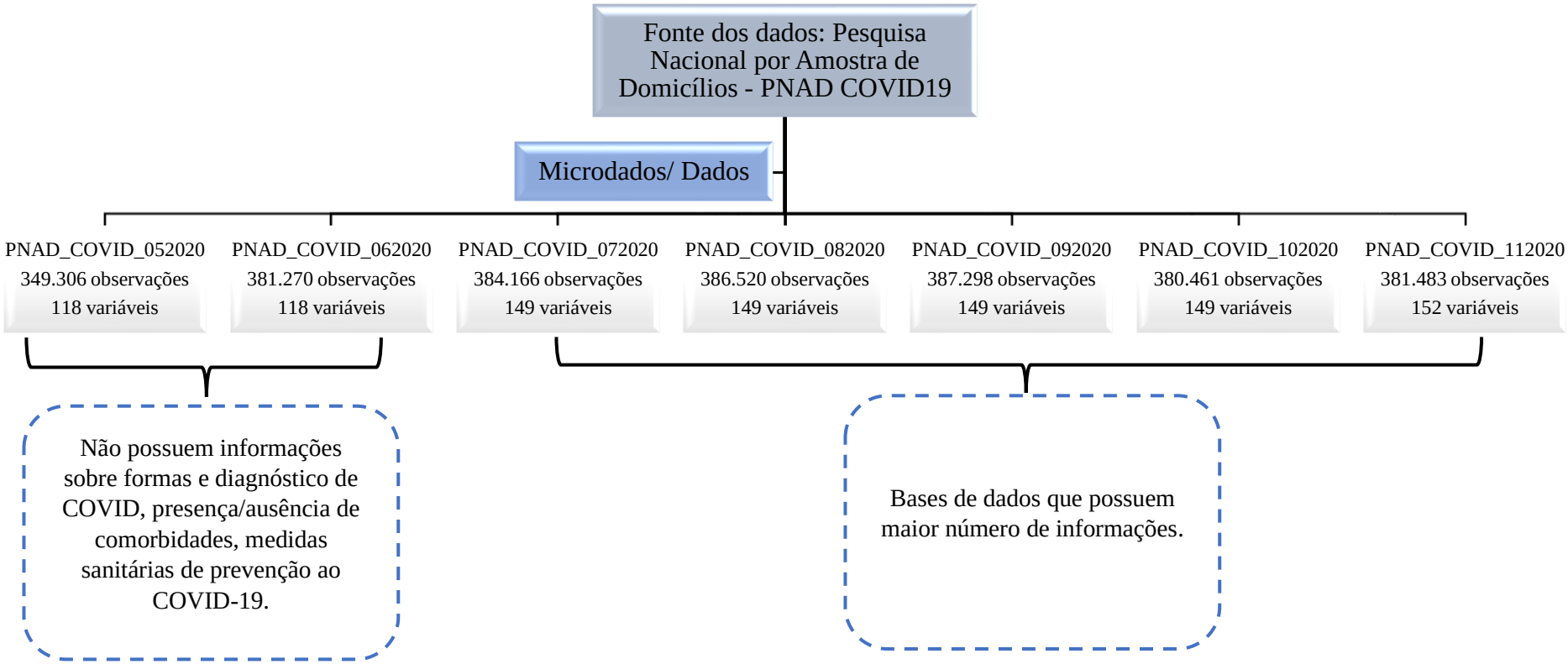
Data: 23/01/2023

Documentação

Descrição das etapas do processo



1 Fluxograma da extração e informações dos bancos de dados



2 Extração e descrição das variáveis

As variáveis foram extraídas diretamente do dicionário da PNAD – COVID-19 contínua, sendo possível identificar suas codificações, descrição e a forma como estão classificadas no banco de dados.

A seguir, está a lista das variáveis de interesse, podendo ou não serem usadas em sua totalidade. Suas descrições estão dispostas no código R.

Quadro com os códigos da variáveis:

UF","CAPITAL","V1022","V1013", "UPA", "Estrato", "V1032", "A001", "A001A", "A002", "A003", "A004", "A005", "B0011", "B0012", "B0013", "B0014", "B0015", "B0016", "B0017", "B0018", "B0019", "B00110", "B00111", "B00112", "B00113", "B002", "B0031", "B0032","B0033","B0034","B0035", "B0036", "B0037","B0041", "B0042", "B0043", "B0044", "B0045", "B0046", "B005", "B006","B007", "B008", "B009A","B009B", "B009C", "B009D", "B009E", "B009F", "B0101", "B0102", "B0103", "B0104","B0105", "B0106", "B011", "C001", "C002", "C003","C008", "C007", "C007A", "C007B", "C010", "C007C", "C007D","C008", "C01012", "C01011", "C012", "C013", "D0051", "F001", "F002A1", "F002A2","F002A3", "F002A4", "F002A5".

3 Perguntas de pesquisa

1. Qual o perfil sociodemográfico desta população? Isto inclui: sexo, idade, etnia/raça, escolaridade, situação de moradia, renda, trabalho e localidade onde residem.
2. Qual o comportamento do perfil sociodemográfico por sexo?
3. Qual o comportamento do perfil sociodemográfico por etnia/raça?
4. Qual o comportamento do perfil sociodemográfico por escolaridade?
5. Qual o comportamento do perfil sociodemográfico por carteira assinada e tipo de emprego?
6. Qual o perfil das pessoas que foram amparadas pelo auxílio emergencial?
7. Qual o comportamento do perfil sociodemográfico por renda e localidade?
8. Existem diferenças entre grupos sociais?
9. Considerando o contexto em que a pesquisa foi realizada, qual o perfil de saúde desta população? Isto inclui: presença de Doenças Crônicas Não Transmissíveis relevantes não só para a carga global de saúde, mas no atual cenário pandêmico?
10. Qual a prevalência de usuários de plano de saúde x usuários do SUS?
11. Quais os principais sintomas relatados que possuem relação com a COVID-19?
12. Qual principal estabelecimento de saúde procurado quando os sintomas de COVID-19 surgiram?
13. Houve complicações por COVID-19? Isto inclui: perfil do participante, internação, cuidados intensivos (sedação, intubação, ventilação mecânica)
14. Quais medidas não farmacológicas foram adotadas para prevenção ao COVID? Há diferenças por sexo? Há diferenças por UF? Há diferenças por período? Há diferenças por idade?
15. Qual a prevalência dos diagnósticos de COVID entre os participantes da pesquisa?
16. Após o resultado do exame, houve adoção de medidas de isolamento?

4 Códigos utilizados e comentados – Software R

O R foi utilizado em todo o processo de manipulação dos dados, extração, limpeza e organização. Os gráficos foram gerados por meio do Power Bi.

Link do repositório no Github: https://github.com/dbspereira/einstein_pnad_covid

```
#------------------------------------------------------------------------------#
# TESTE ANALISTA DE DADOS - EINSTEIN
# BASE DE DADOS UTILIZADA: PNAD COVID
# CANDIDATA: DÉBORA BORGES DOS SANTOS PEREIRA
#------------------------------------------------------------------------------#


#------------------------------------------------------------------------------#
# Bibliotecas  utilizadas
#------------------------------------------------------------------------------#
library(COVIDIBGE) #pacote disponível para manipulação dos dados pnadcovid
library(tidyverse) #manipulação dos dados
library(survey) #ponderação
library(srvyr) #ponderação
library(stats) #contém funções para cálculos estatísticos
library(tidyr) #utilizado na manipulação de dados
library(psych) #utilizado na manipulação de dados
library(dplyr) #utilizado na manipulação de dados
library(readxl) #leitura de bancos em excel
library(haven) #faz parte do tidyverse também para leitura de bancos
library(openxlsx) #abrir arquivos em xlsxs
library(readxl) #ler arquivos em xlsx
library(writexl) #exportar arquivos em xlsx
library(ggplot2) #para gráficos
library(plotly) #para gerar gráfico dinâmicos
library(readr) #para ler bases
library(naniar) #percentual de missing
library(rstatix) #para tratar outliers quando for viável
```

```
library(nortest) #teste normalidade quando for necessário
library(gtsummary) #tabula regressões


#------------------------------------------------------------------------------#
# Diretório utilizado para armazenar arquivos, se necessário.

setwd("C:/Users/Débora/Desktop/Teste_Einstein")
# ------------------------------------------------------------------------------#
# Definindo limite de memória para compilação do programa
#------------------------------------------------------------------------------#

aviso <- getOption("warn") #set da memória
options(warn = -1)
memory.limit(size = 200000)
options(warn = aviso) #para não aparecer o aviso de "warn" sem necessidade
rm(aviso)
#aumento de memória para melhorar a performance

#------------------------------------------------------------------------------#
# Definindo opção de exibição de números sem exponencial
aviso <- getOption("warn")
options(warn = -1)
options(scipien=100) #retirar o formato exponencial de algumas variáveis (renda)
options(warn = aviso) #para não aparecer o aviso de "warn" sem necessidade
rm(aviso)


#------------------------------------------------------------------------------#
# Importação online das bases de dados
```

```
#-------------------------------------------------------------------------#
pnadcovid_5 <- get_covid(year=2020, month=5, design = F)
pnadcovid_6 <- get_covid(year=2020, month=6, design = F)

#Obs: bases 5 e 6 não contemplam informações sobre testes de covid,
#presença de comorbidades, estratégias de prevenção à covid e ente outras.
#contudo, será utilizada para as analises descritivas que contemplam as informações #gerais

pnadcovid_7 <- get_covid(year=2020, month=7, design = F)
pnadcovid_8 <- get_covid(year=2020, month=8, design = F)
pnadcovid_9 <- get_covid(year=2020, month=9, design = F)
pnadcovid_10 <- get_covid(year=2020, month=10, design = F)
pnadcovid_11 <- get_covid(year=2020, month=11, design = F)

# Observação: existe também a forma de importação da base de dados offline, que
#pode ser realizada através do próprio diretório, utilizando o caminho do local,
#mas especificamente para esta análise, optei por extrair os bancos de forma online
#pelo pacote COVIDIBGE, através da função "get_covid". O argumento "design" desta função
#pode ser utilizado para ponderação das respostas e tornar os resultados nacionalmente
#representativos da população. Por enquanto, optei por não utilizá-lo a priori.
#-------------------------------------------------------------------------#


#-------------------------------------------------------------------------#
# Unindo os bancos:
# -------------------------------------------------------------------------#

merged_dataset <- bind_rows(pnadcovid_5,
                            pnadcovid_6,
```

```
                pnadcovid_7,
                pnadcovid_8,
                pnadcovid_9,
                pnadcovid_10,
                pnadcovid_11
                )
#Observação: também é possível unir o banco através das funções merge, rbind,
#por exemplo.
#------------------------------------------------------------------------------#

#------------------------------------------------------------------------------#
# Seleção das variáveis de interesse, códigos fornecidos pelo dicionário
# da pesquisa.
# Criação de nova base de dados, para preservar o banco original.
#------------------------------------------------------------------------------#

#Criação de uma lista com as variáveis de interesse:

variaveis_interesse <- c("UF","CAPITAL","V1022","V1013", "UPA", "Estrato", "V1032",
                         "A001", "A001A", "A002", "A003", "A004", "A005",
                         "B0011", "B0012", "B0013", "B0014", "B0015", "B0016",
                         "B0017", "B0018",
                         "B0019", "B00110", "B00111", "B00112", "B00113",
                         "B002", "B0031", "B0032","B0033","B0034","B0035",
                         "B0036", "B0037","B0041", "B0042", "B0043", "B0044",
                         "B0045", "B0046", "B005", "B006","B007", "B008",
                         "B009A","B009B", "B009C", "B009D", "B009E", "B009F",
                         "B0101", "B0102", "B0103", "B0104","B0105", "B0106",
                         "B011",
                         "C001", "C002", "C003","C008", "C007", "C007A", "C007B",
```

```
                "C010", "C007C", "C007D","C008", "C01012", "C01011",
                "C012", "C013", "D0051",
                "F001", "F002A1", "F002A2","F002A3", "F002A4", "F002A5"
                )

# Extraindo as variáveis de dentro do banco
pnad_covid <- subset(merged_dataset, select= c(variaveis_interesse))

# Caso esteja ocupando muito a memória do PC, remova os bancos que não serão utilizados
# através a função rm()

#-----------------------------------------------------------------------------#
# Renomeando as colunas do banco

pnad_covid <- pnad_covid %>%
  rename("Idade" = "A002",
         "tipo_moradia" = "V1022",
         "mes_pesquisa" = "V1013",
         "Raca_Etnia" = "A004",
         "Sexo" = "A003",
         "Escolaridade" = "A005",
         "Tipo_emprego" = "C007",
         "Area_emprego" = "C007A",
         "carteira_assinada" = "C007B",
         "funcao_trabalho" = "C007C",
         "faixa_rendimento" = "C01011",
         "valor_reais" = "C01012",
         "horas_trabalho" = "C008",
         "domicilio_situacao" = "F001",
         "home_office" = "C013",
```

"auxilio_emergencial" = "D0051",

"sintoma_febre" = "B0011",

"sintoma_tosse" = "B0012",

"dor_de_garganta" = "B0013",

"dificuldade_respirar" = "B0014",

"dor_de_cabeça" = "B0015",

"dor_no_peito" = "B0016",

"nausea" = "B0017",

"congestionamento_nasal" = "B0018",

"fadiga" = "B0019",

"perda_olfato_paladar" = "B00111",

"diarreia" = "B00113",

"procurou_estab_saude" = "B002",

"recuperou_em_casa" = "B0031",

"automedicacao" = "B0034",

"visita_sus" = "B0035",

"buscou_atendimento_sus" = "B0041",

"ps_sus_upa" = "B0042",

"hosp_sus" = "B0043",

"amb_privado_fa" = "B0044",

"ps_privado_fa" = "B0045",

"hosp_privado_fa" = "B0046",

"internacao" = "B005",

"sedado_intubacao_resp" = "B006",

"plano_saude" = "B007",

"teste_covid" = "B008",

"swab_teste" = "B009A",

"result_swab" = "B009B",

"teste_rapido" = "B009C",

"result_teste_rapido" = "B009D",

```
        "teste_sorologico" = "B009E",
        "result_sorologico" = "B009F",
        "diabetes" = "B0101",
        "hipertensao" = "B0102",
        "doencas_respiratorias" = "B0103",
        "doencas_cardiovasculares"= "B0104",
        "depressao" = "B0105",
        "cancer" = "B0106",
        "medida_restricao" = "B011",
        "trabalhou_bico" = "C001",
        "afastou_temporario" = "C002",
        "motivo_afastamento" = "C003",
        "sabao_detergente" = "F002A1",
        "alcool_70" = "F002A2",
        "mascaras" = "F002A3",
        "luvas" = "F002A4",
        "agua_sanitaria" = "F002A5")

#conferindo se a nomeacao ocorreu bem
colnames(pnad_covid)
#-----------------------------------------------------------------------------#
# Conferindo os valores ausentes do banco de dados.

# NA = valores ausentes
# NAN = not a number(valor indefinido)

sapply(pnad_covid, function(x) sum(is.na(x)))
sapply(pnad_covid, function(x) sum(is.nan(x)))

gg_miss_var(pnad_covid, show_pct = TRUE) #mostrando % de missings por variável
```

```
n_var_miss(pnad_covid)

# Uma quantidade grande de valores ausentes, nas análises, seria necessário
# omiti-los. Uma alternativa é omiti-los através deste comando, ou remove-los
# na execução dos comandos

na.omit(pnad_covid)



#------------------------------------------------------------------------------#
# Categorizando variáveis necessárias:
#------------------------------------------------------------------------------#

#Idade categórica

pnad_covid$idadecat <- cut(pnad_covid$Idade,
                           breaks = c(0, 15, 25, 35, 45, 55, 65, Inf),
                           labels = c("0-14", "15-24", "25-34", "35-44",
                                      "45-54", "55-64", "65+"))



#definindo que será um fator
pnad_covid$idadecat <- as.factor(pnad_covid$idadecat)

#definindo como categoria de referência, se caso utilizar algum glm.

pnad_covid$idadecat <- relevel(pnad_covid$idadecat, ref = "35-44")

#Rendimento salarial
```

```
pnad_covid$rendimento_cat <- factor(pnad_covid$faixa_rendimento,
                                    levels = c("0", "1", "2", "3", "4",
                                               "5", "6", "7", "8", "9"),
                                    labels = c("0-100",
                                               "101-300",
                                               "301-600",
                                               "601-800",
                                               "801-1600",
                                               "1601-3000",
                                               "3001-10000",
                                               "10001-50000",
                                               "50001-100000",
                                               "100000+"))
#definindo como categoria de referência, se caso utilizar algum glm.
pnad_covid$rendimento_cat <- relevel(pnad_covid$rendimento_cat, ref = "801-1600")

#Categorizando o mes da pesquisa

pnad_covid$mes_cat <- factor(pnad_covid$mes_pesquisa,
                             levels = c("5", "6", "7", "8", "9",
                                        "10", "11"),
                             labels = c("Maio-2020",
                                        "Junho-2020",
                                        "Julho-2020",
                                        "Agosto-2020",
                                        "Setembro-2020",
                                        "Outubro-2020",
                                        "Novembro-2020"))
```

```
#-------------------------------------------------------------------------------#
# Criação do objeto amostral
# importante para balancear os dados, entre os não respondentes e respondentes,
# permitindo a exploração das informações para serem nacionalmente representativas.


base_ponderada <- pnad_covid %>% as_survey_design(ids = UPA,
                                      strata = Estrato,
                                      weights = V1032, nest = TRUE)

#UPA - unidade primária de amostragem
#Strata = estrato da amostragem
# weights =  variável de peso pós-estratificação
# nest = aninhamento seja dentro do estrato de amostragem

#-------------------------------------------------------------------------------#
# Observando a distribuição de algumas variáveis contínuas


# Idade
svyhist(formula=~as.numeric(Idade), design=base_ponderada, main="Histograma",
      xlab="Distribuição da idade (em anos) da população"
      )


#Curva com simetria à esquerda


# QQPLOT (GRÁFICO DE DISTRIBUIÇÃO NORMAL)
```

```
svyqqmath(~Idade, design=base_ponderada, null=qnorm,
        na.rm=TRUE,
        xlab="Expected",ylab="Observed",
        abline(0,1))



# Horas por semana trabalhada

svyhist(formula=~as.numeric(horas_trabalho), design=base_ponderada,
main="Histograma",
      xlab="Horas de trabalho por semana")

# QQPLOT (GRÁFICO DE DISTRIBUIÇÃO NORMAL)

svyqqmath(~horas_trabalho, design=base_ponderada, null=qnorm,
        na.rm=TRUE,
        xlab="Expected",ylab="Observed",
        abline(0,1))



#Curva com boa simetria, tendendo levemente à esquerda



# Remuneração mensal do salário

svyhist(formula=~as.numeric(valor_reais),
design=base_ponderada, main="Histograma",
      xlab="Salário mensal"
      )
```

```
# QQPLOT (GRÁFICO DE DISTRIBUIÇÃO NORMAL)


svyqqmath(~valor_reais, design=base_ponderada, null=qnorm,
        na.rm=TRUE,
        xlab="Expected",ylab="Observed",
        abline(0,1))



################################################################################
#########
# Teste de normalidade - Anderson-Darling
ad.test(pnad_covid$valor_reais)
ad.test(pnad_covid$Idade)
ad.test(pnad_covid$horas_trabalho)


#Para amostras grandes como n> 5000, os testes de normalidade perdem força.
#resultados p<0.005 (distribuição não normal)
#Hoje recomenda-se que para avaliar normalidade de amostras na saúde se utilizem
#de formas visuais como box-plot e histograma, outros testes mais fortes
#como o Shapiro WIlk são limitados a amostras até 500 observações.
#estamos trabalahando com uma amostra ponderada e muito grande, não é necessário
#se procupar nesse momento.

#------------------------------------------------------------------------------#
# ANÁLISE DE OUTLIERS
#------------------------------------------------------------------------------#

svyboxplot(formula=A002~1, design=base_ponderada,
```

```
        main="Boxplot da idade",
       )


svyboxplot(formula=C008~1, design=base_ponderada,
        main="Horas por semana trabalhada",
        )


svyboxplot(formula=valor_reais~1, design=base_ponderada,
        main="Salário mensal",
        )


# BOXPLOT COM PLOTLY
#porém, sem a ponderação

plot_ly(pnad_covid, y = pnad_covid$Idade, type = "box")


# Obs: em casos em que os outliers são muito preocupantes que enviesam a informação
# é importante eliminá-los, fora do pacote survey, existe outra função para isso.

# ANÁLISE DE OUTLIERS
boxplot(pnad_covid$Idade)

# Tratando os outliers

# Identificando os outliers
```

```
pnad_covid %>% identify_outliers(Idade)

# Excluindo os outliers
outliers <- c(boxplot.stats(pnad_covid$Idade)$out)
pnad_covid <- mola2[-c(which(pnad_covid$Idade %in% outliers)),  ]

#conferindo se foi deletado
boxplot(pnad_covid$Idadeo)




#-------------------------------------------------------------------------------#
# Verificando as classes das variáveis do banco
#-------------------------------------------------------------------------------#

str(pnad_covid)



#-------------------------------------------------------------------------------#
# Estimando totais e prevalências
#-------------------------------------------------------------------------------#

#A função do pacote para a estimação de totais populacionais é a svytotal.
# Sua sintaxe precisa de três parâmetros principais:
#O nome da variável que se deseja calcular o total, precedido por um ~;
#O nome do objeto do plano amostral (base_ponderada);
#A opção na.rm=TRUE, que remove as observações onde a variável é não-aplicável.

#Variáveis numéricas
```

```
#Idade

total_idade<- svytotal(x=~Idade, design=base_ponderada, na.rm=TRUE)
print(total_idade)

#coeficiente de variação
cv(object=total_idade)

#intervalo de confiança
confint(object=total_idade, level=0.95)



#Renda
total_renda <- svytotal(x=~valor_reais, design=base_ponderada, na.rm=TRUE)
print(total_renda)

#coeficiente de variação
cv(object=total_renda)

#intervalo de confiança
confint(object=total_renda, level=0.95)

#Horas trabalhadas

total_horas <- svytotal(x=~horas_trabalho, design=base_ponderada, na.rm=TRUE)
print(total_horas)

#coeficiente de variação
cv(object=total_horas)
```

```
#intervalo de confiança
confint(object=total_horas, level=0.95)


#Estimando médias:

media_renda <- svymean(x=~valor_reais, design=base_ponderada, na.rm=TRUE)
print(media_renda)

cv(object=media_renda)
confint(object=media_renda, level=0.95)



media_idade <- svymean(x=~Idade, design=base_ponderada, na.rm=TRUE)
print(media_idade)

cv(object=media_idade)
confint(object=media_idade, level=0.95)

media_horas <- svymean(x=~horas_trabalho, design=base_ponderada, na.rm=TRUE)
print(media_horas)
cv(object=media_horas)
confint(object=media_horas, level=0.95)


#Média de renda por UF

mediaRendaUF <- svyby(formula=~valor_reais, by=~UF, design=base_ponderada,
FUN=svymean, na.rm=TRUE)
```

```
mediaRendaUF


#Média de renda por sexo

mediaRendaSX <- svyby(formula=~valor_reais, by=~Sexo, design=base_ponderada, FUN=svymean, na.rm=TRUE)

mediaRendaSX


#Média de renda por raça/etnia

mediaRendaet <- svyby(formula=~valor_reais, by=~Raca_Etnia, design=base_ponderada, FUN=svymean, na.rm=TRUE)

mediaRendaet


#Média de renda por tipo de emprego

mediaRendaemp <- svyby(formula=~valor_reais, by=~Tipo_emprego, design=base_ponderada, FUN=svymean, na.rm=TRUE)

mediaRendaemp


# Estimando a frequencia relativa de homens e mulheres em cada nivel de instrucao

freqSexoInstr <- svyby(formula=~Escolaridade, by=~Sexo,
                       design=base_ponderada, FUN=svymean,
                       na.rm=TRUE, vartype=NULL)
print(freqSexoInstr, row.names=FALSE)
```

```
# Esbocando boxplot do numero de horas trabalhadas por sexo
svyboxplot(formula=horas_trabalho ~ Sexo,
        design=base_ponderada,
        all.outliers=TRUE)
```

```
# Esbocando grafico de dispersao entre numero de horas trabalhadas e renda mensal habitual
svyplot(formula=valor_reais ~ horas_trabalho, design=base_ponderada, style="transparent", xlab="Horas habitualmente trabalhadas", ylab="Rendimento habitual")
```

```
#Proporções entre variáveis categóricas:

#Sociodemograficas

base_ponderada %>% group_by(Sexo) %>%
  summarise(n=survey_total(vartype="ci"),
          na.rm=TRUE)  %>%
          mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(Raca_Etnia) %>%
  summarise(n=survey_total(vartype="ci"),
          na.rm=TRUE)  %>%
  mutate(pct=(n/sum(n)*100))
```

```
base_ponderada %>% group_by(Escolaridade) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(idadecat) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(idadecat) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(trabalhou_bico) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(Tipo_emprego) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(Area_emprego) %>%
  summarise(n=survey_total(vartype="ci"),
```

```
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(carteira_assinada) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(funcao_trabalho) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(faixa_rendimento) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(home_office) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(auxilio_emergencial) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
```

```
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(tipo_moradia) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(domicilio_situacao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

#Saúde

base_ponderada %>% group_by(diabetes) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(hipertensao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(doencas_respiratorias) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))
```

```
base_ponderada %>% group_by(doencas_cardiovasculares) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(depressao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(cancer) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

#Covid / sintomas

base_ponderada %>% group_by(sintoma_febre) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(sintoma_tosse) %>%
  summarise(n=survey_total(vartype="ci"),
```

```
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(dor_de_garganta) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(dificuldade_respirar) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(dor_de_cabeça) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(dor_no_peito) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(nausea) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(congestionamento_nasal) %>%
```

```
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(fadiga) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(diarreia) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(plano_saude) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

#Medida tomada quando surgiu os sintomas

base_ponderada %>% group_by(procurou_estab_saude) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(recuperou_em_casa) %>%
```

```
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(automedicacao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(visita_sus) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(buscou_atendimento_sus) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(ps_sus_upa) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(hosp_sus) %>%
  summarise(n=survey_total(vartype="ci"),
```

```
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(amb_privado_fa) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(ps_privado_fa) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(hosp_privado_fa) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(internacao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(sedado_intubacao_resp) %>%
  summarise(n=survey_total(vartype="ci"),
```

```
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(teste_covid) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(swab_teste) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(result_swab) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))


base_ponderada %>% group_by(teste_rapido) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))
```

```
base_ponderada %>% group_by(result_teste_rapido) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(teste_sorologico) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(result_sorologico) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))




base_ponderada %>% group_by(medida_restricao) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



# Medidas de combate e prevenção ao covid (nao farmacológica)



base_ponderada %>% group_by(sabao_detergente) %>%
```

```
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(alcool_70) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



base_ponderada %>% group_by(mascaras) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(luvas) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))

base_ponderada %>% group_by(agua_sanitaria) %>%
  summarise(n=survey_total(vartype="ci"),
            na.rm=TRUE)  %>%
            mutate(pct=(n/sum(n)*100))



#-------------------------------------------------------------------------
# EXEMPLO DE TESTE DE HIPÓTESES
# -------------------------------------------------------------------------
```

PRESSUPOSTOS:

H0= as médias salariais são iguais

H1= as médias salariais são diferentes

#Obs, por default, os grupos devem ser de ordem binária

Pergunta 1: há diferenças de rendimentos mensais por sexo,home_office, tipo de
#moradia (urbana/rural)?

```
svyttest(formula=valor_reais~Sexo, design=base_ponderada)
```

#Resultado:

Rejeitamos a hipótese nula (não há diferenças entre os grupos) e aceitamos

a alterntiva, pois existem diferenças das médias salariais entre os grupos

Estatisticamente significativo

2.2e-16 é a notação científica de 0,00000000000000022,

o que significa que é muito próximo de zero.

resultado t teste = p <0.05

```
svyttest(formula=as.numeric(valor_reais)~home_office, design=base_ponderada)
```

#Resultado:

Rejeitamos a hipótese nula (não há diferenças entre os grupos) e aceitamos

a alterntiva, pois existem diferenças das médias salariais entre os grupos

Estatisticamente significativo

```
# 2.2e-16 é a notação científica de 0,00000000000000022,
# o que significa que é muito próximo de zero.
# resultado t teste = p <0.05



svyttest(formula=as.numeric(valor_reais)~tipo_moradia, design=base_ponderada)

#Resultado:
# Rejeitamos a hipótese nula (não há diferenças entre os grupos) e aceitamos
# a alterntiva, pois existem diferenças das médias salariais entre os grupos
# Estatisticamente significativo
# 2.2e-16 é a notação científica de 0,00000000000000022,
# o que significa que é muito próximo de zero.
# resultado t teste = p <0.05



#------------------------------------------------------------------------#
# EXPOTANDO OS BANCOS PARA SEREM UTILIZADOS NO POWER BI
# ------------------------------------------------------------------------#


# exportando o banco completo

write.csv(merged_dataset, file = "merged_dataset.csv", row.names = FALSE)

#exportando todo o banco com as variáveis de interesse de maio a novembro

write.csv(pnad_covid, file = "pnad_covidfull.csv", row.names = FALSE)
```

#filtrando um novo banco somente a partir do mes de julho

```
pnad_covid$mes_pesquisa <- as.numeric(pnad_covid$mes_pesquisa)
pnad_covid_jul_nov <- filter(pnad_covid, mes_pesquisa >= 7)
table(pnad_covid_jul_nov$mes_pesquisa)

write.csv(pnad_covid_jul_nov, file = "pnad_covid_jul_nov.csv", row.names = FALSE)
```

#----------#

Códigos Power Bi utilizados e comentados

Criação de variável única de DCNT:

#Criando uma estrutura com múltiplas condições a partir das demais variáveis

```
DCNT =
IF(
   pnad_covidfull[doencas_cardiovasculares] == "Sim" ||
   pnad_covidfull[diabetes] == "Sim" || pnad_covidfull[hipertensao] == "Sim" ||
pnad_covidfull[doencas_respiratorias] == "Sim", "Sim", "Nâo"
 )
```

#Criação de variável única para teste de COVID:

```
Diagnóstico_Covid =
IF(
   pnad_covidfull[result_sorologico] == "Positivo" ||
   pnad_covidfull[result_swab] == "Positivo" || pnad_covidfull[result_teste_rapido] == "Positivo", "Positivo",
"Negativo"
 )
```

#Criação de variável única para sintomas de COVID:

```
Sintomas_Covid = IF(
   pnad_covidfull[dificuldade_respirar] == "Sim" ||
   pnad_covidfull[diarreia] == "Sim" || pnad_covidfull[dor_de_cabeça] == "Sim" || pnad_covidfull[dor_de_garganta]
== "Sim"|| pnad_covidfull[perda_olfato_paladar] == "Sim"|| pnad_covidfull[fadiga]== "Sim" ||
```

```
pnad_covidfull[dor_no_peito] == "Sim" || pnad_covidfull[nausea] == "Sim" || pnad_covidfull[sintoma_febre] ==
"Sim" || pnad_covidfull[sintoma_tosse] == "Sim", "Sim","Não"
  )
```

FIM

#----------#