

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

POR DENTRO DA PNAD CONTÍNUA

Uma introdução
ao tratamento de
dados usando o R



POR DENTRO DA PNAD CONTÍNUA

**Uma introdução
ao tratamento de
dados usando o R**

**Reitor**

José Daniel Diniz Melo

Vice-Reitor

Henio Ferreira de Miranda

Diretoria Administrativa da EDUFRN

Maria da Penha Casado Alves (Diretora)

Helton Rubiano de Macedo (Diretor Adjunto)

Bruno Francisco Xavier (Secretário)

Conselho Editorial

Maria da Penha Casado Alves

(Presidente)

Judithe da Costa Leite (Secretária)

Adriana Rosa Carvalho

Alexandro Teixeira Gomes

Elaine Cristina Gavioli

Everton Rodrigues Barbosa

Fabrício Germano Alves

Francisco Wildson Confessor

Gilberto Corso

Gleydson Pinheiro Albano

Gustavo Zampier dos Santos Lima

Izabel Souza do Nascimento

Josenildo Soares Bezerra

Ligia Rejane Siqueira Garcia

Lucélio Dantas de Aquino

Marcelo de Sousa da Silva

Márcia Maria de Cruz Castro

Márcio Dias Pereira

Martin Pablo Cammarota

Nereida Soares Martins

Roberval Edson Pinheiro de Lima

Tatyana Mabel Nobre Barbosa

Tercia Maria Souza de Moura Marques

Editoração

Helton Rubiano de Macedo (Editor)

Kamyla Álvares Pinto (Editora)

Revisão

Wildson Confessor (Coordenador)

Renata Coutinho (Colaboradora)

Design editorial

Rafael Campos (Capa)

Cassiano José Bezerra Marques Trovão (Miolo)

Antonio Hermes Marques da Silva Júnior (Miolo)

Pintura da capa: Operários, 1933, Tarsila do Amaral

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

POR DENTRO DA PNAD CONTÍNUA

Uma introdução
ao tratamento de
dados usando o R





Fundada em 1962, a Editora da UFRN (EDUFRN) permanece até hoje dedicada à sua principal missão: produzir livros com o fim de divulgar o conhecimento técnico-científico produzido na Universidade, além de promover expressões culturais do Rio Grande do Norte. Com esse objetivo, a EDUFRN demonstra o desafio de aliar uma tradição de seis décadas ao espírito renovador que guia suas ações rumo ao futuro.

Publicação digital financiada com recursos do Fundo Editorial da UFRN. A seleção da obra foi realizada pelo Conselho Editorial da EDUFRN, com base em avaliação cega por pares, a partir dos critérios definidos no Edital nº06/2021, para a linha editorial Recursos didático-pedagógicos.

Coordenadoria de Processos Técnicos
Catalogação da Publicação na Fonte.
UFRN / Biblioteca Central Zila Mamede

Trovão, Cassiano José Bezerra Marques
Por dentro da PNAD contínua [recurso eletrônico] : uma introdução ao tratamento de dados usando R / Cassiano José Bezerra Marques Trovão, Antonio Hermes Marques da Silva Júnior. – Dados eletrônicos (1 arquivo : 78 KB). – Natal, RN : EDUFRN, 2022.

Modo de acesso: World Wide Web
<<http://repositorio.ufrn.br>>
Título fornecido pelo criador do recurso
ISBN 978-65-5569-269-3

1. Pesquisa Nacional por Amostra de Domicílios Contínua. 2. R (Linguagem de Programação de Computador) 3. Fatores socioeconômicos – Estatísticas. I. Silva Júnior, Antonio Hermes Marques da. II. Título.

CDD 300.72
RN/UF/BCZM 2021/19 CDU 303.064

Elaborado por Gersoneide de Souza Venceslau – CRB-15/311

Todos os direitos desta edição reservados à EDUFRN – Editora da UFRN
Av. Senador Salgado Filho, 3000 | Campus Universitário
Lagoa Nova | 59.078-970 | Natal/RN | Brasil
e-mail: contato@editora.ufrn.br | www.editora.ufrn.br
Telefone: 84 3342 222

Prefácio

Este livro sintetiza um esforço metodológico pouco presente na história das Ciências Sociais Aplicadas no Brasil: a apresentação da metodologia do principal levantamento socioeconômico anual realizado no país e alguns percursos metodológicos estatísticos para a exploração das informações por ele produzidas.

Nos últimos 25 anos, tanto a produção quanto a exploração estatística das informações socioeconômicas (micrdados) passaram por uma transformação substantiva. De um ambiente dependente de *mainframes* de grande porte para a produção e a exploração das informações, ocorreu a migração para um ambiente significativamente mais simples e acessível tanto para os pesquisadores quanto para os estudantes, o dos computadores tipo PC. Esses pequenos, mas potentes, equipamentos passaram a ser o principal meio para produção e exploração dos dados. Para tanto, foram sendo adaptados e difundidos os *softwares* de tratamento de bases de dados para esses equipamentos como o *Statistical Analysis System (SAS)*, o *Statistical Package for the Social Sciences (SPSS)* e o *Statistical Software for Data Science (STATA)*. Mais recentemente, vêm sendo desenvolvidos e utilizados softwares livres como o **REDATAM**, um acrônimo para *RECuperación de DATos para Áreas pequeñas por Microcomputador*, e o *R Project for Statistical Computing*. Sendo este último o *software* adotado no presente livro.

O resultado dessa transformação foi a emergência da ampla facilidade de acesso aos micrdados e a sua exploração em termos estatísticos, como, também, o importante desenvolvimento do escopo e da complexidade das análises socioeconômicas. Laboratórios com equipamentos individuais para suporte a disciplinas aplicadas de ensino passaram a estar disponíveis nas universidades, estimulando e dando maior qualidade à formação estatística dos estudantes de graduação e pós-graduação nos cursos de Ciências Sociais Aplicadas.

Tanto na Europa quanto nos Estados Unidos, a maior presença desse tipo de formação nas universidades estimulou a produção de livros-textos ou de sites especializados em análise metodológica de bases de dados, a difusão de rotinas¹ para a elaboração de indicadores estatísticos e a publicação de ensaios metodológicos e analíticos focados na produção e na análise de indicadores socioeconômicos. O exemplo dessa iniciativa é o *Integrated Public Use Microdata Series* (IPUMS)², sediado na Universidade de Minnesota nos Estados Unidos, que há 25 anos organiza e disponibiliza os microdados dos censos demográficos de diversos países, articula uma rede de pesquisadores com o objetivo de estimular o uso e o desenvolvimento metodológico sobre essas bases de dados e difunde a produção científica com esse foco. Uma iniciativa mais recente, focando no tema da desigualdade, é o *World Inequality Database* (WID)³. Diversas outras iniciativas também poderiam ser mencionadas, pois terminam por orientar diversos temas em políticas públicas.

Este livro pertence a esse esforço de pesquisadores das Ciências Sociais Aplicadas, cujos resultados têm sido de enorme importância para a definição, a gestão e a avaliação das diversas políticas públicas sociais no país. É fundamental ressaltar a importância desse esforço, por ele ser uma experiência ainda inédita no Brasil. Ainda não é prática comum no país a publicação de livros-textos com esse objetivo nas Ciências Sociais. Ao contrário, as editoras privadas e públicas ainda resistem em publicar livros dessa natureza. Ademais, o próprio sistema de avaliação acadêmica não valoriza essa forma de esforço científico. Explicita-se, portanto, a enorme importância deste livro em um país altamente carente de instrumentos e de recursos humanos capacitados para o desenvolvimento das políticas públicas sociais, esperando que ele estimule outros pesquisadores a realizarem esforços semelhantes.

O livro aborda, em especial, a possibilidade de análise estatística dos temas: mercado de trabalho, desigualdade e pobreza. Os primeiros capítulos apresentam as diversas versões metodológicas da Pesquisa Nacional por Amostra de Domicílios desde sua criação em 1967, as possibilidades metodológicas da versão atual do levantamento, isto é, a Pesquisa Anual por Amostra de Domicílios Contínua, para a elaboração e a análise de indicadores sobre mercado de trabalho, além dos limites

¹ Rotinas são sequências de comando de programação que, executadas, produzem indicadores estatísticos.

² Ver <https://www.ipums.org>.

³ Ver <https://wid.world/>.

metodológicos desses levantamentos.

Em seguida, dois capítulos são dedicados às características, ao potencial do *software R* e a como utilizá-lo para ler e importar os microdados da PNAD Contínua, disponíveis na página do IBGE, por meio de uma sequência de rotinas consolidadas no pacote **PNADcIBGE**.

Os quatro capítulos seguintes exploram como tratar os microdados da PNAD Contínua por pacotes estatísticos vinculados ao *software R*. Apresentam tanto a estrutura de variáveis da PNAD quanto os comandos do *R* e a lógica das rotinas para a elaboração dos indicadores. Isto é, permitem de modo didático que o usuário adquira conhecimento tanto da própria PNAD Contínua quanto do modo de explorá-la com o objetivo de elaborar indicadores estatísticos para uma análise da situação de desigualdade e pobreza no país. O resultado desse esforço é apresentado no último capítulo.

O conjunto dos capítulos propicia ao leitor conhecer a PNAD Contínua, adquirir conhecimento sobre o *R*, utilizando-o para o tratamento de dados e a elaboração de indicadores estatísticos desejados para uma análise socioeconômica focada no mercado de trabalho, nas desigualdades e na pobreza no Brasil.

Como anteriormente apontado, o livro sintetiza um esforço relevante dos autores para produzir um roteiro preciso para estudantes e pesquisadores que tenham interesse em explorar os dados da PNAD Contínua com vistas à análise dos temas socioeconômicos por ele enfocado. É uma experiência pouco difundida no meio acadêmico brasileiro, apesar da sua elevada relevância.

Parabéns aos autores por esta contribuição acadêmica e que ela estimule o desenvolvimento de outras semelhantes, tão necessárias e fundamentais para a informação, a gestão e a avaliação das políticas públicas sociais no país.

Claudio Salvadori Dedecca

Professor Titular Aposentado em Economia Social e do Trabalho

Universidade Estadual de Campinas – UNICAMP

Sumário

1	Introdução	15
2	A Pesquisa Nacional por Amostra de Domicílios (PNAD): um breve histórico	17
2.1	Um breve histórico de uma das principais fontes de dados para pesquisas no campo das Ciências Sociais Aplicadas no Brasil	17
2.2	Microdados, notas técnicas e dicionários: onde encontrar?	22
3	Estatísticas do trabalho: definições conceituais, evolução histórica e a PNAD Contínua	23
3.1	Um breve histórico da evolução dos princípios estatísticos que orientaram a construção dos principais indicadores de mercado de trabalho	23
3.2	Conceitos e indicadores para o tratamento da condição em relação a força de trabalho, participação econômica, ocupação e desocupação .	25
3.3	Renda corrente: tipos de rendimentos na PNAD Contínua	38
4	Potencialidades e limitações da PNAD Contínua: a construção de indicadores sociais, de pobreza e de desigualdades	42
4.1	Breves notas sobre estatísticas sociais	43
4.2	Por dentro do dicionário da PNAD Contínua: potencialidades e limites	48
5	Uma introdução ao ambiente e à linguagem de programação R	56
5.1	Por que usar o R?	57
5.2	RStudio	57
5.3	Primeiros passos: <i>download</i> e instalação	59
5.3.1	O <i>software</i> R	59

5.3.2	O <i>software RStudio</i>	59
5.4	Citação	60
5.5	Exercícios	62
5.6	O ambiente R	62
5.7	A área de trabalho e seus objetos	63
5.8	O editor de Scripts do RStudio	63
5.9	Pacotes do R	66
5.10	Exercícios	67
5.11	Comandos básicos	67
5.11.1	Estilo de programação	67
5.11.2	Operações e expressões: atribuição	69
5.11.3	Operações e expressões: impressão automática (<i>Autoprinting</i>)	70
5.11.4	Operações e expressões: espaços em branco	71
5.11.5	Operações e expressões: objetos	71
5.12	Exercícios	73
5.13	Vetores	75
5.13.1	Sequências	75
5.13.2	Operações com vetores	77
5.13.3	Acessando elementos ou subconjuntos de um vetor por meio de índices	79
5.13.4	Inverter a ordem de um vetor	80
5.14	Funções e argumentos	81
5.15	Data.frames	81
5.15.1	Indexação de data.frames	83
5.15.2	Seleção condicional de observações e transformações	84
5.16	Exercícios	87
5.17	Matrizes	88
5.17.1	Função matrix()	88
5.17.2	Matriz de zeros	90
5.17.3	Matriz diagonal e matriz identidade	90
5.17.4	Sintaxe	90
5.17.5	Multiplicação de matrizes	91
5.17.6	Adição de matrizes	93
5.17.7	Outras operações	93
5.17.8	As funções cbind() e rbind()	94

5.17.9	Nomear linhas e colunas de uma matriz	97
5.17.10	Submatrizes	98
5.17.11	Mais sobre matrizes	100
5.18	Listas	101
5.18.1	Sintaxe	102
5.18.2	Operações com listas: indexação	103
5.19	Fatores	106
5.20	Exercícios	108
6	Importação, leitura e salvamento dos microdados da PNAD Contínua: o pacote PNADcIBGE	109
6.1	Instalação	109
6.2	O pacote PNADcIBGE : funções básicas	110
6.3	Importação e <i>download</i> dos microdados da antiga PNAD anual	122
7	Tratamento dos microdados da PNAD Contínua: o pacote tidyverse	136
7.1	Instalação	136
7.2	O pacote dplyr : funções básicas	137
7.3	Síntese de dados	140
7.4	Agrupamento de casos	141
7.5	Tratamento de casos	142
7.5.1	Extração de casos	143
7.5.2	Arranjo de casos	145
7.5.3	Adição de casos	146
7.6	Tratamento de variáveis	147
7.6.1	Extração de variáveis	147
7.6.2	Criação de variáveis	148
7.6.3	Funções vetorizadas	150
7.6.4	Outras funções de síntese	153
7.6.5	Combinação/fusão de tabelas	157
7.7	Exercícios	164
8	Tratamento dos microdados da PNAD Contínua: o pacote survey	166

8.1	O escopo do pacote survey	166
8.2	Instalação do pacote e <i>download</i> dos dados	167
8.3	Principais funções do pacote survey para análises descritivas	169
8.4	Exemplos comparativos de análise usando os pacotes survey e dplyr .	170
8.4.1	Estimativa da contagem populacional para dados agrupados para variáveis categóricas	171
8.4.2	Estimativa da contagem populacional para dados agrupados por mais de uma variável categórica	171
8.4.3	Estimativa da contagem populacional com interação entre variáveis categóricas de agrupamento	172
8.4.4	Estimativa do somatório dos valores de variáveis contínuas	173
8.4.5	Estimativa da média ponderada para variáveis contínuas	174
8.4.6	Estimativa ponderada da proporção de pessoas no total populacional com uma variável categórica de agrupamento	175
8.4.7	Estimativa ponderada da proporção de pessoas no total populacional com mais de uma variável categórica de agrupamento, sem interação .	176
8.4.8	Estimativa ponderada da proporção de pessoas no total populacional com mais de uma variável categórica de agrupamento, com interação	176
8.4.9	Estimativa da contagem populacional para variáveis categóricas com casos omissos	177
8.4.10	Estimativa da proporção no total populacional para variáveis categóricas com casos omissos	178
8.4.11	Estimativas ponderadas de estatísticas para subconjuntos de dados .	179
8.4.12	Estimativa da proporção entre contagens populacionais para mais de uma variável categórica	179
8.4.13	Outras estimativas	180
8.4.14	Outras estimativas para subconjuntos populacionais	182
8.5	Exercícios	183
9	Uma ponte entre o SPSS e o R: o pacote expss	185
9.1	Instalação	186
9.2	Tratamento de dados	186
9.3	O pacote expss : algumas funções básicas	186
9.3.1	Aplicação de rótulos	187

9.3.2	Criação e transformação de variáveis	188
9.3.3	Aplicação de operações condicionadas	190
9.3.4	Recodificação de variáveis	192
9.4	Criação de tabelas	194
9.4.1	A função <code>fre</code>	195
9.4.2	A função <code>tables</code>	196
9.4.3	A função <code>cro</code>	211
9.5	Apresentação e formatação de tabelas: a interface com outros <i>softwares</i>	219
9.5.1	Console do RStudio e RStudio Viewer	219
9.5.2	Excel	222
9.5.3	L ^A T _E X	224
9.6	Exercícios	227
10	Interface gráfica: o pacote <code>ggplot2</code>	228
10.1	Definições básicas	228
10.2	Instalação e base de dados	229
10.3	Componentes básicos	230
10.4	Tipos básicos de apresentação de dados na forma gráfica	230
10.5	Representação gráfica dos valores de uma única variável contínua	232
10.6	Representação gráfica dos valores de uma única variável discreta	236
10.7	Representação gráfica dos valores de duas variáveis quaisquer	238
10.7.1	Duas variáveis contínuas (x e y)	238
10.7.2	Uma variável discreta e uma variável contínua	241
10.7.3	Duas variáveis discretas	243
10.7.4	Alterações estéticas e incorporação de variáveis para definição de subgrupos	244
10.7.5	Apresentação de distintos tipos de gráficos na mesma imagem	251
10.7.6	Séries temporais	255
10.7.7	Apresentação gráfica de frequências relativas (participações)	259
10.7.8	Transformações e ajustes para melhorar a apresentação dos gráficos	264
10.7.9	Outras modificações estéticas	270
10.8	Salvando gráficos	282
10.9	Exercícios	283

11	PNAD Contínua: Indicadores sociais, de mercado de trabalho, de pobreza e de desigualdades	284
11.1	Indicadores de mercado de trabalho	284
11.1.1	Bases de dados	284
11.1.2	A construção dos indicadores	285
11.2	Indicadores sociais e de condições de vida	289
11.3	Indicadores de pobreza, desigualdades e insuficiência socioeconômica	298
11.3.1	Indicadores de pobreza	298
11.3.2	Desigualdade de renda corrente	314
11.3.3	Índice do Nível de Insuficiência Socioeconômica (INIS)	321
11.4	Exercícios	328
	Referências Bibliográficas	330
	Lista de Figuras	337
	Lista de Tabelas	339

1 Introdução

Este é um livro didático que apresenta noções básicas para o tratamento e a exploração de bases de dados a partir da utilização do ambiente e da linguagem de programação R, com foco especial nas informações da Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua), do Instituto Brasileiro de Geografia e Estatística (IBGE), e na concepção de indicadores socioeconômicos e de mercado de trabalho.

O principal objetivo deste material é possibilitar um primeiro contato do leitor com a utilização de microdados, especialmente para aqueles que desejam aprofundar seus conhecimentos ou realizar pesquisas e estudos baseados nas informações estatísticas socioeconômicas disponibilizadas pelas PNADs Contínuas.

Pretende-se com esse esforço que o presente livro didático sirva para orientar estudos e pesquisas nas distintas esferas das Ciências Sociais, dedicando atenção especial a temas relevantes para pesquisas nas áreas de Economia Social e do Trabalho. Desse modo, este livro é recomendado para pesquisadores nas distintas esferas das Ciências Sociais Aplicadas, mas principalmente para alunos de graduação e pós-graduação.

Além dessa breve introdução, o livro está estruturado em dez capítulos:

- Capítulo 1 - Faz-se uma breve apresentação das Pesquisas Nacionais por Amostra de Domicílios, uma das principais fontes de dados para o tratamento de temas relevantes para pesquisas em Economia Social e do Trabalho e, também, para as Ciências Sociais como um todo, pois traz um conjunto de informações que permitem análises sobre: condições de vida e socioeconômicas, mercado de trabalho, renda, pobreza e desigualdades.
- Capítulo 2 - Apresentam-se os principais conceitos utilizados em grande parte

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

dos estudos, presentes na literatura, sobre mercado de trabalho, além de novos indicadores que podem ser construídos a partir dos dados da PNAD Contínua. Essa pesquisa, instituída no país a partir de 2012, ampliou o potencial de análise e investigação permitindo o aprofundamento de análise de distintos fenômenos que têm ganhado relevância nas sociedades capitalistas contemporâneas, especialmente em países subdesenvolvidos.

- Capítulo 3 - Debate-se o potencial da PNAD Contínua para o estudo de indicadores sociais, de condição de vida, pobreza e desigualdades.
- Capítulo 4 - Faz-se uma breve apresentação da linguagem de programação R, introduzindo ao leitor sua estrutura e forma, para permitir um primeiro contato com o ambiente de trabalho em R e o uso do RStudio(RStudio Team, 2021).
- Capítulo 5 - É apresentado o pacote em R desenvolvido pelo IBGE, que permite a abertura e o *download* das bases de dados nos formatos **survey.design** e no tradicional **data.frame**, além de arquivos auxiliares (dicionários, tabelas de deflatores, anexos metodológicos etc.).
- Capítulo 6 - Apresentam-se algumas das ferramentas de análise para o tratamento dos microdados da PNAD Contínua a partir do pacote **tidyverse**.
- Capítulo 7 - Introduzem-se alguns elementos a respeito da potencialidade do pacote **survey** para análise de microdados de pesquisas baseadas em planos amostrais complexos, como é o caso da PNAD Contínua.
- Capítulo 8 - São apresentados alguns dos recursos do pacote **expss** para o tratamento/transformação de dados, bem como a construção e a apresentação de tabelas, tão relevantes para análises de estatísticas descritivas. Cabe destacar que esse é um pacote que emula o uso de um tradicional *software* estatístico utilizado nas Ciências Sociais, a saber, o **SPSS**.
- Capítulo 9 - Faz-se a apresentação de um dos principais pacotes em R para construção de gráficos, a saber, o **ggplot2**.
- Capítulo 10 - São apresentadas distintas formas para se calcular indicadores sociais, de mercado de trabalho, de pobreza monetária e multidimensional e de desigualdades.

2 A Pesquisa Nacional por Amostra de Domicílios (PNAD): um breve histórico

O presente Capítulo procura fazer um resgate histórico das principais transformações ocorridas na PNAD ao longo do tempo, desde sua concepção na segunda metade dos anos 1960. Tais mudanças ampliaram seu escopo, sua abrangência e seu alcance, garantindo um papel de expressivo destaque para inúmeras pesquisas e estudos que buscaram mapear as condições socioeconômicas dos domicílios e de vida de seus moradores. Sua relevância se materializa, também, por dar suporte e orientação à elaboração de políticas públicas no país nas últimas décadas.

2.1 Um breve histórico de uma das principais fontes de dados para pesquisas no campo das Ciências Sociais Aplicadas no Brasil

A PNAD foi implantada e pensada, no Brasil, enquanto um sistema de pesquisas por amostra probabilística de domicílios em 1967. Sua abrangência nacional tinha por objetivo contribuir para elucidar questões relevantes em áreas como: demografia, saúde, consumo alimentar e nutrição, condições de habitação, consumo de bens e equipamentos domésticos, educação, cultura, além de temas relacionados ao mundo do trabalho.

No final dos anos 1960, a pesquisa contemplava as atuais Regiões Geográficas Nordeste, Sudeste e Sul, além do Distrito Federal. Nos anos em que houve realização dos Censos Demográficos (1970, 1980, 1991, 2000 e 2010), a PNAD foi interrompida. Na década de 1970, mais especificamente em 1973, a pesquisa am-

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

pliou sua abrangência geográfica e passou a contemplar, além das Regiões Nordeste, Sudeste e Sul, as áreas urbanas das atuais Regiões Norte e Centro-Oeste.

No início dos anos 1980, a pesquisa passou a cobrir todo o país, com exceção das áreas rurais dos estados da Região Norte (Rondônia, Acre, Amazonas, Roraima, Pará e Amapá), que representavam cerca de 3% da população brasileira. Após a realização do Censo de 1991, a PNAD voltou a ser realizada (1992 em diante) sem qualquer alteração em sua abrangência demográfica. No entanto, a partir de 2004, seu escopo foi ampliado mais uma vez para abarcar as áreas rurais da Região Norte.

Após 2013, a pesquisa passou a investigar, de forma mais profunda, alguns aspectos referentes ao acesso à tecnologia da informação e comunicação (TIC), com destaque para a Internet em banda larga e, também, para a recepção de diferentes modalidades de sinais de televisão. Seu objetivo era avaliar o período em que se verificava a transição do sistema analógico de TV para o digital.

A impossibilidade de se investigar todos esses temas de forma recorrente fez com que a pesquisa fosse estruturada da seguinte forma: uma Pesquisa Básica, Pesquisas Suplementares e algumas Pesquisas Especiais para contemplar de forma intermitente diversos e distintos temas.

A Pesquisa Básica procura trazer informações que permitem a investigação de fenômenos considerados mais relevantes para mensurar a evolução do nível socioeconômico da população brasileira. As Pesquisas Suplementares produzem informações mais profundas sobre temas de caráter mais permanente, o que possibilita, de forma complementar à Pesquisa Básica, a investigação de outros assuntos de interesse social e econômico para o país. Já as Pesquisas Especiais cobrem assuntos de complexidade ainda maior.

Desde sua implantação, os temas habitação e trabalho foram contemplados pela Pesquisa Básica e definidos como permanentes. Ambos seriam complementados por informações a respeito de características demográficas, educacionais e de rendimentos.

As Pesquisas Suplementares da PNAD, ao longo de sua história, permitiram a investigação da migração interna, da fecundidade, do consumo alimentar, do orçamento familiar (Estudo Nacional da Despesa Familiar - ENDEF), da mobilidade social, da cor ou raça das pessoas (1976), da mortalidade, do consumo de Energia (1979), da saúde (1981), da educação (1982), da previdência (1983), da situação da população menor de idade (1985), dos métodos anticoncepcionais (1986), do acesso

POR DENTRO DA PNAD CONTÍNUA

a serviços de saúde, da suplementação alimentar, do associativismo e da participação político-social (1988, ano da promulgação da chamada Constituição Cidadã, no bojo do processo de redemocratização nacional).

Nos anos 1990 e no período de 2001 a 2008, os suplementos, para além da migração e da fecundidade, passaram a contemplar: em 1992, ensino supletivo, nupcialidade e trabalho infantil de crianças de 5 a 9 anos de idade; em 1996, mobilidade social; em 1998, saúde e trabalho das crianças de 5 a 9 anos de idade; em 2003, participação em programas voltados para educação; em 2004, acesso a transferências de renda de programas sociais e segurança alimentar; em 2005, acesso à internet e posse de telefone móvel celular para uso pessoal; em 2007, aspectos complementares da educação de jovens e adultos; em 2008, tabagismo; em 2009, vitimização e Justiça (temas investigados apenas em 1988); e, em 2013, acesso a sinal digital de televisão aberta.

A partir de 2009, outro conjunto de perguntas passou a integrar os questionários da pesquisa, a saber, aspectos relacionados ao acesso por parte dos domicílios às TIC's, bem como seu uso individual pelos seus moradores. Nesse conjunto de dados, o foco na Internet de banda larga e a distinção quanto à posse de aparelhos eletrônicos utilizados para acessar a internet como: microcomputador, telefone móvel celular, *tablet* e outros, apresentam-se como as principais mudanças ocorridas na pesquisa. A ideia por trás dessa alteração, ocorrida em temas fundamentais para a compreensão da sociedade nos tempos modernos, remete ao fato de o Brasil ter passado por transformações significativas do ponto de vista do acesso a um conjunto de bens e serviços associados a tais tecnologias. Domicílios com *tablets* ou com acesso à banda larga, tanto por meio de tecnologias fixas (DSL, cabo de televisão por assinatura, cabo de fibra óptica, satélite e rádio) quanto de móveis (3G e 4G), passaram a representar uma parcela expressiva do total de domicílios no Brasil, após a década de 2010. Além disso, a PNAD passou a investigar a posse, nos domicílios, de equipamentos de televisão de tela fina, serviços de televisão por assinatura, recepção de sinal digital de televisão aberta e antena parabólica.

Sua última publicação enquanto pesquisa de periodicidade anual foi em 2017, com os dados referentes a 2015. A PNAD nesse formato deu lugar à Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua), cujo início ocorreu em 2012.

A nova pesquisa passou a permitir que se acompanhasssem as flutuações

conjunturais da força de trabalho (curto e médio prazos) por meio de sua divulgação trimestral, sem perder o poder informativo de aspectos mais estruturais (longo prazo), contemplados em sua divulgação agregada anualmente.

Em grande medida, seu planejamento foi pensado para atender à necessidade da construção de indicadores conjunturais mais amplos e de periodicidade trimestral para a força de trabalho. Nesse novo formato, mantiveram-se as principais informações necessárias para o estudo do desenvolvimento socioeconômico brasileiro, porém acrescentando outros elementos que passaram a fazer parte do escopo da pesquisa.

A PNAD Contínua investiga trimestralmente mais de 210 mil domicílios em um conjunto relativamente maior de municípios, quando comparada à sua versão anterior. Segundo o IBGE (2019), isso possibilita um ganho considerável em termos de precisão das estimativas. As informações de caráter conjuntural a respeito da força de trabalho, captadas pelas entrevistas realizadas nos domicílios e divulgadas trimestralmente, são complementadas por um conjunto de informações adicionais de cunho mais estrutural divulgadas, apenas, anualmente.

Nesse escopo mais estrutural, de longo prazo, também é possível investigar características dos domicílios associadas às condições de habitação, à existência de bens duráveis, além de rendimentos de outras fontes. Vale destacar que, na divulgação trimestral, as informações são apenas referentes aos rendimentos do trabalho (habituais e efetivos). Para além de prover informações sobre a renda das pessoas, um dos principais objetivos da PNAD Contínua é permitir a construção de indicadores sobre condições e qualidade de vida da população, de forma a servir como guia e referência para a construção de políticas públicas no país.

Outras informações sobre temas relevantes no campo da Economia do Trabalho também se mostram presentes na pesquisa. Dentre esses, destacam-se: trabalho infantil, outras formas de trabalho, como a produção para o próprio consumo, trabalho voluntário, cuidado de pessoas do domicílio/familiares e afazeres domésticos. Essas informações são divulgadas, apenas, anualmente.

A respeito da força de trabalho, recortes analíticos podem ser elaborados a partir de características demográficas, de educação, regionais, de situação dos domicílios etc. Esses temas são investigados em um trimestre específico ou em parte da amostra a cada trimestre na PNAD Contínua. Posteriormente, tais informações são acumuladas a fim de gerar resultados anuais. São produzidos, também, alguns

POR DENTRO DA PNAD CONTÍNUA

indicadores sobre outros temas suplementares de periodicidade variável.

A unidade de investigação principal da pesquisa continua sendo o domicílio. Implantada, experimentalmente, em outubro de 2011, passou a assumir um caráter definitivo em janeiro de 2012 com abrangência para todo o território nacional. Sua amostra foi pensada para produzir resultados para o total do Brasil, para as Grandes Regiões, suas Unidades da Federação, Regiões Metropolitanas que possuem municípios de capitais, para a Região Integrada de Desenvolvimento (RIDE) da Grande Teresina, e para os próprios municípios das capitais dos estados.

A pesquisa vem, de forma gradual, ampliando o número de informações e indicadores investigados. Sua divulgação tem distintas periodicidades: 1) mensal, para um conjunto restrito de indicadores associados à força de trabalho, sendo limitada ao nível geográfico do Brasil; 2) trimestral, para todos os níveis geográficos contemplados pela pesquisa, com a ressalva de que esta se limita a temas associados à população e à força de trabalho; e 3) anual, para os demais temas permanentes e indicadores complementares à força de trabalho. De forma variável, outros temas ou tópicos permanentes são pesquisados ocasionalmente.

Tecnicamente,

Os indicadores mensais utilizam as informações dos últimos três meses consecutivos da pesquisa, existindo, entre um trimestre móvel e o seguinte, repetição das informações de dois meses. Assim, os indicadores da PNAD Contínua produzidos mensalmente não refletem a situação de cada mês, mas, sim, a situação do trimestre móvel que finaliza a cada mês (IBGE, 2021a).

Cabe destacar que os dados anuais são obtidos acumulando-se as informações de visitas específicas aos domicílios ao longo do ano, ou concentradas em determinado trimestre.

Os temas e tópicos suplementares, pesquisados em trimestres específicos do ano, são: educação (2º trimestre); e acesso à televisão e à internet além da posse de telefone móvel celular para uso pessoal (4º trimestre). Já os temas pesquisados ao longo do ano em visitas específicas são: habitação, características gerais dos moradores, informações adicionais da força de trabalho (1ª visita); outras formas de trabalho (afazeres domésticos, cuidados de pessoas, produção para o próprio consumo e trabalho voluntário, trabalho de crianças e adolescentes (5ª visita); e rendimentos de outras fontes (1ª e 5ª visitas).

2.2 Microdados, notas técnicas e dicionários: onde encontrar?

Os microdados de todas as PNADs são de domínio público, de livre acesso, e estão disponíveis no site oficial do IBGE¹. Mais recentemente, o instituto passou a divulgar esses microdados, bem como toda a documentação correspondente, para os anos de 1976 a 2015².

No caso da PNAD Contínua, os microdados podem, também, ser acessados diretamente pelo site do IBGE, na parte a ela referente³, na seção específica denominada “Microdados”.

Para aqueles que desejarem aprofundar seus conhecimentos a respeito dos conceitos e das técnicas de construção do plano amostral complexo das PNADs, é de fundamental importância a leitura das notas técnicas mais gerais e as que acompanham os microdados para cada ano específico; afinal, notam-se, em todos os anos, algumas modificações como, por exemplo, o nome e/ou, no conteúdo de variáveis, a incorporação de novos temas, como aqueles que aparecem em distintos suplementos, dentre outras.

Mais que isso, tanto para a produção de estudos e análises a respeito do mercado de trabalho quanto para a construção de indicadores sociais e/ou de condições de vida, recomenda-se a leitura atenta dos dicionários das bases de dados específicas, trimestrais e anuais (agrupadas por visita ou por trimestres), para que se tenha uma compreensão mais ampla do escopo da pesquisa em termos do conjunto de variáveis e de seus conteúdos. Esse esforço permite ao pesquisador conhecer não apenas a potencialidade, mas, também, as limitações dessas pesquisas para a análise do objeto ou do fenômeno social pesquisado.

¹ <https://www.ibge.gov.br>

² Esses dados estão disponíveis no site do IBGE, podendo ser acessado por meio do seguinte endereço: <https://www.ibge.gov.br/estatisticas/sociais/trabalho/9127-pesquisa-nacional-por-amostra-de-domicilios?=&t=downloads>

³ <https://www.ibge.gov.br/estatisticas/sociais/trabalho/17270-pnad-continua.html?=&t=microdados>

3 Estatísticas do trabalho: definições conceituais, evolução histórica e a PNAD Contínua

Neste capítulo, pretende-se apresentar, brevemente, a evolução dos principais indicadores utilizados na literatura sobre Economia do Trabalho no Brasil, a terminologia usada para sua elaboração e os avanços metodológicos possibilitados pela PNAD Contínua. Mais que isso, procura-se explorar as potencialidades dessa pesquisa para capturar não apenas os indicadores básicos relativos à força de trabalho como também a subutilização da força de trabalho, um fenômeno que se mostra cada vez mais presente na sociedade brasileira e que tem chamado a atenção de pesquisadores e estudiosos sobre o Mundo do Trabalho. Ademais, breves comentários serão feitos a respeito do seu potencial para a mensuração de um conjunto de indicadores sociais, que visam contribuir para a qualificação e a quantificação de fenômenos associados a transformações das condições de vida, da pobreza e das desigualdades.

3.1 Um breve histórico da evolução dos princípios estatísticos que orientaram a construção dos principais indicadores de mercado de trabalho

Como mostrou Dedecca (2006), a evolução do capitalismo no século XX, em um contexto de consolidação e criação do Estado de Bem-Estar Social, passou a

exigir um conjunto amplo de informações socieconômicas para orientar as políticas públicas em áreas como saúde, educação e trabalho.

Com a criação da Organização Internacional do Trabalho (OIT), no pós Primeira Guerra Mundial, mais precisamente em 1919, conformou-se um espaço permanente de debate sobre a construção de estatísticas socieconômicas no âmbito do mercado de trabalho. Após a Segunda Guerra Mundial, a Organização das Nações Unidas (ONU) ampliou a abrangência da construção e o escopo da produção de informações econômicas, passando a orientar as nações para se construir, com base em conceitos e normas, as diretrizes para a contabilidade social, com foco na mensuração do produto e da renda agregados.

Na esfera social, as recomendações e orientações focaram na padronização dos censos demográficos, culminando, assim, na construção dos Sistemas Nacionais de Estatística em boa parte dos países. A partir da metade do século XX, dessa forma, a preocupação concentrou-se: 1) na configuração dos levantamentos estatísticos; e 2) nos conceitos básicos para o tratamento dos distintos temas socioeconômicos.

Ainda, segundo Dedecca (2006), as orientações a respeito das formas e dos conteúdos que passaram a definir a construção e a captação de estatísticas socioeconômicas partiam de duas dimensões metodológicas básicas:

1. Definição das categorias básicas a respeito do trabalho.
2. Definição do Sistema de Contas Nacionais, da Classificação de Atividade Econômica e da Classificação de Uso do Tempo (classificações socioeconômicas).

Foram essas dimensões que permitiram o estabelecimento de critérios para definir o conceito de trabalho do ponto de vista estatístico e para refletir aquilo que a sociedade considerava como trabalho. Esse procedimento carregava desafios de duas ordens:

1. “heterogeneidade das configurações econômicas e sociais entre nações, que estabelece diferentes estágios de desenvolvimento econômico e social e, portanto, uma diversidade enorme de conformações específicas de organização da atividade produtiva e dos mercados de trabalho” (DEDECCA, 2006, p. 107).
2. sistemáticas mudanças econômicas e sociais, que se processam de modo distinto, com especificidades para cada país, que exigiam “uma atividade de per-

POR DENTRO DA PNAD CONTÍNUA

manente atualização das categorias e classificações adotadas” (DEDECCA, 2006, p. 107).

A despeito de se reconhecer esses desafios, o Brasil tem seguido as orientações e recomendações internacionais estabelecidas pela ONU. Os levantamentos socioeconômicos nacionais têm procurado estabelecer metodologias compatíveis com as práticas que orientam a construção de indicadores estatísticos utilizados em outros países, porém sem perder de vista as especificidades e a configuração socioeconômica nacionais.

Nessa trajetória de aprofundamento dos conhecimentos sobre o mundo do trabalho, destacam-se, como fontes de informações socioeconômicas no Brasil, os levantamentos estatísticos domiciliares dos Censos Demográficos e das Pesquisas Nacionais por Amostra de Domicílios. Nessas pesquisas, “a noção de trabalho associa-se às atividades de produção de bens e serviços mercantis e não-mercantis, desde que as últimas provoquem aumento do padrão de consumo ou melhoria do bem-estar” (DEDECCA, 2006, p. 108).

Não se pode deixar de mencionar que, nem todas as formas de trabalho têm sua renda mensurada por esses levantamentos. A título de exemplo, as pesquisas captam rendimentos monetários associados a ocupações específicas, porém, para alguns rendimentos implícitos e não-monetários, nota-se que muitos não podem ser captados adequadamente, como é o caso da produção para o próprio uso.

3.2 Conceitos e indicadores para o tratamento da condição em relação a força de trabalho, participação econômica, ocupação e desocupação

Distintos fatores biológicos, culturais e sociais podem afetar a inserção da população na vida produtiva. Isso faz com que nem todas as pessoas se encontrem disponíveis para exercer algum tipo de trabalho. As lutas dos trabalhadores, a consolidação da legislação trabalhista e a ampliação da segurança social também podem afetar essa inserção. Por exemplo, esses fatores fizeram com que o avanço da organização social no capitalismo do século XX restringisse o uso de força de trabalho de crianças e idosos. Outras parcelas da população, também, podem não se encontrar disponíveis para trabalhar por motivos diversos, como são os casos dos

presidiários e das pessoas com deficiências incapacitantes.

A proibição do trabalho infantil, por lei, e a universalização dos sistemas educacionais fez com que se passasse a determinar uma idade mínima para ingressar no mercado de trabalho. No Brasil, esse limite foi definido pela Constituição Federal em 16 anos, seguindo orientação internacional da ONU. A exceção são os jovens aprendizes que podem iniciar sua vida laboral a partir dos 14 anos em ocupações específicas.

Em termos conceituais, as pesquisas em Economia do Trabalho passaram a dividir a população total em:

1. **População em Idade Ativa (PIA):** representa a parcela da População Total (PT) com idade entre 16 e 65 anos. Esse é o segmento que representa o recurso humano potencial máximo que uma sociedade pode dispor para realizar suas atividades produtivas. Cabe destacar que, em países subdesenvolvidos como o Brasil, onde o avanço do Estado de Bem-Estar social foi limitado, a PIA utilizava como referência as pessoas de 10 anos ou mais sem estabelecer critérios de idade máxima para a participação na atividade econômica¹.
2. **População Economicamente Ativa (PEA):** partindo-se da ideia de que nem todos com 10 anos ou mais de idade estão disponíveis para entrar no mercado de trabalho, como é o caso de estudantes, doentes, trabalhadores domésticos que realizam serviços exclusivamente para sua própria família sem remuneração, aposentados e presos, decidiu-se chamar de PEA a parcela da PIA que pode ser considerada em exercício de alguma atividade produtiva ou em busca de participar do processo de produção. O conceito que define a PEA, assim, está associado àquele que define a oferta de trabalho existente no sistema econômico e social. A partir da constatação de que a situação mais comum no modo de produção capitalista é a de que oferta e demanda por trabalho não assumem um mesmo valor, sendo a primeira, recorrentemente, maior que a segunda, subdividi-se a PEA em Pessoas Ocupadas (PO) e Pessoas em Situação de Desemprego (PD).
3. **População Ocupada (PO):** a ocupação se manifesta por meio de distintos

¹ “A adoção de uma PIA de dimensões mais amplas deve-se à recorrência do trabalho infantil e também da presença de idosos na vida produtiva, em razão do não-acesso à proteção social” (DEDECÇA, 2006, p. 110).

POR DENTRO DA PNAD CONTÍNUA

regimes ou relações de trabalho. Os exemplos mais evidentes no capitalismo são: o assalariamento, a ocupação autônoma/independente (conta própria) e a ocupação sob forma de empregador. Algumas outras categorias aparecem em menor proporção, como é o caso dos trabalhadores não remunerados em ajuda a negócio da família, dos trabalhadores que produzem para o próprio consumo ou constroem para seu próprio uso. Segundo Dedecca (2006), esses regimes ocupacionais podem ser desenvolvidos formalmente, quando cumprem normas legais que regulamentam a atividade produtiva e o mercado de trabalho, ou, informalmente, quando ficam à margem da regulação e da proteção social. Nos estudos sobre mercado de trabalho, a formalidade é normalmente atribuída ao assalariamento com registro em carteira de trabalho, ao trabalho por conta própria com contribuição para a Previdência Social e à ocupação de empregadores que constituíram uma empresa legal com registro no CNPJ. Já a informalidade é definida pelo oposto, ou seja, pelos trabalhadores assalariados sem carteira de trabalho assinada, autônomos sem contribuição previdenciária e empregadores sem registro formal da empresa. Essa definição tem sido usada para caracterizar a maior parte das pessoas, dentro desses últimos segmentos, cujas atividades ocorrem sem proteção legal e, em grande medida, em condições relativamente mais precárias.

4. **População Desempregada (PD):** essa é uma condição que caracteriza pessoas que não estão envolvidas em qualquer atividade produtiva, mas que tomaram providência para buscar uma oportunidade de trabalho no período de referência da pesquisa.

No Brasil, antes da consolidação da PNAD Contínua como principal pesquisa para retratar as características da força de trabalho, havia três formas de mensuração do desemprego:

1. **Desemprego Aberto:** conceito presente no Censo Demográfico e na Pesquisa Nacional por Amostra de Domicílio anual (até 2015), em que se define desempregada a pessoa que não realizou atividade produtiva pressionando permanentemente o mercado de trabalho com procura por ocupação nos últimos 30 dias.
2. **População inativa ligada marginalmente à PEA:** conceito mensurado pela extinta Pesquisa Mensal de Emprego (PME), em que se considerava a

população inativa que se encontrava disponível para trabalhar nos últimos 358 dias.

3. **População inativa em situação de desalento:** conceito, também mensurada pela PME, em que se considera como desalentadas aquelas pessoas que deixaram de procurar trabalho nos seis meses anteriores à pesquisa, por se sentirem desestimuladas em razão das condições do mercado de trabalho.

Na Pesquisa de Emprego e Desemprego (PED), do Dieese em parceria com a Fundação SEADE, outras nomenclaturas eram utilizadas para a mensuração do desemprego e para outros fenômenos que se manifestam em paralelo ao desemprego aberto. Na metodologia da PED, além da possibilidade de se calcular a taxa de desemprego aberto para regiões metropolitanas específicas, era possível calcular o chamado desemprego oculto por trabalho precário ou por desalento.

Como deixou claro Dedecca (2006), essas formas de expandir e aprofundar as análises do campo da Economia do Trabalho para além do desemprego, surgem da:

[...] necessidade de se ter maior cuidado na classificação da condição de desemprego em sociedades em que o sistema de proteção social ao desemprego é inexistente ou limitado. Em tais economias, o desempregado, em geral, deve financiar a procura de trabalho, resolver o problema de renda decorrente da perda do emprego anterior e estabelecer, de maneira autônoma, uma estratégia de busca de uma nova ocupação. Nesse sentido, a procura pode estar associada à existência de alguma atividade remunerada realizada de maneira irregular e descontínua, impedindo que o desempregado utilize completamente seu tempo para procurar uma oportunidade de trabalho (DEDECCA, 2006, p. 115).

Com base nessa terminologia, os indicadores básicos para qualquer análise sobre o mercado de trabalho eram os seguintes:

$$\text{Taxa de Participação} = \frac{\text{População Economicamente Ativa}}{\text{População em Idade Ativa}} \quad (3.1)$$

$$\text{Taxa de Ocupação} = \frac{\text{População Ocupada}}{\text{População Economicamente Ativa}} \quad (3.2)$$

POR DENTRO DA PNAD CONTÍNUA

$$\text{Taxa de Desemprego Aberto} = \frac{\text{População em Desemprego Aberto}}{\text{População Economicamente Ativa}} \quad (3.3)$$

$$\text{Taxa de Assalariamento} = \frac{\text{População Assalariada}}{\text{População Ocupada}} \quad (3.4)$$

A partir da PNAD Contínua, o IBGE introduziu uma nova terminologia a respeito das características do trabalho, incorporando as formas ocultas captadas pela PED, mas que não faziam parte da antiga PNAD anual. Esse movimento foi feito na direção de melhorar e ampliar a captação do que se denominou subutilização da força de trabalho. Nesse ponto, é necessário apresentar essa nova nomenclatura, para deixar evidenciadas as principais diferenças.

A definição de trabalho na PNAD Contínua contempla, também, distintas formas de produção de bens e serviços para terceiros, ou para consumo próprio, podendo ser entendida, segundo o IBGE (2020b), a partir das seguintes classificações:

1. **Trabalho em ocupação** remunerado (dinheiro ou produtos) na produção de bens e serviços ou trabalho sem remuneração em ajuda à atividade econômica de membro do domicílio ou familiares. “O item [...] trabalho em ocupação, apresenta a forma de trabalho adotada para definir a força de trabalho. Esse conceito de trabalho em ocupação, utilizado a partir do quarto trimestre de 2015, já está ajustado à Resolução I da 19^a Conferência Internacional de Estatísticos do Trabalho - CIET”² (IBGE, 2020b, p. 127).
2. **Trabalho na produção para o próprio consumo** domiciliar ou de familiares em atividades produtivas ligadas à agropecuária, à transformação de produtos minerais e florestais, à fabricação de bens de uso doméstico e à construção, ampliação ou reparos realizados na própria moradia.
3. **Trabalho voluntário** (não compulsório) não remunerado, realizado por pelo menos uma hora na semana em benefício de terceiros (não moradoras do domicílio e não familiares).

² Para mais detalhes sobre a Resolução I da 19^a Conferência Internacional de Estatísticos do Trabalho - CIET, ver ILO (2013).

4. **Trabalho sem remuneração no cuidado e/ou em apoio ou auxílio a outras pessoas** que não se encontram capazes de realizar tarefas laborais de forma independente (crianças, idosos, pessoas que necessitam de cuidados especiais etc.).
5. **Trabalho não remunerado em afazeres domésticos** em benefício próprio e/ou de outros moradores do domicílio.

A nomenclatura “População em Idade Ativa” foi alterada para “Pessoas em Idade de Trabalhar”, cuja definição refere-se às pessoas de 14 anos ou mais de idade na data de referência da pesquisa.

As pessoas ocupadas são assim classificadas por terem trabalhado por pelo menos uma hora completa de forma remunerada em dinheiro ou em produtos na semana de referência. São ocupados, também, aqueles que exerceram algum trabalho sem remuneração no caso de ajuda à atividade econômica de algum membro do domicílio ou de familiares. Ou, ainda, pessoas que possuíam trabalho remunerado, mas se encontravam afastadas de forma temporária na semana da pesquisa (a exemplo das que estavam de férias, folga, licença remunerada etc. ou cujo afastamento era inferior a quatro meses anteriores ao último dia da semana de referência da pesquisa. A partir do quarto trimestre de 2015, esse conceito foi adequado àquele definido por ILO (2013), 19^a CIET³.

As pessoas desempregadas passaram a ser definidas como Pessoas Desocupadas, também em conformidade com as normativas definidas por ILO (2013). Na PNAD Contínua, desocupadas são as pessoas sem trabalho em ocupação na semana de referência da pesquisa e que tomaram, efetivamente, alguma providência para conseguir trabalho nos últimos 30 dias anteriores à pesquisa. Mas, mais que isso, essas pessoas têm de estar disponíveis para assumir o posto de trabalho na semana. Aquelas que haviam conseguido um trabalho, mas cujo início se daria em menos de quatro meses após o último dia da semana da pesquisa, também são consideradas desocupadas⁴.

³ “Os ajustes ocorreram nos aspectos referentes ao trabalho sem remuneração direta ao trabalhador e à caracterização como ocupadas das pessoas que tinham trabalho remunerado do qual estavam temporariamente afastadas na semana de referência” (IBGE, 2020a, p. 4).

⁴ “Anteriormente, no que se refere às pessoas que não tomaram providência efetiva para conseguir trabalho no período de referência de 30 dias porque já o haviam conseguido para começar após a semana de referência, não havia limite de tempo fixado para assumir o trabalho” (IBGE, 2020a,

POR DENTRO DA PNAD CONTÍNUA

O ato de procurar trabalho passou a ser definido como a tomada de alguma providência efetiva para conseguir trabalho. Mais precisamente, são:

[...] o contato estabelecido com empregadores; a prestação de curso; a inscrição em concurso; a consulta a agência de emprego, sindicato ou órgão similar; a resposta a anúncio de emprego; a solicitação de trabalho a parente, amigo, colega ou por meio de anúncio; a tomada de medida para iniciar o próprio negócio mediante a procura de local, equipamento ou outros pré requisitos; a solicitação de registro ou licença para funcionamento do empreendimento etc. (IBGE, 2019, p. 28).

Na PNAD Contínua, as Pessoas em Idade de Trabalhar, a antiga PIA, são classificadas em relação à sua situação de ocupação em:

1. **Ocupadas:** aquelas que trabalharam por pelo menos uma hora completa de forma remunerada em dinheiro ou em produtos na semana de referência.
2. **Não ocupadas:** aquelas que se encontravam desocupadas ou fora da força de trabalho.

A condição em relação à força de trabalho é uma forma de classificar as pessoas em dois subgrupos⁵:

1. **Pessoas na força de trabalho:** pessoas ocupadas ou desocupadas.
2. **Pessoas fora da força de trabalho:** pessoas que não estavam ocupadas nem desocupadas.

Segundo o IBGE (2020b), o conceito de ocupação refere-se ao cargo, função, profissão ou ofício exercido pelas pessoas na semana de referência da pesquisa. O IBGE utiliza uma definição própria para classificar essas ocupações, a saber, a Classificação de Ocupações para Pesquisas Domiciliares - COD⁶. Essa classificação foi

p. 4).

⁵ Todas essas definições levam em consideração a semana de referência da pesquisa.

⁶ “A CBO-Domiciliar mantém-se idêntica à ISCO-08 no nível mais agregado (grande grupo) e regrupa alguns subgrupos principais, subgrupos e grupos de base, considerando as especificidades nacionais e as dificuldades de sua captação com precisão nas pesquisas domiciliares” (IBGE, 2020b, p. 127).

desenvolvida para as pesquisas domiciliares e está baseada na International Standard Classification of Occupations (ISCO-08), da Organização Internacional do Trabalho (OIT).

As atividades produtivas são classificadas a partir da finalidade ou do ramo de negócio, empresa ou entidade do empreendimento (empresa, instituição, entidade, firma, negócio ou trabalho desenvolvido individualmente ou com ajuda de terceiros). No caso dos trabalhadores por conta própria, essa classificação depende da natureza da ocupação exercida pela pessoa.

Todas as atividades são classificadas de acordo com a Classificação Nacional de Atividades Econômicas Domiciliar - CNAE-Domiciliar 2.0, uma adaptação realizada pelo IBGE com base na Classificação Nacional de Atividades Econômicas - CNAE 2.0⁷.

As pessoas podem, também, ser classificadas segundo sua posição na ocupação. Esse é um conceito que se refere à “relação de trabalho existente entre a pessoa e o empreendimento em que trabalha” na semana de referência (IBGE, 2020b, p. 141).

De acordo com IBGE (2020b), definem-se quatro categorias para as posições na ocupação:

1. **Empregados:** definidos como as pessoas que trabalhavam para um empregador (Pessoa Física ou Jurídica), com jornada de trabalho e remuneração em dinheiro, mercadorias, produtos ou benefícios como: moradia, alimentação, roupas, treinamento etc. São considerados empregados, também, aqueles que prestavam serviço militar obrigatório, sacerdotes, padres, pastores e outros tipos de clérigos. Nessa categoria, são incluídos os trabalhadores domésticos remunerados (em dinheiro ou benefícios) em um ou mais domicílios.
2. **Trabalhadores por conta própria:** definidos como aqueles cujo trabalho é a exploração do próprio empreendimento, sozinhos ou com sócio (os), empre-

⁷ “A CNAE-Domiciliar 2.0 mantém-se idêntica à CNAE 2.0 nos níveis mais agregados (seção e divisão), com exceção das divisões do comércio em que não se distingue o atacado do varejo, reagrupa classes onde o detalhamento foi considerado inadequado para as pesquisas domiciliares e desagrega algumas atividades de interesse para as pesquisas domiciliares. CNAE 2.0 tem como referência a International Standard Industrial Classification of all Economic Activities - ISIC (Clasificación Industrial Internacional Uniforme de todas las Actividades Económicas - CIU), 4^a revisão, das Nações Unidas” (IBGE, 2020b, p. 128).

POR DENTRO DA PNAD CONTÍNUA

gados, porém podendo contar com a ajuda de familiares (trabalhador familiar auxiliar).

3. **Empregadores:** são as pessoas cujo trabalho é explorar um empreendimento próprio, a partir da contratação de ao menos um empregado.
4. **Trabalhadores familiares auxiliares:** são aquelas pessoas que trabalharam sem remuneração em alguma atividade econômica por, ao menos, uma hora na semana, ajudando algum membro do domicílio ou familiar.

Para os empregados, são definidas três subcategorias:

1. Empregados com carteira de trabalho assinada.
2. Militares e funcionários públicos estatutários.
3. Empregados sem carteira de trabalho assinada.

Esse mesmos empregados podem ser classificados em:

1. **Setor público:** definidos como empreendimentos de qualquer esfera do governo federal, estadual ou municipal, autarquias e/ou empresas públicas, inclusive as de economia mista.
2. **Setor privado:** definidos como empreendimentos no setor privado.

Do ponto de vista metodológico-analítico, IBGE (2021), em suas notas técnicas, apresenta um conjunto de indicadores básicos para análise do mercado de trabalho com nomenclaturas, ligeiramente, diferentes das empregadas anteriormente nas pesquisas domiciliares no país. Dentro esses, destacam-se:

$$\text{Taxa de Participação} = \frac{\text{População na Força de Trabalho}}{\text{População em Idade de Trabalhar}} \quad (3.5)$$

$$\text{Nível de Ocupação} = \frac{\text{População Ocupada}}{\text{População em Idade de Trabalhar}} \quad (3.6)$$

$$\text{Taxa de Ocupação} = \frac{\text{População Ocupada}}{\text{População na Força de Trabalho}} \quad (3.7)$$

$$\text{Nível de Desocupação} = \frac{\text{População Desocupada}}{\text{População em Idade de Trabalhar}} \quad (3.8)$$

$$\text{Taxa de Desocupação} = \frac{\text{População Desocupada}}{\text{População na Força de Trabalho}} \quad (3.9)$$

As transformações no capitalismo contemporâneo exigem constantes atualizações das formas de mensuração dos fenômenos que se manifestam no Mundo do Trabalho. Alguns desses fenômenos só podiam, até certo ponto, ser parcialmente captados pela PED da Fundação SEADE, em parceria com o DIEESE. Nesse sentido, uma das principais contribuições da PNAD Contínua para os estudos em Economia do Trabalho foi a incorporação de distintas perguntas em seus questionários para permitir uma avaliação mais precisa de alguns desses fenômenos.

Seguindo as recomendações apresentadas na Resolução I da 19^a CIET⁸, realizada pela OIT, o IBGE passou a divulgar informações sobre o que se denominou Subutilização da Força de trabalho. De acordo com o IBGE (2016), a subutilização é um conceito desenvolvido para complementar os estudos e o monitoramento do mercado de trabalho. Essa formulação ampliou o escopo de análise para além do conceito básico de desocupação/desemprego. Seu objetivo é fornecer uma estimativa mais precisa da efetiva demanda por trabalho em termos de ocupação.

São três os elementos que compõem a chamada população subutilizada. Mutuamente exclusivos, dois desses componentes integram a Força de Trabalho (FT), a saber, as Pessoas Subocupadas por Insuficiência de Horas Trabalhadas (PSIHT) e as Pessoas Desocupadas (PD). O terceiro componente foi denominado Força de Trabalho Potencial (FTP).

Segundo a ILO (2013), os países, especialmente os subdesenvolvidos e periféricos, devem adotar indicadores capazes de proporcionar uma análise mais robusta acerca do complexo fenômeno da subutilização da força de trabalho. Para além da tradicional Taxa de Desocupação, definida anteriormente, podem ser adotados os seguintes indicadores:

- 1. Taxa combinada da subocupação por insuficiência de horas trabalhadas**

⁸ Ver ILO (2013).

POR DENTRO DA PNAD CONTÍNUA

lhadas e da desocupação (TCSIED)

$$TCSIED = \frac{\text{Subocupados por Insuficiência de Horas} + \text{Desocupados}}{\text{Força de Trabalho}} \quad (3.10)$$

2. Taxa Combinada da Desocupação e da Força de Trabalho Potencial (TCDeFTP)

$$TCDeFTP = \frac{\text{Desocupados} + \text{Força de Trabalho Potencial}}{\text{Força de Trabalho Ampliada}} \quad (3.11)$$

3. Taxa Composta da subutilização da Força de Trabalho (TCSFT)

$$TCSFT = \frac{\text{Subocupados por Insuficiência de Horas} + \text{Desocupados} + \text{Força de Trabalho Potencial}}{\text{Força de Trabalho Ampliada}} \quad (3.12)$$

Para entender esses indicadores, é necessário que se conceituem os elementos que os conformam. Primeiramente, segundo o IBGE (2016), tem-se as pessoas subocupadas por insuficiência de horas trabalhadas. Para serem assim classificadas, essas devem atender, simultaneamente, a quatro condições na semana de referência:

1. Ter 14 anos ou mais de idade.
2. Trabalhar habitualmente menos de 40 horas em todos os seus trabalhos.
3. Desejar trabalhar mais horas que as habitualmente trabalhadas.
4. Estar disponível para trabalhar mais horas em 30 dias, a contar do primeiro dia da semana de referência.

Ainda dentro da força de trabalho, têm-se as Pessoas Desocupadas, que são aquelas com 14 anos ou mais de idade, sem trabalho remunerado na semana de referência, que tomaram alguma providência efetiva para conseguir-no nos últimos 30 dias e que estavam disponíveis para assumir um posto de trabalho na semana da pesquisa. As pessoas sem ocupação que não tomaram providência efetiva para conseguir trabalho porque já haviam conseguido⁹, também, são consideradas Desocupadas.

⁹ O início desse trabalho tem de se dar após a semana de referência e em um prazo de, no máximo, 3 meses.

Os subutilizados que se encontravam fora da força de trabalho¹⁰, mas que possuíam potencial para se tornar força de trabalho, foram denominados Força de Trabalho Potencial (FTP). Assim, a FTP representa o conjunto das pessoas de 14 anos ou mais de idade, não ocupadas ou desocupadas na semana de referência, com potencial para trabalhar, o que contempla:

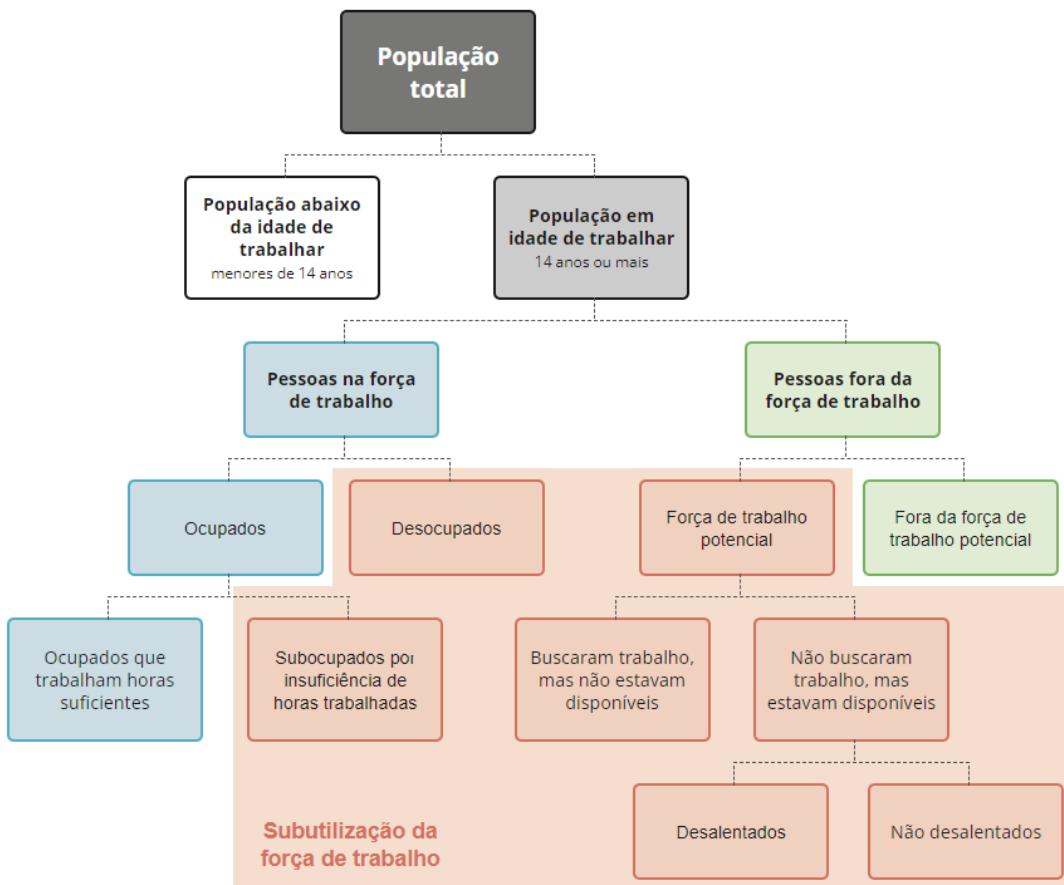
1. As pessoas que buscaram de forma efetiva um trabalho, mas não se encontravam disponíveis para trabalhar.
2. As pessoas que não buscaram trabalho efetivamente, porém gostariam de ter uma ocupação, estando disponíveis para trabalhar.

Por fim, definiu-se o conceito de Força de Trabalho Ampliada, que representa o somatório da Força de Trabalho com a Força de Trabalho Potencial. Isso fica evidente a partir da observação da Figura 3.1, que resume as principais divisões da população e sua condição em relação à força de trabalho segundo os critérios estabelecidos na própria PNAD Contínua.

¹⁰ Como exemplo de pessoas fora da força de trabalho e fora da força de trabalho potencial, tem-se: as pessoas que cuidam da casa, mas que não trabalham fora; adolescentes em idade escolar; aposentados; pessoas que não têm interesse ou condições de trabalhar etc.

POR DENTRO DA PNAD CONTÍNUA

Figura 3.1: Divisões da população e da força de trabalho



Fonte: IBGE (2010).

Cabe destacar, ainda, a existência de um fenômeno que tem ganhado destaque a partir do final da década de 2010, a saber, o desalento¹¹. Essa é uma das categorias que compõem a FTP. Segundo o IBGE (2020a), desalentadas são as pessoas que se encontravam, na semana da pesquisa, fora da força de trabalho, estando disponíveis para assumir um trabalho, porém desestimuladas a tomar providência para conseguir uma posição no mercado de trabalho. Nessa definição, o desestímulo se deve a motivos como: não conseguir trabalho adequado; não ter experiência profissional ou qualificação; inexistência de oportunidades de trabalho na localidade em que reside; ou impossibilidade de conseguir trabalho por ser considerada muito jovem ou muito idosa.

¹¹ Também chamado de desemprego por desalento.

Em termos metodológicos, a análise do desalento na PNAD Contínua pode ser feita a partir de três indicadores:

1. Taxa de desalento na força de trabalho ampliada

$$\text{Taxa de Desalento na} \quad = \frac{\text{Pessoas Desalentadas}}{\text{Força de Trabalho Ampliada}} \quad (3.13)$$

2. Percentual de desalentados na população fora da força de trabalho

$$\text{Proporção de Desalentados na} \quad = \frac{\text{Pessoas Desalentadas}}{\text{População Fora da Força de Trabalho}} \quad (3.14)$$

3. Percentual de desalentados na Força de Trabalho Potencial

$$\text{Proporção de Desalentados na} \quad = \frac{\text{Pessoas Desalentadas}}{\text{Força de Trabalho Potencial}} \quad (3.15)$$

3.3 Renda corrente: tipos de rendimentos na PNAD Contínua

A PNAD Contínua é uma pesquisa que produz informações sobre rendimentos correntes habituais e efetivos do trabalho com periodicidade mensal, para todo o Brasil, e trimestral, para um nível de desagregação que contempla: Unidades da Federação, Regiões Metropolitanas e Municípios das capitais. Outras fontes de renda como, por exemplo, programas sociais, aposentadorias, juros e aluguéis são divulgadas, apenas, anualmente.

Os rendimentos mensais habitualmente recebidos no trabalho principal ou nos demais trabalhos obtidos na semana de referência são os rendimentos mensais que as pessoas estavam acostumadas a receber em um mês completo de trabalho. Já o rendimento efetivo é aquele que, de fato, foi recebido do mês em que a semana de referência estava inserida.

Os rendimentos habituais do trabalho são especialmente úteis para captar informações sobre o padrão de vida do empregador, do tra-

POR DENTRO DA PNAD CONTÍNUA

balhador por conta própria e dos empregados sem carteira assinada cujos rendimentos efetivos variam todo mês e/ou com grande amplitude, podendo, inclusive, ser zero em mais de um mês e depois tomar valores elevados. Isso pode ocorrer especialmente (mas não somente) na agricultura, onde o empreendedor pode trabalhar por meses até a colheita e venda da sua produção. De fato, as flutuações do rendimento dos trabalhadores por conta própria e empregadores são reconhecidas internacionalmente e amplamente reportadas, sendo as dificuldades ainda maiores na agricultura (IBGE, 2021, p. 102-103).

De acordo com o IBGE (2021), na mensuração dos rendimentos correntes brutos do trabalho recebidos por empregados (em dinheiro, produtos ou mercadorias) e por trabalhadores domésticos (em dinheiro), não são computados os valores recebidos na forma de benefícios não monetários como: “moradia, alimentação, roupas, vales alimentação, refeição ou transporte etc.” (IBGE, 2021, p. 44).

O rendimento bruto do trabalho recebido em dinheiro, na PNAD Contínua, assume as seguintes formas:

[...] salário, vencimento, gratificação, ajuda de custo, resarcimento, salário família, anuênio, quinquênio, bonificação, participação nos lucros, horas extras, adicional noturno, adicional de insalubridade, participação anual nos lucros, 13º salário, 14º salário etc.), sem excluir o salário família e nem os pagamentos efetuados por meio administrativo, tais como: contribuição para instituto de previdência, imposto de renda, pensão alimentícia, contribuição sindical, previdência privada, seguro e planos de saúde etc. (IBGE, 2021, p. 44).

Já quanto aos rendimentos nas formas de produtos ou mercadorias, oriundos de atividades produtivas como agricultura, pecuária, caça, silvicultura, exploração florestal, pesca e aquicultura, a contabilização se dá por meio de seus preços de mercado, não sendo considerada a parcela dos bens destinada ao próprio consumo domiciliar.

Para os rendimentos do trabalho de empregadores e de trabalhadores por conta própria, considera-se a renda corrente habitual em dinheiro, nos casos em que

há registro para a pessoa que explora o trabalho em questão, excluindo-se os pagamentos efetuados “por meio administrativo (tal como: contribuição para instituto de previdência, imposto de renda, pensão alimentícia, previdência privada, seguro e planos de saúde etc.)” (IBGE, 2021, p. 44). Esses são valores que podem ser fixos ou variáveis quando assumem a forma de um percentual dos lucros obtido no negócio explorado. No caso dos empreendimentos não organizados ou estruturados, o registro em valor monetário é feito pela diferença entre as receitas do negócio e suas despesas, que incluem: “pagamento de empregados, matéria prima, energia elétrica, telefone, equipamentos e outros investimentos etc.” (IBGE, 2021, p. 44).

Quanto à retirada em produtos ou mercadorias, o valor monetário é definido pelos preços dos produtos retirados nas mesmas atividades produtivas consideradas para os empregados, sendo calculado pela “diferença entre o valor de mercado dos produtos ou mercadorias e as despesas necessárias para a sua produção, excluindo-se a parcela destinada ao próprio consumo do domicílio.” Já, no caso de pessoa licenciada por instituto de previdência, “é o rendimento bruto recebido como benefício (auxílio-doença, auxílio por acidente de trabalho etc.)” (IBGE, 2021, p. 44).

Por fim, cabe destacar que a PNAD Contínua traz informações sobre outras fontes de rendimentos em dinheiro não oriundos do trabalho ou obtidos esporadicamente. Esses são os casos dos prêmios de “loteria, venda de bem móvel ou imóvel, saque do Fundo de Garantia do Tempo de Serviço - FGTS, restituição do imposto de renda, herança, indenização de seguro etc.” (IBGE, 2021, p. 46).

As outras fontes de renda captadas pela pesquisa estão organizadas nas seguintes categorias:

1. Programas Sociais referentes ao Benefício de Prestação Continuada (BPC), à Lei Orgânica da Assistência Social - LOAS, ao Bolsa Família e a outros programas sociais do governo nas esferas federal, estadual ou municipal.
2. Aposentadorias ou pensões de institutos de previdências de qualquer esfera de governo, incluindo os valores recebidos do Fundo de Assistência ao Trabalhador Rural (FUNRURAL), além dos “pagamentos efetuados por meio administrativo, tais como: contribuição para instituto de previdência, imposto de renda, pensão alimentícia, contribuição sindical, empréstimo consignado, seguro e planos de saúde etc.” (IBGE, 2021, p. 46).
3. Seguro-desemprego ou seguro defeso.

POR DENTRO DA PNAD CONTÍNUA

4. Pensão alimentícia, doação ou mesada.
5. Aluguel ou arrendamento, “inclusive sublocação, ou arrendamento de móveis, imóveis, máquinas, equipamentos, animais etc.” (IBGE, 2021, p. 46).
6. Outros rendimentos.

Essa última categoria, isto é, os “Outros rendimentos” contemplam o seguinte:

[...] bolsa de estudo ou programa educacional; caderneta de poupança; aplicações financeiras; complementação ou suplementação de aposentadoria paga por entidades seguradoras ou fundos de pensão; pensão paga por caixa de assistência social, entidade seguradora ou fundo de pensão, na qualidade de beneficiária de outra pessoa; programa social privado; parceria; direitos autorais; exploração de patentes etc. (IBGE, 2021, p. 46).

A combinação das distintas formas de rendimentos recebeu na PNAD Contínua a nomenclatura “Rendimento de Todas as Fontes.” Esse é o valor monetário declarado pelas pessoas de 14 anos ou mais de idade, que compreende “a soma do rendimento mensal habitualmente recebido de todos os trabalhos e do rendimento recebido de outras fontes no mês de referência” (IBGE, 2021, p. 47).

O chamado “Rendimento Domiciliar” nada mais é que a soma dos rendimentos de todas as fontes de todos os moradores do domicílio, excluindo-se as pessoas cuja condição no domicílio eram as de pensionistas, empregados domésticos ou seus familiares. O “Rendimento Domiciliar *Per Capita*” é divisão deste último pelo número total de moradores do domicílio, também, se excluindo as pessoas que não pertencem ao domicílio, seguindo o critério anterior.

É com base nessas informações que se devem ter claras as potencialidades e, também, as limitações dos dados da PNAD Contínua para o tratamento de fenômenos como desigualdade de renda corrente e pobreza monetária no país. É evidente que outras dimensões desses fenômenos podem ser tratadas a partir de distintas informações disponíveis em tal pesquisa. Esse é o caso da dimensão educacional, das características pessoais e das condições habitacionais e do consumo de bens individuais, temas que serão tratados nos próximos capítulos¹².

¹² Sobre o tema Desigualdades e Insuficiência Socieconômica, ver Trovão e Araújo (2021). A metodologia desenvolvida por esses autores será abordada no último capítulo do presente livro.

4 Potencialidades e limitações da PNAD Contínua: a construção de indicadores sociais, de pobreza e de desigualdades

Na segunda metade dos anos 1960, o professor de Administração de Empresas na Escola de Negócios e de Administração de Harvard, Raymond A. Bauer, fez um apelo importante para que se ampliasse o uso de pesquisas por amostragem para coletar informações estatísticas sociais básicas. Para ele, isso era fundamental por permitir que se traçassem tendências e se mensurasse, de forma mais precisa, o progresso social e econômico, no bojo da busca por atender às demandas crescentes da sociedade (BAUER, 1966).

Nas últimas décadas, o estudo e o desenvolvimento de indicadores sociais passou a fazer parte da agenda das políticas públicas em âmbito internacional. Por meio da contribuição de instituições como o Programa das Nações Unidas para o Desenvolvimento (PNUD) e o Banco Mundial, avançou-se significativamente nesse sentido. Esse esforço ganhou um impulso por parte da contribuição da academia, onde pesquisas passaram a ser elaboradas para ampliar o escopo das análises a respeito de fenômenos que se mostram cada vez mais complexos, à medida que o modo de produção capitalista se transforma.

Com isso em mente, procura-se, neste capítulo, apresentar o potencial e, também, as limitações da PNAD Contínua para a construção de indicadores sociais. O objetivo é contribuir para explorar essa importante fonte de dados, dando suporte para futuras pesquisas e estudos que se dediquem ao entendimento da evolução socioeconômica de um país tão heterogêneo e desigual quanto o Brasil.

4.1 Breves notas sobre estatísticas sociais

Segundo Bauer (1966), a importância prática das estatísticas sociais e das pesquisas por amostragens são reconhecidas desde primeira realização dos Censos. Inicialmente, essas pesquisas eram realizadas com propósitos tributários ou para a mensuração do potencial da força militar dos países. No entanto, ao longo do tempo, foram sendo incorporadas questões relevantes para avaliar e orientar as sociedades em transformação. Para esse autor, "Nossa capacidade de planejar com antecedência e avaliar o que fizemos depende de nossa capacidade de avaliar como estamos em relação a como éramos" (BAUER, 1966, p. 339).

Após a Segunda Guerra Mundial, criou-se um sistema de indicadores para avaliar o desempenho econômico e orientar o comando sobre o desempenho da economia. No entanto, estudiosos passaram a considerar que as informações econômicas se mostravam insuficientes para avaliar o desempenho social e a evolução do próprio bem-estar social. A insatisfação das pessoas relacionadas à deficiência de informações sociais se dava, segundo Bauer (1966), por três motivos:

1. A necessidade de aperfeiçoar os indicadores e seus objetos de análise existentes, para evitar conclusões equivocadas a respeito do comportamento de determinadas variáveis.
2. A necessidade de se aprofundar o entendimento de fenômenos que são fonte de preocupação recorrente e que, constantemente, são objeto de avaliações essenciais, mas sobre os quais ou não existem séries históricas sistematicamente coletadas, ou essas são tendenciosas.
3. A existência de aspectos das sociedades que os cientistas sociais consideravam fundamentais, mas que não pareciam receber a atenção devida por parte dos cientistas não sociais.

Segundo o IBGE (2020b), o debate iniciado por Raymond A. Bauer:

[...] foi uma resposta ao momento político específico pelo qual passava os Estados Unidos, marcado, principalmente, pelo crescimento das reivindicações por direitos civis e oposição à participação americana na Guerra do Vietnã. Em um contexto de ampliação do dinamismo econômico, queda do desemprego e introdução

de políticas de proteção social – como as políticas de combate à pobreza – o aumento das tensões sociais desafiava o governo e analistas que tinham nos indicadores econômicos as principais ferramentas para o monitoramento da dinâmica social do país (IBGE, 2020b, p. 7).

Esse debate ganhou fôlego no final do século XX, momento em que cresciam as preocupações da sociedade com a inadequação das pesquisas para a construção de estatísticas e indicadores sociais capazes de quantificar e qualificar a evolução socioeconómica das nações. Boa parte desses indicadores eram orientados, predominantemente, pela ideia de bem-estar enquanto sinônimo de consumo e/ou de acesso à renda monetária, uma visão ancorada na teoria utilitarista e levada a cabo por distintas escolas de pensamento econômico no campo conservador. Nesse sentido, passou-se a estimular o aprimoramento de pesquisas, de métricas de mensuração e da própria análise social.

As recorrentes demandas sociais por direitos e por acesso a bens e serviços que garantem cidadania em uma sociedade cada dia mais complexa e com maiores desafios evidenciaram que as respostas dadas pelos indicadores existentes eram insuficientes. A incorporação de novos indicadores sociais passou a se destacar na pauta das políticas públicas. Segundo o IBGE (2020b), isso garantiu o pontapé inicial para o “movimento de indicadores sociais” nos Estados Unidos. Tal movimento espalhou-se para outros países da Europa e para o Japão, dando inicio à produção de relatórios cujo foco era o monitoramento das condições de vida nos diversos países. Ademais, fez com que os indicadores sociais assumissem uma posição central para o diagnóstico e a implementação de diversas políticas públicas.

Os estudos de Osberg e Sharpe (2002, 2006, 2011) demonstraram a importância de se considerar o bem-estar econômico para além da dimensão da renda corrente. Para esses autores, era extremamente necessária a adoção de critérios que se mostrassem capazes de captar a evolução das condições de distribuição dos recursos econômicos produzidos nas sociedades.

Sua formulação para um indicador de bem-estar complexo e multidimensional parte de quatro dimensões:

1. Fluxos efetivos de consumo *per capita*, que incluem o consumo de bens e serviços comercializados, serviços governamentais, fluxos efetivos de produção familiar *per capita*, lazer e mudanças no ciclo de vida.

POR DENTRO DA PNAD CONTÍNUA

2. Acumulação social líquida de estoques de recursos produtivos, incluindo a acumulação líquida de capital tangível, estoque habitacional, mudanças líquidas no valor dos estoques de recursos naturais, custos ambientais, mudanças líquidas no nível de endividamento externo, acumulação de capital humano e estoque de investimento em pesquisa e desenvolvimento
3. Distribuição de renda, incluindo medidas de intensidade da pobreza (incidência e profundidade) e de desigualdade de renda.
4. Insegurança econômica quanto à perda do emprego, ao desemprego, à doença, à desintegração familiar e à pobreza na velhice.

O debate travado a respeito de temas como bem-estar, pobreza e desigualdade reconheceu o aspecto multidimensional desses fenômenos. Sua incorporação pela agenda das políticas públicas de organismos internacionais ganhou força com as contribuições dos trabalhos de Amartya Sen e Mahbul ul Haq, no contexto da busca pela definição do conceito de desenvolvimento humano no âmbito do PNUD (UNDP, 1994).

Para Sen (1999), não apenas a definição de tais fenômenos como também seu enfrentamento passava pelo entendimento e o fortalecimento das capacidades individuais. Como reflexo, pode-se observar uma crescente utilização de abordagens multidimensionais em pesquisas que exploravam tais fenômenos e que passaram a propor e utilizar múltiplos indicadores sociais. Como destaque especial, tem-se o conjunto de estudos desenvolvidos pelo PNUD, após os anos 1990, que consolidaram distintas metodologias para a mensuração do desenvolvimento humano no âmbito das ações e dos estudos levados a cabo pela Organização das Nações Unidas (ONU)¹.

As privações ou insuficiências socioeconômicas passaram a assumir um papel relevante no campo acadêmico e das políticas públicas pensadas para lidar com esses fenômenos. Os diferentes níveis de acesso aos diversos elementos fundamentais para a manutenção da vida, em todas as dimensões que conformam as desigualdades, a pobreza e o desenvolvimento humano, mostram-se fundamentais para a construção de indicadores adequados a sua mensuração. Foi nesse sentido que o debate sobre indicadores socioeconômicos evoluiu, tomando uma direção no sentido do aprofundamento das concepções metodológicas para além da renda. Isso foi necessário, pois

¹ A esse respeito, ver Trovão (2015, cap. 1).

a materialização desses fenômenos se dá em distintas esferas da vida em sociedade como o bem-estar, a liberdade, a qualidade de vida, a saúde e a longevidade, a participação política, o lazer, a segurança, as oportunidades etc.

No Brasil, o IBGE iniciou a produção de indicadores sociais a partir do início da década de 1970, mais precisamente em 1973, por meio do Grupo Projeto de Indicadores Sociais (GPIS). O objetivo era sistematizar a produção de estatísticas sociais no Brasil. Mais que isso, era produzir indicadores sociais que levassem em consideração as relações sistêmicas entre as distintas dimensões relevantes para a análise social a partir de uma perspectiva histórica.

No final dos anos 1990, o IBGE publicou a primeira edição da chamada Síntese de Indicadores Sociais, com objetivo de apresentar um quadro sintético das condições sociais da população brasileira. De modo mais amplo, essa proposta procurava orientar a formulação de políticas públicas nos níveis dos governos federal, estaduais e municipais.

Essa Síntese de Indicadores Sociais (SIS) surgiu em um contexto de crescente preocupação com a promoção de políticas públicas de combate à pobreza e às desigualdades. O esforço do IBGE contribuiu para aprimorar e aprofundar o monitoramento não apenas das condições de vida no país, mas também das próprias políticas de educação, de mercado de trabalho, de distribuição de renda, de habitação, de saúde etc.

Nesse mesmo sentido, tal iniciativa contribuiu para o monitoramento da evolução de distintos grupos sociais, especialmente, daqueles mais expostos às vicissitudes econômicas, isto é, os mais vulneráveis (mulheres, crianças, pretos ou pardos etc.).

Em sua trajetória, a SIS passou a incorporar distintos temas relevantes para a análise socioeconômica bem como novos indicadores, o que contribuiu para a evolução teórica e prática no campo metodológico dos estudos sobre as principais questões sociais do país.

Apesar de não desprezar a importância de outras fontes de informação, as PNADs e, após 2012, a PNAD Contínua mantiveram-se como as principais fontes de informação para construção de indicadores sociais no Brasil, especialmente, nos intervalos de realização dos Censos Demográficos.

Para retratar a estrutura social brasileira, o IBGE, em sua Síntese de Indicadores Sociais, passou a retratar três dimensões:

POR DENTRO DA PNAD CONTÍNUA

1. Estrutura econômica e mercado de trabalho.
2. Padrão de vida e distribuição de renda.
3. Educação.

A relevância da utilização da PNAD para esse fim está em sua capacidade de proporcionar a construção de uma ampla gama de indicadores, a partir de distintos recortes analíticos. Esses são os casos das dimensões territoriais, cuja desagregação permite ao pesquisador descer ao nível das capitais das unidades da federação na PNAD Contínua e ao das dimensões individuais de cor/raça, de gênero, de idade e de nível educacional. Nesse sentido, essa pesquisa permite que se avaliem e se definam distintas formas de agrupamentos populacionais definidos a partir do interesse e do objeto de pesquisa. Outros exemplos de recortes analíticos guardam relação com a posição das pessoas em relação à força de trabalho, à ocupação ou, ainda, com a situação dos domicílios, isto é, a localidade da moradia (rural/urbana).

Nas dimensões estrutura econômica e mercado de trabalho, o IBGE (2020b) destaca a possibilidade de se analisar:

1. A dinâmica do mercado de trabalho.
2. As desigualdades estruturais quanto às características de inserção dos trabalhadores.
3. As relações contratuais de trabalho (informalidade/formalidade e/ou trabalho intermitente).
4. O desemprego e a subutilização da força de trabalho.
5. As condições dos mais vulneráveis (pretos ou pardos, mulheres e jovens etc.).

Na esfera da análise do padrão de vida e da distribuição de renda, o IBGE (2020b) destaca a potencialidade dessa fonte de informação para conceber indicadores relacionados:

1. À distribuição do rendimento do trabalho e de outras fontes.
2. Ao acesso a bens e serviços de uso coletivo (condições habitacionais).

3. À pobreza monetária e multidimensional.
4. À posse de bens de uso individual.

Na dimensão educacional, a PNAD Contínua permite a construção de indicadores como:

1. Frequência escolar.
2. Proporção da população com acesso à rede pública e privada de ensino.
3. Nível de instrução.
4. Taxa de analfabetismo.

No entanto, deve-se atentar para o fato de que “[...] enquanto algumas informações passaram a ser captadas em todos os trimestres – caso do bloco de trabalho – outras se restringiram a trimestres específicos – bloco de educação – ou em entrevistas ao longo do ano – casos de habitação e rendimentos de outras fontes”. Isso faz com que parte das informações só esteja disponível em periodicidade anual (IBGE, 2020b, p. 9).

4.2 Por dentro do dicionário da PNAD Contínua: potencialidades e limites

Os microdados da PNAD Contínua são disponibilizados de duas formas: trimestral e anual. Em ambas, o IBGE permite a desagregação territorial das informações segundo Unidades da Federação (UF), Municípios da capital e Regiões Metropolitanas (RM) ou para a Região Administrativa Integrada de Desenvolvimento (RID) de Teresina. Além disso, como mencionado anteriormente, do ponto de vista territorial, é possível desagregar os microdados por Situação do Domicílio.

Na divulgação trimestral, pode-se trabalhar com um conjunto expressivo de informações sobre:

1. Características gerais dos moradores que contemplam, dentre outras informações:

POR DENTRO DA PNAD CONTÍNUA

- número de pessoas no domicílio;
- condição no domicílio (pessoa responsável pelo domicílio, cônjuge ou companheiro(a), filho(a), genro ou nora, pai, mãe, padrasto ou madrasta, irmão ou irmã, avós, pensionista, empregado(a) doméstico(a), parente do(a) empregado(a) doméstico(a) etc.);
- sexo (homem ou mulher);
- idade do morador na data de referência;
- cor ou raça (branca, preta, amarela, parda, indígena).

2. Características de educação para os moradores de 5 anos ou mais de idade englobando, dentre outros elementos:

- saber ler e escrever;
- frequentar escola, se a escola que frequenta é da rede privada ou pública;
- qual o curso que frequenta (pré-escola, alfabetização de jovens e adultos, ensino fundamental, ensino médio, superior – graduação, mestrado, doutorado);
- qual é o ano/série/semestre que frequenta;
- qual foi o curso mais elevado que frequentou anteriormente;
- qual foi o último ano/série/semestre que concluiu com aprovação, no curso que frequentou anteriormente.

3. Características de trabalho das pessoas de 14 anos ou mais de idade, que incluem informações como:

- trabalhou ou estagiou, durante pelo menos 1 hora, em alguma atividade remunerada em dinheiro ou em produtos, mercadorias, moradia, alimentação etc. na semana de referência;
- fez algum bico ou trabalhou em alguma atividade ocasional remunerada durante pelo menos 1 hora;
- ajudou durante pelo menos 1 hora, sem receber pagamento, no trabalho remunerado de algum morador do domicílio ou de parente;

- tinha algum trabalho remunerado do qual estava temporariamente afastado;
- o motivo que levou ao afastamento desse trabalho (Férias, folga ou jornada de trabalho variável, Licença maternidade, Licença remunerada por motivo de doença ou acidente da própria pessoa, Outro tipo de licença remunerada por estudo, paternidade, casamento, licença prêmio etc., Afastamento do próprio negócio/empresa por motivo de gestação, doença, acidente etc., sem ser remunerado por instituto de previdência, Fatores ocasionais por tempo, paralisação nos serviços de transportes etc., Greve ou paralisação);
- durante o tempo de afastamento, continuou a receber ao menos uma parte do pagamento; por quanto tempo estava afastado do trabalho (de 1 mês a menos de 1 ano, de 1 ano a menos de 2 anos, mais de 2 anos).

4. Características ocupacionais, por meio de dados que explicitam:

- quantos trabalhos tinha na semana de referência;
- qual a ocupação (cargo ou função) baseada na Classificação de Ocupações para as Pesquisas Domiciliares (COD);
- se nesse trabalho era Trabalhador doméstico, Militar do exército, da marinha, da aeronáutica, da polícia militar ou do corpo de bombeiros militar, Empregado do setor privado, Empregado do setor público (inclusive empresas de economia mista), Empregador, Conta própria ou Trabalhador familiar não remunerado;
- qual a atividade dessa ocupação, baseada na Relação de Códigos de Atividades da CNAE-Domiciliar;
- com quantos empregados trabalhava nesse negócio/empresa;
- se o negócio/empresa era registrado no Cadastro Nacional da Pessoa Jurídica - CNPJ;
- que tipo de local funcionava esse negócio/empresa (loja, escritório, galpão etc., fazenda, sítio, granja, chácara etc., ou não tinha estabelecimento para funcionar;

POR DENTRO DA PNAD CONTÍNUA

- onde exercia normalmente esse trabalho (estabelecimento de outro negócio/empresa, local designado pelo empregador, cliente ou freguês, domicílio de empregador, patrão, sócio ou freguês, domicílio de residência, em local exclusivo para o desempenho da atividade, domicílio de residência, sem local exclusivo para o desempenho da atividade, em veículo automotor (táxi, ônibus, caminhão, automóvel, embarcação etc.), em via ou área pública (rua, rio, manguezal, mata pública, praça, praia etc.) ou outro local;
- se era contratado(a) como empregado temporário;
- se era servidor público estatutário (federal, estadual ou municipal);
- se tinha carteira de trabalho assinada nessa ocupação;
- se contribuía para instituto de previdência;
- quantas horas trabalhava normalmente e efetivamente em todos os trabalhos que possuía;
- se gostaria de trabalhar mais horas do que as horas que efetivamente trabalhou no(s) trabalho(s) que tinha na semana de referência;²
- se estaria disponível para trabalhar mais do que as horas que efetivamente trabalhou no(s) trabalho(s) que tinha;

5. Características daqueles sem ocupação, incluindo, dentre outras informações:

- se tomou alguma providência para conseguir trabalho, seja um emprego ou um negócio próprio;
- qual foi a principal providência que tomou para conseguir trabalho;
- para quem não tomou providência para conseguir trabalho, se gostaria de ter trabalhado na semana de referência;
- o principal motivo para não ter tomado providência para conseguir trabalho (Conseguiu proposta de trabalho para começar após a semana de referência, estava aguardando resposta de medida tomada para conseguir trabalho, desistiu de procurar por não conseguir encontrar trabalho, achava que não iria encontrar trabalho por ser muito jovem ou muito

² Questão que define a condição de subocupação em caso de resposta afirmativa.

idoso, tinha que cuidar de filho(s), de outro(s) dependente(s) ou dos afazeres domésticos, estudo, incapacidade física, mental ou doença permanente, outro motivo);

- até o dia da pesquisa, por quanto tempo estava sem qualquer trabalho e tentando conseguir trabalho (Menos de 1 mês, De 1 mês a menos de 1 ano, De 1 ano a menos de 2 anos, 2 anos ou mais);
- por quanto tempo vinha procurando trabalho.

6. Registro sobre rendimentos de pessoas ocupadas, informando:

- o tipo de remuneração habitualmente recebida no trabalho principal e em todos os trabalhos para pessoas de 14 anos ou mais de idade;
- o valor em R\$ do rendimento mensal habitual e efetivo do trabalho principal e de todos os trabalhos para pessoas de 14 anos ou mais de idade (apenas para pessoas que receberam em dinheiro, produtos ou mercadorias no trabalho principal).

Para além desse conjunto de informações de periodicidade trimestral, a divulgação anual da PNAD Contínua traz alguns dados adicionais sobre:

1. Características da habitação e de acesso à Tecnologia de Informação e Comunicação (TIC), que agregam informações sobre:

- o tipo de domicílio (casa, apartamento, habitação em casa de cômodos, cortiço ou cabeça de porco);
- o material que predomina na construção das paredes externas deste domicílio (alvenaria com revestimento/taipa com revestimento, alvenaria sem revestimento, taipa sem revestimento, madeira apropriada para construção (aparelhada), madeira aproveitada, outro material);
- o material que predomina na cobertura (telhado) deste domicílio (telha sem laje de concreto, telha com laje de concreto, somente laje de concreto, madeira apropriada para construção, zinco, alumínio ou chapa metálica, outro material);

POR DENTRO DA PNAD CONTÍNUA

- o material que predomina no piso deste domicílio (cerâmica, lajota ou pedra, madeira apropriada para construção, cimento, terra, outro material);
- número de cômodos no domicílio;
- o número de cômodos servindo permanentemente como dormitório para os moradores deste domicílio;
- a principal forma de abastecimento de água utilizada neste domicílio (rede geral de distribuição, poço profundo ou artesiano, poço raso, freático ou cacimba, fonte ou nascente, água da chuva armazenada, outra);
- a frequência com que a água proveniente de rede geral esteve disponível para este domicílio (diariamente, de 4 a 6 dias na semana, de 1 a 3 dias na semana, outra);
- se faz uso de reservatório, caixa d'água, cisterna, para armazenar a água;
- se a água utilizada neste domicílio chega canalizada ou não é canalizada;
- o número de banheiros (com chuveiro ou banheira e vaso sanitário ou privada) de uso exclusivo dos moradores existentes neste domicílio, inclusive os localizados no terreno ou na propriedade;
- a utilização de sanitário ou buraco para dejeções, inclusive os localizados no terreno ou na propriedade (cercado por paredes de qualquer material);
- o destino do esgoto do banheiro (rede geral, rede pluvial, fossa séptica ligada à rede, fossa séptica não ligada à rede, fossa rudimentar, vala, rio, lago ou mar);
- o destino dado ao lixo (coletado diretamente por serviço de limpeza, coletado em caçamba de serviço de limpeza, queimado (na propriedade), enterrado (na propriedade), jogado em terreno baldio ou logradouro, outro destino);
- se utiliza energia elétrica e qual(is) sua(s) origem(ns) (rede geral, gerador, placa solar, eólica);
- a frequência habitual da energia elétrica proveniente de rede geral;
- o combustível utilizado para a preparação dos alimentos (não utiliza combustível/não prepara alimentos, gás de botijão, gás encanado, lenha ou carvão, energia elétrica, outro combustível);

- a propriedade do domicílio (próprio de algum morador já pago, próprio de algum morador ainda pagando, alugado, cedido por empregador, cedido por familiar, cedido de outra forma, outra condição);
- o valor mensal em R\$ do aluguel ou da prestação paga, ou que deveria ter sido paga, no mês de referência;
- a posse de bens de uso individual ou por membros do domicílio como: telefone móvel celular para uso pessoal, telefone fixo convencional, geladeira, máquina de lavar roupa, televisão, serviço de televisão por assinatura, televisão com antena parabólica, microcomputador (inclusive *laptop, notebook, ultrabook ou netbook*);
- o acesso à internet no domicílio por meio de microcomputador, *tablet*, telefone móvel celular, televisão ou outro equipamento, acesso à internet neste domicílio e por qual meio (microcomputador de mesa ou portátil, *tablet*, telefone móvel celular, televisão, outro equipamento eletrônico);
- a posse de automóvel ou motocicleta de uso pessoal.

2. Registro sobre rendimentos efetivamente recebidos de outras fontes, proporcionando informações sobre:

- se recebeu e o valor em R\$ dos rendimentos oriundos de programas sociais (Benefício Assistencial de Prestação Continuada – BPC-LOAS, Programa Bolsa Família, outros programas sociais do governo);
- se recebeu e o valor em R\$ dos rendimentos de aposentadoria ou pensão de instituto de previdência federal (INSS), estadual, municipal ou do governo federal, estadual, municipal;
- se recebeu e o valor em R\$ dos rendimentos de seguro-desemprego, seguro defeso;
- se recebeu e o valor em R\$ dos rendimentos nas formas pensão alimentícia, doação ou mesada em dinheiro de pessoa que não morava no domicílio;
- se recebeu e o valor em R\$ dos rendimentos de aluguel ou arrendamento;
- se recebeu e o valor em R\$ de outros rendimentos (bolsa de estudos, rendimento de caderneta de poupança, aplicações financeiras etc.).

POR DENTRO DA PNAD CONTÍNUA

Por fim, cabe destacar que os microdados anuais da PNAD Contínua possuem, também, um conjunto de variáveis derivadas que ampliam e/ou sintetizam informações assumindo formas como, por exemplo:

- rendimento recebido em todas as fontes (habitual de todos os trabalhos e efetivo de outras fontes, apenas para pessoas que receberam em dinheiro, produtos ou mercadorias em qualquer trabalho ou que receberam rendimentos em dinheiro de outras fontes);
- rendimento efetivo domiciliar (exclusive rendimentos em cartão/tíquete transporte ou alimentação e exclusive o rendimento das pessoas cuja condição na unidade domiciliar era pensionista, empregado doméstico ou parente do empregado doméstico);
- rendimento efetivo domiciliar *per capita*;
- rendimento domiciliar (habitual de todos os trabalhos e efetivo de outras fontes);
- rendimento domiciliar *per capita* (habitual de todos os trabalhos e efetivo de outras fontes).

Após essa imersão nos dicionários da PNAD Contínua, apresentam-se os passos iniciais e o primeiro contato com uma das mais utilizadas linguagens de programação, em distintas áreas do conhecimento, da atualidade, o R.

5 Uma introdução ao ambiente e à linguagem de programação R

A linguagem de programação R foi desenvolvida na Universidade de Auckland, por Ross Ihaka¹ e Robert Gentleman². Foi chamada de R em virtude das iniciais de seus criadores Ross e Robert. Atualmente, o projeto é mantido e desenvolvido pelo *R Development Core Team*. Ele foi iniciado em 1992, e a versão inicial foi lançada em 1995. No ano de 2000, a versão beta estável 1.0.0 foi disponibilizada ao público.

O R é uma implementação do S, outra linguagem de programação estatística desenvolvida no *Bell Labs*, que remonta aos anos 1970. Após o lançamento inicial do R, muitas pessoas decidiram adotá-lo e trabalhar em seu código para melhorar seus recursos.

Em 1995, tornou-se uma linguagem de código aberto, sendo permitido a qualquer pessoa modificá-la e aprimorá-la. Isso aconteceu porque Martin Mächler convenceu os criadores da linguagem a usar a licença GNU³ para tornar o R livre.

Em meados de 1997, uma pequena equipe chamada R Core Team foi desenvolvida para modificar o código-fonte do R, que está em operação até hoje.

Várias iniciativas surgiram em torno do R. Em 2007, uma empresa chamada Revolution Computing (hoje conhecida como Revolution Analytics, mais tarde vendida para a Microsoft), foi fundada para fornecer suporte e extensões para o R, mais especificamente, para *big data*. Em 2011, uma empresa chamada RStudio disponibi-

¹ Ver <http://www.stat.auckland.ac.nz/~ihaka/>.

² Ver <http://gentleman.fhcrc.org/>.

³ O sistema operacional GNU é um sistema de *software* livre compatível com o Unix. Vale destacar que GNU significa “GNU’s Not Unix”.

POR DENTRO DA PNAD CONTÍNUA

lizou ao público um produto de mesmo nome.

O *RStudio* é um ambiente de desenvolvimento integrado para o R, fornecendo outras ferramentas em linguagem R). Tanto o *RStudio* quanto o que costumava ser o **Revolution R Open**, agora **Microsoft R Open**, são *softwares* livres.

5.1 Por que usar o R?

O R é distribuído sob os termos da Licença Pública Geral (*General Public License – GPL*) GNU, nas versões 2 e 3. Sua implicação é que, em termos gerais, a GPL baseia-se em quatro liberdades:

1. Executar o programa para qualquer propósito (liberdade nº 0).
2. Estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade nº 1). O acesso ao código-fonte é um pré-requisito para esta liberdade.
3. Redistribuir cópias de modo a ajudar outros usuários (liberdade nº 2).
4. Aperfeiçoar o programa e liberar seus aperfeiçoamentos, de modo que toda a comunidade se beneficie deles (liberdade nº 3). O acesso ao código-fonte é um pré-requisito para esta liberdade.

Para saber mais sobre licenças “oficialmente autorizadas” para o R, basta acessar o seguinte endereço, no *site* do *R-Project*: <https://www.r-project.org/Licenses/>.

5.2 RStudio

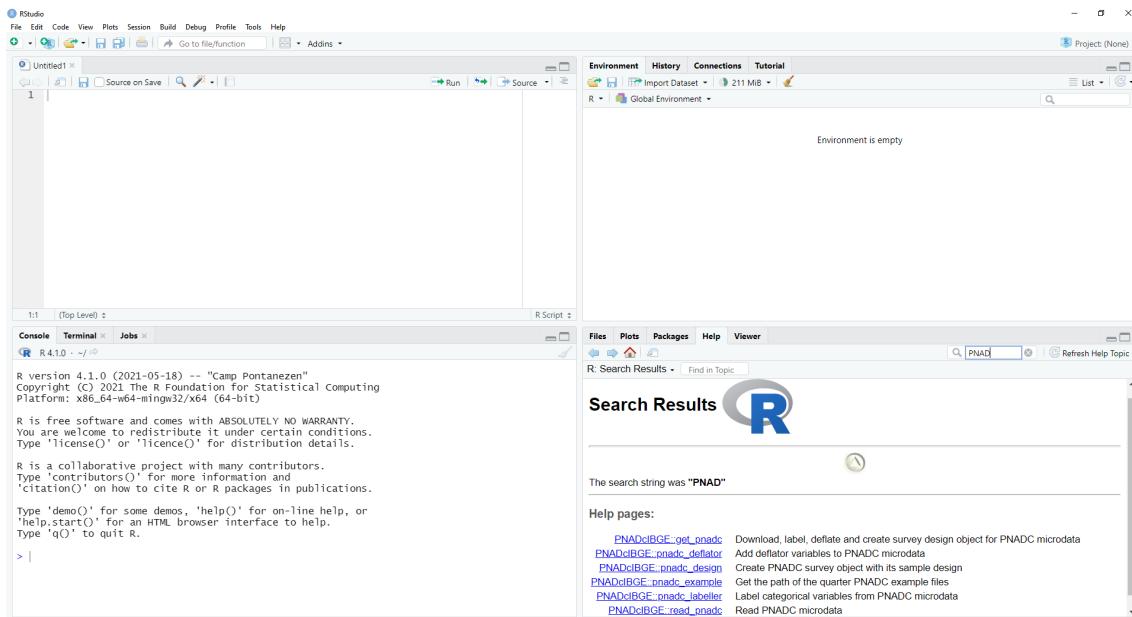
Trabalhar com linguagens de programação requer ferramentas apropriadas para facilitar o processo, a saber, editores de texto, gerenciadores de arquivos, navegadores da WEB etc. O *RStudio* (RStudio Team, 2021) é um IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) que aprimora o R padrão, adicionando diversas funcionalidades, tais como:

- um poderoso editor de texto com realce de sintaxe, complementação automática de parênteses e teclas de atalho;

- um navegador de espaços de trabalho;
- um visualizador de dados;
- um visualizador de histórico de comandos;
- um gerenciador de arquivos;
- um gerenciador de pacotes;
- uma guia de gráficos;
- uma guia de ajuda integrada.

Tudo isso é apresentado ao usuário de forma compacta e intuitiva em apenas uma tela personalizável (ver Figura 5.1). Para baixar o RStudio, basta acessar o site <https://www.rstudio.com/products/rstudio/> e escolher a opção *Download RStudio Desktop*, versão *Free*.

Figura 5.1: Tela inicial do R “dentro” do *RStudio*.



Fonte: Elaborada pelo autor.

O RStudio é organizado em quatro painéis. Por padrão, o painel superior esquerdo é o editor de *Scripts*, onde se visualiza e edita o código-fonte dos programas

POR DENTRO DA PNAD CONTÍNUA

em R. O painel inferior esquerdo é geralmente o painel do console do R, no qual se pode digitar comandos e visualizar as mensagens de saída. Os painéis à direita têm várias guias diferentes que mostram informações sobre seu código. Pode-se alterar a localização dos painéis e as guias em que são exibidas as informações por meio do seguinte caminho no sistema operacional Windows: *Tools > Global Options > Pane Layout*⁴.

5.3 Primeiros passos: *download* e instalação

5.3.1 O *software R*

Para fazer o *download* do R, basta acessar <http://www.r-project.org> e seguir os seguintes passos:

- Clique na opção ‘CRAN’ (menu do lado esquerdo, abaixo da palavra ‘Download’).
- Escolha um dos espelhos (na dúvida, escolha o primeiro – ‘Cloud 0’).
- Clique em ‘Download R for Windows’ (caso esse seja o sistema operacional em uso).
- Depois, clique em ‘base’.
- Por fim, clique em ‘Download R X.X.X for Windows’ (em que X.X.X corresponde à versão mais atual do R compatível com o sistema operacional em uso).
- Após o *download*, vá até a pasta de destino e execute o instalador do R e clique em ‘Avançar’ quando necessário.

5.3.2 O *software RStudio*

O próximo passo é instalar o RStudio.

⁴ Ou o caminho *Preferences > Pane Layout* em um Mac.

- Acesse o site <https://rstudio.com/products/rstudio/download/#download>.
- Clique em ‘Download RStudio for Windows’ e faça o *download* do arquivo *.exe.
- Vá até a pasta de destino do *download*, execute o instalador do RStudio e clique em ‘Avançar’ quando necessário.

5.4 Citação

Para citar o R em algum trabalho acadêmico, pode-se recorrer a uma função muito útil chamada `citation()`, que pode ser executada no editor de texto (*Script*) ou no próprio console.

```
citation()
```

To cite R in publications use:

```
R Core Team (2021). R: A language and environment for
statistical computing. R Foundation for Statistical
Computing, Vienna, Austria. URL
https://www.R-project.org/.
```

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {R: A Language and Environment for Statistical
            Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2021},
```

POR DENTRO DA PNAD CONTÍNUA

```
url = {https://www.R-project.org/},  
}
```

We have invested a lot of time and effort in creating R,
please cite it when using it for data analysis. See also
'citation(''pkgname'')' for citing R packages.

Para citar qualquer um dos seus muitos pacotes, basta informar o nome do pacote como argumento da função `citation()`, como no seguinte exemplo:

```
citation(package = "PNADcIBGE")
```

To cite package 'PNADcIBGE' in publications use:

Douglas Braga and Gabriel Assuncao (2021). PNADcIBGE:
Downloading, Reading and Analysing PNADC Microdata. R
package version 0.6.4.

<https://CRAN.R-project.org/package=PNADcIBGE>

A BibTeX entry for LaTeX users is

```
@Manual{,  
  title = {PNADcIBGE: Downloading, Reading and Analysing  
    PNADC Microdata},  
  author = {Douglas Braga and Gabriel Assuncao},  
  year = {2021},  
  note = {R package version 0.6.4},  
  url = {https://CRAN.R-project.org/package=PNADcIBGE},  
}
```

5.5 Exercícios

1. Tente calcular a soma de 2 mais 2 no R.
2. Tente calcular a raiz quadrada de 10 no R.
3. O R possui uma extensa documentação, incluindo um manual denominado *An Introduction to R*. Use o painel de ajuda do RStudio para localizar esse manual⁵.

5.6 O ambiente R

O R oferece um conjunto de recursos de programação integrados para o tratamento de dados, cálculo e exibição gráfica. Entre outras funcionalidades, destacam-se:

- instruções para o tratamento e o armazenamento de dados;
- um conjunto de operadores simples para cálculos matriciais;
- uma expressiva coleção de ferramentas integradas para análise de dados;
- recursos gráficos para análise e exibição de dados diretamente na tela do computador ou para inclusão em documentos de texto;
- uma linguagem de programação bem desenvolvida, simples e eficaz (chamada S) que inclui condicionais, *loops*, funções recursivas definidas pelo usuário e recursos de entrada e saída⁶.

O termo “ambiente” tem a intenção de caracterizá-lo como um sistema totalmente planejado e consistente, em vez de um *software* que requer o acréscimo incremental de ferramentas muito específicas e inflexíveis, como é comum em outros *softwares* de análise de dados.

⁵ Ver Venables *et al.* (2009).

⁶ A maioria das funções fornecidas pelo sistema são escritas na linguagem S.

POR DENTRO DA PNAD CONTÍNUA

O R é um veículo para novos métodos de análise de dados em desenvolvimento. Ele se desenvolveu rapidamente e foi ampliado por uma significativa coleção de pacotes. No entanto, a maioria dos programas escritos em R são essencialmente efêmeros, escritos para uma única parte da análise de dados.

5.7 A área de trabalho e seus objetos

Os comandos a seguir listam os objetos armazenados na memória.

```
ls()  
objects()
```

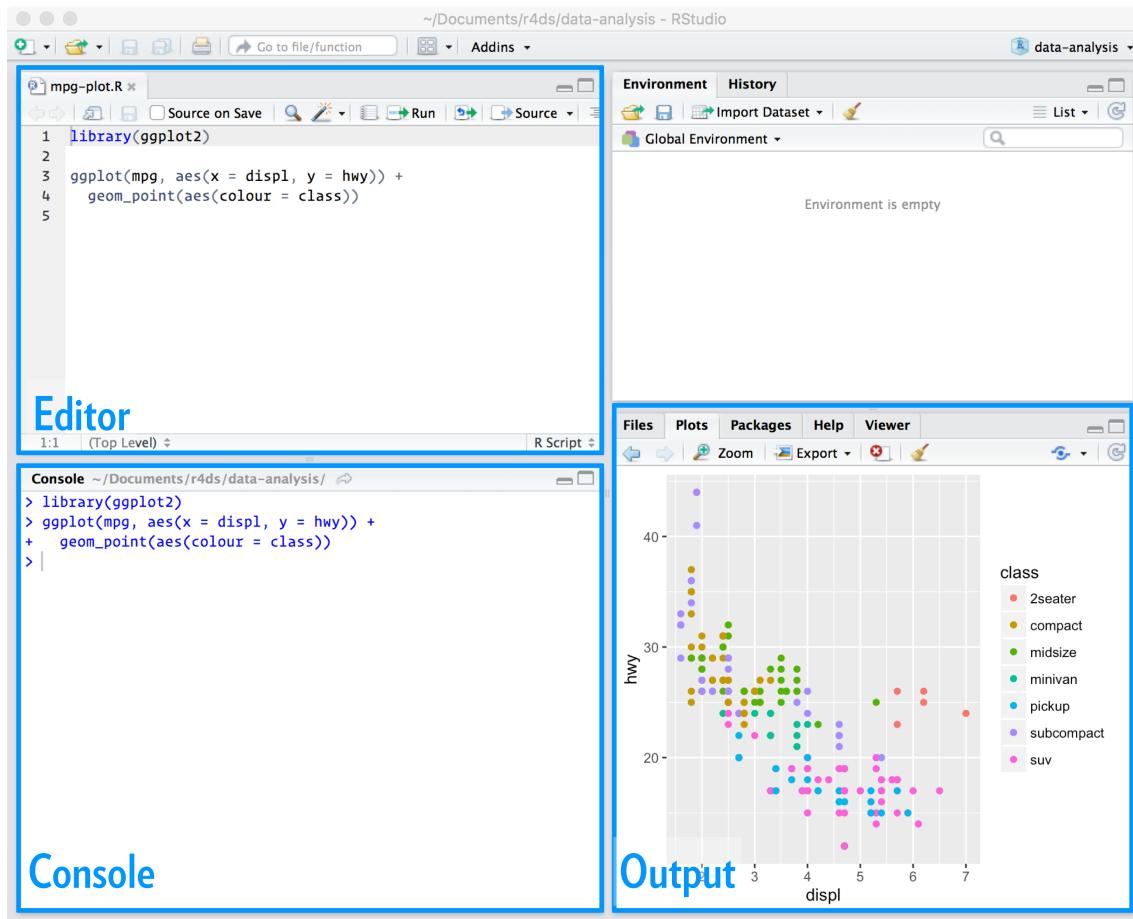
Para remover todos os objetos da memória, pode-se usar a seguinte sintaxe:

```
rm(list = ls())
```

5.8 O editor de **Scripts** do RStudio

Até agora, provavelmente todos os comandos foram aplicados diretamente no console. Esse é um ótimo lugar para começar, mas percebe-se uma relativa limitação na medida em que se torna necessário produzir gráficos e rotinas mais complexas. Para ter mais espaço para trabalhar, uma ótima ideia é usar o editor de *Scripts*. Para abri-lo, basta clicar no menu ‘Arquivo’, selecione ‘Novo arquivo’ e, em seguida, *Script R*. Pode-se usar, também, o atalho do teclado Cmd/Ctrl + Shift + N. Com isso, podem ser vistos quatro painéis (ver Figura 5.2).

Figura 5.2: Identificação dos painéis do R “dentro” do *RStudio*.



Fonte: Elaborada pelo autor.

O editor de *Scripts* é um ótimo espaço para digitar o código que se deseja aplicar. Pode continuar utilizando o console, porém depois de escrevê-lo e de verificar que este funciona, recomenda-se que se copie e cole a rotina aplicada no editor de *Scripts*. O *RStudio* salva automaticamente o conteúdo do editor quando se sai do programa. Ao carregar o programa novamente, o *Script* é retomado automaticamente. No entanto, é uma boa ideia salvar seus *Scripts* regularmente além de fazer o seu *backup*.

A chave para usar o editor de *Scripts* de forma eficaz é memorizar um dos atalhos de teclado mais importantes: Cmd/Ctrl+Enter. Esse atalho permite executar o código no console.

POR DENTRO DA PNAD CONTÍNUA

Exemplo:

- Selecione o código apresentado abaixo;
- Se o seu cursor estiver em `,`, por exemplo, pressione Cmd/Ctrl+Enter;
- O R executará o comando completo que gera o objeto `not_cancelled`;
- O R também moverá o cursor para a próxima instrução (começando com `not_cancelled |>`);
- Isso facilita a execução de seu *Script* completo pressionando repetidamente Cmd/Ctrl+Enter.

```
library(dplyr)
library(nycflights13)

not_cancelled <- flights |>
  filter(!is.na(dep_delay), !is.na(arr_delay))

not_cancelled |>
  group_by(year, month, day) |>
  summarise(mean = mean(dep_delay), .groups = "drop")
```

Em vez de executar expressão por expressão, pode-se executar o *Script* completo de uma única vez: Cmd/Ctrl+Shift+S. Fazer isso regularmente é uma ótima maneira de verificar se todas as partes importantes do código foram executadas.

Recomenda-se que sempre se inicie o *Script* com os pacotes necessários. Dessa forma, se a intenção for compartilhar o código com outras pessoas, elas poderão ver facilmente quais pacotes precisam instalar.

De agora em diante, recomenda-se fortemente começar pelo editor e praticar seus atalhos de teclado. Com o passar do tempo e a prática, enviar qualquer código para o console dessa forma se tornará tão natural, a ponto de ser imperceptível.

5.9 Pacotes do R

Existem **muitos** pacotes (também conhecidos como **bibliotecas**) para o R. Boa parte deles está disponível no repositório oficial do R, conhecido como Comprehensive R Archive Network (CRAN)⁷. No entanto, existem muitos outros pacotes disponíveis em outras fontes.

Para verificar os pacotes instalados no R, basta que se execute o seguinte comando:

```
.packages(all.available = TRUE)
```

Para carregar um pacote, basta utilizar os comandos `library()` ou `require()` (basta um). Se o pacote não estiver instalado, o comando retornará uma mensagem de erro.

```
library(NomeDoPacote)
```

```
# Error in library(NomeDoPacote) :  
#   there is no package called 'NomeDoPacote'
```

Para instalar o pacote que se deseja, pode-se utilizar o comando `install.packages()`. Deve-se observar que o nome do pacote deve vir entre aspas:

```
install.packages("NomeDoPacote") # Obviamente NomeDoPacote  
# não existe no CRAN!
```

Para acessar funções específicas, não disponíveis na instalação básica do R, é necessário **instalar** um pacote/biblioteca. Além disso, faz-se necessário **carregar**

⁷ Ver <https://cran.r-project.org/>

POR DENTRO DA PNAD CONTÍNUA

o pacote desejado a cada sessão R. Depois de carregado, passa-se a ter acesso a tudo o que esse oferece (funções, conjuntos de dados etc.).

5.10 Exercícios

1. Acesse a conta do Twitter do *RStudio Tips*, endereço <https://twitter.com/rstudiotips> e encontre dicas que pareçam interessantes. Pratique como usá-las!
2. Que outros erros comuns o diagnóstico do RStudio pode relatar? Leia as dicas e informações no endereço <https://support.rstudio.com/hc/en-us/articles/205753617-Code-Diagnostics> para descobrir.
3. Obtenha a ajuda para a função `q`. Qual o propósito dessa função?
4. Abra a ajuda da função `detach`. Identifique e informe um exemplo de como utilizá-la para fazer o inverso da função `library`.
5. Utilize a função `help.search` para descobrir como remover um ou mais pacotes instalados.
6. Possivelmente, haverá um erro no primeiro exemplo da Seção 2. Instale o pacote `dplyr` e execute novamente o código daquele exemplo.

5.11 Comandos básicos

5.11.1 Estilo de programação

Muitas linguagens de computador – principalmente as modernas, incluindo o R – permitem uma grande flexibilidade na maneira como se escreve um código. Porém, isso pode tornar difícil para um programador ler o código escrito por outro. Para resolver esse problema, muitas organizações têm “guias de estilo” para as linguagens que seus programadores usam.

Existem dois guias de estilos para o R que são razoavelmente bem aceitos entre os usuários mais experientes. O primeiro é disponibilizado pelo Google⁸ e o último é uma modificação deste realizada por Hadley Wickham⁹, autor dos pacotes **ggplot2**¹⁰, **dplyr**¹¹ e muitos outros¹², além de cientista-chefe do RStudio¹³. No presente livro, procura-se seguir esses guias na medida do possível.

O R não necessita de demarcador de fim de comando (como, por exemplo, ;). Caso a linha de programação esteja sintaticamente completa, o interpretador do R irá executar o comando. Caso contrário, o R irá continuar na próxima linha.

Exemplo:

```
2 + 2
2 +
2
(2
+ 2)
```

Contudo, é possível usar o ponto e vírgula com outra finalidade. O símbolo ';' , no R, serve para colocar vários comandos em uma única linha. Entretanto, o guia de estilo do Google não recomenda.

Exemplo:

```
2 + 2; pi; "bla bla bla"
```

⁸ Ver <https://google.github.io/styleguide/Rguide.xml>.

⁹ Ver <http://adv-r.had.co.nz/Style.html>.

¹⁰ Ver Wickham *et al.* (2016).

¹¹ Ver Wickham *et al.* (2021).

¹² Coletivamente, esses pacotes são chamados de **tidyverse** (WICKHAM *et al.*, 2019).

¹³ Ver <https://www.rstudio.com/>.

5.11.2 Operações e expressões: atribuição

O símbolo `<-` representa o comando de atribuição, sendo preferível ao símbolo `=`. O primeiro é mais difícil de digitar em um código longo, mas tanto Google como Hadley Wickham recomendam usar o primeiro operador (o código-fonte do R também utiliza exaustivamente o `<-`). O motivo é que o operador de atribuição para a linguagem S, que precedeu o R, era o `<-` por décadas, até que John Chambers decidiu adicionar o `=`¹⁴.

Estranhamente, o símbolo `->` também é um operador de atribuição. Ademais, existe uma função chamada `assign`, além de operadores `<<-` e `->>`.

Exemplos:

```
obj <- 2
obj
obj = 3
obj
4 -> obj
obj
obj <<- 5
obj
6 ->> obj
obj
assign("obj", 7)
obj
```

Todos os operadores de atribuição produzem exatamente o mesmo resultado nos exemplos acima. No entanto, em outros contextos, tais operadores implicam instruções diferentes. Dentro do corpo de uma função, por exemplo, a atribuição com operador `<<-` realiza uma operação diferente de quando se usa `<-`. Já, a função `assign`, com argumentos opcionais, permite maior flexibilidade na atribuição e especificação mais precisa de onde a atribuição é feita. Em resumo, recomenda-se que se utilize o `<-` para atribuições.

¹⁴ Alguns dizem que foi para agradar os programadores em C++.

5.11.3 Operações e expressões: impressão automática (*Autoprinting*)

Um comando em R, que não é uma atribuição, ao ser executado em nível superior (fora de uma função), imprime automaticamente o valor do resultado. Portanto, a maneira mais simples de visualizar o que está armazenado em um objeto do R é executar seu nome no console.

Exemplo:

```
x <- 1
x
(x <- 1) # equivalente às duas linhas acima!
```

Se o objeto for muito grande para visualizar, as funções `head`, `tail` e `summary` podem ser úteis.

Exemplos:

```
x <- rnorm(10000)
x
head(x)
tail(x)
summary(x)
```

Esse recurso é chamado de impressão automática (*autoprinting*) e, embora pareça muito simples, na verdade, é uma implementação complexa, internamente. Tal recurso automático pode ser alterado usando a função `invisible`, que faz com que aquilo que normalmente seria impresso no console seja ocultado. Na verdade, a função `invisible` modifica as chamadas ocultas das funções `print` e `show` (para objetos S4), que são funções genéricas, cujos resultados variam de acordo com o tipo (ou a classe) do objeto. Essa explicação pode não fazer sentido neste momento, porém isso será retomado nos próximos capítulos do presente livro. A Seção 1.6 do manual *R Internals*¹⁵ explica esses conceitos com detalhes. Em resumo, ao realizar

¹⁵ Ver <https://cloud.r-project.org/doc/manuals/r-release/R-ints.html#Autoprinting>.

POR DENTRO DA PNAD CONTÍNUA

uma atribuição, o resultado armazenado não será impresso no console, a menos que se utilize uma sintaxe para tal finalidade.

Contudo, o fato de o resultado de uma operação de atribuição não ser impresso automaticamente não significa que o R não executou a instrução, ou que o resultado não existe. Por exemplo, no trecho de código abaixo, está sendo informado ao R para realizar a atribuição `y <- 2` e, em seguida, atribuir o resultado a `x`.

Exemplos:

```
x <- y <- 2
x
y
```

5.11.4 Operações e expressões: espaços em branco

Como em C e C++, os espaços em branco não são necessários em códigos do R. Existem algumas situações em que, se as instruções forem executadas sem espaçamento, o resultado muda, mas tais casos são exceção. Então, tanto faz digitar:

```
x <- 2
# ou
x<-2
```

Porém, os guias de estilo recomendam o uso de espaços em branco para facilitar a leitura, independentemente do R precisar ou não.

5.11.5 Operações e expressões: objetos

No R, tudo que existe é um *objeto* e tudo que realiza uma ação é uma chamada de função¹⁶.

¹⁶ Ver John Chambers, citado na Seção 6.3 do livro *Advanced R* de Hadley Wickham.

Programação orientada a objetos

No contexto do R, ao chamarmos as estruturas de programação de “objetos”, não significa que estamos trabalhando com Programação Orientada a Objetos (POO). Os objetos do R não são como os objetos do C++. No R, **tudo** é considerado objeto, e, em C++, apenas instâncias de classes criadas com o operador `new` são objetos. Neste sentido, para o R, números, funções, trechos da linguagem R (expressões em R) também são objetos. Em contrapartida, em C++, essas estruturas definitivamente não são. A linguagem de programação C não possui nenhum objeto, seja no sentido do C++ ou no sentido do R.

O R não é uma linguagem particularmente voltada a POO. De fato, ela tem três sistemas de POO diferentes no núcleo do R (o programa sem qualquer pacote adicional). E tem vários outros sistemas de POO em pacotes que acompanham a instalação, bem como os disponibilizados no CRAN. Porém, nenhum desses sistemas de POO atua como o do C++. Além disso, nenhum dos sistemas de POO é necessário para a maioria da programação em R, sendo que a maior parte da programação em R não é POO.

Portanto, por enquanto, não se deve focar naquilo que se aprende sobre POO em outros cursos de Linguagens de Programação, pois certos “vícios” de POO podem atrapalhar o aprendizado do R.

O R como uma linguagem dinâmica

O R é uma linguagem de programação **dinâmica**, o que significa que:

1. Qualquer nome pode ser atribuído a objetos de qualquer tipo. Ao contrário de C e C++, não é preciso “declarar” o tipo de objeto que acompanha cada nome.
2. Não é necessário compilar o código no sentido de C e C++. É possível, simplesmente, digitar ou copiar e colar comandos do R no console e assim obter o resultado esperado.

Isso torna o R significativamente mais simples de programar do que C ou C++, por exemplo.

POR DENTRO DA PNAD CONTÍNUA

No entanto, o R é uma linguagem tão poderosa quanto C++, apesar de não ser tão poderosa quanto C. É possível programar sistemas operacionais (Windows, MacOS e Linux) em C, mas não em qualquer outra linguagem de nível mais alto.

Reprodutibilidade

Embora seja possível digitar ou copiar e colar comandos do R no console e ver o que acontece, isso não é recomendável. Portanto, recomenda-se que se digite todo o código do R em um arquivo e o execute em um ambiente limpo¹⁷. Dependendo da complexidade do código, talvez seja o caso de executá-lo em outro computador de forma remota. O comando para fazer isso a partir de uma linha de comando do sistema operacional é:

```
R CMD BATCH -- vanilla codigo.R
```

Aqui, o `codigo.R` é um arquivo com os comandos. Um arquivo chamado `codigo.Rout` será produzido. Ou seja, todos os comandos do R em `codigo.R` serão executados, e os resultados serão salvos em `codigo.Rout`. Isso permite rodar o mesmo código em diversos computadores, núcleos de processamento e/ou instâncias a fim de reduzir o tempo de processamento de rotinas computacionalmente intensivas.

Exceto para a “aleatoriedade” da geração de números aleatórios (que pode ser removida definindo-se sementes), todo código em R é, em tese, 100% reproduzível. Na prática, tal afirmação nem sempre é verdadeira em virtude da obsolescência de determinados comandos e conflitos de versões tanto do R como de seus pacotes.

5.12 Exercícios

1. Complete o código abaixo no editor de *Scripts* do RStudio de forma a atribuir o valor 42 ao objeto `x`. Execute o código e veja se o valor 42, armazenado em `x`, aparece.

¹⁷ Lembre-se do comando `rm(list=ls())`.

```
x <-  
x
```

2. Suponha que se esteja preparando uma salada de frutas que contém 5 bananas e se deseja armazenar a quantidade de bananas em uma variável com o nome bananas. Lembre-se: para atribuir um número ou um objeto a uma variável em R, deve-se usar o operador de atribuição `<-`.
 - (a) Digite o seguinte código: `bananas <- 5` para atribuir o valor 5 a bananas.
 - (b) Digite: `bananas` abaixo do segundo comentário.
 - (c) Utilize o atalho Cmd/Ctrl+Enter para executar o código e observe o resultado no console: é possível ver que o número 5 está impresso. Portanto, o R agora vincula a variável bananas ao valor 5.
3. Para uma boa salada de frutas, são necessárias laranjas, também. Então, caso se decida adicionar 6 laranjas, o reflexo é criar imediatamente a variável laranjas e atribuir o valor 6 a ela. Em seguida, para calcular quantas frutas se tem no total, graças às duas variáveis, pode-se codificar de uma forma fácil e clara da seguinte forma: `bananas + laranjas`.
 - (a) Atribua a laranjas o valor 6.
 - (b) O R permite combinar as variáveis bananas e laranjas em uma nova variável frutas. Crie esta nova variável frutas, que é a quantidade total de frutas em sua cesta.
4. O ditado popular diz “não se devem misturar bananas com laranjas”. Porém foi o que se acabou de fazer! As variáveis bananas e laranjas continham um número no exercício anterior. O operador `+` pode trabalhar com o que se chama, no R, de variáveis numéricas. Suponha que realmente se queira adicionar “bananas” e “laranjas” e atribuir um texto à variável laranjas (por exemplo, “seis”). A atribuição da variável frutas agora contém a adição de uma variável numérica e uma variável categórica. Isso não se mostra possível.

POR DENTRO DA PNAD CONTÍNUA

- (a) Utilize o conhecimento das questões anteriores para implementar o problema no R. Certifique-se de que entendeu o porquê de o código não funcionar.
- (b) Ajuste o código de forma que o R entenda que se tem 6 laranjas e, portanto, uma cesta de frutas com 11 itens.

Dica: deve-se atribuir o valor numérico 6 à variável `laranjas` sobrescrevendo então o valor da palavra “seis”. Deve-se Observar como as aspas são usadas para indicar que “seis” é uma palavra (conjunto de caracteres).

5.13 Vetores

A função `c()` é usada para criar um vetor a partir de seus argumentos, que tanto podem ser escalares quanto vetores. Alguns exemplos são dados abaixo:

```
x <- c(2,3,4,5)  
x
```

```
# [1] 2 3 4 5
```

```
y <- c(x,6,7,8)  
y
```

```
# [1] 2 3 4 5 6 7 8
```

5.13.1 Sequências

O operador `:` produz uma sequência de valores inteiros com incrementos de 1. Por exemplo, no código abaixo, a variável `xx` armazena o vetor da sequência de números inteiros de 1 a 10. Porém, não é necessário acrescentar a função `c()`, como em `xx <- c(1:10)`.

```
xx <- 1:10  
xx
```

```
# [1] 1 2 3 4 5 6 7 8 9 10
```

É possível, também, gerar um vetor da sequência decrescente dos inteiros de 100 a 1.

```
xx <- 100:1
```

```
xx
```

```
# [1] 100 99 98 97 96 95 94 93 92 91 90 89
# [13] 88 87 86 85 84 83 82 81 80 79 78 77
# [25] 76 75 74 73 72 71 70 69 68 67 66 65
# [37] 64 63 62 61 60 59 58 57 56 55 54 53
# [49] 52 51 50 49 48 47 46 45 44 43 42 41
# [61] 40 39 38 37 36 35 34 33 32 31 30 29
# [73] 28 27 26 25 24 23 22 21 20 19 18 17
# [85] 16 15 14 13 12 11 10 9 8 7 6 5
# [97] 4 3 2 1
```

Os números entre colchetes indicam a posição do vetor naquele ponto (a linha começa pelo elemento da posição indicada entre colchetes).

Função seq()

Os três argumentos da função correspondem, respectivamente, ao valor inicial, ao valor final e ao incremento da sequência. Pode-se escrever, assim:
`seq(from=1,to=10,by=2)`.

```
seq(1,10,2)
```

```
# [1] 1 3 5 7 9
```

Caso a sequência seja decrescente, o valor atribuído ao argumento `by` precisa ser negativo.

```
seq(10,1,-2)
```

```
# [1] 10 8 6 4 2
```

POR DENTRO DA PNAD CONTÍNUA

O último argumento indica, agora, o comprimento da sequência, o resultado dessa função é mostrado abaixo. O incremento é $\frac{26}{3} = 8,6667$.

```
seq(1, 27, length = 4)
```

```
# [1] 1.000000 9.666667 18.333333 27.000000
```

Função rep()

```
rep(1,10)
```

```
# [1] 1 1 1 1 1 1 1 1 1 1
```

```
x <- 10  
rep(c(1,2), x)
```

```
# [1] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
```

```
rep(0:1, c(4,3))
```

```
# [1] 0 0 0 0 1 1 1
```

5.13.2 Operações com vetores

```
x <- 1:10  
x
```

```
# [1] 1 2 3 4 5 6 7 8 9 10
```

```
x + 2
```

```
# [1] 3 4 5 6 7 8 9 10 11 12
```

```
sqrt(x)

# [1] 1.000000 1.414214 1.732051 2.000000 2.236068
# [6] 2.449490 2.645751 2.828427 3.000000 3.162278
```

```
y <- 1:10
y
```

```
# [1] 1 2 3 4 5 6 7 8 9 10
```

```
x + y
```

```
# [1] 2 4 6 8 10 12 14 16 18 20
```

No caso de operações efetuadas elemento a elemento, se os vetores tiverem tamanhos diferentes, os elementos do menor vetor serão repetidos até atingir o do maior vetor, como no exemplo abaixo:

```
x <- 1:10
x

# [1] 1 2 3 4 5 6 7 8 9 10
```

```
y <- c(1,2,3)
y
```

```
# [1] 1 2 3
```

```
x + y
```

```
# Warning in x + y: longer object length is not a multiple of
# shorter object length
```

```
# [1] 2 4 6 5 7 9 8 10 12 11
```

POR DENTRO DA PNAD CONTÍNUA

5.13.3 Acessando elementos ou subconjuntos de um vetor por meio de índices

Índices são usados para fazer referência a partes de um vetor. São utilizados colchetes e os índices são contados a partir de 1.

```
x <- 1:10  
x[4] <- 500
```

No código acima, substitui-se o valor da posição quatro do vetor `x` por 500. Exibir os valores da terceira e da quarta posições do vetor `x`; `x[3,4]` não funcionará, pois `x` será interpretado como matriz.

```
x[c(3,4)]
```

```
# [1] 3 500
```

Pode-se atribuir os três primeiros valores de `x` à variável `w` (`w` será um vetor):

```
w <- x[1:3]  
is.vector(w) #Verificar se 'w' é um vetor.
```

```
# [1] TRUE
```

Pode-se atribuir os valores das posições 2, 3 e 5 do vetor `y` à variável `z`:

```
y <- 10:1  
z <- y[c(2,3,5)]
```

Ao se acessar uma posição inexistente, o resultado será *Not Available* (NA), pois a dimensão do vetor `x` é 10, no exemplo a seguir:

```
x <- 1:10  
x[12]
```

```
# [1] NA
```

É possível, também, o uso de índices negativos.

```
x <- x[-n]
```

O R define isso como sendo todo o vetor x exceto o n -ésimo elemento, como no exemplo a seguir:

```
x <- 1:10
```

```
x
```

```
# [1] 1 2 3 4 5 6 7 8 9 10
```

```
y <- x[-3]
```

```
y
```

```
# [1] 1 2 4 5 6 7 8 9 10
```

```
x[-c(3,4,5)]
```

```
# [1] 1 2 6 7 8 9 10
```

Observação:

Não se pode usar índices negativos e positivos ao mesmo tempo.

5.13.4 Inverter a ordem de um vetor

A função `rev(x)` inverte o vetor x, como a seguir:

```
x <- c(1,2,3,4,5)
```

```
x
```

```
# [1] 1 2 3 4 5
```

```
rev(x)
```

```
# [1] 5 4 3 2 1
```

```
x + rev(x)
```

```
# [1] 6 6 6 6 6
```

5.14 Funções e argumentos

Funções no R são definidas pelo comando `function`. A estrutura padrão é a seguinte:

```
ao_quadrado <- function(x){  
  x2 <- x * x  
  return(x2)  
}  
ao_quadrado(3)
```

```
# [1] 9
```

Nesse caso, `x` é um argumento. É possível, ainda, criar funções que não necessitam de argumentos.

```
que_horas_sao <- function(){  
  return(paste0("São ", format(Sys.time(), "%H:%M"), "."))  
}  
que_horas_sao()  
  
# [1] "São 08:25."
```

5.15 Data.frames

Um `data.frame` é uma das classes de objetos do R utilizadas para armazenamento de conjuntos de dados. A `tibble` é uma releitura moderna do `data.frame`. Seu nome vem da antiga função `tbl_df()` do `dplyr`. Inicialmente, esses objetos

eram criados com a função `tbl_df()`, que era mais facilmente pronunciada como “tibble diff”. Hoje, as funções do `tidyverse` trabalham automaticamente com a forma `tibble`. Os dados contidos no objeto de classe `data.frame` chamado `milsa`, no exemplo a seguir, foram extraídos de Morettin e Bussab (2017).

```
library(readr)
library(dplyr)
# Arquivo com dados de Bussab & Morettin: Estatística Básica.
url <- "https://raw.githubusercontent.com/fernandomayer/data/master/"
arq <- "milsa.csv"

link <- paste0(url, arq)
milsa <- read_csv(link)
class(milsa)

# [1] "spec_tbl_df" "tbl_df"        "tbl"           "data.frame"
```

Existem duas diferenças principais no uso de um `data.frame` em comparação a uma `tibble`: visualização e particionamento.

As `tibbles` tem um método de visualização refinado que mostra apenas as primeiras 10 linhas e todas as colunas que cabem na tela. Isso torna muito mais fácil trabalhar com grandes volumes de dados. Além de seu nome, cada coluna relata seu tipo, um bom recurso emprestado da função `str()`:

```
milsa

# A tibble: 36 x 8
#   Funcionario Est.civil Inst Filhos Salario Anos Meses
#       <dbl> <chr>    <chr>  <dbl>   <dbl> <dbl> <dbl>
# 1 1 solteiro  1o Grau    NA     4      26     3
# 2 2 casado    1o Grau    1      4.56    32    10
# 3 3 casado    1o Grau    2      5.25    36     5
# 4 4 solteiro  2o Grau    NA     5.73    20    10
# 5 5 solteiro  1o Grau    NA     6.26    40     7
# 6 6 casado    1o Grau    0      6.66    28     0
```

POR DENTRO DA PNAD CONTÍNUA

```
# 7      7 solteiro 1o Grau     NA    6.86    41     0
# 8      8 solteiro 1o Grau     NA    7.39    43     4
# 9      9 casado   2o Grau     1    7.59    34    10
# 10    10 solteiro 2o Grau     NA    7.44    23     6
# ... with 26 more rows, and 1 more variable: Regiao <chr>
```

5.15.1 Indexação de data.frames

As **tibbles**, assim como os **data.frames**, permitem indexação de variáveis pelo nome utilizando o operador **\$**, tal como:

```
milsa$Filhos
```

```
# [1] NA 1 2 NA NA 0 NA NA 1 NA 2 NA NA 3 0 NA 1 2 NA
# [20] NA 1 NA NA 0 2 2 NA 0 5 2 NA 1 3 NA 2 3
```

Ao se tentar acessar uma variável que não existe, aparecerá uma mensagem de erro, como no seguinte exemplo:

```
milsa$UF
```

```
# Warning: Unknown or uninitialised column: 'UF'.
# NULL
```

As **tibbles** usam os operadores **[** e **[[**, em que: **[** retorna outra **tibble**, e **[[** retorna um vetor.

```
milsa[ , 1]
```

```
# > A tibble: 36 x 1
#   Funcionario
#       <dbl>
# 1 1
# 2 2
# 3 3
# 4 4
```

```
# 5      5
# 6      6
# 7      7
# 8      8
# 9      9
# 10     10
# # ... with 26 more rows
```

```
milsa[[1]]
```

```
# [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
# [20] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
```

5.15.2 Seleção condicional de observações e transformações

A função `filter()`¹⁸ permite “filtrar” uma `tibble` de acordo com uma condição como, por exemplo:

```
milsa %>%
  filter(Est.civil == "casado")
```

```
# A tibble: 20 x 8
#   Funcionario Est.civil Inst      Filhos Salario  Anos Meses
#   <dbl> <chr>    <chr>      <dbl>  <dbl> <dbl> <dbl>
# 1 2 casado  1o Grau     1  4.56  32  10
# 2 3 casado  1o Grau     2  5.25  36   5
# 3 6 casado  1o Grau     0  6.66  28   0
# 4 9 casado  2o Grau     1  7.59  34  10
# 5 11 casado 2o Grau     2  8.12  33   6
# 6 14 casado 1o Grau     3  8.95  44   2
# 7 15 casado 2o Grau     0  9.13  30   5
# 8 17 casado 2o Grau     1  9.77  31   7
# 9 18 casado 1o Grau     2  9.8   39   7
# 10 21 casado 2o Grau    1  11.1  30   9
```

¹⁸ Essa função será mais bem explorada nos próximos capítulos.

POR DENTRO DA PNAD CONTÍNUA

```
# 11      24 casado    Superior     0  12.8    26    1
# 12      25 casado    2o Grau      2  13.2    32    5
# 13      26 casado    2o Grau      2  13.6    35    0
# 14      28 casado    2o Grau      0  14.7    29    8
# 15      29 casado    2o Grau      5  14.7    40    6
# 16      30 casado    2o Grau      2  16.0    35   10
# 17      32 casado    2o Grau      1  16.6    36    4
# 18      33 casado    Superior     3  17.3    43    7
# 19      35 casado    2o Grau      2  19.4    48   11
# 20      36 casado    Superior     3  23.3    42    2
# ... with 1 more variable: Regiao <chr>
```

```
milsa %>%
  filter(Est.civil == "solteiro" & Inst == "2o Grau")
```

```
# A tibble: 6 x 8
#   Funcionario Est.civil Inst     Filhos Salario Anos Meses
#       <dbl> <chr>     <chr>     <dbl>   <dbl> <dbl> <dbl>
# 1        4 solteiro  2o Grau     NA    5.73   20   10
# 2       10 solteiro  2o Grau     NA    7.44   23    6
# 3       13 solteiro  2o Grau     NA    8.74   37    5
# 4       16 solteiro  2o Grau     NA    9.35   38    8
# 5       20 solteiro  2o Grau     NA   10.8    37    4
# 6       22 solteiro  2o Grau     NA   11.6    34    2
# ... with 1 more variable: Regiao <chr>
```

```
milsa %>%
  filter(Anos >= 30 & Inst == "2o Grau")
```

```
# > A tibble: 15 x 8
#   Funcionario Est.civil Inst     Filhos Salario Anos Meses
#       <dbl> <chr>     <chr>     <dbl>   <dbl> <dbl> <dbl>
# 1        9 casado    2o Grau     1    7.59   34   10
# 2       11 casado    2o Grau     2    8.12   33    6
# 3       13 solteiro  2o Grau     NA    8.74   37    5
```

```
# 4      15 casado  2o Grau     0   9.13   30    5
# 5      16 solteiro 2o Grau    NA   9.35   38    8
# 6      17 casado  2o Grau     1   9.77   31    7
# 7      20 solteiro 2o Grau    NA  10.8    37    4
# 8      21 casado  2o Grau     1  11.1    30    9
# 9      22 solteiro 2o Grau    NA  11.6    34    2
# 10     25 casado  2o Grau     2  13.2    32    5
# 11     26 casado  2o Grau     2  13.6    35    0
# 12     29 casado  2o Grau     5  14.7    40    6
# 13     30 casado  2o Grau     2  16.0    35   10
# 14     32 casado  2o Grau     1  16.6    36    4
# 15     35 casado  2o Grau     2  19.4    48   11
# ... with 1 more variable: Regiao <chr>
```

A função `mutate`¹⁹ permite criar, modificar ou remover colunas em uma tibble:

```
# cria a coluna 'Sal.cor' e remove as colunas 'Meses' e 'Filhos'
correcao <- 9.03
milsa %>%
  mutate(Sal.cor = Salario * correcao) %>%
  mutate(Filhos = NULL, Meses = NULL)
```

```
# > A tibble: 36 x 7
#   Funcionario Est.civil Inst     Salario  Anos Regiao  Sal.cor
#   <dbl> <chr>     <chr>     <dbl> <dbl> <chr>     <dbl>
# 1       1 solteiro  1o Grau     4      26 interi~  36.1
# 2       2 casado   1o Grau    4.56    32 capital  41.2
# 3       3 casado   1o Grau    5.25    36 capital  47.4
# 4       4 solteiro  2o Grau    5.73    20 outro   51.7
# 5       5 solteiro  1o Grau    6.26    40 outro   56.5
# 6       6 casado   1o Grau    6.66    28 interi~  60.1
# 7       7 solteiro  1o Grau    6.86    41 interi~  61.9
# 8       8 solteiro  1o Grau    7.39    43 capital  66.7
```

¹⁹ Essa função será mais bem explorada nos próximos capítulos.

POR DENTRO DA PNAD CONTÍNUA

```
# 9           9 casado   2o Grau    7.59    34 capital    68.5
# 10          10 solteiro 2o Grau    7.44    23 outro      67.2
# ... with 26 more rows

# substitui 'Salario' por 'Salario * correcao'
milsa %>%
  mutate(Salario = Salario * correcao)

# > A tibble: 36 x 8
#   Funcionario Est.civil Inst     Filhos Salario  Anos Meses
#   <dbl> <chr>     <chr>     <dbl>    <dbl> <dbl> <dbl>
# 1 1         solteiro 1o Grau     NA     36.1   26    3
# 2 2         casado   1o Grau     1     41.2   32   10
# 3 3         casado   1o Grau     2     47.4   36    5
# 4 4         solteiro 2o Grau     NA     51.7   20   10
# 5 5         solteiro 1o Grau     NA     56.5   40    7
# 6 6         casado   1o Grau     0     60.1   28    0
# 7 7         solteiro 1o Grau     NA     61.9   41    0
# 8 8         solteiro 1o Grau     NA     66.7   43    4
# 9 9         casado   2o Grau     1     68.5   34   10
# 10 10        solteiro 2o Grau     NA     67.2   23    6
# ... with 26 more rows, and 1 more variable: Regiao <chr>
```

5.16 Exercícios

1. Gere uma sequência de 1 a 50 com a função `seq` utilizando um único argumento.
2. Crie um vetor `Idade` com os valores 22, 25, 18, 20.
3. Verifique a ajuda da função `sort`. Aplique ao vetor `Idade` que acabou de ser criado. Explique o resultado obtido.
4. Verifique a ajuda da função `rank` e também aplique ao vetor `Idade`. Qual a relação entre `sort(Idade)` e `rank(Idade)`?

5. Crie um vetor **Nomes** com os nomes "Tiago", "Mateus", "Olivia", "Estela" e um vetor **Genero** com as observações "M", "M", "F", "F".
6. Crie uma **tibble** com os vetores **Idade**, **Nomes** e **Genero**. Atribua o resultado a **Dados**.
7. Modifique a variável **Idade** em **Dados** para exibir valores em meses.
8. Crie quatro funções distintas que tomam um vetor numérico como único argumento e retornam:
 - O último valor. Deve-se usar `[` ou `[[?]`
 - Os elementos em posições pares.
 - Cada elemento, exceto o último valor.
 - Apenas números pares (e sem valores ausentes). Teste as funções nas variáveis numéricas do conjunto de dados **milsa**.

5.17 Matrizes

5.17.1 Função `matrix()`

A função `matrix(v, nrow = 1, ncol = c)` comporta um conjunto de argumentos em que `v` é um vetor, `1` é o número de linhas, e `c` é o número de colunas. Além de `v`, é necessário especificar, apenas, ou `nrow` ou `ncol`. Essa função recebe um vetor como argumento e o transforma em uma matriz de acordo com as dimensões especificadas (ver os exemplos a seguir).

```
x <- 1:24
xmat <- matrix(x, nrow = 3)
xmat

#      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
# [1,]     1     4     7    10    13    16    19    22
# [2,]     2     5     8    11    14    17    20    23
# [3,]     3     6     9    12    15    18    21    24
```

POR DENTRO DA PNAD CONTÍNUA

Caso o usuário não defina o nome do argumento (por exemplo, `xmat <- matrix(x, 3)`), por padrão, a função será executada com o preenchimento sendo feito por coluna. Deve-se reparar que esse mesmo resultado pode ser obtido utilizando as seguintes formas: informando o argumento `ncol` (`xmat <- matrix(x, ncol = 3)`); ou deixando um espaço em branco na posição do primeiro argumento (`xmat <- matrix(x, , 3)`). Em todos os casos, o preenchimento será executado por coluna.

```
xmat <- matrix(x, ncol = 3)
xmat

#      [,1] [,2] [,3]
# [1,]    1    9   17
# [2,]    2   10   18
# [3,]    3   11   19
# [4,]    4   12   20
# [5,]    5   13   21
# [6,]    6   14   22
# [7,]    7   15   23
# [8,]    8   16   24
```

A opção de utilizar o argumento `byrow = TRUE` força o preenchimento por linha, como no exemplo a seguir:

```
xmat <- matrix(x, ncol = 3, byrow = TRUE)
xmat

#      [,1] [,2] [,3]
# [1,]    1    2    3
# [2,]    4    5    6
# [3,]    7    8    9
# [4,]   10   11   12
# [5,]   13   14   15
# [6,]   16   17   18
# [7,]   19   20   21
# [8,]   22   23   24
```

5.17.2 Matriz de zeros

Para criar uma matriz com zeros, basta fazer:

```
matrix(0, 2, 2)
```

```
#      [,1] [,2]
# [1,]    0    0
# [2,]    0    0
```

5.17.3 Matriz diagonal e matriz identidade

Função

Para criar uma matriz diagonal, basta utilizar a função `diag()`.

5.17.4 Sintaxe

A função `diag(k, nrow = 1, ncol = c)` tem como argumentos: `k`, que é o elemento da diagonal; `nrow`, o número de linhas; e `ncol`, o número de colunas. No entanto, é necessário identificar, apenas, o número de linhas (verificar que o exemplo a seguir é equivalente à `diag(5, 3, 3)`).

```
diag(5, 3)
```

```
#      [,1] [,2] [,3]
# [1,]    5    0    0
# [2,]    0    5    0
# [3,]    0    0    5
```

A forma padrão dessa função é `k = 1`. Assim, `diag(3)` é equivalente à `diag(1, 3)` (ver exemplo a seguir).

```
diag(3)
diag(1, 3)
```

POR DENTRO DA PNAD CONTÍNUA

```
#      [,1] [,2] [,3]
# [1,]    1    0    0
# [2,]    0    1    0
# [3,]    0    0    1
```

Caso o primeiro argumento seja um vetor, então, serão os elementos desse vetor que formarão a diagonal da matriz (ver o exemplo a seguir).

```
diag(c(1, 2, 3))
```

```
#      [,1] [,2] [,3]
# [1,]    1    0    0
# [2,]    0    2    0
# [3,]    0    0    3
```

5.17.5 Multiplicação de matrizes

A multiplicação de matrizes é fundamental para diversas áreas das Ciências Sociais Aplicadas como, por exemplo, na elaboração das Matrizes de Insumo-Produto, peças fundamentais na construção das Contas Nacionais²⁰. Considere os seguintes exemplos:

```
x <- matrix(1:4, 2)
y <- matrix(4:1, 2)
x

#      [,1] [,2]
# [1,]    1    3
# [2,]    2    4
```

²⁰ “Uma Matriz de Insumo-Produto é compreendida, normalmente, como uma matriz de coeficientes técnicos diretos que apresenta o quanto determinada atividade econômica necessita consumir das demais atividades para que possa produzir uma unidade monetária adicional. A partir desta matriz é desenvolvido o modelo de Leontief que possibilita calcular a produção de cada atividade a partir de uma demanda final exógena” (IBGE, 2018, p. 9).

```
y
```

```
#      [,1] [,2]
# [1,]    4    2
# [2,]    3    1
```

Para multiplicar elemento por elemento de `x` e `y`, basta fazer:

```
x*y
```

```
#      [,1] [,2]
# [1,]    4    6
# [2,]    6    4
```

A seguir, tem-se a multiplicação padrão:

```
x %*% y
```

```
#      [,1] [,2]
# [1,]   13    5
# [2,]   20    8
```

Nos casos abaixo, é executado, de duas formas, o chamado Produto de Kronecker, cuja aplicação nas Ciências Econômicas pode ser vista em Stern *et al.* (2020): *Otimização e Processos Estocásticos Aplicados à Economia e Finanças*.

```
kronecker(x, diag(2))
```

```
#      [,1] [,2] [,3] [,4]
# [1,]    1    0    3    0
# [2,]    0    1    0    3
# [3,]    2    0    4    0
# [4,]    0    2    0    4
```

```
kronecker(diag(2), x)
```

```
#      [,1] [,2] [,3] [,4]
# [1,]    1    3    0    0
# [2,]    2    4    0    0
# [3,]    0    0    1    3
# [4,]    0    0    2    4
```

5.17.6 Adição de matrizes

Para somar duas matrizes, basta fazer:

```
x + y
```

```
#      [,1] [,2]
# [1,]    5    5
# [2,]    5    5
```

5.17.7 Outras operações

Algumas operações podem ser feitas em cada um dos elementos de uma matriz (ver exemplo a seguir).

```
2 * x
```

```
#      [,1] [,2]
# [1,]    2    6
# [2,]    4    8
```

```
x / 2
```

```
#      [,1] [,2]
# [1,]  0.5  1.5
# [2,]  1.0  2.0
```

```
sqrt(x)

#           [,1]      [,2]
# [1,] 1.000000 1.732051
# [2,] 1.414214 2.000000
```

Outras operações como determinante, transposta e inversa, também, podem ser necessárias. Esses exemplos aparecem demonstrados na sequência de comandos abaixo:

```
det(x)
```

```
# [1] -2
```

```
t(x)
```

```
#           [,1]      [,2]
# [1,] 1 2
# [2,] 3 4
```

```
solve(x)
```

```
#           [,1]      [,2]
# [1,] -2 1.5
# [2,] 1 -0.5
```

5.17.8 As funções cbind() e rbind()

As funções `cbind()` e `rbind()` permitem a adição de colunas ou linhas a uma matriz. Considere os exemplos a seguir:

```
x <- matrix(1:10, 5)
x
```

POR DENTRO DA PNAD CONTÍNUA

```
#      [,1] [,2]
# [1,]    1    6
# [2,]    2    7
# [3,]    3    8
# [4,]    4    9
# [5,]    5   10

m <- cbind(x, 11:15)
m

#      [,1] [,2] [,3]
# [1,]    1    6   11
# [2,]    2    7   12
# [3,]    3    8   13
# [4,]    4    9   14
# [5,]    5   10   15

m <- rbind(m, c(100, 200, 300))
m

#      [,1] [,2] [,3]
# [1,]    1    6   11
# [2,]    2    7   12
# [3,]    3    8   13
# [4,]    4    9   14
# [5,]    5   10   15
# [6,] 100  200  300

m1 <- cbind(x, 1990:1994)
m1

#      [,1] [,2] [,3]
# [1,]    1    6 1990
# [2,]    2    7 1991
# [3,]    3    8 1992
# [4,]    4    9 1993
# [5,]    5   10 1994
```

```
m2 <- cbind(x, 2 * x)
m2

#      [,1] [,2] [,3] [,4]
# [1,]     1     6     2    12
# [2,]     2     7     4    14
# [3,]     3     8     6    16
# [4,]     4     9     8    18
# [5,]     5    10    10    20
```

Agora, considere o seguinte vetor w :

```
w <- c(.1, .2, .3, .2, .1)
```

Para ver w como uma matriz, basta fazer:

```
w <- as.matrix(w)
w

#      [,1]
# [1,]  0.1
# [2,]  0.2
# [3,]  0.3
# [4,]  0.2
# [5,]  0.1

m3 <- cbind(x, w, 2*x, w*x)
m3

#          w
# [1,] 1 6 0.1 2 12 0.1 0.6
# [2,] 2 7 0.2 4 14 0.4 1.4
# [3,] 3 8 0.3 6 16 0.9 2.4
# [4,] 4 9 0.2 8 18 0.8 1.8
# [5,] 5 10 0.1 10 20 0.5 1.0
```

POR DENTRO DA PNAD CONTÍNUA

Pode-se adicionar uma nova linha à matriz `m2`.

```
q <- c(4, 4, 8, 8)
```

Atente para a correção: `q` é um vetor coluna, assim, deve-se usar `t(q)` ou `rbind(m2, c(4, 4, 8, 8))`.

```
m4 <- rbind(m2, q)
m5 <- matrix(1:6, 2)
```

5.17.9 Nomear linhas e colunas de uma matriz

Para nomear as colunas, pode-se fazer:

```
colnames(x) <- c("c1", "c2")
x
```

```
#      c1 c2
# [1,]  1  6
# [2,]  2  7
# [3,]  3  8
# [4,]  4  9
# [5,]  5 10
```

Para nomear as linhas, pode-se fazer:

```
rownames(x) <- paste0("l", 1:5)
```

```
x
```

```
#      c1 c2
# l1  1  6
# l2  2  7
# l3  3  8
# l4  4  9
# l5  5 10
```

Para nomear linhas e colunas, simultaneamente, pode-se fazer:

```
dimnames(x) <- list(c("a1", "a2", "a3", "a4", "a5"), c("b1", "b2"))
x

#      b1 b2
# a1   1  6
# a2   2  7
# a3   3  8
# a4   4  9
# a5   5 10
```

Para executar, tudo ao mesmo tempo, pode-se fazer:

```
x1 <- matrix(1:6, 3, 2, dimnames = list(letters[1:3], letters[1:2]))
x1

#      a b
# a 1  4
# b 2  5
# c 3  6
```

5.17.10 Submatrizes

Para criar uma matriz qualquer `z` de dimensões 4×6 , pode-se executar o seguinte comando:

```
z <- matrix(1:24, 4)
z

#      [,1] [,2] [,3] [,4] [,5] [,6]
# [1,]     1     5     9    13    17    21
# [2,]     2     6    10    14    18    22
# [3,]     3     7    11    15    19    23
# [4,]     4     8    12    16    20    24
```

Agora, caso se deseje atribuir o elemento da linha 2 com a coluna 3 a uma variável `z23`, pode-se fazer:

POR DENTRO DA PNAD CONTÍNUA

```
z23 <- z[2,3]
```

```
z23
```

```
# [1] 10
```

Para investigar e mostrar os elementos da linha 2, por exemplo, faz-se:

```
z[2, ]
```

```
# [1] 2 6 10 14 18 22
```

Nesse caso, o resultado é um vetor. Já, para mostrar a coluna 3, faz-se:

```
z[, 3]
```

```
# [1] 9 10 11 12
```

Nesse caso, o resultado também é um vetor. Já se se deseja investigar, selecionar ou mostrar apenas as linhas 1 e 3, pode-se aplicar o seguinte comando:

```
z[c(1, 3), ]
```

```
# [,1] [,2] [,3] [,4] [,5] [,6]
# [1,]    1     5     9    13    17    21
# [2,]    3     7    11    15    19    23
```

Para selecionar, mostrar ou investigar as colunas 4 e 6, por exemplo, basta fazer:

```
z[, c(4, 6)]
```

```
# [,1] [,2]
# [1,]   13   21
# [2,]   14   22
# [3,]   15   23
# [4,]   16   24
```

Nos dois casos anteriores, o resultado é uma matriz. Agora, se o desejo for a criação de uma submatriz composta pelas linhas 2, 3 e 4 e pelas colunas 4, 5 e 6 da matriz `z`, basta que se faça:

```
z[2:4, 4:6]
```

```
#      [,1] [,2] [,3]
# [1,]    14   18   22
# [2,]    15   19   23
# [3,]    16   20   24
```

No exemplo a seguir, o que se tem é um comando que mostra todas as colunas de `z`, mas apenas as linhas cujo valor da segunda coluna de `z` é maior que 6²¹.

```
y <- z[z[, 2] > 6, ]
y

#      [,1] [,2] [,3] [,4] [,5] [,6]
# [1,]    3    7   11   15   19   23
# [2,]    4    8   12   16   20   24
```

5.17.11 Mais sobre matrizes

Para obter a dimensão da matriz, aplica-se:

```
dim(z)
```

```
# [1] 4 6
```

A função `summary` é significativamente relevante para o cálculo de estatísticas básicas para cada coluna da matriz. Isso serve para o pesquisador que desejar um resumo de todos os elementos da matriz. Nesse caso, basta fazer: `summary(as.numeric(z))`.

²¹ Esse comando funciona como um filtro para matrizes.

POR DENTRO DA PNAD CONTÍNUA

```
summary(z)
```

	V1	V2	V3	V4
# Min.	:1.00	Min. :5.00	Min. : 9.00	Min. :13.00
# 1st Qu.	:1.75	1st Qu.:5.75	1st Qu.: 9.75	1st Qu.:13.75
# Median	:2.50	Median :6.50	Median :10.50	Median :14.50
# Mean	:2.50	Mean :6.50	Mean :10.50	Mean :14.50
# 3rd Qu.	:3.25	3rd Qu.:7.25	3rd Qu.:11.25	3rd Qu.:15.25
# Max.	:4.00	Max. :8.00	Max. :12.00	Max. :16.00
	V5	V6		
# Min.	:17.00	Min. :21.00		
# 1st Qu.	:17.75	1st Qu.:21.75		
# Median	:18.50	Median :22.50		
# Mean	:18.50	Mean :22.50		
# 3rd Qu.	:19.25	3rd Qu.:23.25		
# Max.	:20.00	Max. :24.00		

Para calcular, por exemplo, a média aritmética dos elementos da segunda coluna de `z`, basta aplicar:

```
mean(z[ , 2])
```

```
# [1] 6.5
```

5.18 Listas

Tecnicamente, uma lista é um vetor. Vetores comuns – aqueles do tipo que usamos até agora neste livro – são chamados de vetores *atômicos*, uma vez que seus componentes não podem ser divididos em componentes menores. Em contraste, as listas são chamadas de vetores *recursivos*.

5.18.1 Sintaxe

A função `list()` assume a seguinte forma básica:

```
list(comp1, comp2, ...)
```

Os argumentos `comp1`, `comp2`, ... na função `list` são os componentes de uma lista, os quais podem ser matrizes, vetores, outras listas ou qualquer outro objeto do R.

No primeiro exemplo de lista que se apresenta a seguir, considera-se um banco de dados de funcionários. Para cada funcionário, pode-se armazenar o nome, o salário e um *booleano* indicando a filiação ao sindicato. Uma vez que se tem três tipos de variáveis diferentes – caractere, numérico e lógico –, a lista, assim, é um objeto possível que permite o armazenamento dessas informações. O banco de dados inteiro pode ser uma lista de listas ou algum outro tipo de lista, como uma `tibble` ou `data.frame`²².

```
j <- list(nome = "Paulo", salario = 5000, filiado = F)
```

A visualização de `j` pode ser feita por completo ou por componente:

```
j
```

```
# $nome
# [1] "Paulo"
#
# $salario
# [1] 5000
#
# $filiado
# [1] FALSE
```

Na verdade, os nomes dos componentes — que são chamados de *tags* nos manuais do R — e.g. `nome`, `salario` etc. são opcionais. Pode-se fazer o seguinte:

²² Essas transformações serão apresentadas nos próximos capítulos.

POR DENTRO DA PNAD CONTÍNUA

```
jalt <- list("Paulo", 5000, F)
jalt
```

```
# [[1]]
# [1] "Paulo"
#
# [[2]]
# [1] 5000
#
# [[3]]
# [1] FALSE
```

No entanto, geralmente o uso de nomes em vez de índices numéricos garante uma maior organização, reduzindo a possibilidade de se cometer erros.

Os nomes dos componentes da lista podem ser abreviados em qualquer extensão possível, sem causar ambiguidade:

```
j$sal
```

```
# [1] 5000
```

Como listas são vetores, essas podem ser criadas, também, por meio da função `vector()`, como no exemplo seguinte:

```
z <- vector(mode = "list")
z[["abc"]] <- 3
z
```

```
# $abc
# [1] 3
```

5.18.2 Operações com listas: indexação

É possível acessar um componente de uma lista a partir de diversas formas (ver os exemplos a seguir).

```
j$salario  
  
# [1] 5000  
  
j[["salario"]]  
  
# [1] 5000
```

Pode-se acessar os componentes de uma lista por meio de seus índices numéricos, tratando a lista como um vetor. No entanto, é necessário observar que, nesse caso, usa-se colchetes duplos em vez de colchetes simples.

```
j[[2]]  
  
# [1] 5000
```

Portanto, existem três maneiras de acessar um componente individual `c` de uma lista e retorná-lo no tipo de dados `c`:

- `lst$c`
- `lst[["c"]]`
- `lst[[i]]`, em que `i` é a posição de `c` em `lst`.

Cada um desses tipos pode ser útil em contextos diferentes (ver os próximos exemplos). Uma alternativa para a segunda e terceira técnicas listadas acima é usar colchetes simples em vez de colchetes duplos.

- `lst["c"]`
- `lst[i]`, em que `i` é a posição de `c` em `lst`.

Portanto, é possível acessar os elementos da lista utilizando colchetes simples, da mesma forma que para vetores. Porém, há uma diferença importante no padrão de indexação vetorial ordinária (atômica). Se colchetes simples `[]` forem usados, o resultado será outra lista – uma sublista do original. Por exemplo, continuando com o exemplo anterior, tem-se o seguinte:

POR DENTRO DA PNAD CONTÍNUA

```
j[1:2]
```

```
# $nome  
# [1] "Paulo"  
#  
# $salario  
# [1] 5000
```

```
j2 <- j[2]  
j2
```

```
# $salario  
# [1] 5000
```

```
str(j2)
```

```
# List of 1  
# $ salario: num 5000
```

A operação de indexação retornou outra lista que consiste nos dois primeiros componentes da lista original `j`. Observe que o resultado obtido faz sentido, uma vez que os colchetes para indexação são funções.

Por outro lado, pode-se usar colchetes duplos `[[]]` para fazer referência a apenas um único componente, com o resultado sendo o tipo desse componente.

```
j[[1:2]]
```

```
# Error in j[[1:2]]: subscript out of bounds
```

```
j2a <- j[[2]]  
j2a
```

```
# [1] 5000
```

```
class(j2a)
```

```
# [1] "numeric"
```

5.19 Fatores

Um fator pode ser visto simplesmente como um vetor que contém mais informação. Essa informação extra consiste em um registro dos valores distintos naquele vetor, chamados de níveis, como no exemplo abaixo:

```
x <- c(5,12,13,12)
xf <- factor(x)
xf
```

```
# [1] 5 12 13 12
# Levels: 5 12 13
```

Nesse caso, todos os valores distintos em `xf` – 5,12 e 13 – são definidos como níveis.

```
str(xf)
```

```
# Factor w/ 3 levels "5","12","13": 1 2 3 2
```

```
unclass(xf)
```

```
# [1] 1 2 3 2
# attr(",levels")
# [1] "5"  "12" "13"
```

O núcleo de `xf` não é `(5,12,13,12)`, mas sim `(1,2,3,2)`. Isso significa que esses dados consistem primeiro em um valor de nível 1, depois em dois valores de nível 2 e nível 3 e, finalmente, em outro valor de nível 2, nessa ordem. Portanto, os dados foram recodificados por nível. Os próprios níveis também são registrados, embora como caracteres como `"5"` em vez de 5.

POR DENTRO DA PNAD CONTÍNUA

O comprimento de um fator ainda é definido em termos do comprimento dos dados, em vez de ser uma contagem do número de níveis (ver o exemplo a seguir).

```
length(xf)
```

```
# [1] 4
```

É possível ainda a antecipação de novos níveis futuros, como pode ser visto a seguir:

```
x <- c(5,12,13,12)
xff <- factor(x, levels = c(5,12,13,88))
xff
```

```
# [1] 5 12 13 12
# Levels: 5 12 13 88
```

```
xff[2] <- 88
xff
```

```
# [1] 5 88 13 12
# Levels: 5 12 13 88
```

Originalmente, `xff` não continha o valor 88, mas, ao defini-lo, cria-se essa possibilidade futura. Mais tarde, realmente pode-se adicionar o valor.

Da mesma forma, não se pode entrar com um nível “ilegal.” Veja o que acontece quando se tenta executar o seguinte comando:

```
xff[2] <- 28
```

```
# Warning in '[<-.factor'('*tmp*', 2, value = 28):
# invalid factor level, NA generated
```

5.20 Exercícios

1. Se `dados` é um `data.frame`, explique o que acontece com `t(dados)` e `t(t(df))`. Realize alguns experimentos, certificando-se de aplicar os comandos a um conjunto de dados com diferentes tipos de variáveis (colunas).
2. O que a função `as.matrix()` faz quando aplicada a um `data.frame` com variáveis (colunas) de tipos diferentes? Como o resultado difere de `data.matrix()`?
3. Explique a relação entre os seguintes objetos:

```
a <- 1:10
b <- list(a, a)
c <- list(b, a, 1:10)
```

4. Explique o que acontece quando se executa o seguinte código:

```
x <- list(1:10)
x[[2]] <- x
x
```

5. O que acontece com um fator quando seus níveis são modificados?

```
f1 <- factor(letters)
levels(f1) <- rev(levels(f1))
f1
```

6 Importação, leitura e salvamento dos microdados da PNAD Contínua: o pacote **PNADcIBGE**

O pacote **PNADcIBGE**, desenvolvido por Braga e Assuncao (2020), foi pensado para fornecer um conjunto de ferramentas para baixar, ler e analisar os microdados da PNAD Contínua. Suas funções permitem baixar não apenas os microdados diretamente do site oficial do IBGE¹ como também a documentação e os arquivos necessários para sua leitura e exploração. Outras análises poderão ser feitas por meio de pacotes como **tidyverse**, desenvolvido por Wickham *et al.* (2019), e **survey**, desenvolvido por Lumley (2019)².

6.1 Instalação

Para instalar o pacote no R, pode-se utilizar o comando `install.packages ("PNADcIBGE")` no console ou no *Script*. Ao fim da instalação, deve-se carregá-lo utilizando o comando `library(PNADcIBGE)` ou `require(PNADcIBGE)` (ver aplicação abaixo).

¹ Ver <https://www.ibge.gov.br/>.

² Os próximos capítulos procuram apresentar algumas das principais funções desses pacotes.

```
install.packages("PNADcIBGE")
library(PNADcIBGE)
```

6.2 O pacote **PNADcIBGE**: funções básicas

O pacote **PNADcIBGE** possui seis funções básicas:

- `get_pnadc`
- `pnadc_deflator`
- `pnadc_design`
- `pnadc_example`
- `pnadc_labeller`
- `read_pnadc`.

O comando `get_pnadc` é a função principal, pois permite ao usuário fazer a importação *online* dos dados para o RStudio. Por meio dela, faz-se o *download* das bases de microdados e de arquivos essenciais como: *inputs*, dicionários e outros arquivos auxiliares. Esses arquivos ficam disponíveis por períodos específicos (ano ou trimestre) ou por visitas (para bases anuais). Essas opções devem ser declaradas como argumentos da função no próprio console ou no *Script* do RStudio. A forma básica dessa função, com suas definições padrões e com todos os argumentos que essa aceita, é a seguinte:

```
get_pnadc(year,
           quarter = NULL,
           interview = NULL,
           topic = NULL,
           vars = NULL,
           defyear = NULL,
```

POR DENTRO DA PNAD CONTÍNUA

```
    defperiod = NULL,  
    labels = TRUE,  
    deflator = TRUE,  
    design = TRUE,  
    savedir = tempdir()  
)
```

É necessário definir, ao menos, o argumento `year`. Nesse caso, todos os outros assumirão suas definições padrões. Diante disso, o resultado será um objeto que contém os microdados do ano escolhido com todas as variáveis, rótulos, com as variáveis de deflator, no formato `design` e com todos os dados auxiliares (arquivos com os dicionários, os anexos e a documentação) salvos em um diretório temporário.

Vale lembrar que os dados anuais podem ser baixados de duas formas, a saber, por entrevista ou por trimestre. Para entender como isso funciona, deve-se analisar cada um dos argumentos da função `get_pnadc`.

- `year`: o ano em que a pesquisa foi realizada. Precisa ser um ano entre 2012 e o ano corrente. Este argumento não aceita vetores.
- `quarter`: o trimestre do ano em que a pesquisa foi realizada. Deve ser um número de 1 a 4. Assim como o argumento anterior, este não aceita vetores. Caso se informe `NULL`, torna-se necessário fornecer um valor para a entrevista (argumento `interview`) ou para o número do tópico (argumento `topic`).
- `interview`: o número da entrevista referente aos dados a serem baixados. Deve ser um número de 1 a 5. Também não aceita que se informe um vetor. Essa opção é utilizada para obter os dados anuais segundo divulgação por entrevista. Quando se informa `NULL` (argumento padrão no caso de não se informar esse argumento), faz-se necessário fornecer um valor em `quarter` ou em `topic`.
- `topic`: o trimestre relacionado ao tópico dos dados a serem baixados. Deve ser um número de 1 a 4. Também não se pode informar um vetor. Ao usar esta opção, obtém-se os dados anuais por trimestre. Caso seja `NULL`, os argumentos `quarter` ou `interview` devem ser fornecidos. Com esse argumento, é possível baixar os dados anuais trimestre a trimestre.

- `vars`: vetor contendo os nomes das variáveis a serem baixadas. O argumento padrão (caso o usuário não informe nada) faz com que o comando baixe todas as variáveis existentes nos microdados.
- `defyear`: o ano dos dados do deflator (disponibilizados pelo próprio IBGE) que serão baixados e incorporados aos microdados. Deve ser um número entre 2017 e o último ano disponível. Não se pode informar um vetor. Caso assuma o valor `NULL`, o ano deflator será definido como o do último ano disponível para os microdados divulgados para a entrevista definida, ou igual ao ano para os microdados do trimestre definido no argumento `topic`. Quando se informa um valor em `quarter`, o argumento `defyear` é ignorado. Este argumento deve ser usado apenas se o deflator for definido como `TRUE`.
- `defperiod`: o trimestre de referência para a variável de deflacionamento a ser baixada junto com os microdados anuais, quando se define o argumento `topic`. Deve ser um número de 1 a 4. Vetor não é aceito. Se for `NULL`, o período do deflator será definido como igual ao `topic`. Quando `quarter` ou `interview` são definidos, este argumento é ignorado, tornando-se funcional apenas quando o argumento `deflator` é definido como `TRUE`. Com esse argumento, é possível adicionar variáveis de deflacionamento nos microdados anuais para os trimestres definidos no argumento `topic`.
- `labels`: um valor lógico, que pode ser `TRUE`, se se deseja que as variáveis categóricas apresentem-se como fatores, com os rótulos correspondentes ao dicionário da pesquisa, ou `FALSE`, caso o desejo seja que a base de dados apresente apenas valores numéricos, inclusive para variáveis categóricas.
- `deflator`: um valor lógico que, se definido como `TRUE`, permite que variáveis relacionadas ao deflator sejam baixadas e adicionadas aos microdados.
- `design`: um valor lógico que, se definido como `TRUE`, retorna um objeto de classe `survey.design`. Esse tipo de objeto é altamente recomendável para análises posteriores a respeito dos intervalos de confiança das estimativas. Esse objeto permite que se explore pesquisas baseadas em planos amostrais complexos, como é o caso da PNAD Contínua. Caso se defina esse argumento como `FALSE`, apenas os microdados no tradicional formato `data.frame` será baixado.

POR DENTRO DA PNAD CONTÍNUA

- `savedir`: diretório para salvar os microdados e o conjunto da documentação e dos arquivos necessários para sua abertura. Caso não se informe esse argumento, a função utiliza um diretório temporário.

A seguir, seguem dois exemplos simples:

- **Primeiro exemplo** – Para baixar os dados referentes à PNAD Contínua trimestral para o 1º trimestre de 2020, pode-se utilizar o seguinte comando:

```
pnadc2021T1_dsg <- get_pnadc(2021,  
                                quarter = 1)
```

Nesse caso, os dados são armazenados no objeto `pnadc2021T1_dsg`, cuja classe é `survey.design`. Para testar, basta que se aplique:

```
class(pnadc2021T1_dsg)
```

```
# [1] "survey.design2" "survey.design"
```

Uma rápida inspeção usando o comando `names()` revela as seguintes informações armazenadas no objeto `pnadc2021T1_dsg`:

```
names(pnadc2021T1_dsg)
```

```
# > [1] "cluster"      "strata"       "has.strata"   "prob"  
#   [5] "allprob"      "call"         "variables"    "fpc"  
#   [9] "pps"          "postStrata"
```

- `cluster`: identificação das unidades primárias de amostragem (UPAs);
- `strata`: identificação dos estratos;

- `has.strata`: indicador lógico que identifica se a pesquisa possui estratificação em alguma de suas etapas. Para o caso da PNAD Contínua, tal indicador é sempre verdadeiro (`TRUE`).
- `prob`: probabilidades de seleção das UPAs;
- `allprob`: matriz de probabilidade dos pesos;
- `call`: objeto de chamamento de função;
- `variables`: objeto do tipo `data.frame`, ou seja, os microdados propriamente ditos.
- `fpc`: fator de correção para populações finitas.
- `pps`: indicador lógico no caso de a pesquisa utilizar processo de amostragem por conglomerado com probabilidade proporcional ao tamanho³. Como a PNAD Contínua não se vale de tal metodologia, o resultado desse indicador é falso (`FALSE`);
- `postStrata`: variáveis de pós-estratificação.

Deve-se reparar que o conjunto dos microdados da PNAD Contínua ficam armazenados no elemento `...$variables` do objeto de classe `survey.design` (`pnadc2021T1_dsg$variables`, no exemplo anterior). Para verificar, basta que se utilize o seguinte comando:

```
names(pnadc2021T1_dsg$variables)
```

```
# >[1] "Ano"          "Trimestre"    "UF"           "Capital"      "RM_RIDE"  
# [6] "UPA"          "Estrato"       "V1008"        "V1014"       "V1016"  
# [11] "V1022"        "V1023"        "V1027"        "V1028"       "V1029"  
# [16] "posest"       "V2001"        "V2003"        "V2005"       "V2007"  
# [21] "V2008"        "V20081"       "V20082"       "V2009"       "V2010"  
# [26] "V3001"        "V3002"        "V3002A"       "V3003"       "V3003A"
```

³ Do inglês, *probability proportional to sample size*.

POR DENTRO DA PNAD CONTÍNUA

# [31]	"V3004"	"V3005"	"V3005A"	"V3006"	"V3006A"
# [36]	"V3007"	"V3008"	"V3009"	"V3009A"	"V3010"
# [41]	"V3011"	"V3011A"	"V3012"	"V3013"	"V3013A"
# [46]	"V3013B"	"V3014"	"V4001"	"V4002"	"V4003"
# [51]	"V4004"	"V4005"	"V4006"	"V4006A"	"V4007"
# [56]	"V4008"	"V40081"	"V40082"	"V40083"	"V4009"
# [61]	"V4010"	"V4012"	"V40121"	"V4013"	"V40132"
# [66]	"V40132A"	"V4014"	"V4015"	"V40151"	"V401511"
# [71]	"V401512"	"V4016"	"V40161"	"V40162"	"V40163"
# [76]	"V4017"	"V40171"	"V401711"	"V4018"	"V40181"
# [81]	"V40182"	"V40183"	"V4019"	"V4020"	"V4021"
# [86]	"V4022"	"V4024"	"V4025"	"V4026"	"V4027"
# [91]	"V4028"	"V4029"	"V4032"	"V4033"	"V40331"
# [96]	"V403311"	"V403312"	"V40332"	"V403321"	"V403322"
# [101]	"V40333"	"V403331"	"V4034"	"V40341"	"V403411"
# [106]	"V403412"	"V40342"	"V403421"	"V403422"	"V4039"
# [111]	"V4039C"	"V4040"	"V40401"	"V40402"	"V40403"
# [116]	"V4041"	"V4043"	"V40431"	"V4044"	"V4045"
# [121]	"V4046"	"V4047"	"V4048"	"V4049"	"V4050"
# [126]	"V40501"	"V405011"	"V405012"	"V40502"	"V405021"
# [131]	"V405022"	"V40503"	"V405031"	"V4051"	"V40511"
# [136]	"V405111"	"V405112"	"V40512"	"V405121"	"V405122"
# [141]	"V4056"	"V4056C"	"V4057"	"V4058"	"V40581"
# [146]	"V405811"	"V405812"	"V40582"	"V405821"	"V405822"
# [151]	"V40583"	"V405831"	"V40584"	"V4059"	"V40591"
# [156]	"V405911"	"V405912"	"V40592"	"V405921"	"V405922"
# [161]	"V4062"	"V4062C"	"V4063"	"V4063A"	"V4064"
# [166]	"V4064A"	"V4071"	"V4072"	"V4072A"	"V4073"
# [171]	"V4074"	"V4074A"	"V4075A"	"V4075A1"	"V4076"
# [176]	"V40761"	"V40762"	"V40763"	"V4077"	"V4078"
# [181]	"V4078A"	"V4082"	"VD2002"	"VD2003"	"VD2004"
# [186]	"VD3004"	"VD3005"	"VD3006"	"VD4001"	"VD4002"
# [191]	"VD4003"	"VD4004"	"VD4004A"	"VD4005"	"VD4007"
# [196]	"VD4008"	"VD4009"	"VD4010"	"VD4011"	"VD4012"

```
# [201] "VD4013"      "VD4014"      "VD4015"      "VD4016"      "VD4017"  
# [206] "VD4018"      "VD4019"      "VD4020"      "VD4023"      "VD4030"  
# [211] "VD4031"      "VD4032"      "VD4033"      "VD4034"      "VD4035"  
# [216] "VD4036"      "VD4037"      "Habitual"    "Efetivo"
```

- **Segundo exemplo** – Para baixar os dados referentes à PNAD Contínua anual do ano de 2019 para a primeira visita, pode-se usar o seguinte comando:

```
pnadc2019_visita1_dsg <- get_pnadc(2019,  
                                      interview = 1)
```

Nesse exemplo, os dados são armazenados no objeto `pnadc2019_visita1_dsg` cuja classe também é `survey.design`. No entanto, para trabalhar com os microdados no formato `data.frame`, pode-se fazer:

```
pnadc2019_visita1_df <- get_pnadc(2019,  
                                      interview = 1,  
                                      design = FALSE)
```

Para verificar a classe do objeto criado, basta que se aplique o comando:

```
class(pnadc2019_visita1_df)
```

```
# > [1] "tbl_df"       "tbl"          "data.frame"
```

Nesse caso, o objeto tem formato `tibble`. Como visto em capítulo anterior, essa é uma abordagem moderna para os tradicionais `data.frames`.

Outra forma de transformar os dados baixados no formato `survey.design` em `data.frame` é a seguinte:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df <- pnadc2019_visita1_dsg$variables
```

Assim sendo, acessa-se o `data.frame` que está contido no objeto `pnadc2019_visita1_dsg`, cuja classe é `survey.design`. Para verificar se deu certo, basta usar os comandos:

```
class(pnadc2019_visita1_df)
```

```
# >[1] "data.frame"
```

```
names(pnadc2019_visita1_df)
```

No caso da função `names()`, o resultado deve ser o mesmo visto anteriormente, quando se utilizou o comando `names(pnadc2021T1$variables)`.

O pacote **PNADcIBGE** também possibilita a transformação do formato `data.frame` em `survey.design`. Para isso, pode-se utilizar a função `pnadc_design`, que cria um objeto com todos os elementos associados ao desenho amostral complexo da PNAD Contínua, que pode ser utilizado para análises socioeconômicas por meio de pacotes como o `survey`. Sua forma básica é a seguinte:

```
pnadc_design(data_pnadc)
```

Seu único argumento é:

- `data_pnadc`: uma `tibble` ou um `data.frame` com os microdados da PNAD Contínua, lidos com a função `read_pnadc`.

Exemplo:

```
pnadc2019_visita1_dsg <- pnadc_design(pnadc2019_visita1_df)
```

O resultado desse comando é um objeto de classe `survey.design` que contém os dados da PNAD Contínua e todos os elementos que definem seu desenho amostral.

Com relação à aplicação de rótulos, pode-se deparar com uma situação em que se opte pelo argumento `labels = FALSE` na função `get_pnadc`, e, posteriormente, se queira aplicar os rótulos. Nesse caso, é possível usar o comando `pnadc_labeller`, que é uma função criada para aplicar os rótulos da variáveis a partir do dicionário da PNAD Contínua. Sua forma básica é a seguinte:

```
pnadc_labeller(data_pnadc,  
                 dictionary.file)
```

Seus argumentos são:

- `data_pnadc`: uma *tibble* que contenha os microdados da PNAD Contínua lidos com a função `read_pnadc`.
- `dictionary.file`: um arquivo `.xls` contendo o dicionário que deve ser baixado no *site* do IBGE⁴.

⁴ Quanto a essa documentação, para os dados trimestrais, deve-se selecionar o arquivo no formato `.zip`, disponível no endereço https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Trimestral/Microdados/Documentacao/; para os microdados anuais por entrevista, é necessário selecionar um arquivo `.xls`, de acordo com a entrevista e o ano desejados, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Microdados/Visita/); e para os microdados anuais por trimestre, deve-se selecionar um arquivo `.xls`, de acordo com o ano e o trimestre desejados, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Microdados/Trimestre/.

POR DENTRO DA PNAD CONTÍNUA

Exemplo:

```
pnadc2019_visita1_df <-  
  pnadc_labeller(pnadc2019_visita1_df,  
                  "C:/Downloads/dicionario_PNADC_mic rodados_  
                  2019_visita1_20201201.xls")
```

O resultado é uma *tibble* com os mic rodados da PNAD Contínua e suas variáveis categóricas dispostas como fatores, com seus respectivos rótulos.

A importação dos dados da PNAD Contínua para o RStudio pode ser feita, também, de forma *offline*. É possível ler os dados da PNAD Contínua a partir de arquivos em formato .txt por meio da função `read_pnadc`, cuja forma básica é seguinte:

```
read_pnadc(microdata,  
           input_txt,  
           vars = NULL)
```

Seus argumentos são:

- `microdata`: um arquivo no formato.txt contendo os mic rodados da PNAD Contínua que podem ser baixados no site do IBGE⁵.

⁵ Para os dados trimestrais deve-se selecionar o arquivo do ano e do trimestre desejado em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Trimestral/Mic rodados/; para os dados anuais por entrevista, pode-se acessar https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Mic rodados/Visita/; e para os dados anuais por trimestre, pode-se acessar https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Mic rodados/Trimestre/.

- `input_txt`: um arquivo no formato `.txt`, compatível com os microdados que se deseja, contendo o *input script* disponibilizado em linguagem de programação SAS⁶.
- `vars`: vetor contendo o nome das variáveis que se deseja analisar. O argumento padrão `NULL` mantém todas as variáveis.

Exemplo com a seleção de apenas duas variáveis:

```
pnadc2019_visita1_df_selec <-
  read_pnadc(microdata =
    "C:/Downloads/PNADC_2019_visita1.txt",
    input_txt =
    "C:/Downloads/input_PNADC_2019_visita1_20200826.txt",
    vars = c("VD4001", "VD4002"))
```

O resultado é uma *tibble* com as variáveis selecionadas dos microdados, incluindo as da estrutura básica da pesquisa, no formato `survey.design`.

Por fim, como os dados de rendimentos devem ser tratados em termos reais, o pacote possibilita a incorporação posterior ao *download* dos arquivos referentes aos deflatores construídos pelo próprio IBGE, especificamente, para a PNAD Contínua. As variáveis referentes a esses deflatores podem ser incorporadas por meio da função `pnadc_deflator`. Essa adiciona variáveis deflatoras aos microdados da PNAD Contínua já existentes.

⁶ Para os dados trimestrais, deve-se baixar o dicionário e o *input* contido em um arquivo `.zip`, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Trimestral/Microdados/Documentacao/; para os dados anuais por entrevista, deve-se selecionar o *input* em formato `.txt` de acordo como o ano e a entrevista adequados, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Microdados/Visita/; e para os dados anuais por trimestre, deve-se selecionar o *input* em formato `.txt` de acordo como o ano e o trimestre adequados, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Microdados/Trimestre/.

POR DENTRO DA PNAD CONTÍNUA

Para deflacionar as variáveis de renda, é recomendado que se utilizem o conjunto de deflatores criados pelo próprio IBGE⁷. Essa é uma função relevante para quem não deseja baixar todos os microdados novamente, quando o IBGE divulga uma nova base de dados (trimestral ou anual), pois isso altera o cálculo do deflacionamento de todos os dados de renda, uma vez que o usual é trazer as informações de renda para valores do último ano ou trimestre disponível (mais recente).

Essa função possui dois argumentos e assume a seguinte forma básica:

```
pnadc_deflator(data_pnadc,  
                 deflator.file)
```

Seus argumentos são:

- **data_pnadc**: uma *tibble* que contenha os microdados da PNAD Contínua.
- **deflator.file**: um arquivo .xls contendo os deflatores (incluindo o caminho para o diretório onde o arquivo se encontra) da pesquisa selecionada⁸.

⁷ A documentação para o deflacionamento dos dados trimestrais pode ser acessada em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Trimestral/Micrdados/Documentacao/PNADCIBGE_Deflator_Trimestral.pdf. Para os dados anuais, baixados por entrevista, deve-se acessar: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Micrdados/Visita/Documentacao_Geral/PNADCIBGE_Deflator_Anual_Visita.pdf. Já para os dados anuais, baixados por trimestre, pode-se visitar o endereço: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Micrdados/Trimestre/Documentacao_Geral/PNADCIBGE_Deflator_Anual_Trimestre.pdf.

⁸ Quanto aos deflatores dos microdados trimestrais, deve-se selecionar o arquivo disponibilizado em formato .zip que se encontra no endereço https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Trimestral/Micrdados/Documentacao/; para os dados anuais, baixados por visita, deve-se baixar o arquivo no formato .xls de acordo com o ano apropriado no endereço https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Micrdados/Visita/Documentacao_Geral/; e, para os dados anuais por trimestre, selecionar um arquivo .xls compatível com o trimestre definido e, dentro da pasta da documentação, baixar o ano desejado, acessando https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Micrdados/Trimestre/.

Exemplo:

```
pnadc2019_visita1_df <-
  pnadc_deflator(pnadc2019_visita1_df,
                  "C:/Downloads/deflator_PNADC_2019.xls")
```

6.3 Importação e *download* dos microdados da antiga PNAD anual

O pacote **lodown**, desenvolvido por Damico (2022)⁹, facilita o processo de *download* e importação dos microdados da PNAD em sua versão anual, que foi encerrada em 2015. Para tanto, devem-se adotar os seguintes passos:

- Instalar o pacote **lodown**.

```
install.packages("devtools", repos = "http://cran.rstudio.com/")
library(devtools)

install_github("ajdamico/lodown", dependencies = TRUE)
library(lodown)
```

- Criar um catálogo com as informações básicas para importação dos microdados disponíveis no site do IBGE.

O pacote **lodown** cria um catálogo com as informações necessárias para a extração dos microdados diretamente da página do IBGE, por meio da função `get_catalog()`.

⁹ Anthony Joseph Damico publicou, ainda, um livro que traz o passo a passo para explorar microdados públicos de distintas fontes de todo o mundo. Para mais detalhes, ver Damico (2019).

POR DENTRO DA PNAD CONTÍNUA

```
# Cria um catálogo com todos os microdados disponíveis  
# da PNAD no site do IBGE  
  
catalogo <-  
  get_catalog("pnad",  
    output_dir = file.path("C:/Bases de Dados",  
      "PNAD"))
```

Até o presente momento, esse catálogo definido pelo pacote **lodown** encontra-se desatualizado devido às mudanças realizadas pelo IBGE nos endereços em que se encontram os microdados. No entanto, verificou-se que o problema encontra-se na definição inicial do endereço. No catálogo do pacote, consta a expressão “ftp:” em vez de “http:”. Esse simples problema pode ser corrigido por meio da função `str_replace()` do pacote **stringr**, desenvolvido por Wickham (2019) e presente no **tidyverse**. Para corrigi-lo, basta aplicar os seguintes comandos¹⁰:

```
catalogo <-  
  catalogo %>%  
  mutate(ftp_folder = str_replace(ftp_folder, "ftp:", "http:"),
        sas_ri = str_replace(sas_ri, "ftp:", "http:"),
        full_url = str_replace(full_url, "ftp:", "http:"))
```

- Baixar todos os dados disponíveis ao mesmo tempo.

Por meio da função `lodown`, é possível fazer o *download* de todos os microdados disponíveis de uma única vez informando à função o argumento “`pnad`” além do diretório de destino dos dados em `output_dir`. Porém, deve-se ter claro que esse procedimento pode demorar significativamente, pois depende da velocidade de conexão e de processamento do computador do usuário.

¹⁰ Deve-se inspecionar o objeto catálogo antes de rodar esse comando para saber se houve alguma atualização e se sua definição foi corrigida.

```
lodown("pnad",
       catalog = catalogo,
       output_dir = file.path("C:/Bases de Dados",
"PNAD"))
```

Por outro lado, é possível baixar os dados para um único período. Depois de criar o catálogo e de corrigi-lo (caso necessário), é possível definir um subconjunto (`subset`) para que a função `lodown()` importe um ano específico. No exemplo a seguir, faz-se o *download* dos microdados do ano de 2015.

```
# Baixa os microdados, salvando no diretório especificado, o ano
# definido no argumento year
lodown("pnad",
       subset(catalogo, year == 2015),
       output_dir = file.path("C:/Bases de Dados",
"PNAD"))
```

Para importar para o Rstudio os microdados gerados e salvos no diretório do HD do computador, basta aplicar os seguintes comandos:

```
pnad_2015_df <-
  readRDS("C:/Bases de Dados/PNAD/2015 main.rds")

# Verificar o formato do objeto pnad_2015_df
class(pnad_2015_df)

# > [1] "data.frame"
```

No entanto, o pacote **lodown** deve ser utilizado com parcimônia, uma vez que sua atualização não é recorrente. Assim, destaca-se que o método tradicional de

POR DENTRO DA PNAD CONTÍNUA

baixar os microdados diretamente do repositório `ftp` do IBGE¹¹ e fazer sua leitura a partir do disco rígido do computador mostra-se uma opção mais segura.

Para tanto, pode-se usar um pacote específico do R que possibilita a tradução das rotinas em linguagem **SAS** disponibilizadas pelo IBGE¹², a saber, o **SAScii**, desenvolvido por Damico (2012). Para instalá-lo e habilitá-lo, basta aplicar os seguintes comandos:

```
install.packages("SAScii")
library(SAScii)
```

Com esse pacote, é possível criar um `data.frame` no R a partir de um arquivo ASCII¹³ e de um conjunto de instruções de importação em linguagem **SAS**.

Uma de suas funções básicas é a `parse.SAScii`, que converte as instruções de importação em **SAS** em argumentos para uma chamada de função do tipo `read.fwf`, do pacote `readr`, desenvolvido por Wickham e Hester (2020). Sua lógica é reconfigurar o bloco `INPUT` de um arquivo de sintaxe (`.sas`) nos argumentos necessários para executar a função `read.fwf` em um conjunto de dados ASCII (`.dat` ou `.txt`, por exemplo).

Seus principais argumentos são os seguintes:

- `sas_ri`: sequência de caracteres contendo a localização das instruções de importação do SAS;
- `beginline`: número da linha no arquivo de importação do SAS onde a instrução `INPUT` começa. Se a palavra `INPUT` aparecer antes do bloco `INPUT` real, a função retornará um erro;

¹¹ Ver http://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_anual/microdados/.

¹² Para anos mais recentes, o IBGE tem disponibilizado arquivos de leitura em R, porém, para os microdados dos anos 1970, 1980 e 1990, só estão disponíveis os dicionários e os *Inputs* em **SAS**.

¹³ ASCII é um tipo de arquivo que contém textos não formatados, podendo ser: caracteres, números, pontuação, tabulações etc. Normalmente, esses arquivos podem ser lidos e editados por meio do “Bloco de notas” da Microsoft.

- `lrecl`: opção LRECL do código em **SAS**, fornecida pelo próprio IBGE. Apenas necessário se a largura do arquivo ASCII for maior do que as colunas reais que contêm os dados (essa opção serve para corrigir os arquivos que possuem espaços vazios no lado direito do banco de dados).

Dentre os principais argumentos da função `read.fwf`, destacam-se:

- `file`: um caminho para um arquivo, uma conexão ou os dados em distintos formatos como: `.dat`; `.txt` ou `.zip`. Estes últimos serão descompactados automaticamente. Arquivos definidos por `http://`, `https://`, `ftp://` ou `ftps://` serão baixados automaticamente;
- `col_positions`: posições das colunas, conforme criado pelas funções `fwf_empty()`, `fwf_widths()` ou `fwf_positions()`. Para ler apenas os campos selecionados, use `fwf_positions()`;
- `col_types`: argumento que pode assumir as formas `NULL`, uma especificação do tipo `cols()` (para informar manualmente os tipos de variáveis) ou uma `string`. Caso esse argumento seja definido como `NULL`, todos os tipos de coluna serão imputados a partir das características das primeiras 1000 linhas do arquivo de entrada. Isso pode ser conveniente (e rápido), mas pode conter muitos erros de palpite do R ao interpretar o tipo da variável. No caso da especificação de coluna definida em `cols()`, deve-se informar manualmente uma especificação para cada variável. Caso se queira apenas ler um subconjunto de variáveis, pode-se usar `cols_only()`. Alternativamente, pode-se utilizar uma representação na forma de uma `string` compacta em que cada caractere representa o tipo de uma variável na ordem que elas são definidas no arquivo de entrada (`c` = character; `i` = integer; `n` = number; `d` = double; `l` = logical; `f` = factor; `D` = date; `T` = date time; `t` = time; `?` = guess; `_` or `-` = skip);
- `progress`: exibe uma barra de progresso. A barra de progresso automática pode ser desabilitada definindo a opção `readr.show_progress = FALSE`.

A seguir, reproduz-se um pedaço (primeiras e últimas linhas) do arquivo `.txt` que o IBGE disponibiliza juntamente com os microdados das PNADs anuais. Com isso, pode-se ter uma noção mais precisa do formato da sintaxe de leitura dos microdados da PNAD em **SAS** que é usado pela função `read.fwt`. Deve-se reparar

POR DENTRO DA PNAD CONTÍNUA

que essa função inicia sua operação a partir da declaração do bloco de instruções INPUT da rotina SAS, na linha informada no argumento beginline.

```
/* LAYOUT PARA LEITURA DOS DADOS DE DOMICILIO DA PNAD 92 */
DATA DOMICS;
    INFILE 'nome do arquivo a ser lido' LRECL=165;
    INPUT
        @1      V0101  $CHAR2.   /* ANO DA PESQUISA          */
        @3      UF      $CHAR2.   /* CODIGO DA UNIDADE DA FEDERACAO */
        @5      CONTROL $CHAR6.  /* NUMERO DE CONTROLE        */
        @11     V0103  $CHAR3.   /* NUMERO DE SERIE          */
        @14     V0104  $CHAR2.   /* TIPO DE ENTREVISTA        */
        @16     V0105      2.    /* TOTAL DE MORADORES        */
        (...)                                 /*                         */
        @123    V4609      9.    /* PROJECAO DE POPULACAO DEPOP */
        @132    V4610      3.    /* INVERSO DA FRACAO          */
        @135    V4611      5.    /* PESO(TOTAL PESSOAS AMOSTRA) */
        @140    V4614     12.    /* Rendimento mensal domiciliar */
        @152    BRANCOS $CHAR14. /* Bytes em branco           */
    ; RUN;
```

Exemplo: Leitura dos dados da PNAD anual de 1992¹⁴.

```
# Leitura dos micrados da PNAD 1992

#Bibliotecas necessárias
library(readr)
library(data.table)
library(SAScii)
library(tidyverse)
```

¹⁴ Para baixar os micrados e a sintaxe de leitura em SAS, deve-se acessar: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_anual/micrados/1992/.

```
#Base de domicílios
# Cria dicionário/input em formato data.frame
dic_dom1992 <-
  parse.SAScii("C:/Bases de Dados/PNAD/1992/Layout/sas/SAS_DOM92.txt",
               beginline = 4,
               lrecl = 165)

#Converte o data.frame em um data.table
setDT(dic_dom1992) # convert to data.table

#Cria a string contendo os tipos de variáveis necessárias para
# informar em col_types
dic_dom1992 <-
  dic_dom1992 %>%
  mutate(id_char = case_when(char == TRUE ~ "c",
                             char == FALSE ~ "n"))

id_char_dom <- paste(dic_dom1992$id_char, collapse = "")

# Lê o arquivo .txt ou .dat usando o objeto dicionário
pnaddom92 <-
  read_fwf("C:/Bases de Dados/PNAD/1992/Dados/DOM92.DAT",
           fwf_widths(dput(dic_dom1992[,width]),
                       col_names=(dput(dic_dom1992[,varname]))),
           progress = interactive(),
           col_types = id_char_dom)

#Verifica as especificações dos tipos de variáveis criadas
spec(pnaddom92)

#Base de pessoas
# Cria dicionário no formato data.frame
dic_pes1992 <-
```

POR DENTRO DA PNAD CONTÍNUA

```
parse.SASci("C:/Bases de Dados/PNAD/1992/Layout/sas/SAS_PES92.txt",
            beginline = 4,
            lrecl = 1552)

#Converte o data.frame em um data.table
setDT(dic_pes1992) # convert to data.table

#Cria a string contendo os tipos de variáveis necessárias para
# informar em col_types
dic_pes1992 <-
  dic_pes1992 %>%
  mutate(id_char = case_when(char == TRUE ~ "c",
                             char == FALSE ~ "n"))

id_char_pes <- paste(dic_pes1992$id_char, collapse = "")

# Lê o arquivo .txt ou .dat usando o objeto dicionário/input
pnadpes92 <-
  read_fwf("C:/Bases de Dados/PNAD/1992/Dados/PES92.DAT",
           fwf_widths(dput(dic_pes1992[,width]),
                       col_names=(dput(dic_pes1992[,varname]))),
           progress = interactive(),
           col_types = id_char_pes)

#Verifica as especificações dos tipos de variáveis criadas
spec(pnadpes92)

#Realiza a fusão das bases de pessoas e domicílios
pnad1992 <-
  inner_join(pnaddom92,pnadpes92,
             by=c("V0101","UF","CONTROL","V0103"))

#Salva os arquivos criados em um diretório do HD no formato .rds
```

```
saveRDS(pnaddom92,  
        "C:/Bases de Dados/PNAD/1992/pnad1992dom.rds")  
  
saveRDS(pnadpes92,  
        "C:/Bases de Dados/PNAD/1992/pnad1992pes.rds")  
  
saveRDS(pnad1992,  
        "C:/Bases de Dados/PNAD/1992/pnad1992.rds")
```

Deve-se ter claro que a fusão das bases de dados por meio da função `inner_join` mantém apenas os casos que aparecem com as chaves definidas nas duas bases de dados. Isso implica que, caso haja algum domicílio vazio (sem registro de moradores) na base de domicílios, esses não farão parte na base agregada.

Deve-se fazer uma ressalva com relação ao uso da função `parse.SASci`, qual seja, ela não pode ser usada em caso de sobreposição de início de colunas/variáveis no arquivo contendo os microdados. Sua aplicação, nesse caso, retornará um erro. Essa situação é recorrente nas bases de dados dos anos 2000, em que o início de uma variável é o mesmo de outra. Por exemplo, as variáveis `UF` (Unidade da Federação) e `V0102` (Número de Controle) possuem a mesma posição inicial no banco de dados da PNAD de 2001. A solução para esses casos é a realização de alterações nos arquivos de dicionário, disponibilizados pelo IBGE no formato `.xls`, para que eles apresentem apenas quatro colunas com as seguintes informações: posição inicial; tamanho, nome da variável e descrição, devendo ser salvos em formato `.csv` (ver Figura 6.1).

A sequência de comandos, a seguir, permite a abertura dos microdados da PNAD anual de 2001, tanto para a base de pessoas quanto de domicílios. Para isso, serão utilizadas as funções `read_csv2`, do pacote `readr` (WICKHAM; HESTER, 2020), e `fwf2csv`, do pacote `descr`, criado por Aquino (2021). A fusão dos dois arquivos em uma única base é realizada com o auxílio do pacote `dplyr`, que será apresentado no Capítulo 7. A importação dos microdados para o R fica a cargo da função `fread`, da biblioteca `data.table` (DOWLE; SRINIVASAN, 2021).

POR DENTRO DA PNAD CONTÍNUA

Figura 6.1: Dicionário original .xls (à esquerda) e arquivo .csv alterado (à direita), necessário para a abertura das bases da PNAD anual que apresentem sobreposição de colunas/variáveis.

Dicionário da PNAD2001-microdados – Arquivo de Domicílios									
Posição Inicial	Tamanho	Código de variável N°	Questão		Categorias				
			Descrição	Tipo	Descrição				
1	4	V0101	Ano de Referência						
PARTE 1 – IDENTIFICAÇÃO E CONTROLE									
5	2	UF	Unidade da Federação		2 primeiras posições do controle				
8	5	V0102	2	Número de controle					
9	13	V0103	3	Número de Série					
10			4		TIPO A UNIDADE OCUPADA				
11					1 Residenciada				
12					2 Fazenda				
13					3 Recusa				
14					4 Outra				
15					TIPO B UNIDADE VAGA				
16	16	V0104	2		5 Em condições de ser habitada				
17					6 Uso ocasional				
18					7 Construção ou reforma				
19					8 Entradas				
20					9 Demóida				
21					10 Não foi encontrada				
22					11 Não residencial				
23					12 Fora do setor				
24									
25	18	V0105	2	5	Total de Moradores				
26	20	V0106	2	6	Total de Moradores de 5 anos ou mais				
27						1 Particular permanente			
28						3 Particular improvisado			
29									

DicDom2001									
A1	B	C	D	E	F	G	H	I	J
1	4 V0101	Ano de Referência							
2	5	2 UF	Unidade de Federação						
3	5	8 V0102	Número de controle						
4	13	3 V0103	Número de Série						
5	16	2 V0104	Tipo de Entrevista						
6	18	2 V0105	Total de Moradores						
7	20	2 V0106	Total de Moradores de 5 anos ou mais						
8	22	1 V0201	Unidade de Domicílio						
9	23	1 V0202	Tipo de domicílio						
10	24	1 V0203	Material predominante das paredes externas						
11	25	1 V0204	Material predominante na cobertura (telhado)						
12	26	2 V0205	Número de cômodos no domicílio						
13	28	2 V0206	Número de cômodos servindo de dormitório						
14	30	5 V0207	Número de quartos de banho						
15	31	12 V0208	Aluguel mensal pago em 09 / 2001						
16	43	1 V0201	Código de valor de aluguel						
17	44	12 V0209	Prestação mensal pago em 09 / 2001						
18	56	1 V0209	Código de valor de prestação						
19	57	1 V0210	O terreno é próprio						
20	58	1 V0211	Água canalizada em, pelo menos, um cômodo						
21	59	1 V0212	Água canalizada em todo o domicílio, a proveniente de						
22	60	1 V0213	Água canalizada da rede de distribuição para a propriedade						
23	61	1 V0214	Água utilizada no domicílio é de poço ou nascente localizado na propriedade						
24	62	1 V0215	Existe banheiro ou sanitário no domicílio ou na propriedade						
25	63	1 V0216	Este banheiro é de uso...						
26	64	1 V0217	De que forma é feito o encanamento deste banheiro ou sanitário						
27	65	1 V0218	Este banheiro é de uso...						
28	66	1 V0219	Forma de iluminação do domicílio						
29	67	1 V0220	Este domicílio tem telefone móvel celular						
30	68	1 V0209	Este domicílio tem telefone fixo						
31	69	1 V0221	Este domicílio tem fogão de 2 ou mais bocas						
32	70	1 V0222	Este domicílio tem fogão de uma boca						
33	71	1 V0223	Este domicílio tem fogão de duas bocas						
34	72	1 V0224	Este domicílio tem algum tipo de filtro d'água						
35	73	1 V0225	Este domicílio tem rádio						
36	74	1 V0226	Este domicílio tem televisão em cores						
37	75	1 V0227	Este domicílio tem televisão em preto e branco						

Fonte: IBGE, Dicionário de variáveis PNAD 2001 Anual (base de domicílios).

```
#Bibliotecas
library(data.table)
library(descr)
library(tidyverse)
library(readxl)

#Base de domicílios
# Cria um objeto com o dicionário alterado em formato .csv
dicdom2001 <-
  read_csv(
    file = "C:/Bases de Dados/PNAD/2001/Dicionário/DicDom2001.csv",
    col_names = F)

#define os nomes das variáveis do dicionário
```

```
colnames(dicdom2001) <- c('inicio', 'tamanho', 'variavel', 'nomes')

#Cria variável que identifica o final da posição de cada variável
dicdom2001 <-
  dicdom2001 %>%
  mutate(end_dom = inicio + tamanho - 1)

# Transforma os microdados em um arquivo .csv com o auxílio do objeto
# contendo o dicionário
fwf2csv(fwffile='C:/Bases de Dados/PNAD/2001/Dados/DOM2001.txt',
         csvfile='PNAD_dom_2001.csv',
         names=dicdom2001$variavel,
         begin=dicdom2001$inicio,
         end=dicdom2001$end_dom)

# Lê os microdados com a função fread do pacote data.table
pnaddom2001 <- fread(input='PNAD_dom_2001.csv',
                      sep='auto',
                      sep2='auto',
                      integer64='double')

#Base de pessoas
# Cria um objeto com o dicionário alterado em formato .csv
dicipes2001 <-
  read_csv2(
    file ="C:/Bases de Dados/PNAD/2001/Dicionário/DicPes2001.csv",
    col_names = F)

#Define os nomes das variáveis do dicionário
colnames(dicipes2001) <- c('inicio', 'tamanho', 'variavel', 'nomes')

#Cria variável que identifica o final da posição de cada variável
dicipes2001 <- dicipes2001 %>%
```

POR DENTRO DA PNAD CONTÍNUA

```
mutate(end_pes = inicio + tamanho - 1)

# Transforma os microdados em um arquivo .csv com o auxílio do objeto
# contendo o dicionário
fwf2csv(fwffile='C:/Bases de Dados/PNAD/2001/Dados/PES2001.txt',
         csvfile='PNAD_pes_2001.csv',
         names=dicpes2001$variavel,
         begin=dicpes2001$inicio,
         end=dicpes2001$end_pes)

# Lê os microdados com a função fread do pacote data.table
pnadpes2001 <- fread(input='PNAD_pes_2001.csv',
                      sep='auto',
                      sep2='auto',
                      integer64='double')

# Realiza a operação de fusão das duas bases pelas variáveis de
# identificação do domicílio
pnad2001 <-
  inner_join(pnaddom2001,pnadpes2001,
             by=c("V0101","UF","V0102","V0103"))

#Salva os arquivos criados no diretório escolhido
saveRDS(pnaddom2001,
         "C:/Bases de Dados/PNAD/2001/pnad2001dom.rds")

saveRDS(pnadpes2001,
         "C:/Bases de Dados/PNAD/2001/pnad2001pes.rds")

saveRDS(pnad2001,
         "C:/Bases de Dados/PNAD/2001/pnad2001.rds")
```

Por fim, cabe destacar que, assim como a PNAD Contínua, a antiga PNAD

anual pode ser tratada e explorada no tradicional formato `data.frame` e no formato `survey.design`.

A seguir, apresenta-se um conjunto de comandos para transformar o objeto `pnad_2015_df`, `data.frame` criado anteriormente, em um outro objeto no formato `survey.design`. Para isso, basta seguir os seguintes passos:

```
# Habilitar a biblioteca survey
library(survey)

# Transformar os dados
options(survey.lonely.psu = "adjust")

pop_types <-
  data.frame(v4609 = unique(pnad_2015_df$v4609),
             Freq = unique(pnad_2015_df$v4609))

# Criar um objeto auxiliar com pre-estratificação
prestratified_design <-
  svydesign(id = ~ v4618,
            strata = ~ v4617 ,
            data = pnad_2015_df ,
            weights = ~ pre_wgt ,
            nest = TRUE)

# Liberar memoria RAM
gc()

# Definir o objeto final com pos-estratificação
pnad_2015_dsg <-
  postStratify(design = prestratified_design,
                strata = ~ v4609,
                population = pop_types)
```

POR DENTRO DA PNAD CONTÍNUA

```
# Remover o objeto auxiliar e liberar memoria  
rm(prestratified_design)  
gc()
```

Com isso, acredita-se que os próximos passos sejam introduzir os conhecimentos básicos necessários para o tratamento dos dados da PNAD. O foco principal dos próximos capítulos serão os microdados da PNAD Contínua. O objetivo é apresentar distintos pacotes do R para facilitar a vida de cientistas e pesquisadores no campo das Ciências Sociais Aplicadas.

7 Tratamento dos microdados da PNAD

Contínua: o pacote **tidyverse**

O **tidyverse**, elaborado por Wickham *et al.* (2019), é uma coleção de pacotes em R projetada para ciência de dados que compartilha uma filosofia de design, de gramática e de estruturas de dados¹. No **tidyverse 1.3.0**, pesquisadores e cientistas contam com a possibilidade de usar as funções de pacotes como: **ggplot2**, que cria gráficos a partir de uma Gramática de Gráficos em que se fornecem dados, mapeiam-se variáveis, define-se uma estética e ajustam-se os detalhes segundo critérios próprios; **dplyr**, que é uma gramática específica para o tratamento de dados; **tidyr**², que conta com um conjunto de funções que possibilitam a organização de dados – foco do presente capítulo; **readr**, que lê dados em formato de planilhas dos tipos **.csv**, **.tsv** e **.fwf**; **tibble**; dentre outros³.

7.1 Instalação

Para instalar o **tidyverse**, basta aplicar os seguintes comandos:

¹ Mais informações sobre o pacote podem ser vistas em <https://tidyverse.tidyverse.org>.

² Ver Wickham (2021).

³ O livro *R for Data Science: Import, Tidy, Transform, Visualize, and Model*, de Wickham e Grolemund (2016), traz um vasto conjunto de aplicações em ciência de dados e está disponível de forma aberta e livre em: <https://r4ds.had.co.nz/>.

```
install.packages("tidyverse")
library(tidyverse)
```

7.2 O pacote **dplyr**: funções básicas

O **dplyr**, desenvolvido por Wickham *et al.* (2021), pode ser definido como uma gramática de tratamento de dados que possui um conjunto expressivo de funções para enfrentar os diversos desafios comuns na vida de pesquisadores e cientistas de dados. Sua principal característica é trabalhar a partir de uma estrutura de construção baseada na conexão de termos por meio dos chamados *pipes*, que funcionam da seguinte forma: $x \%>\% y$, que nada mais é do que a representação simbólica de $f(y) \Rightarrow f(x, y)$ ⁴.

Não se pode deixar de mencionar que esse pacote permite o tratamento de `data.frames` e `tibbles`, em que cada variável encontra-se em sua própria coluna e cada caso em sua própria linha. Nesse sentido, o **dplyr** permite o tratamento de casos e de variáveis, isto é, de linhas e colunas.

Com o objetivo de ilustrar algumas das funcionalidades do **dplyr**, serão utilizados os dados do quarto trimestre de 2020 da PNAD Contínua do IBGE para algumas variáveis selecionadas. Para baixar o conjunto de dados, pode-se aplicar o seguinte comando:

⁴ Para mais detalhes sobre o funcionamento dos *pipes*, ver Bache e Wickham (2020).

```
pnadc2020T4 <-
  get_pnadc(2020, quarter = 4,
             design = FALSE, labels = T,
             vars = c("Ano", "Trimestre", "UF",
                     "UPA", "Estrato", "V1028",
                     "V2007", "V2009", "V2010",
                     "V3007", "VD3004", "VD4001",
                     "VD4002", "VD4005", "VD4019",
                     "VD4020", "VD4035", "Habitual",
                     "Efetivo"))
```

Deve-se reparar que o objeto de nome pnadc`2020T4` recebe a importação dos dados, direto do site do IBGE, no formato `data.frame`, com os respectivos *labels*. A descrição das variáveis selecionadas pode ser vista na Tabela 7.1.

Tabela 7.1: Descrição de variáveis da PNAD Contínua trimestral.

Variável	Descrição
UF	Unidade da Federação
V1028	Peso trimestral com correção de não entrevista com pós-estratificação pela projeção de população
V2007	Sexo
V2009	Idade do morador na data de referência
V2010	Cor ou raça
V3007	Ja concluiu algum outro curso de graduação?
VD3004	Nível de instrução mais elevado alcançado (pessoas de 5 anos ou mais de idade)
VD4001	Condição em relação à força de trabalho na semana de referência para pessoas de 14 anos ou mais de idade
VD4002	Condição de ocupação na semana de referência para pessoas de 14 anos ou mais de idade
VD4005	Pessoas desalentadas na semana de referência

POR DENTRO DA PNAD CONTÍNUA

Descrição de variáveis da PNAD Contínua trimestral (continuação).

Variável	Descrição
VD4019	Rendimento mensal habitual de todos os trabalhos para pessoas de 14 anos ou mais de idade (apenas para pessoas que receberam em dinheiro, produtos ou mercadorias em qualquer trabalho)
VD4020	Rendimento mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (apenas para pessoas que receberam em dinheiro, produtos ou mercadorias em qualquer trabalho)
VD4035	Horas efetivamente trabalhadas na semana de referência em todos os trabalhos para pessoas de 14 anos ou mais de idade
Habitual	Deflator para obtenção dos valores reais, a preços médios do último trimestre divulgado, das variáveis de rendimento habitual
Efetivo	Deflator para obtenção dos valores reais, a preços médios do último trimestre divulgado, das variáveis de rendimento efetivo

Fonte: IBGE, Documentação da PNAD Contínua trimestral, Dicionário de variáveis (2020)

Caso o pacote **tidyverse** não tenha sido habilitado, pode-se carregar apenas o **dplyr**. Basta aplicar o seguinte comando:

```
library(dplyr)
```

Dentre suas principais funções pensadas para tratar conjuntos de dados nos formatos **data.frame** e/ou **tibble**, destacam-se algumas a seguir:

- **mutate()**: adiciona novas variáveis em função de variáveis existentes ou não;
- **select()**: possibilita a escolha de variáveis com base em seus nomes, para serem mantidas ou excluídas do conjunto de dados;
- **filter()**: permite escolher casos com base em seus valores, podendo ser utilizado para variáveis categóricas, contínuas ou uma combinação de distintos tipos de variáveis;

- `summarise()`: possibilita a criação/transformação de variáveis por meio de uma fórmula de cálculo predefinida;
- `arrange()`: permite alterar/reordenar as linhas de um conjunto de dados;
- `group_by()`: possibilita que se execute qualquer uma dessas operações “por grupo”.

7.3 Síntese de dados

As funções de síntese permitem o cálculo de distintas estatísticas por colunas, isto é, variáveis, criando uma nova tabela com os resultados. Esse tipo de função toma um vetor como entrada e retorna um valor síntese.

- `summarise`: cria uma nova tabela com uma ou mais linhas para cada combinação de variáveis de agrupamento; se não houver variáveis de agrupamento, a saída terá uma única linha resumindo todas as observações definidas na entrada. A tabela de saída conterá uma coluna para cada variável de agrupamento e uma coluna para cada uma das estatísticas sintetizadas especificadas.

Exemplo: Cria uma tabela síntese com o rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (`VD4020`) na PNAD Contínua trimestral:

```
pnadc2020T4 %>%  
  summarise("Rendimento médio" =  
            weighted.mean(VD4020,  
                            w = V1028,  
                            na.rm = TRUE))
```

- `count`: permite contar valores únicos de uma ou mais variáveis (fatores), por meio da função `group_by`⁵. Essa função conta o número de linhas do conjunto de dados de acordo com a variável definida.

⁵ A função `group_by()` será apresentada posteriormente.

POR DENTRO DA PNAD CONTÍNUA

A gramática para essa função pode assumir duas formas:

1. Forma básica:

```
df %>%
  count (a, b)
```

2. Forma equivalente:

```
df %>%
  group_by (a, b) %>%
  summarise (n = n ())
```

Exemplo: Cria uma tabela síntese com a contagem de pessoas segundo Unidades da Federação na PNAD Contínua trimestral:

```
pnadc2020T4 %>%
  count (UF,
        wt = V1028,
        name = "Número de pessoas")
```

7.4 Agrupamento de casos

O uso da função `group_by` permite criar uma cópia de uma tabela agrupada por uma ou mais variáveis. O `dplyr` trata cada grupo separadamente e combina os resultados.

- `group_by`: converte uma `tbl` existente em uma `tbl` agrupada, na qual as operações são realizadas “por grupo”.

Exemplo: Cria uma tabela síntese com o rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020), por Unidade da Federação, na PNAD Contínua trimestral:

```
pnadc2020T4 %>%
  group_by(UF) %>%
  summarise ("Rendimento médio" =
    weighted.mean(VD4020,
      w = V1028,
      na.rm = TRUE))
```

- `rowwise`: permite que se calculem valores de variáveis em um conjunto de dados, linha por linha.

Exemplo: Adiciona uma variável com a média do rendimento médio mensal habitual e efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4019 e VD4020, respectivamente):

```
pnadc2020T4 %>%
  rowwise() %>%
  mutate ("Média efetivo/habitual" =
    mean(c_across(VD4019:VD4020)))
```

7.5 Tratamento de casos

Deve ficar claro que tratar/alterar casos é transformar as linhas de um conjunto de dados. Isso pode se dar por meio de extração, arranjo ou adição de casos/linhas.

POR DENTRO DA PNAD CONTÍNUA

7.5.1 Extração de casos

As funções que alteram as linhas retornam um subconjunto de casos na forma de uma nova tabela. Dentre as principais funções disponíveis destacam-se:

- `filter`: extrai linhas de acordo com uma determinada condição.

Exemplo: Extrai os dados para o Rio Grande do Norte:

```
pnadc2020T4 %>%  
  filter(UF == "Rio Grande do Norte")
```

Os principais operadores relacionais admitidos por essa função são:

```
x < y  
x > y  
x <= y  
x >= y  
x == y  
x != y
```

Os principais operadores lógicos admitidos são:

```
! x  
x & y  
x && y  
x | y  
x || y  
xor(x, y)  
isTRUE(x)  
isFALSE(x)
```

Ambos os tipos de operadores podem ser passados enquanto elementos condicionais para filtrar (`filter()`) o conjunto de dados originais. Para saber mais sobre esses operadores, basta aplicar o seguinte comando no console ou no *Script*:

```
?Comparison  
?base::Logic
```

- `distinct`: extrai linhas de acordo com uma determinada condição.

Exemplo: Extrai dados não duplicados para as variáveis UPA, Estrato e V1008 (variáveis de identificação dos domicílios da PNAD Contínua), soltando como resultado uma nova base cujo total de casos é o total de domicílios da amostra:

```
pnadc2020T4 %>%  
  distinct(UPA, Estrato, V1008, .keep_all = T)
```

- `slice`: extrai linhas de acordo com suas posições.

Exemplo: Extrai dados para as primeiras 1000 linhas:

```
pnadc2020T4 %>%  
  slice(1:1000, .preserve = T)
```

- `slice_sample`: extrai casos de forma randomizada para um número específico de casos (n) ou para uma proporção definida (prop).

Exemplo: Extrai dados randomizados para 25% da população, isto é, da amostra expandida pelos pesos (V1028)⁶, quando se utiliza o argumento `weight_by`:

⁶ Para selecionar essa mesma proporção da amostra, basta utilizar o argumento `weight_by = NULL`.

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2020T4 %>%
  slice_sample(weight_by = V1028, prop = .25)
```

- `slice_min`: extrai casos com os menores valores de uma variável específica;
- `slice_max`: extrai casos com os maiores valores de uma variável específica;
- `slice_head`: extrai casos para os primeiros registros da amostra;
- `slice_tail`: extrai casos para os últimos registros da amostra.

Exemplos: Extrai dados mínimos, máximos, do topo e da base do conjunto dos dados da PNAD trimestral:

```
pnadc2020T4 %>%
  slice_min(order_by = VD4020, prop = .10)

pnadc2020T4 %>%
  slice_max(order_by = VD4020, prop = .10)

pnadc2020T4 %>%
  slice_head(n = 1000)

pnadc2020T4 %>%
  slice_tail(n = 1000)
```

7.5.2 Arranjo de casos

É possível ordenar as posições das linhas de um conjunto de dados por meio da função `arrange()`. Tal arranjo pode se dar pelos valores de uma ou mais variáveis (colunas). A ordenação ascendente não exige a definição de qualquer argumento. Já se o usuário desejar ordenar de forma descendente os dados, é necessário aplicar a função `desc()`.

- `arrange`: ordena os casos a partir de uma ou mais variáveis específicas.

Exemplo: Ordenando os dados a partir das variáveis de identificação do domicílio de forma ascendente e descendente, respectivamente:

```
pnadc2020T4 %>%
  arrange(UPA,Estrato, V1008)

pnadc2020T4 %>%
  arrange(desc(UPA,Estrato, V1008))
```

7.5.3 Adição de casos

Talvez seja necessário, por algum motivo, adicionar linhas com valores a uma tabela ou a um conjunto de dados. Para isso, existe uma função denominada `add_row()`.

Exemplo: Adiciona casos a um objeto denominado `obj` definido a partir da função `distinct()` apresentada anteriormente:

```
obj <- pnadc2020T4 %>%
  distinct(Ano,V2007)

obj

## # A tibble: 2 x 2
#   Ano    V2007
#   <chr> <fct>
# 1 2020  Mulher
# 2 2020  Homem

obj %>%
  add_row(Ano = "2020", V2007 = "Total")
```

```
## # A tibble: 3 x 2
#   Ano    V2007
#   <chr> <chr>
# 1 2020  Mulher
# 2 2020  Homem
# 3 2020  Total
```

7.6 Tratamento de variáveis

Tratar/alterar variáveis é transformar colunas de um conjunto de dados. Isso pode se dar por meio da extração, da transformação de múltiplas variáveis de uma única vez ou da criação de novas variáveis.

7.6.1 Extração de variáveis

As funções de extração de colunas retornam um conjunto de colunas como um vetor ou uma tabela. Dentre elas, destacam-se:

- `pull`: função semelhante a `$`, que se mostra mais útil porque funciona com pipes em `data.frames`, além de possibilitar a nomeação da saída.
- `select`: permite a seleção de variáveis em um `data.frame`, usando como referência os nomes das variáveis (por exemplo, `a:f` seleciona todas as colunas de “a” à esquerda até “f” à direita). Podem-se usar funções-predicado como `is.numeric` para selecionar variáveis com base em suas propriedades.
- `relocate`: altera as posições das colunas, usando a mesma sintaxe da função `select()` para facilitar a movimentação de blocos de colunas simultaneamente.

Exemplos:

1. Extrai os pesos da PNAD Contínua trimestral como um vetor:

```
pnadc2020T4 %>%  
  pull(V1028)
```

2. Extrai as variáveis de identificação dos domicílios e os pesos da PNAD Contínua trimestral como uma tabela:

```
pnadc2020T4 %>%  
  select(UPA, Estrato, V1008, V1028)
```

3. Repositiona as variáveis de identificação dos domicílios e os pesos da PNAD Contínua trimestral para o final do `data.frame`:

```
pnadc2020T4 %>%  
  relocate(UPA, Estrato, V1008, V1028,  
    .after = last_col())
```

7.6.2 Criação de variáveis

A criação de variáveis implica a aplicação de funções vetorizadas por colunas. Em outras palavras, essas funções tomam vetores como estrada, produzindo como resultado um vetor de mesmo tamanho.

- `mutate`: adiciona novas variáveis e preserva as existentes (novas variáveis sobrescrevem variáveis existentes com o mesmo nome);
- `transmute`: adiciona novas variáveis e elimina as existentes.

Exemplos:

1. Cria uma variável que divide a população em 10 grupos iguais ordenados, segundo rendimento médio mensal efetivo de todos os trabalhos para pessoas de

POR DENTRO DA PNAD CONTÍNUA

14 anos ou mais de idade (VD4020) com o auxílio do pacote **dineq**⁷ – mantendo todas as outras:

```
pnadc2020T4 %>%
  mutate(Decis =
    ntile.wtd(VD4020,
               n = 10,
               weights = V1028))
```

2. Cria uma variável que divide a população em 10 grupos iguais ordenados segundo rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020) com o auxílio do pacote **dineq** – excluindo todas as outras:

```
pnadc2020T4 %>%
  transmute(Decis =
    ntile.wtd(VD4020,
               n = 10,
               weights = V1028))
```

Ainda com relação ao tratamento de colunas, destaca-se que o pacote **dplyr** permite renomear variáveis por meio da função `rename`.

- `rename`: altera nomes de variáveis usando a sintaxe `novo_nome = antigo_nome`.

Exemplo: Renomeia as variáveis de peso (V1027 e V1028) para identificar se são com ou sem pós-estratificação:

```
pnadc2020T4 %>%
  rename(peso_sem_pos_estrat = V1027,
         peso_com_pos_estrat = V1028)
```

⁷ Esse pacote foi desenvolvido por Schulenberg (2018).

7.6.3 Funções vetorizadas

Como deve ter ficado claro na seção anterior, quando se apresentaram as funções `mutate()` e `transmute()`, podem-se aplicar funções vetorizadas, isto é, para colunas ou variáveis de um `data.frame`, para criar novas colunas/variáveis. Dentre seus tipos, destacam-se algumas funções com os seguintes objetivos: cálculos cumulativos (por exemplo, `cumsum`); ordenamento (por exemplo, `ntile` ou `ntiles.wtd`()); operações matemáticas (adição, subtração, multiplicação, divisão, logarítmicas, comparações lógicas etc.), dentre outras.

- `cumsum`: retorna um vetor cujos elementos são as somas cumulativas dos elementos da variável passada como argumento.

Exemplo: Cria uma variável com o resultado da soma acumulada do rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020):

```
pnadc2020T4 %>%
  arrange(VD4020) %>%
  mutate(soma_acum = cumsum(VD4020))
```

- `ntile`: classifica, subdividindo o vetor de entrada em ‘n’ intervalos. O tamanho dos intervalos pode variar.

Exemplo: Cria uma variável de ordenamento – ponderado – com base nos centésimos da distribuição do rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020)⁸:

```
pnadc2020T4 %>%
  mutate(percentil = ntile(VD4020*V1028, n=100))
```

⁸ Deve-se notar que essa é uma forma distinta de calcular percentis, como se definiu, anteriormente, por meio da função `ntiles.wtd()`.

POR DENTRO DA PNAD CONTÍNUA

Exemplo: Cria uma variável lógica que indica: se verdadeiro, VD4019 > VD4020, se falso, os demais casos:

```
pnadc2020T4 %>%
  mutate("Habitual maior que efetivo?" =
    VD4019 > VD4020)
```

Dentre outras funções vetorizadas, podem-se destacar algumas bastante úteis para o tratamento de dados:

- **if_else**: comparada à função-base do sistema `ifelse()`, esta função verifica se verdadeiro e falso são do mesmo tipo, permitindo que se alterem os valores a partir do cumprimento da condição definida.

Exemplo: Criando uma variável (`nova_cor` que agrupa em “Negros” e “Não negros” a variável `V2010`:

```
pnadc2020T4 %>%
  mutate(nova_cor =
    if_else(V2010 %in% c("Preta", "Parda"),
    "Negros",
    "Não negros"))
```

- **case_when**: permite vetorizar várias instruções do tipo `if_else()`. É um equivalente em R da instrução SQL CASE WHEN. Se nenhum caso coincidir, NA é retornado.

Exemplo: Cria faixas de salários mínimos para o rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020):

```
pnadc2020T4 %>%
  mutate(id_mult_sm =
    case_when(VD4020 >= 0 & VD4020 < 1045 ~ "Menos de 1 sm",
              VD4020 == 1045 ~ "1 sm",
              VD4020 > 1045 & VD4020 <= 2*1045 ~ "Mais de 1 a 2 sm",
              VD4020 > 2*1045 & VD4020 <= 3*1045 ~ "Mais de 2 a 3 sm",
              VD4020 > 3*1045 & VD4020 <= 5*1045 ~ "Mais de 3 a 5 sm",
              VD4020 > 5*1045 ~ "Mais de 5 sm"))
```

- `coalesce`: dado um conjunto de vetores, a função encontra o primeiro valor não ausente em cada posição do vetor.

Exemplo: Cria uma variável transformada para o rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020), transformando os casos NA em zero:

```
pnadc2020T4 %>%
  mutate(VD4020_transf =
    coalesce(VD4020, 0))
```

- `na_if`: converte valores quaisquer para NAs

Exemplo: Substituindo a categoria “Ignorado” da variável raça/cor (v2010) por NA:

```
pnadc2020T4 %>%
  mutate(V2010 =
    na_if(V2010, "Ignorado"))
```

7.6.4 Outras funções de síntese

Existe um amplo conjunto de funções que sumarizam informações de vetores, isto é, variáveis em um conjunto de dados, o que permite a criação de uma nova tabela com os resultados desejados. Dentre elas, destacam-se:

- `count`: conta valores únicos de uma ou mais variáveis⁹.

Exemplo: Contar o número de homens e mulheres (`v2007`) na população:

```
pnadc2020T4 %>%  
  count(V2007, wt = V1028)
```

Outra forma se realizar contagens de casos se dá por meio da função `n()`, para dados amostrais, ou `sum()`, para dados populacionais, obtidos a partir da expansão da amostra por meio de uma variável de peso (`V1028`, por exemplo).

- `n`: retorna informações de tamanho do grupo ou da variável. Precisa ser aplicada internamente às funções `summarize()` e/ou `mutate()`.
- `sum`: retorna a soma de todos os valores passados como argumento.

Exemplos:

1. Conta o número de homens e mulheres (`v2007`) na amostra (`n()`):

```
pnadc2020T4 %>%  
  group_by(V2007) %>%  
  summarise(n = n())
```

2. Conta o número de homens e mulheres (`v2007`) na população (`sum()`):

⁹ Essa função foi detalhada em seção anterior.

```
pnadc2020T4 %>%  
  group_by(V2007) %>%  
  summarise(n = sum(V1028))
```

Podem-se produzir estatísticas de posição para variáveis contínuas por meio de funções como: `mean()`, `median()` ou `quantile()` (para dados amostrais) ou `weighted.mean()`, `weighted.median()` ou `weighted.quantile()` (para dados populacionais ponderados). Suas descrições podem ser vistas a seguir:

- `mean`: calcula a média aritmética amostral;
- `weighted.mean`: calcula a média ponderada;
- `median`: calcula a mediana amostral;
- `weighted.median`: calcula a mediana ponderada;
- `quantile`: calcula quantis amostrais;
- `weighted.quantile`: calcula quantis ponderados.

Exemplos:

1. Calcula o rendimento médio para o rendimento médio mensal efetivo de todos os trabalhos (ponderado) para pessoas de 14 anos ou mais de idade (VD4020):

```
pnadc2020T4 %>%  
  summarise(média =  
            weighted.mean(VD4020,  
                            w = V1028,  
                            na.rm =TRUE))
```

2. Calcula o rendimento mediano para o rendimento médio mensal efetivo de todos os trabalhos (ponderado) para pessoas de 14 anos ou mais de idade (vd4020)¹⁰:

¹⁰ O cálculo da mediana ponderada exige a instalação do pacote `spatstat`, desenvolvido por Baddeley e Turner (2005).

POR DENTRO DA PNAD CONTÍNUA

```
# Instala o pacote spatstat
install.packages("spatstat")
library(spatstat)

pnadc2020T4 %>%
  summarise(mediana =
    weighted.median(VD4020,
                     w = V1028,
                     na.rm =TRUE))
```

3. Calcula quantis selecionados para o rendimento médio mensal efetivo de todos os trabalhos (ponderados) para pessoas de 14 anos ou mais de idade (VD4020):

```
pnadc2020T4 %>%
  summarise(quantis =
    weighted.quantile(VD4020,
                      w = V1028,
                      probs = seq(0,1,0.25),
                      na.rm = TRUE))
```

Outras estatísticas descritivas, como estatísticas de ordem (máximo e mínimo), medidas de dispersão (variância e desvio-padrão), também podem ser calculadas usando o **dplyr**, tanto para dados amostrais quanto populacionais (ponderados).

- **max**: calcula o valor máximo de um vetor;
- **min**: calcula o valor mínimo de um vetor;
- **var**: calcula a variância amostral;
- **weighted.var**: calcula a variância ponderada;
- **sd**: calcula o desvio-padrão amostral;

- `weighted.sd`: calcula o desvio-padrão ponderado¹¹.

Exemplos:

1. Calcula os valores máximo e mínimo do rendimento médio mensal efetivo de todos os trabalhos (ponderados) para pessoas de 14 anos ou mais de idade (VD4020):

```
pnadc2020T4 %>%
  summarise(máximo = max(VD4020,
                           na.rm = TRUE),
             mínimo = min(VD4020,
                           na.rm = TRUE))
```

2. Calcula a variância e o desvio-padrão do rendimento médio mensal efetivo de todos os trabalhos (ponderados) para pessoas de 14 anos ou mais de idade (VD4020):

```
#Instalando o pacote radiant
install.packages("radian")
library(radian)

pnadc2020T4 %>%
  summarise(variância = weighted.var(VD4020,
                                       w = V1028,
                                       na.rm = TRUE),
            desvio_padrão = weighted.sd(VD4020,
                                         wt = V1028,
                                         na.rm = TRUE),
            desvio_padrão2 = sqrt(variância))
```

¹¹ Para a utilização dessa função, é necessária a instalação do pacote **radian**, elaborado por Nijs (2021).

7.6.5 Combinação/fusão de tabelas

A combinação de tabelas é uma forma de se combinar os resultados obtidos por meio do tratamento de dados. A seguir, serão apresentadas algumas das principais formas de se mesclar, fundir e agregar dados por linhas e por colunas.

- `bind_cols`: retorna uma única tabela com os dados dispostos lado a lado. Os tamanhos das colunas precisam ser idênticos. As colunas não serão compatibilizadas por uma chave de identificação.

Exemplo: Combina variáveis (número de pessoas e rendimento médio) por sexo (v2007):

```
tbl1 <- pnadc2020T4 %>%
  count(V2007,
        wt = V1028,
        name = "Número de pessoas")

tbl2 <- pnadc2020T4 %>%
  group_by(V2007) %>%
  summarise("Rendimento Média" =
            weighted.mean(VD4020,
                           w = V1028,
                           na.rm = T))

tbl3 <- bind_cols(tbl1,tbl2)
## A tibble: 2 x 4
#   V2007...1 `Número de pessoas` V2007...3 `Renda Média`
#   <fct>           <dbl> <fct>           <dbl>
# 1 Homem            100542865. Homem       2683.
# 2 Mulher           111109504. Mulher      2218.
```

- `bind_rows`: retorna uma única tabela com os dados dispostos de forma empilhada. A função define uma coluna chave de identificação automaticamente, caso esta não seja informada.

Exemplo: Combina linhas (número de pessoas e rendimento médio) por sexo (v2007):

```
tbl4 <- bind_cols(tbl1,tbl2)

## # A tibble: 4 x 3
#   V2007  `Número de pessoas`  `Renda Média`
#   <fct>      <dbl>          <dbl>
# 1 Homem     100542865.        NA
# 2 Mulher    111109504.        NA
# 3 Homem     NA              2683.
# 4 Mulher    NA              2218.
```

As tabelas também podem ser agregadas de forma relacionada. Existe um conjunto de funções que permite esse tipo de fusão, que ocorre por meio da compatibilização de valores correspondentes.

- `inner_join`: inclui todos os casos de x e y, caso haja correspondência;
- `left_join`: funde os casos de y em x, mantendo todos os casos de x;
- `right_join`: funde os casos de x em y, mantendo todos os casos de y;
- `full_join`: inclui todos os casos de x ou y, independentemente de correspondência.

Exemplos:

1. Junta os objetos `tbl1` e `tbl2`, dos exemplos anteriores, em que há correspondência de casos para a variável `v2007`¹²:

¹² Deve-se reparar que, por qualquer método, o resultado será o mesmo por conta da estrutura dos objetos.

POR DENTRO DA PNAD CONTÍNUA

```
inner_join(tbl1, tbl2, by = "V2007")

left_join(tbl1, tbl2, by = "V2007")

right_join(tbl1, tbl2, by = "V2007")

full_join(tbl1, tbl2, by = "V2007")
```

2. Junta os objetos `tbl1` e `tbl5`, criados a partir da contagem de pessoas por raça/cor (`V2010`)¹³:

```
#Cria o arquivo tbl5
tbl5 <- pnadc2020T4 %>%
  count(V2010,
    wt = V1028,
    name = "Número de pessoas")

# Junta os arquivos

inner_join(tbl1, tbl5)

# Joining, by = "Número de pessoas"
# A tibble: 0 x 3
# ... with 3 variables: V2007 <fct>, Número
# de pessoas <dbl>, V2010 <fct>

left_join(tbl1, tbl5)

# Joining, by = "Número de pessoas"
# A tibble: 2 x 3
```

¹³ Deve-se notar que, dentre as opções, o método mais adequado para possibilitar a junção de tabelas sem compatibilidade é por meio da função `full_join`.

```
# V2007 'Número de pessoas' V2010
# <fct> <dbl> <fct>
# 1 Homem 100542865. NA
# 2 Mulher 111109504. NA

right_join(tbl1, tbl5)

# Joining, by = "Número de pessoas"
# A tibble: 6 x 3
# V2007 'Número de pessoas' V2010
# <fct> <dbl> <fct>
# 1 NA 93833532. Branca
# 2 NA 18395459. Preta
# 3 NA 1578133. Amarela
# 4 NA 97144209. Parda
# 5 NA 644014. Indígena
# 6 NA 57022. Ignorado

full_join(tbl1, tbl5)

# Joining, by = "Número de pessoas"
# A tibble: 8 x 3
# V2007 'Número de pessoas' V2010
# <fct> <dbl> <fct>
# 1 Homem 100542865. NA
# 2 Mulher 111109504. NA
# 3 NA 93833532. Branca
# 4 NA 18395459. Preta
# 5 NA 1578133. Amarela
# 6 NA 97144209. Parda
# 7 NA 644014. Indígena
# 8 NA 57022. Ignorado
```

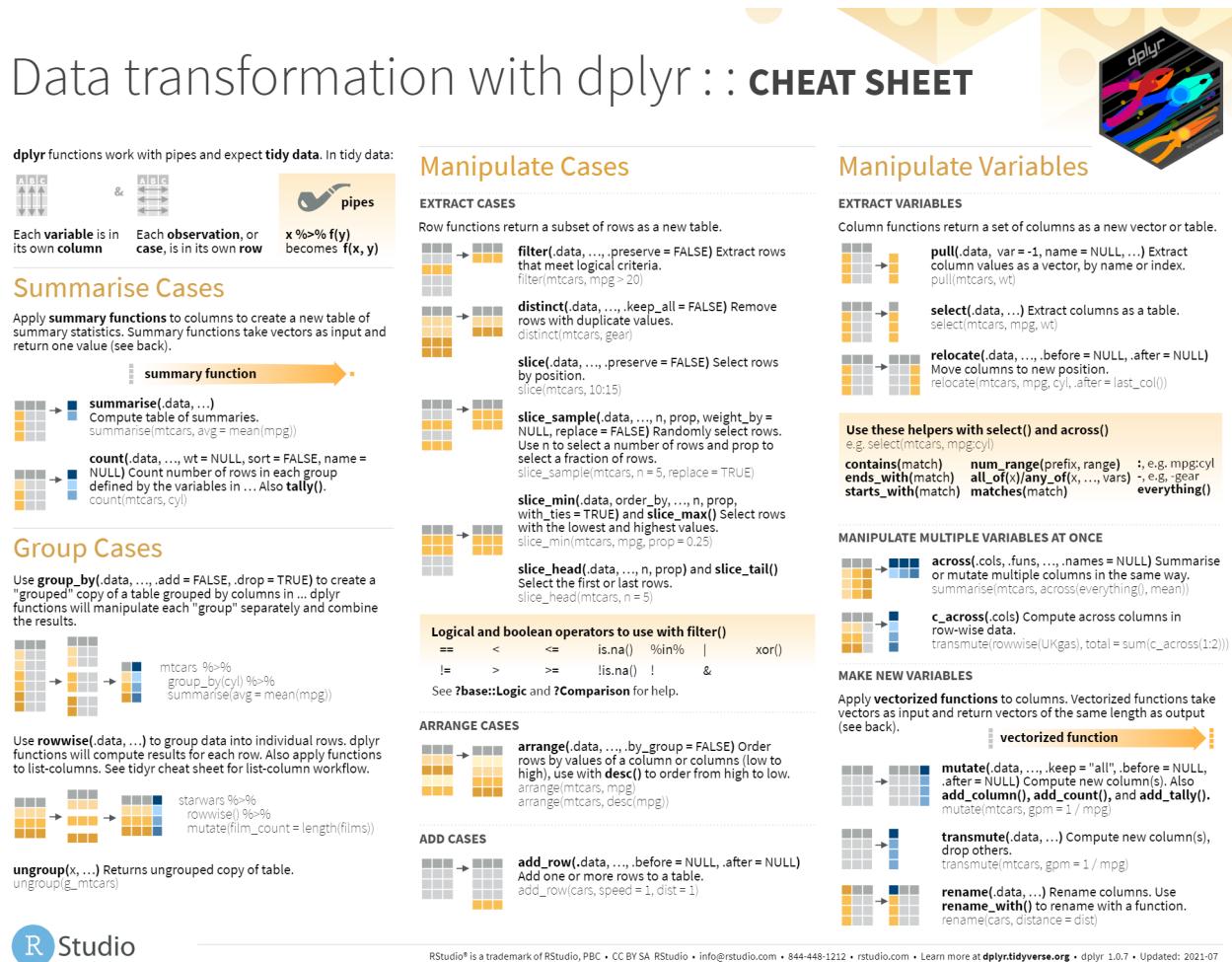
POR DENTRO DA PNAD CONTÍNUA

O conjunto dos elementos que foram apresentados no presente capítulo pode ser encontrado, com outros exemplos e aplicações no *site* do pacote **tidyverse**¹⁴. Ademais, reproduz-se, nas Figuras 7.1 e 7.2, um material-síntese (Folha de Cola – *Cheat Sheet*) desenvolvido por Wickham *et al.* (2019), que reúne as principais funções do **dplyr** e suas aplicações para o tratamento e a exploração de dados¹⁵.

¹⁴ Ver <https://www.tidyverse.org/>.

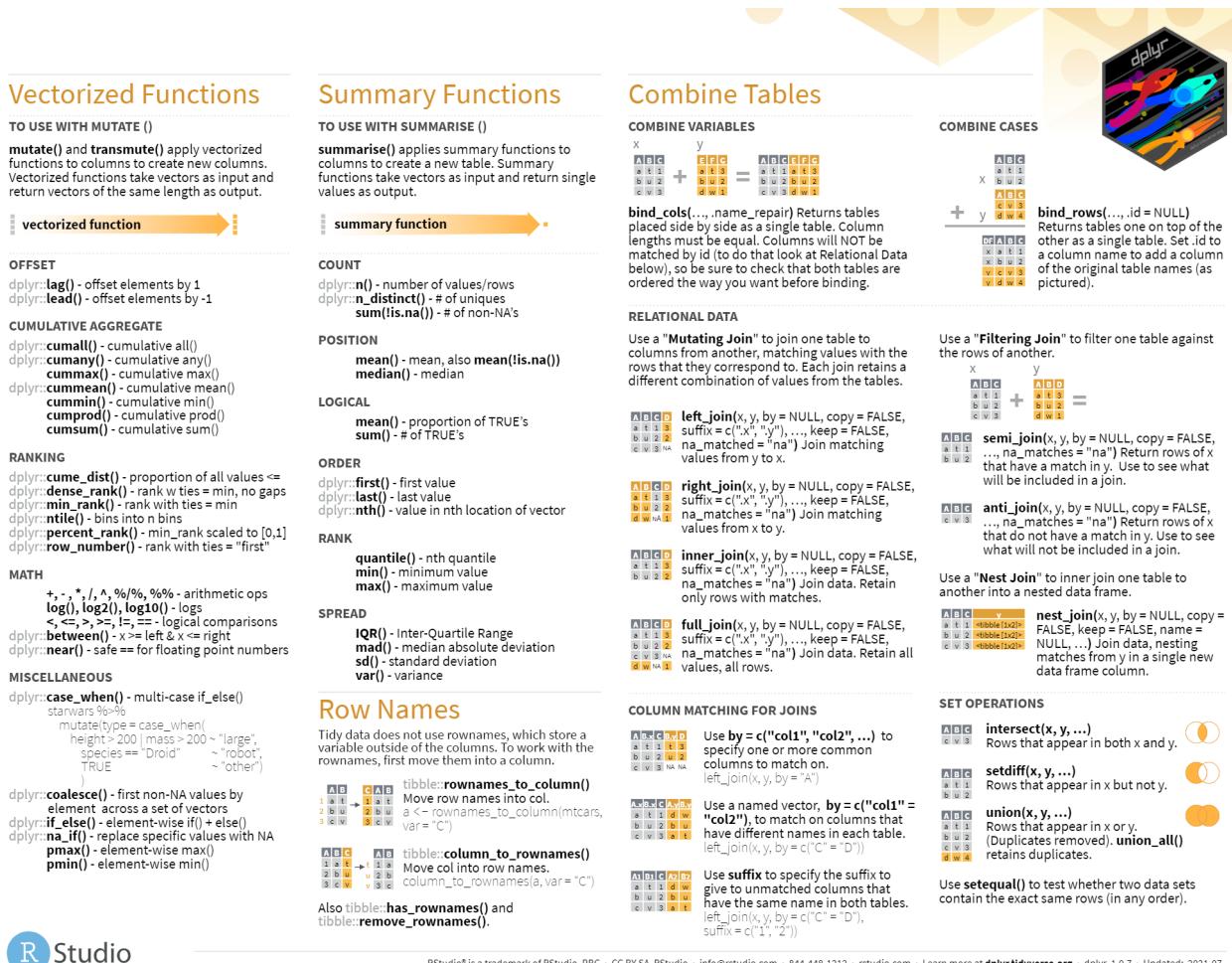
¹⁵ Para uma melhor visualização do material, recomenda-se acessar: <https://github.com/rstudio/cheatsheets/blob/master/data-transformation.pdf>.

Figura 7.1: Folha de cola para o tratamento de dados – **dplyr** (parte 1).



Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

Fonte: RSTUDIO (2021a)

Figura 7.2: Folha de cola para o tratamento de dados – **dplyr** (parte 2).RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at dplyr.tidyverse.org • dplyr 1.0.7 • Updated: 2021-07

Fonte: RSTUDIO (2021a).

7.7 Exercícios

1. Crie uma variável na base de dados da PNAD Contínua trimestral com o valor real (deflacionado para o último trimestre disponível) do rendimento médio habitual do trabalho principal.
2. Crie um subconjunto de dados com base nos dados da PNAD Contínua trimestral para mulheres residentes de domicílios localizados no meio rural.
3. Crie um subconjunto de dados aleatórios com apenas 10% dos casos na amostra na PNAD Contínua.
4. Ordene a PNAD Contínua de acordo com os valores do rendimento médio habitual do trabalho principal, do maior para o menor.
5. Crie um subconjunto de dados, a partir da PNAD Contínua trimestral utilizada nos exercícios anteriores, com apenas as variáveis sexo e cor/raça.
6. Crie uma variável lógica (VERDADEIRO/FALSO) que indica se a pessoa é mulher.
7. Calcule o total de pessoas por Unidades da Federação para a PNAD Contínua do quarto trimestre de 2020.
8. Calcule o total de homens e mulheres agrupados por cor/raça para a PNAD Contínua do quarto trimestre de 2020.
9. Calcule o rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020), apenas para os indivíduos que se declararam de cor Preta e Parda à PNAD Contínua do quarto trimestre de 2020.
10. Calcule o total de pessoas por Unidades da Federação para a PNAD Contínua do quarto trimestre de 2020.
11. Calcule o total de pessoas ocupadas e desocupadas para a PNAD Contínua do quarto trimestre de 2020.

POR DENTRO DA PNAD CONTÍNUA

12. Calcule a proporção de pessoas ocupadas e desocupadas para a PNAD Contínua do quarto trimestre de 2020 (DICA: utilize a função `mutate()`).

8 Tratamento dos microdados da PNAD Contínua: o pacote **survey**

Segundo Lumley (2004), a análise de pesquisas com amostras complexas, como é o caso das PNADs, tem sido tradicionalmente um espaço para *softwares* especializados, o que faz com que o interesse pelo desenvolvimento de instrumentais em linguagem R se mostre cada vez mais expressivo.

Na maioria das pesquisas em estatística, os dados são considerados aleatórios, e o pesquisador precisa modelar sua distribuição. Assim, a vantagem da visão baseada em *design* é que o procedimento de amostragem está sob controle e conhecimento do pesquisador (LUMLEY, 2004).

O pacote de código R denominado **survey**, desenvolvido por Lumley (2019), fornece recursos (funções) para analisar dados de pesquisas como a PNAD Contínua, pois leva em conta seu plano amostral complexo.

No presente capítulo, o objetivo é explorar algumas das funções desse pacote, especialmente aquelas associadas à análise descritiva introdutória, comparando sua aplicação com a forma apresentada no Capítulo 7, no qual se analisou a potencialidade do pacote **dplyr** para a produção de estimativas populacionais (ponderadas) para diversas estatísticas.

8.1 O escopo do pacote **survey**

Dentre os principais métodos implementados no pacote, destacam-se:

- Médias, totais, razões, quantis, tabelas de contingência, modelos de regressão,

POR DENTRO DA PNAD CONTÍNUA

modelos log-lineares, curvas de sobrevivência, testes de classificação, para toda a amostra e para sua expansão;

- Cálculo de variâncias por linearização de Taylor ou por pesos replicados (BRR, *jackknife*, *bootstrap*, *bootstrap* multiestágio, ou fornecidos pelo usuário);
- Amostragem multiestágio com ou sem reposição;
- Amostragem PPT (Probabilidade Proporcional ao Tamanho) com ou sem reposição: estimadores de Horvitz-Thompson e Yates-Grundy e uma gama de aproximações;
- Pós-estratificação, *raking/calibração* generalizada, estimativas de Regressões Generalizadas, corte de pesos;
- Planos de amostragem em duas fases;
- Pesos estimados para estimadores aumentados por PPI (Ponderação por Probabilidade Inversa);
- Gráficos;
- Suporte ao uso de múltiplos dados imputados;
- Objetos de design com suporte de banco de dados para grandes conjuntos de dados (incluindo pesos replicados);
- Algum suporte para processamento paralelo em computadores com processadores com múltiplos núcleos;
- Análise multivariada: componentes principais, análise fatorial (ainda em caráter experimental);
- Teste da razão de verossimilhanças (Rao-Scott) para Modelos Lineares Generalizados (MLG), modelos de Cox e modelos log-lineares.

8.2 Instalação do pacote e *download* dos dados

Para instalar o pacote **survey**, basta executar o seguinte comando no R.

```
install.packages("survey")
library(survey)
```

A apresentação comparada de algumas das principais funções do **survey** está baseada, no presente capítulo, na base de dados da PNAD Contínua do quarto trimestre de 2020 para os formatos **survey.design** e **data.frame**.

Para fazer o *download* e definir os objetos do mesmo banco de dados nos dois formatos a serem utilizados, basta aplicar os seguintes comandos:

```
# Formato survey.design
pnadc2020T4_dsg <-
  get_pnadc(2020,4)

# Formato data.frame / tibble
pnadc2020T4_df <-
  as_tibble(pnadc2020T4_dsg$variables)
```

Deve-se notar que foram adicionados dois sufixos ao nome **pnadc2020T4_***. O sufixo **_dsg** serve para designar o banco de dados em formato **survey.design** e o sufixo **_df**, para o formato **data.frame**. Para verificar a classe dos dois objetos, pode-se usar a função **class()**:

```
class(pnadc2020T4_dsg)

# > [1] "survey.design2" "survey.design"

class(pnadc2020T4_df)

# > [1] "tbl_df"      "tbl"        "data.frame"
```

8.3 Principais funções do pacote **survey** para análises descritivas

O conjunto de instrumentos denominado “**surveystatus**” contém um conjunto de funções para o cálculo de estimativas de distintas estatísticas descritivas, aplicáveis a pesquisas com planos amostrais complexos. Dentre elas, destacam-se:

- **svymean**: estima médias de variáveis contínuas para a população (ponderado pelo peso pós-estratificação – **v1028** no caso da PNAD Contínua trimestral), além de possibilitar estimativas para razões entre contagem de subpopulações;
- **svyvar**: estima variâncias de variáveis contínuas para a população (ponderadas pelo peso pós-estratificação);
- **svytotal**: estima o somatório dos valores/categorias de variáveis contínuas ou categóricas, sendo que estas últimas devem ser passadas como recorte de análise (ponderado pelo peso pós-estratificação);
- **svyratio**: estima razões entre variáveis (ponderadas pelo peso pós-estratificação);
- **svyquantile**: estima quantis e seus intervalos de confiança (ponderados pelo peso pós-estratificação);
- **svyby**: calcula estatísticas em subconjuntos a partir de recortes definidos por fatores (ponderadas pelo peso pós-estratificação).

Os principais argumentos dessas funções podem ser vistos a seguir:

- **x**: fórmula, vetor ou matriz de dados;
- **design**: objeto no formato **survey.design** ou **svyrep.design**;
- **na.rm**: casos com valores omissos devem ser descartados?;
- **influence**: deve ser retornada uma matriz de funções de influência (principalmente para apoiar a função **svyby**)?;
- **rho**: parâmetro para o estimador de variância de Fay em um design Balanced Repeated Replicates (BRR);

- `return.replicates`: deve ser retornados média e total baseados em pesos replicados?;
- `deff`: retorna o efeito de design;
- `object`: retorna o resultado de uma das outras funções-síntese definidas;
- `quietly`: não avisa quando não houver efeito de design calculado;
- `estimate.only`: não calcula erros-padrões (útil quando `svyvar` é usada para estimar efeitos de design);
- `parm`: especifica quais parâmetros devem receber intervalos de confiança, se um vetor de números ou um vetor de nomes. Sua não definição implica que todos os parâmetros serão considerados;
- `level`: o nível de confiança desejado;
- `df`: graus de liberdade para distribuição `t` no intervalo de confiança, usar `degf` (`design`) para o número de Unidades Primárias de Amostragem (UPA) menos o número de estratos;
- `names`: vetor de cadeias de caracteres contendo os nomes;

8.4 Exemplos comparativos de análise usando os pacotes `survey` e `dplyr`

Nesta seção, serão apresentadas algumas aplicações e a forma gramatical de algumas das funções discutidas no Capítulo 7, no qual foi introduzido o pacote `dplyr`.

A fim de facilitar a apresentação e a comparação dos resultados obtidos com os dois pacotes, será aplicada a função `options()` para que as saídas no console sejam parecidas, assumindo a forma numérica e não notação científica. Para tanto, basta aplicar o seguinte comando:

POR DENTRO DA PNAD CONTÍNUA

```
options(OutDec=",",
       scipen=100,
       digits=4,
       big.mark = ".")
```

8.4.1 Estimativa da contagem populacional para dados agrupados para variáveis categóricas

Exemplo: Estimativa da contagem da população (ponderada) por subgrupo, definido pela variável sexo (v2007) com os dois pacotes:

```
#survey
svytotal(~V2007,
         pnadc2020T4_dsg,
         na.rm = T)

coef(svytotal(~V2007,
             pnadc2020T4_dsg,
             na.rm = T))

#dplyr
pnadc2020T4_df %>%
  count(V2007,
        wt=V1028)
```

8.4.2 Estimativa da contagem populacional para dados agrupados por mais de uma variável categórica

Exemplo: Estimativa da contagem da população por sexo (v2007) e por raça/cor (v2010) – deve-se reparar que a definição dos resultados empilhados, sem

a interação das variáveis, é feita pelo símbolo matemático de adição (+) informado entre as variáveis:

```
#survey
svytotal(~V2007 + V2010,
          pnadc2020T4_dsg,
          na.rm = T)

coef(svytotal(~V2007 + V2010,
              pnadc2020T4_dsg,
              na.rm = T))

#dplyr

bind_rows(pnadc2020T4_df %>%
          count(V2007,
                wt=V1028),
          pnadc2020T4_df %>%
          count(V2010,
                wt=V1028))
```

8.4.3 Estimativa da contagem populacional com interação entre variáveis categóricas de agrupamento

Exemplo: Estimativa da contagem da população a partir da interação entre as variáveis sexo (`V2007`) e raça/cor (`V2010`) – deve-se reparar que a definição dos resultados obtidos de forma aninhada, com a interação entre as variáveis, é feita pela função `interaction()`:

```
#survey
svytotal(~interaction(V2007, V2010),
          pnadc2020T4_dsg,
```

POR DENTRO DA PNAD CONTÍNUA

```
na.rm = T)

coef(svytotal(~interaction(V2007, V2010),
               pnadc2020T4_dsg,
               na.rm = T))

#dplyr
pnadc2020T4_df %>%
  group_by(V2010) %>%
  count(V2007,
        wt = V1028)

#Forma alternativa (dplyr)
pnadc2020T4_df %>%
  count(V2010, V2007,
        wt = V1028)
```

Cabe ressaltar que a função `svytotal()` proporciona, também, estimativas para variáveis contínuas, pois realiza o somatório dos valores das variáveis.

8.4.4 Estimativa do somatório dos valores de variáveis contínuas

Exemplo: Estimativa do somatório do rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (`VD4020`):

```
#survey
svytotal(~VD4020,
          pnadc2020T4_dsg,
          na.rm = T)

# >      total       SE
# VD4020 208657531708 3054008058
```

```
#Apresentando apenas a estimativa pontual
coef(svymtotal(~VD4020,
  pnadc2020T4_dsg,
  na.rm = T))
```

```
#dplyr
pnadc2020T4_df %>%
  summarise("Somatório" =
    sum(VD4020*V1028,
      na.rm = T))
```

```
# > A tibble: 1 x 1
#   `renda mensal efetiva total`<dbl>
#   1 208657531708.
```

8.4.5 Estimativa da média ponderada para variáveis contínuas

Exemplo: Estimativa do rendimento médio mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020):

```
#survey
svymean(~VD4020,
  pnadc2020T4_dsg,
  na.rm = T)

coef(svymean(~VD4020,
  pnadc2020T4_dsg,
  na.rm = T))
```

POR DENTRO DA PNAD CONTÍNUA

```
#dplyr  
pnadc2020T4_df %>%  
  summarise("media da renda" =  
            weighted.mean(VD4020,  
                            w = V1028,  
                            na.rm = T))
```

Quando a função svymean() é aplicada a variáveis categóricas, o resultado é a estimativa da proporção ponderada das categorias no total da população.

8.4.6 Estimativa ponderada da proporção de pessoas no total populacional com uma variável categórica de agrupamento

Exemplo: Estimativa da proporção de pessoas por sexo (v2007) no total da população:

```
#survey  
svymean(~V2007,  
        pnadc2020T4_dsg,  
        na.rm = T)  
  
coef(svymean(~V2007,  
             pnadc2020T4_dsg,  
             na.rm = T))  
  
#dplyr  
pnadc2020T4_df %>%  
  count(V2007,wt=V1028) %>%  
  mutate(freq = n/sum(n)) %>%  
  select("V2007","freq")
```

8.4.7 Estimativa ponderada da proporção de pessoas no total populacional com mais de uma variável categórica de agrupamento, sem interação

Exemplo: Estimativa da proporção de pessoas por sexo (v2007) e por raça/cor (v2010) no total da população, sem interação entre as variáveis:

```
#survey
svymean(~V2007 + V2010,
         pnadc2020T4_dsg,
         na.rm = T)

coef(svymean(~V2007 + V2010,
             pnadc2020T4_dsg,
             na.rm = T))

#dplyr
bind_rows(pnadc2020T4_df %>%
           count(V2007, wt = V1028) %>%
           mutate(freq = n/sum(n)) %>%
           select("V2007", "freq"),
           pnadc2020T4_df %>%
           count(V2010, wt = V1028) %>%
           mutate(freq = n/sum(n)) %>%
           select("V2010", "freq"))
```

8.4.8 Estimativa ponderada da proporção de pessoas no total populacional com mais de uma variável categórica de agrupamento, com interação

Exemplo: Estimando a proporção de pessoas por sexo (v2007) e por raça/cor (v2010) no total da população, com interação entre as variáveis:

```
#survey
svymean(~interaction(V2007, V2010),
        pnadc2020T4_dsg,
        na.rm = T)

coef(svymean(~interaction(V2007, V2010),
             pnadc2020T4_dsg,
             na.rm = T))

#dplyr
pnadc2020T4_df %>%
  count(V2007, V2010, wt = V1028) %>%
  mutate(freq = n/sum(n)) %>%
  select("V2007", "V2010", "freq")
```

8.4.9 Estimativa da contagem populacional para variáveis categóricas com casos omissos

Algumas variáveis da PNAD Contínua possuem casos omissos devido ao fato de elas representarem questões da pesquisa para as quais as pessoas respondentes não precisam prestar informações. Para que esses valores omissos sejam explicitados como uma categoria da variável, pode-se aplicar a função `fct_explicit_na()`, do pacote `tidyverse/dplyr`. Com isso, fornece-se a esses valores ausentes um nível de fator explícito, garantindo que eles sejam considerados nos cálculos realizados. Vale ressaltar que, no `survey`, os casos omissos não são contabilizados para o resultado estimado.

Exemplo: Estimativa do total de pessoas ocupadas e desocupadas (VD400 2):

```
#survey
coef(svytotal(~VD4002,
```

```
pnadc2020T4_dsg,  
na.rm=TRUE))  
  
#Explicitando NAs  
pnadc2020T4_df <-  
  pnadc2020T4_df %>%  
    mutate(VD4002 = fct_explicit_na(VD4002))  
  
#dplyr  
pnadc2020T4_df %>%  
  count(VD4002,  
    wt = V1028)
```

8.4.10 Estimativa da proporção no total populacional para variáveis categóricas com casos omissos

Exemplos: Estimativa da proporção de pessoas ocupadas e desocupadas no total da força de trabalho (VD4002) – é necessário reparar que os casos omissos não devem ser considerados nesse cálculo:

```
#survey (exclusão de casos omissos automaticamente,  
# usando o argumento na.rm = TRUE)  
coef(svymean(~VD4002, pnadc2020T4_dsg, na.rm=TRUE))  
  
#dplyr  
pnadc2020T4_df %>%  
  filter(VD4002 != "(Missing)") %>%  
  count(VD4002, wt = V1028) %>%  
  mutate(freq = n/sum(n)) %>%  
  select("VD4002","freq")
```

POR DENTRO DA PNAD CONTÍNUA

Deve-se atentar para o fato de esse cálculo representar o conceito de taxa de desocupação para os dados da PNAD Contínua, quando se observa a proporção de desocupados no total (força de trabalho, que é o somatório de ocupados mais desocupados).

8.4.11 Estimativas ponderadas de estatísticas para subconjuntos de dados

No pacote **survey**, a definição de um subconjunto de dados se dá por meio da função `subset()`. Sua aplicação é similar à função `filter()` do **dplyr**.

Exemplo: Estimativa do rendimento mensal habitual do trabalho principal (`VD4016`) para pessoas pretas ou pardas (`V2010`) com 14 anos ou mais de idade:

```
#survey
coef(svymean(~VD4016,
              subset(pnadc2020T4_dsg,
                     V2010 == "Preta" | V2010 == "Parda"),
                     na.rm = TRUE))

#dplyr
pnadc2020T4_df %>%
  filter(V2010 == "Preta" | V2010 == "Parda") %>%
  summarise("RHTP" = weighted.mean(VD4016,
                                    w=V1028,
                                    na.rm = TRUE))
```

8.4.12 Estimativa da proporção entre contagens populacionais para mais de uma variável categórica

Como visto anteriormente, a função `svyratio()` permite o cálculo de razões entre o somatório dos componentes de variáveis categóricas e o total. Com o uso dessa função, é possível estimar de outra maneira a taxa de desocupação. Isso se dá por meio da relação entre as variáveis da PNAD Contínua “Condição em relação à

força de trabalho na semana de referência para pessoas de 14 anos ou mais de idade” (VD4001) e “Condição de ocupação na semana de referência para pessoas de 14 anos ou mais de idade” (VD4002).

Exemplo: Estimativa da taxa de desocupação para o total do Brasil por meio da função svyratio() e sua forma correlata com o dplyr:

```
#survey
coef(svyratio(~VD4002 == "Pessoas desocupadas",
              ~VD4001 == "Pessoas na força de trabalho",
              pnadc2020T4_dsg,
              na.rm = T))

#dplyr
#Explicitando NAs
pnadc2020T4_df <-
  pnadc2020T4_df %>%
  mutate(VD4001 = fct_explicit_na(VD4001),
         VD4002 = fct_explicit_na(VD4002))

#Calculando a taxa
pnadc2020T4_df %>%
  filter(VD4002 != "(Missing)" & VD4001 != "(Missing)") %>%
  count(VD4002, VD4001, wt = V1028) %>%
  mutate(taxa = n/sum(n)*100) %>%
  filter(VD4002 == "Pessoas desocupadas") %>%
  select("taxa")
```

8.4.13 Outras estimativas

Outras estatísticas como o rendimento mediano ou outros quantis para variáveis contínuas, também, são possíveis de serem calculadas com o pacote survey.

POR DENTRO DA PNAD CONTÍNUA

Exemplo 1: Estimativa do valor do rendimento mediano mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020)¹:

```
#survey
coef(svyquantile(~VD4020,
                  pnadc2020T4_dsg,
                  quantiles = .5,
                  na.rm = T))

#dplyr e spatstat
pnadc2020T4_df %>%
  summarise("mediana da renda" =
            weighted.quantile(VD4020,
                               w = V1028,
                               probs = 0.5,
                               na.rm = TRUE))
```

Exemplo 2: Estimativas de outros quantis para o rendimento mensal efetivo de todos os trabalhos para pessoas de 14 anos ou mais de idade (VD4020):

```
#survey
coef(svyquantile(~VD4020,
                  pnadc2020T4_dsg,
                  quantiles = c(.1,.25,.5,.75,.9,.99),
                  na.rm = T))
```

¹ O cálculo dos quantis no pacote **dplyr** pode ser realizado com o auxílio do pacote **spatstat**.

```
#dplyr e spatstat
pnadc2020T4_df %>%
  summarise("p.1" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.1,
                                         na.rm = TRUE),
            "p.25" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.25,
                                         na.rm = TRUE),
            "p.5" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.5,
                                         na.rm = TRUE),
            "p.75" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.75,
                                         na.rm = TRUE),
            "p.9" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.9,
                                         na.rm = TRUE),
            "p.99" = weighted.quantile(VD4020,
                                         w = V1028,
                                         probs = 0.99,
                                         na.rm = TRUE))
```

8.4.14 Outras estimativas para subconjuntos populacionais

Por meio da função `svyby()`, é possível fazer cálculos para subconjuntos da população a partir da interação de variáveis categóricas definidas como unidades de análise (variáveis de agrupamento).

POR DENTRO DA PNAD CONTÍNUA

Exemplo: Estimativa da taxa de desocupação por sexo (v2007) e por raça/cor (v2010), a partir da interação entre as duas variáveis:

```
#survey
coef(svyby(~VD4002 == "Pessoas desocupadas",
           by = ~interaction(V2007,V2010),
           pnadc2020T4_dsg,
           svyratio,
           denominator = ~VD4001 == "Pessoas na força de trabalho",
           na.rm = T))

#dplyr
pnadc2020T4_df %>%
  filter(VD4002 != "(Missing)" & VD4001 != "(Missing)") %>%
  group_by(V2007,V2010) %>%
  count(VD4002, VD4001, wt = V1028) %>%
  group_by(V2007,V2010) %>%
  mutate(taxa = n/sum(n)) %>%
  filter(VD4002 == "Pessoas desocupadas") %>%
  select(V2007,V2010,taxa)
```

8.5 Exercícios

1. Calcule o rendimento mensal efetivo de todos os trabalhos para mulheres de 14 anos ou mais de idade com os pacotes **survey** e **dplyr**.
2. Calcule a taxa de desocupação para pessoas com idade de 25 anos ou mais, utilizando os pacotes **survey** e **dplyr**.
3. Calcule a frequência relativa de cada nível de instrução para homens pardos com mais de 30 anos, utilizando os pacotes **survey** e **dplyr**.

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

4. Calcule a frequência relativa de homens e mulheres em cada nível de instrução, utilizando os pacotes **survey** e **dplyr**.
5. Calcule a frequência relativa de cada nível de instrução para homens e mulheres, utilizando os pacotes **survey** e **dplyr**.

9 Uma ponte entre o SPSS e o R: o pacote **expss**

Muitos pesquisadores em ciências sociais e economia tiveram seu primeiro contato com bases de dados em um *software* chamado **SPSS**, que possui uma interface relativamente amigável, permitindo ao usuário executar comandos por meio de cliques em caixas e janelas específicas. A utilização de seus recursos nesse formato contava com a possibilidade de se salvar os comandos executados em *Scripts*, com uma linguagem própria de programação, também mediante uma sequência de cliques.

O pacote **expss**, desenvolvido por Demin (2020), permite o cálculo e a exibição de tabelas com suporte a rótulos no estilo **SPSS**, com suas múltiplas marcações, seus alinhamentos, para estimativas pontuais ponderadas (expansão amostral), para variáveis de múltiplas respostas e testes de significância. Seus recursos permitem a produção de tabelas em ‘Shiny’, ‘*.xlsx’ e R, por exemplo. Traz métodos para o tratamento de variáveis com suporte a rótulos para funções R e de outros pacotes. Além disso, o pacote traz um conjunto de funções para a transformação e o tratamento de dados em formatos **SPSS** e **Excel**, como: ‘RECODE’, ‘COUNT’, ‘COMPUTE’, ‘DO IF’ etc. Com esse pacote, os usuários podem migrar do tratamento de dados em **SPSS** e **Excel** para o R.

No presente capítulo, o objetivo é apresentar o pacote **expss**, que faz uma ponte entre o **SPSS** e o R, destacando e emulando as principais características do **SPSS** para tratar dados, além de criar e produzir distintos resultados na forma de tabelas.

9.1 Instalação

Para instalar e habilitar o **expss**, podem-se aplicar os seguintes comandos:

```
install.packages("expss")
library(expss)
```

9.2 Tratamento de dados

O pacote **expss** possui distintas ferramentas para tratar, alterar e transformar dados. Nesta seção, serão apresentados alguns exemplos elaborados para o tratamento dos microdados em formato `data.frame` da PNAD Contínua anual de 2019, utilizada e armazenada no objeto `pnadc2019_visita1_df`, em capítulos anteriores.

Para baixar novamente os dados, basta realizar os seguintes comandos:

```
pnadc2019_visita1_df <- get_pnadc(2019,
                                      interview=1,
                                      design=FALSE,
                                      labels = FALSE)
```

9.3 O pacote **expss**: algumas funções básicas

- `apply_labels`: permite que se atribuam rótulos a variáveis categóricas.
- `compute`: avalia a expressão `expr` no contexto dos dados em formato `data.frame` e retorna os dados originais possivelmente modificados;

POR DENTRO DA PNAD CONTÍNUA

- `calculate`: avalia a expressão `expr` no contexto dos dados em formato `data.frame` e retorna o valor da expressão avaliada sem criar uma variável. A função `use_labels` é um atalho para calcular com o argumento `use_labels` habilitado como `TRUE`;
- `do_if`: modifica apenas as linhas para as quais `cond` é igual a `TRUE`. As outras linhas permanecem inalteradas. Variáveis recém-criadas, também, terão valores apenas nas linhas para as quais `cond` foi definido como `TRUE`. Nas demais linhas, `NAs` são introduzidos. Esta função tenta imitar a função “`DO IF () . . . END IF.`” do SPSS.

9.3.1 Aplicação de rótulos

Nesta seção, serão apresentados apenas alguns exemplos de como aplicar rótulos aos microdados da PNAD Contínua, com base em sua documentação, mais especificamente, em seus dicionários de variáveis.

- `apply_labels`: aplica rótulos a variáveis categóricas.

Exemplo:

```
pnadc2019_visita1_df =  
  apply_labels(pnadc2019_visita1_df,  
    V2007 = "Sexo",  
    V2007 = c("Homem" = 1,  
             "Mulher"=2),  
    V2010 = "Cor ou raça",  
    V2010 = c("Branca" = 1,  
             "Preta" = 2,  
             "Amarela" = 3,  
             "Parda" = 4,  
             "Indígena" = 5,  
             "Ignorado" = 9),  
    V1022 = "Situação do domicílio",  
    V1022 = c("Urbana" = 1, "Rural" = 2))
```

9.3.2 Criação e transformação de variáveis

Como se sabe, no contexto do tratamento dos dados, diversas transformações no banco de dados podem ser necessárias. Para isso, o pacote **expss** dispõe de um conjunto de funções, que são exploradas na sequência.

- `compute`: utilizada para criação e transformação de variáveis.

Exemplos:

1. Somar os elementos nas linhas de distintas variáveis:

```
compute(pnadc2019_visita1_df, {  
    y_social = sum_row(V5001A2,V5002A2,V5003A2)  
    var_lab(y_social) = "Rendimentos de programas sociais"  
})
```

2. criar um novo arquivo com a nova variável definida. Para tanto, basta atribuir o resultado da função `compute` a um novo objeto qualquer (por exemplo, “`nova_base`”).

```
nova_base <-  
compute(pnadc2019_visita1_df, {  
    y_social = sum_row(V5001A2,V5002A2,V5003A2)  
    var_lab(y_social) = "Rendimentos de programas sociais"  
})
```

3. Transformar o arquivo original que se está tratando.

```
pnadc2019_visita1_df <-  
compute(pnadc2019_visita1_df, {  
    y_social = sum_row(V5001A2,V5002A2,V5003A2)  
    var_lab(y_social) = "Rendimentos de programas sociais"  
})
```

POR DENTRO DA PNAD CONTÍNUA

Essa função, também, possibilita o cálculo de variáveis. Porém, esses cálculos não são realizados caso um dos valores seja `NA` (ver exemplos a seguir).

1. Comando que não produz o resultado desejado (existência de `NAs`).

```
nova_base <-
  compute(pnadc2019_visita1_df, {
    y_social = V5001A2+V5002A2+V5003A2
    var_lab(y_social) = "Rendimentos de programas sociais"
  })
```

2. Para corrigir esse problema, é possível recodificar as variáveis, substituindo `NAs` por 0 (zero). Basta utilizar o comando `recode`¹.

```
#Recodificar na mesma variável

pnadc2019_visita1_df$V5001A2 <-
  expss::recode(pnadc2019_visita1_df$V5001A2,
    NA ~ 0, TRUE ~ copy)

#Recodificar em outra variável

pnadc2019_visita1_df$V5001A2_nova <-
  expss::recode(pnadc2019_visita1_df$V5001A2,
    NA ~ 0, TRUE ~ copy)
```

Ainda é possível calcular, por exemplo, o valor da soma dos programas sociais como feito anteriormente, porém retornando apenas um vetor com os valores. Para isso, basta realizar os seguintes comandos:

¹ Mais exemplos serão apresentados posteriormente nesse capítulo.

```
calculate(pnadc2019_visita1_df,  
         sum_row(V5001A2,V5002A2,V5003A2))
```

Em alguns casos, pode ser necessário criar uma nova variável vazia dentro da programação usando, por exemplo, a função `.new_var`. Essa função cria uma variável de comprimento `.N` (valor igual ao número de casos dos dados) com `NAs`, para ser usada nas expressões dentro das funções `compute` e `calculate`. Para adicionar uma variável qualquer com `NAs`, pode-se aplicar o seguinte comando:

```
pnadc2019_visita1_df <-  
  compute(pnadc2019_visita1_df, {  
    n_var = .new_var()  
    n_var[] = 1  
  })
```

No entanto, isso não se mostra imperativo, podendo se fazer apenas:

```
pnadc2019_visita1_df$n_var <-  
  compute(pnadc2019_visita1_df, {  
    n_var = 1  
  })
```

9.3.3 Aplicação de operações condicionadas

No tratamento de dados, é bastante comum a necessidade de se utilizar condicionais para realizar algum comando, como é o caso da criação de variáveis restrinvidas por uma determinada condição. Se, por exemplo, for necessário adicionar uma variável que seja condicionada a outras quaisquer, então, pode-se utilizar a função `do_if`.

POR DENTRO DA PNAD CONTÍNUA

- `do_if`: possibilita transformações ou alterações condicionadas nos dados, e associadas a outras variáveis ou funções.

Exemplo Criar uma variável com a soma dos rendimentos de programas sociais, retornando uma variável condicionada à `V2007 = 2` (sexo igual a Mulher):

```
do_if(pnadc2019_visita1_df,
      V2007 == 2, {
        y_social = sum_row(V5001A2,V5002A2,V5003A2)
      })
```

Utilizar condicionantes pode ser útil para a criação de variáveis de identificação condicionadas a uma relação específica.

Exemplo: Criar uma variável de identificação que receba o valor para os casos em que a variável sexo assuma a forma da categoria “Mulher” `V2007 = 2`. Para isso, pode-se fazer:

```
pnadc2019_visita1_df <-
  compute(pnadc2019_visita1_df, {
    id_var = ifelse(V2007 == 2, 1, NA)
  })

# Alternativa
pnadc2019_visita1_df <-
  compute(pnadc2019_visita1_df, {
    id_var = ifs(V2007 == 2 ~ 1)
  })
```

A função `ifs` ainda permite transformações para variáveis contínuas, como no exemplo a seguir:

```
pnadc2019_visita1_df$faixa_rend <-
  ifs(pnadc2019_visita1_df$VD5005 >= 0 &
      pnadc2019_visita1_df$VD5005 < 1000 ~ 1,
      pnadc2019_visita1_df$VD5005 == 1000 ~ 2,
      pnadc2019_visita1_df$VD5005 > 1000 ~ 3,
      TRUE ~ NA)
```

O código pode ficar mais limpo seguindo o seguinte exemplo:

```
pnadc2019_visita1_df <-
  pnadc2019_visita1_df %>%
    mutate(faixa_rend =
      ifs(VD5005 >= 0 &
          VD5005 < 1000 ~ 1,
          VD5005 == 1000 ~ 2,
          VD5005 > 1000 ~ 3,
          TRUE ~ NA))
```

9.3.4 Recodificação de variáveis

O comando `recode`, também, permite criar variáveis a partir de outras, recodificando-as:

- `recode`: transforma e recodifica variáveis.

Exemplos:

1. Números inteiros em `string`:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df$RG <-
  expss::recode(pnadc2019_visita1_df$UF,
    11:17 ~ "Norte",
    21:29 ~ "Nordeste",
    31:35 ~ "Sudeste",
    41:43 ~ "Sul",
    50:53 ~ "Centro-Oeste"
  )
```

2. Números inteiros em números inteiros:

```
pnadc2019_visita1_df$RG <-
  expss::recode(pnadc2019_visita1_df$UF,
    11:17 ~ 1,
    21:29 ~ 2,
    31:35 ~ 3,
    41:43 ~ 4,
    50:53 ~ 5,
    TRUE ~ NA
  )
```

3. *Strings* em números inteiros:

```
pnadc2019_visita1_df$RG_transform <-
  expss::recode(pnadc2019_visita1_df$RG,
    "Norte" ~ 1,
    "Nordeste" ~ 1,
    "Sudeste" ~ 2,
    "Sul" ~ 1,
    "Centro-Oeste" ~ 1)
```

4. *Strings* em *Strings*:

```
pnadc2019_visita1_df$RG_transform <-
  expss::recode(pnadc2019_visita1_df$RG,
    "Norte" ~ "BR - SE",
    "Nordeste"~ "BR - SE",
    "Sudeste" ~ "SE",
    "Sul"~ "BR - SE",
    "Centro-Oeste" ~ "BR - SE")
```

Após a transformação realizada no exemplo acima (números inteiros em números inteiros), pode ser necessária a adição de rótulos à nova variável categórica criada. Para isso, podem-se utilizar os seguintes comandos:

```
var_lab(pnadc2019_visita1_df$RG) = "RG"
val_lab(pnadc2019_visita1_df$RG) = c("Norte" = 1,
                                      "Nordeste" = 2,
                                      "Sudeste" = 3,
                                      "Sul" = 4,
                                      "Centro-Oeste" = 5)
```

9.4 Criação de tabelas

Nas análises descritivas, é bastante comum o uso da forma de tabelas para a apresentação dos resultados de uma pesquisa. Essas tabelas podem assumir distintas formas e conter diversas informações. Nesta seção, serão apresentadas algumas possibilidades para se explorar os dados da PNAD contínua.

9.4.1 A função `fre`

Uma das formas mais básicas e úteis de se inspecionar uma variável contida em um `data.frame` é por meio da função `fre`, contida no pacote `expss`. Essa função retorna um `data.frame` com seis colunas contendo: rótulos ou valores, contagens, porcentagem válida (excluindo `NA`), porcentagem total (com `NA`), porcentagem de respostas (para coluna única, `x` é igual à porcentagem válida) e porcentagem cumulativa.

Sua forma básica é a seguinte:

```
fre(  
  x,  
  weight = NULL,  
  drop_unused_labels = TRUE,  
  prepend_var_lab = FALSE,  
  stat_lab = getOption("expss.fre_stat_lab",  
  c("Count", "Valid percent", "Percent",  
    "Responses, %", "Cumulative responses, %"))  
)
```

A seguir, apresentam-se dois exemplos para a variável `RG` (Região Geográfica) criada anteriormente:

1. Dados amostrais:

```
fre(pnadc2019_visita1_df$RG,  
     weight = NULL)
```

2. Dados ponderados:

```
fre(pnadc2019_visita1_df$RG,  
     weight = pnadc2019_visita1_df$V1032)
```

9.4.2 A função `tables`

A construção de tabelas por meio do pacote **expss** consiste no uso de, ao menos, três funções encadeadas pelo operador `%>%` (tubo ou pipe)². Primeiramente, é necessário especificar as variáveis para as quais as estatísticas serão calculadas, a partir da função `tab_cells`. Na sequência, é possível calcular algumas estatísticas a partir da função `tab_stat_*`. Por fim, para criar a tabela, é necessário usar a função `tab_pivot`.

É possível, ainda, dividir as estatísticas por colunas por meio da função `tab_cols` ou, por linhas, com `tab_rows`. Pode-se, também, ordenar o resultado da tabela criada com as funções `tab_sort_asc`, eliminar linhas/colunas vazias com `drop_rc` e/ou transpor os resultados com `tab_transpose`.

Geralmente, a tabela resultante é apenas um `data.frame` em que se podem usar distintas operações arbitrárias para modificá-lo, como variáveis de coluna/linha adicionais, peso (ponderação), apresentar valores omissos e subgrupos para distintos recortes de análise. As estatísticas calculadas são sempre apresentadas nas últimas células do `data.frame`. A função `tab_pivot` define diferentes estatísticas e onde os rótulos das estatísticas aparecerão – dentro ou fora das linhas/colunas.

A forma básica para a construção de tabelas com a função `tables` é a seguinte:

```
dataset %>%
  tab_cells (variable) %>%
  tab_stat_... () %>%
  tab_pivot ()
```

De acordo com a própria documentação do pacote disponível no site do CRAN³, é possível que se criem distintas tabelas, cujo conteúdo pode ser definido por estatísticas de contagem de casos, de média, moda, mediana, desvio-padrão etc.

Quanto à forma que essa tabela de resultados assumirá, pode-se constatar que essas podem apresentar: valores absolutos (amostrais ou expandidos com a estimativa populacional) ou em percentual.

A seguir, são apresentados alguns exemplos e propostos alguns exercícios:

² Esse operador foi apresentado no Capítulo 7.

³ <https://cran.r-project.org/web/packages/expss/vignettes/examples.html>

POR DENTRO DA PNAD CONTÍNUA

→ Tabelas cruzadas entre as variáveis v2007 (sexo) e RG (Região Geográfica)⁴.

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_stat_cases(total_statistic = "u_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

⁴ Para a construção dessas tabelas, deve-se notar que serão necessários alguns comandos específicos: `tab_weight()` – para definição do peso usado para a estimativa populacional; `tab_stat_...()` – para a definição do resultado (percentual ou absoluto) das tabelas. Dentre as estatísticas possíveis a serem calculadas para o total das tabelas `total_statistic()`, destacam-se: `u_cases`, `u_cpct`, `u_rpct`, `u_tpct` (para os casos não ponderados) e `w_cases`, `w_responses`, `w_cpct`, `w_rpct`, `w_tpct` (para casos ponderados).

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_rpct(total_statistic = "w_rpct",
                 total_label = "Total") %>%
  tab_pivot()
```

- Tabelas cruzadas entre as variáveis V2007 (sexo), RG (Região Geográfica) e V1022 (Situação do Domicílio).

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%
  tab_cells(V1022) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_stat_cases(total_statistic = "u_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

POR DENTRO DA PNAD CONTÍNUA

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(V1022) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(V1022) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(V1022) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
```

```
tab_weight(weight = V1032) %>%  
  tab_stat_rpct(total_statistic = "w_rpct",  
                total_label = "Total") %>%  
  tab_pivot()
```

- Tabelas cruzadas entre as variáveis RG (Região Geográfica), V2007 (sexo) e V1022 (Situação do Domicílio) para múltiplas colunas.

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%  
  tab_cells(RG) %>%  
  tab_cols(total(), V2007, V1022) %>%  
  tab_stat_cases(total_statistic = "u_cases",  
                 total_label = "Total") %>%  
  tab_pivot()
```

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%  
  tab_cells(RG) %>%  
  tab_cols(total(), V2007, V1022) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_cases(total_statistic = "w_cases",  
                 total_label = "Total") %>%  
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007, V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007, V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_rpct(total_statistic = "w_rpct",
                 total_label = "Total") %>%
  tab_pivot()
```

- Tabelas cruzadas entre as variáveis RG (Região Geográfica), v2007 (sexo) e v1022 (Situação do Domicílio) para Variáveis aninhadas nas colunas.

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V1022) %>%
  tab_stat_cases(total_statistic = "u_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_rpct(total_statistic = "w_rpct",
                 total_label = "Total") %>%
  tab_pivot()
```

POR DENTRO DA PNAD CONTÍNUA

- Tabelas cruzadas entre as variáveis RG (Região Geográfica), V2007 (sexo) e V1022 (Situação do Domicílio) para múltiplos aninhamentos nas colunas.

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), list(V2007,V2010) %nest% V1022) %>%
  tab_stat_cases(total_statistic = "u_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), list(V2007,V2010) %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), list(V2007,V2010) %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
```

```
total_label = "Total") %>%  
tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%  
  tab_cells(RG) %>%  
  tab_cols(total(), list(V2007,V2010) %nest% V1022) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_rpct(total_statistic = "w_rpct",  
                total_label = "Total") %>%  
  tab_pivot()
```

- Tabelas cruzadas entre as variáveis RG (Região Geográfica), v2007 (sexo) e v1022 (Situação do Domicílio) para múltiplos aninhamentos nas colunas – outra versão.

1. Contagem dos casos na amostra:

```
pnadc2019_visita1_df %>%  
  tab_cells(RG) %>%  
  tab_cols(total(), V2007 %nest% V2010 %nest% V1022) %>%  
  tab_stat_cases(total_statistic = "u_cases",  
                 total_label = "Total") %>%  
  tab_pivot()
```

2. Contagem da população estimada baseada no peso com pós-estratificação pela projeção de população:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V2010 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Percentual por colunas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V2010 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

4. Percentual por linhas da população estimada baseada no peso com pós-estratificação pela projeção de população:

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007 %nest% V2010 %nest% V1022) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_rpct(total_statistic = "w_rpct",
                 total_label = "Total") %>%
  tab_pivot()
```

Outra possibilidade é a adição de estatísticas referentes a variáveis contínuas, para serem calculadas no corpo da tabela por meio da função `tab_stat_fun()`. Dentre as estatísticas possíveis, destacam-se:

- `w_mean`: média ponderada de um vetor numérico;
 - `w_sd`: desvio-padrão da amostra ponderada de um vetor numérico;
 - `w_var`: variância da amostra ponderada de um vetor numérico;
 - `w_se`: erro padrão ponderado de um vetor numérico;
 - `w_median`: mediana ponderada de um vetor numérico;
 - `w_mad`: desvio absoluto médio ponderado da mediana de um vetor numérico;
 - `w_sum`: soma ponderada de um vetor numérico;
 - `w_n`: número ponderado de valores de um vetor numérico;
 - `w_cov`: matriz de covariância ponderada de uma matriz numérica e/ou `data.frame`;
 - `w_cor`: matriz de correlação de *Pearson* ponderada de uma matriz numérica e/ou `data.frame`;
 - `w_pearson`: atalho para `w_cor`; `w_spearman` (matriz ponderada de correlação de *Spearman* de uma matriz numérica e/ou `data.frame`).
- Tabelas cruzadas entre as variáveis `RG` (Região Geográfica) e `V2007` (sexo) com o cálculo de estatísticas a partir da variável `VD5005` (Rendimento efetivo domiciliar *per capita*).

1. Rendimento médio:

```
pnadc2019_visita1_df %>%  
  tab_cells(VD5005) %>%  
  tab_cols(total(), V2007) %>%  
  tab_rows(RG, total()) %>%
```

POR DENTRO DA PNAD CONTÍNUA

```
tab_weight(weight = V1032) %>%
  tab_stat_fun(Média = w_mean) %>%
  tab_pivot()
```

2. Soma dos Rendimentos (Massa):

```
pnadc2019_visita1_df %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun(Soma = w_sum) %>%
  tab_pivot()
```

3. Desvio-padrão do rendimento médio:

```
pnadc2019_visita1_df %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun("Desvio-Padrão" = w_sd) %>%
  tab_pivot()
```

4. Soma e Média dos rendimentos na mesma tabela:

```
pnadc2019_visita1_df %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007) %>%
```

```
tab_rows(RG, total()) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_fun(Média = w_mean,  
               Soma = w_sum) %>%  
  tab_pivot()
```

Existe, ainda, a possibilidade de se filtrar os dados para definir o conteúdo da tabela de forma a que esse seja condicionado a um subgrupo específico da amostra/população. Para isso, é necessário que se use a função `tab_subgroup()`. A seguir, seguem dois exemplos:

1. Número de Pessoas Desocupadas (VD4002= 2) por RG (Região Geográfica), V2007 (sexo) e V1022 (Situação do Domicílio):

```
pnadc2019_visita1_df %>%  
  tab_subgroup(VD4002==2) %>%  
  tab_cells(RG) %>%  
  tab_cols(total(), V2007 %nest% V1022) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_cases(total_statistic = "w_cases",  
                 total_label = "Total") %>%  
  tab_pivot()
```

2. Rendimento médio efetivo domiciliar *per capita* para pessoas desocupadas (VD4002= 2) por RG (Região Geográfica), V2007 (sexo) e V1022 (Situação do Domicílio):

```
pnadc2019_visita1_df %>%  
  tab_subgroup(VD4002==2) %>%  
  tab_cells(VD5005) %>%  
  tab_cols(total(), V2007) %>%
```

POR DENTRO DA PNAD CONTÍNUA

```
tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun(Média = w_mean) %>%
  tab_pivot()
```

Podem-se combinar, na mesma tabela, informações de população com informações de variáveis contínuas como nos exemplos a seguir:

1. Número total de pessoas e em percentual por RG (Região Geográfica) e V2007 (sexo):

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Número de pessoas e rendimento médio efetivo domiciliar *per capita* por RG (Região Geográfica) e V2007 (sexo):

```
pnadc2019_visita1_df %>%
  tab_cells(RG) %>%
  tab_cols(total(), V2007) %>%
  tab_weight(weight = V1032) %>%
```

```
tab_stat_cases(total_statistic = "w_cases",
               total_label = "Total") %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun(Média = w_mean) %>%
  tab_pivot()
```

A utilização do comando `list()`, internamente às funções `tab_cols()` ou `tab_rows()`, permite que se criem subtotais nas colunas ou nas linhas, como nos exemplos a seguir:

1. Rendimento médio efetivo domiciliar *per capita* por RG (Região Geográfica), `v2007` (sexo) e `v1022` (Situação do Domicílio) com subtotal nas colunas:

```
pnadc2019_visita1_df %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007 %nest% list(V1022, total())) %>%
  tab_rows(RG) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun(Média = w_mean) %>%
  tab_pivot()
```

2. Rendimento médio efetivo domiciliar *per capita* por RG (Região Geográfica), `v2007` (sexo) e `v1022` (Situação do Domicílio) com subtotal nas linhas:

```
pnadc2019_visita1_df %>%
  tab_cells(VD5005) %>%
  tab_cols(total(), V2007) %>%
  tab_rows(RG %nest% list(V1022, total())) %>%
```

```
tab_weight(weight = V1032) %>%
  tab_stat_fun(Média = w_mean) %>%
  tab_pivot()
```

9.4.3 A função `cro`

Uma das ferramentas mais tracionais do SPSS era a chamada “Tabela de referência cruzada” com suporte a rótulos, pesos e variáveis de resposta múltipla. No `expss`, distintas funções servem a esse propósito, como são os casos das funções: `cro` e `cro_cases`, que permitem a construção de tabelas cruzadas para contagem de casos (amostrais ou ponderados). Também é possível a construção de tabelas com a porcentagem na coluna (`cro_cpct`) ou na linha (`cro_rpct`). Deve-se destacar que essas funções fornecem resultados com base na porcentagem de casos válidos (se tiver pelo menos um valor não NA).

A função `cro_tpct` constrói uma tabela de contingência da porcentagem da tabela, cuja base para a porcentagem, também, é o número de casos válidos. As funções do tipo `calc_cro_*` são iguais às anteriores, mas trazem como primeiro argumento o conjunto dos dados a serem tratados.

Suas formas básicas são as seguintes:

```
cro(
  cell_vars,
  col_vars = total(),
  row_vars = NULL,
  weight = NULL,
  subgroup = NULL,
  total_label = NULL,
  total_statistic = "u_cases",
  total_row_position = c("below", "above", "none")
)

cro_cases(
  cell_vars,
```

```
col_vars = total(),
row_vars = NULL,
weight = NULL,
subgroup = NULL,
total_label = NULL,
total_statistic = "u_cases",
total_row_position = c("below", "above", "none")
)

cro_cpct(
  cell_vars,
  col_vars = total(),
  row_vars = NULL,
  weight = NULL,
  subgroup = NULL,
  total_label = NULL,
  total_statistic = "u_cases",
  total_row_position = c("below", "above", "none")
)

cro_rpct(
  cell_vars,
  col_vars = total(),
  row_vars = NULL,
  weight = NULL,
  subgroup = NULL,
  total_label = NULL,
  total_statistic = "u_cases",
  total_row_position = c("below", "above", "none")
)

cro_tpct(
  cell_vars,
  col_vars = total(),
```

POR DENTRO DA PNAD CONTÍNUA

```
row_vars = NULL,  
weight = NULL,  
subgroup = NULL,  
total_label = NULL,  
total_statistic = "u_cases",  
total_row_position = c("below", "above", "none")  
)
```

As funções do tipo `calc_cro_*` assumem as mesmas formas das apresentadas anteriormente, com uma única diferença, a saber, seu primeiro argumento é o nome do objeto contendo o conjunto de dados⁵. A seguir, é apresentado uma única forma básica, de exemplo, pois as demais seguem a mesma regra⁶:

```
calc_cro(  
  data,  
  cell_vars,  
  col_vars = total(),  
  row_vars = NULL,  
  weight = NULL,  
  subgroup = NULL,  
  total_label = NULL,  
  total_statistic = "u_cases",  
  total_row_position = c("below", "above", "none")  
)
```

As funções do tipo `cro_*` necessitam ser chamadas dentro da função `calculate`, como no exemplo a seguir:

Exemplo: Tabela cruzada entre `RG` (Região Geográfica) e `v2007` (sexo) usando a função `cro_cases` associada à `calculate`:

```
calculate(pnadc2019_visita1_df,  
         cro_cases(
```

⁵ Nesse caso, o objeto contendo o `data.frame` pode ser passado por meio de um *pipe* – `%>%`.

⁶ Para mais detalhes, basta digitar, no console do R, o comando `?calc_cro`.

```
cell_vars = RG,  
col_vars = list(total(),V2007),  
row_vars = NULL,  
weight = V1032,  
subgroup = NULL,  
total_label = "Total",  
total_statistic = "w_cases",  
total_row_position = c("below"))
```

Já, para as funções do tipo `calc_cro_*`, isso não é necessário. Basta que se informe, como primeiro argumento, o conjunto de dados, como no exemplo a seguir:

Exemplo: Tabela cruzada entre `RG` (Região Geográfica) e `V2007` (sexo):

```
calc_cro_cases(pnadc2019_visita1_df,  
                cell_vars = RG,  
                col_vars = list(total(),V2007),  
                row_vars = NULL,  
                weight = V1032,  
                subgroup = NULL,  
                total_label = "Total",  
                total_statistic = "w_cases",  
                total_row_position = c("below"))
```

É importante reparar que os resultados são os mesmos. Isso indica que a forma `calc_cro_*` funciona como um atalho para as funções do tipo `cro_*`.

Exemplos:

1. Tabela cruzada entre `RG` (Região Geográfica) e `V2007` (sexo) com percentuais nas colunas:

POR DENTRO DA PNAD CONTÍNUA

```
calc_cro_cpct(pnadc2019_visita1_df,
               cell_vars = RG,
               col_vars = list(total(),V2007),
               row_vars = NULL,
               weight = V1032,
               subgroup = NULL,
               total_label = "Total",
               total_statistic = "w_cpct",
               total_row_position = c("below"))
```

2. Tabela cruzada entre RG (Região Geográfica) e V2007 (sexo) com percentuais nas linhas:

```
calc_cro_rpct(pnadc2019_visita1_df,
               cell_vars = RG,
               col_vars = list(total(),V2007),
               row_vars = NULL,
               weight = V1032,
               subgroup = NULL,
               total_label = "Total",
               total_statistic = "w_rpct",
               total_row_position = c("below"))
```

3. Tabela cruzada entre RG (Região Geográfica) e V2007 (sexo) com percentuais em relação ao total geral:

```
calc_cro_tpct(pnadc2019_visita1_df,
               cell_vars = RG,
               col_vars = list(total(),V2007),
               row_vars = NULL,
               weight = V1032,
```

```
subgroup = NULL,  
total_label = "Total",  
total_statistic = "w_tpct",  
total_row_position = c("below"))
```

Por fim cabe destacar que as funções do tipo `cro` podem ser usadas para o cálculo de estatísticas de variáveis contínuas. Dentre elas, destacam-se:

- `cro_mean`: calcula a média (NAs são sempre omitidos);
- `cro_sum`: calcula a soma (NAs são sempre omitidos);
- `cro_median`: calcula a mediana (NAs são sempre omitidos);
- `cro_mean_sd_n`: calcula simultaneamente a média, o desvio-padrão e o N;
- `cro_pearson`: calcula a correlação de Pearson da primeira variável em cada `data.frame`, definida no argumento `cell_vars` com outras variáveis;
- `cro_spearman`: calcula a correlação de Spearman da primeira variável em cada `data.frame` definida no argumento `cell_vars` com outras variáveis;
- `cro_fun`: cria uma tabela cruzada com estatísticas personalizadas definidas pelo argumento `fun`. O tratamento dos NAs depende da função definida.

Suas formas básicas são as seguintes:

```
#Usar com a função calculate  
cro_fun(  
  cell_vars,  
  col_vars = total(),  
  row_vars = total(label = ""),  
  weight = NULL,  
  subgroup = NULL,  
  fun,  
  ...)
```

POR DENTRO DA PNAD CONTÍNUA

```
unsafe = FALSE
)

calc_cro_fun(
  data,
  cell_vars,
  col_vars = total(),
  row_vars = total(label = ""),
  weight = NULL,
  subgroup = NULL,
  fun,
  ...,
  unsafe = FALSE
)
```

O termo `fun`, no nome da função, pode ser substituído pela estatística que se deseja como, por exemplo, `mean`, `median`, `sum` etc. Isso leva à não necessidade de se utilizar o argumento `fun` internamente à função. Note a diferença nos dois exemplos a seguir:

1. Tabela cruzada entre `RG` (Região Geográfica) e `V2007` (sexo) para o rendimento médio efetivo domiciliar *per capita*, usando a forma `calc_cro_fun`:

```
calc_cro_fun(pnadc2019_visita1_df,
              cell_vars = VD5005,
              col_vars = list(total(),V2007),
              row_vars = list(RG, total(label = "Total")),
              weight = V1032,
              subgroup = NULL,
              fun   = w_mean,
              unsafe = FALSE)
```

2. Tabela cruzada entre `RG` (Região Geográfica) e `V2007` (sexo) para o rendimento médio efetivo domiciliar *per capita*, usando a forma `calc_cro_mean`:

```
calc_cro_mean(pnadc2019_visita1_df,
               cell_vars = VD5005,
               col_vars = list(total(),V2007),
               row_vars = list(RG, total(label = "Total")),
               weight = V1032,
               subgroup = NULL)
```

Deve-se notar que, como se definiu no argumento `weight` uma variável de ponderação (`V1032`), então é necessário que a função utilizada no argumento `fun` seja uma das opções abaixo:

```
w_mean(x, weight = NULL, na.rm = TRUE)
w_median(x, weight = NULL, na.rm = TRUE)
w_var(x, weight = NULL, na.rm = TRUE)
w_sd(x, weight = NULL, na.rm = TRUE)
w_se(x, weight = NULL, na.rm = TRUE)
w_mad(x, weight = NULL, na.rm = TRUE)
w_sum(x, weight = NULL, na.rm = TRUE)
w_n(x, weight = NULL, na.rm = TRUE)
unweighted_valid_n(x, weight = NULL)
valid_n(x, weight = NULL)
w_max(x, weight = NULL, na.rm = TRUE)
w_min(x, weight = NULL, na.rm = TRUE)
w_cov(x, weight = NULL, use = c("pairwise.complete.obs",
                                "complete.obs"))
w_cor(x, weight = NULL, use = c("pairwise.complete.obs",
                                "complete.obs"))
w_pearson(x, weight = NULL, use = c("pairwise.complete.obs",
                                    "complete.obs"))
w_spearman(x, weight = NULL, use = c("pairwise.complete.obs",
                                       "complete.obs"))
```

Vale reparar, ainda, que o resultado é exatamente o mesmo, seja usando a função `cro`, dentro de `calculate`, ou `calc_cro`.

9.5 Apresentação e formatação de tabelas: a interface com outros softwares

9.5.1 Console do RStudio e RStudio Viewer

Até aqui, todos os resultados tiveram como saída o console do R, em sua forma padrão. No entanto, o pacote **expss** traz algumas opções para controlar seu comportamento em termos de exibição. Essas opções podem ser definidas por meio de funções especiais como:

- `expss_digits`: número de dígitos após o separador decimal que serão mostrados nas tabelas. `NULL` é o padrão e significa um dígito, enquanto que `NA` significa nenhum arredondamento;
- `expss_enable_value_labels_support`: por padrão, todas as variáveis usarão rótulos para os níveis das variáveis categóricas definidas como fatores;
- `expss_output....`: por padrão, as tabelas são impressas no console. Pode-se alterar esse comportamento alterando esta opção. Existem cinco opções possíveis: “rnotebook”, “viewer”, “comment”, “raw” ou “huxtable”.
 1. `expss_output_rnotebook()`: a primeira opção é útil quando você executa seu código no *Script*, e a saída é renderizada em um HTML, apresentado na janela de visualização.
 2. `expss_output_viewer`: esta opção também apresentará as tabelas no visualizador do RStudio⁷.
 3. `expss_output_commented()`: imprime a saída padrão para o console com o símbolo de comentário `#` no início de cada linha. Com isso pode-se facilmente copiar e colar sua saída no *Script*.
 4. `expss_output_raw()`: desativa qualquer formatação, e todas as tabelas são impressas como `data.frames`.

⁷ As funções `expss_fix_encoding_on` e `expss_fix_encoding_off` podem contribuir para solucionar problemas com a codificação de caracteres no RStudio Viewer no Windows.

5. `expss_output_huxtable()`: opção de saída de impressão `huxtable` por meio da biblioteca **huxtable**⁸.

Todas as distintas opções de visualização foram pensadas para o exemplo apresentado anteriormente, referente à “Tabela cruzada entre RG (Região Geográfica) e V2007 (sexo) para o rendimento médio efetivo domiciliar *per capita*, usando a forma `calc_cro_mean`”. Para gerá-la, basta realizar os seguintes comandos:

```
calc_cro_mean(pnadc2019_visita1_df,  
              cell_vars = VD5005,  
              col_vars = list(total(),V2007),  
              row_vars = list(RG, total(label = "Total")),  
              weight = V1032,  
              subgroup = NULL)
```

Deve-se destacar que todas essas funções precisam ser aplicadas antes do comando que gera a tabela para que se altere o comportamento do pacote e, assim, a forma de exibição.

Exemplos:

```
expss_output_huxtable()  
#inserir o comando para gerar a tabela  
  
expss_output_viewer()  
#inserir o comando para gerar a tabela  
  
expss_output_rnotebook()  
#inserir o comando para gerar a tabela
```

⁸ Esse pacote foi desenvolvido por Hugh-Jones (2021).

POR DENTRO DA PNAD CONTÍNUA

O resultado dessas funções pode ser visualizado nas Figuras 9.1 (a) e (b), respectivamente.

Figura 9.1: Distintas opções de visualização de tabelas com o pacote **expss**.

(a) Saída no console da tabela – função `expss_output_huxtable()`

		#Total	Sexo	
RG	Norte	VD5e+03	Homem	Mulher
Nordeste		905	901	909
Sudeste		1.83e+03	1.84e+03	1.81e+03
Sul		1.78e+03	1.8e+03	1.76e+03
Centro-Oeste		1.65e+03	1.68e+03	1.63e+03
Total	VD5e+03	Total	1.48e+03	1.48e+03
			1.47e+03	

(b) Saída no R Viewer da tabela – funções `expss_output_viewer()` e `expss_output_rnotebook()`

RG	Norte	VD5005	#Total		Sexo	
			Homem	Mulher	Homem	Mulher
Norte	VD5005	895.46	897.05	893.92		
Nordeste	VD5005	905.07	900.96	908.87		
Sudeste	VD5005	1825.97	1842.86	1810.48		
Sul	VD5005	1779.88	1800.46	1760.29		
Centro-Oeste	VD5005	1649.87	1675.77	1625.26		
Total	VD5005	1475.09	1484.91	1465.93		

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

As outras opções geram resultados no console, como demonstrado na sequência:

```
expss_output_default()
expss_digits(digits = 2)
#inserir o comando para gerar a tabela
```

			#Total	Sexo	
				Homem	Mulher
----- ----- ----- ----- ----- -----					
RG	Norte	VD5005	895.46	897.05	893.92
	Nordeste	VD5005	905.07	900.96	908.87
	Sudeste	VD5005	1825.97	1842.86	1810.48
	Sul	VD5005	1779.88	1800.46	1760.29
	Centro-Oeste	VD5005	1649.87	1675.77	1625.26
Total	VD5005	1475.09	1484.91	1465.93	

```
expss_output_commented()  
#Aqui, deve-se inserir o comando para gerar a tabela
```

#				#Total	Sexo	
#					Homem	Mulher
#	-----	-----	-----	-----	-----	-----
#	RG	Norte	VD5005	895.46	897.05	893.92
#		Nordeste	VD5005	905.07	900.96	908.87
#		Sudeste	VD5005	1825.97	1842.86	1810.48
#		Sul	VD5005	1779.88	1800.46	1760.29
#		Centro-Oeste	VD5005	1649.87	1675.77	1625.26
#	Total		VD5005	1475.09	1484.91	1465.93

```
expss_output_raw()  
#Aqui, deve-se inserir o comando para gerar a tabela
```

row_labels	#Total	Sexo	Homem	Sexo	Mulher
RG Norte VD5005	895.4608	897.0500		893.9166	
RG Nordeste VD5005	905.0652	900.9624		908.8720	
RG Sudeste VD5005	1825.9745	1842.8554		1810.4754	
RG Sul VD5005	1779.8847	1800.4597		1760.2947	
RG Centro-Oeste VD5005	1649.8662	1675.7715		1625.2576	
Total VD5005	1475.0865	1484.9099		1465.9255	

9.5.2 Excel

Para exportar tabelas elaboradas com o pacote `expss` para um arquivo `*.xlsx`, é necessário instalar o pacote `openxlsx`, criado por Schauberger e Walker (2021).

Para instalá-lo, basta digitar no console ou rodar, a partir do *Script*, os seguintes comandos:

POR DENTRO DA PNAD CONTÍNUA

```
install.packages ("openxlsx")
library(openxlsx)
```

No sistema Windows, pode ser necessária a instalação do RTools, que pode ser baixado no *site* do CRAN: RTools⁹.

Inicialmente, faz-se necessário armazenar o resultado em forma de tabela, como a última definida no exemplo da seção anterior, e armazená-lo em um objeto. Nesse caso, o objeto receberá o nome `tabela_exemplo`.

```
tabela_exemplo <- calc_cro_mean(pnadc2019_visita1_df,
  cell_vars = VD5005,
  col_vars = list(total(),V2007),
  row_vars = list(RG, total(label = "Total")),
  weight = V1032,
  subgroup = NULL)
```

Em seguida, faz-se necessário criar uma pasta de trabalho – `pt` – e adicionar, a ela, uma planilha – `pl`.

```
pt = createWorkbook ()
pl = addWorksheet (pt, "Planilha1")
```

Para Exportar, usa-se a função `xl_write()`. Além disso, é necessário especificar a pasta de trabalho e a planilha, anteriormente criadas.

```
xl_write (tabela_exemplo, pt, pl)
```

⁹ <https://cran.r-project.org/bin/windows/Rtools/>

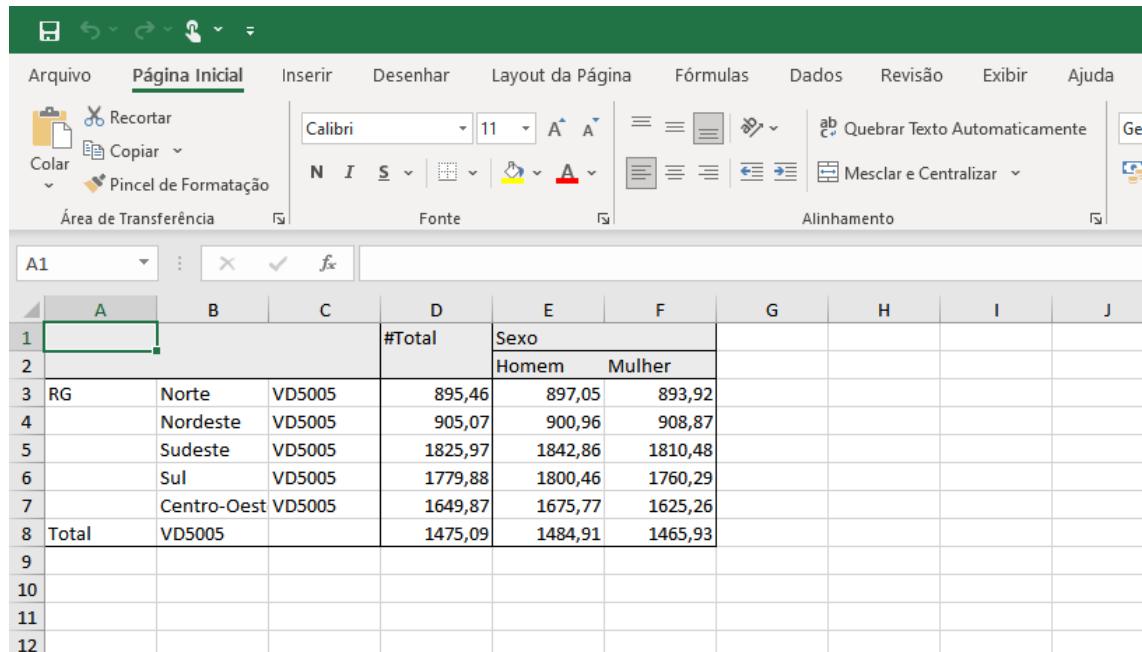
Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

Finalmente, deve-se salvar a pasta de trabalho com a tabela em arquivo `*.xlsx`, informando o diretório em que ele será salvo, usando a função `saveWorkbook`.

```
saveWorkbook (pt, "C:/Downloads/tabela_exemplo.xlsx", overwrite = TRUE)
```

O resultado pode ser visto na Figura 9.2.

Figura 9.2: Apresentação da tabela armazenada no objeto `tabela_exemplo` por meio do `openxlsx`



The screenshot shows a Microsoft Excel spreadsheet with a green header bar containing the ribbon tabs: Arquivo, Página Inicial, Inserir, Desenhar, Layout da Página, Fórmulas, Dados, Revisão, Exibir, and Ajuda. The 'Página Inicial' tab is selected. Below the ribbon, there are standard toolbar icons for Recortar (Cut), Copiar (Copy), Colar (Paste), and Pincel de Formatação (Format Painter). The font and alignment groups are visible on the right side of the ribbon. The main worksheet area has a table with the following data:

	A	B	C	D	E	F	G	H	I	J
1				#Total	Sexo					
2					Homem	Mulher				
3	RG	Norte	VD5005	895,46	897,05	893,92				
4		Nordeste	VD5005	905,07	900,96	908,87				
5		Sudeste	VD5005	1825,97	1842,86	1810,48				
6		Sul	VD5005	1779,88	1800,46	1760,29				
7		Centro-Oeste	VD5005	1649,87	1675,77	1625,26				
8	Total		VD5005	1475,09	1484,91	1465,93				
9										
10										
11										
12										

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

9.5.3 L^AT_EX

A função `kable` do pacote `knitr`, criado por Xie (2014), é um gerador de tabelas que possui um *design* simples em formato L^AT_EX. Ele apenas gera tabelas para dados estritamente retangulares, como matrizes e `data.frames` ou `tibbles`. É possível formatar células, número de dígitos, alinhamento de colunas, além de outras características por meio de argumentos para personalizar a aparência das tabelas. Sua forma básica é a seguinte:

POR DENTRO DA PNAD CONTÍNUA

```
kable(x,
      format,
      digits =getOption("digits"),
      row.names = NA,
      col.names = NA,
      align,
      caption = NULL,
      label = NULL,
      format.args = list(),
      escape = TRUE, ...)
```

Para gerar o resultado do exemplo anterior (`tabela_exemplo`) usando a função `kable`, basta usar o seguinte comando:

```
kable(tabela_exemplo, format = "latex")

\begin{tabular}{l|r|r|r}
\hline
row_labels & \#Total & 1 & 2 \\
\hline
Centro-Oeste|VD5005 & 1649.8662 & 1675.7715 & 1625.2576\\
\hline
Nordeste|VD5005 & 905.0652 & 900.9624 & 908.8720\\
\hline
Norte|VD5005 & 895.4608 & 897.0500 & 893.9166\\
\hline
Sudeste|VD5005 & 1825.9745 & 1842.8554 & 1810.4754\\
\hline
Sul|VD5005 & 1779.8847 & 1800.4597 & 1760.2947\\
\hline
Total|VD5005 & 1475.0865 & 1484.9099 & 1465.9255\\
\hline
\end{tabular}
```

Pode-se editar, ainda, o resultado apresentado no console do RStudio para melhorar a apresentação da tabela que será mostrada no documento em pdf (ver exemplo de edição a seguir).

```
\begin{table}
\caption{Rendimento médio efetivo domiciliar per capita por Região Geográfica e sexo. Brasil, 2019}
\centering
\begin{tabular}{t}[t]{lccc}
\toprule
Região Geográfica & Total & Homem & Mulher\\
\midrule
Centro-Oeste|VD5005 & 1.650 & 1.676 & 1.625\\
Nordeste|VD5005 & 905 & 901 & 909\\
Norte|VD5005 & 895 & 897 & 894\\
Sudeste|VD5005 & 1.826 & 1.843 & 1.810\\
Sul|VD5005 & 1.780 & 1.800 & 1.760\\
\addlinespace
Total|VD5005 & 1.475 & 1.485 & 1.466\\
\bottomrule
\multicolumn{4}{l}{\rule{0pt}{1em}\textit{Fonte: Elaboração própria}}\\
&a partir da PNAD Contínua anual de 2019 -- visita 1}\\
\end{tabular}
\end{table}
```

Assim, a forma final editada para apresentação em um documento .pdf pode ser visualizada na Tabela 9.1.

Por fim, destaca-se que esse pacote pode contribuir de forma significativa para a elaboração de tabelas de distintas formas e com diversos tipos de estatística. Tudo isso para um número expressivo de possibilidades de recortes de análise.

POR DENTRO DA PNAD CONTÍNUA

Tabela 9.1: Rendimento médio efetivo domiciliar *per capita* por Região Geográfica e sexo no Brasil (2019)

Região Geográfica	Total	Homem	Mulher
Centro-Oeste VD5005	1.650	1.676	1.625
Nordeste VD5005	905	901	909
Norte VD5005	895	897	894
Sudeste VD5005	1.826	1.843	1.810
Sul VD5005	1.780	1.800	1.760
Total VD5005	1.475	1.485	1.466

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

9.6 Exercícios

1. Calcule o total de pessoas ocupadas nas unidades da federação do Nordeste segundo UF e sexo.
2. Calcule o rendimento real médio das pessoas ocupadas nas unidades da federação do Nordeste segundo UF e sexo.
3. Calcule o número de pessoas desalentadas segundo sexo, cor/raça e Regiões Geográficas de forma combinada.
4. Calcule a massa real de rendimentos de todas as fontes das pessoas desalentadas segundo sexo, cor/raça e Regiões Geográficas de forma combinada.

10 Interface gráfica: o pacote **ggplot2**

O pacote **ggplot2** foi desenvolvido por Wickham *et al.* (2016) para a construção de gráficos estatísticos que permitem a combinação de componentes independentes, não se limitando a conjuntos de gráficos predefinidos. Ele permite que se criem novos gráficos adaptados ao objeto de estudo, por possuir uma gramática subjacente baseada na chamada Gramática de Gráficos (WILKINSON, 2013).

Essa gramática estabelece quais variáveis do conjunto de dados serão “mapeadas” como atributos estéticos (eixos, cor, forma, tamanho) e repassadas subsequentemente a objetos geométricos (pontos, linhas, barras). Os gráficos em **ggplot2**, também, comportam transformações estatísticas de dados além da incorporação de informações sobre o sistema de coordenadas.

De forma geral, os gráficos são compostos por um conjunto de dados, por informações que se deseja visualizar e um mapeamento, isto é, a descrição de como as variáveis serão mapeadas conforme os atributos estéticos a serem definidos. Assim, a ideia básica é a de que se pode construir todos os gráficos com os mesmos componentes: os dados, um sistema de coordenadas e os **geoms** (marcas visuais que representam pontos de dados).

10.1 Definições básicas

São cinco os componentes possíveis de mapeamento:

- **layer**: as camadas são conjuntos de elementos geométricos (**geoms**) e de transformações estatísticas (**stats**). Os **geoms** são os componentes visuais dos gráficos (pontos, linhas, barras, áreas, polígonos etc.), e as transformações esta-

POR DENTRO DA PNAD CONTÍNUA

tísticas são a síntese de dados, que podem ocorrer por meio, por exemplo, de definições de agrupamentos, contagem de observações (histograma) ou ajustes de modelos lineares;

- **scales**: função que dimensiona os valores no espaço de dados para valores no espaço estético, incluindo o uso de cores, formas ou tamanhos. Essa função também permite a definição e a forma de legendas e de eixos, possibilitando a leitura dos valores nos gráficos;
- **coord**: representa o sistema de coordenadas e descreve como essas serão mapeadas no plano do gráfico. Fornece eixos e linhas de grade para ajudar a leitura dos dados representados no gráfico. Comumente se usam as coordenadas cartesianas, porém vários outros tipos estão disponíveis, o que inclui coordenadas polares e mapas, por exemplo;
- **facet**: essa função especifica a divisão da exibição dos subconjuntos de dados a serem representados nos gráficos que devem aparecer em separado segundo o critério escolhido;
- **theme**: permite o controle fino da aparência dos gráficos, como o tamanho da fonte e a cor de fundo.

10.2 Instalação e base de dados

O pacote **ggplot2** está contido no pacote **tidyverse**, porém pode ser instalado individualmente e aplicado por meio dos seguintes comandos:

```
install.packages("ggplot2")
library(ggplot2)
```

Todos os exemplos do presente capítulo estão baseados no conjunto de dados da PNAD Contínua anual para a visita 1, já utilizado nos capítulos anteriores.

Assim, o objeto necessário para dar prosseguimento é o mesmo que foi definido como `pnadc2019_visita1_df`.

Caso seja necessário obter os dados novamente, basta utilizar o seguinte comando:

```
pnadc2019_visita1_df <-  
  get_pnadc(2019,  
             interview = 1,  
             design = FALSE,  
             labels = FALSE)
```

10.3 Componentes básicos

Cada gráfico elaborado com o pacote **ggplot2** tem três componentes principais:

- **data**: o conjunto de dados a serem explorados;
- **aes**: o conjunto de mapeamentos estéticos que incluem variáveis e propriedades visuais;
- **geom**: a definição de uma ou mais camadas que descrevem como tratar/apresentar as observações a partir de uma função do tipo `geom`.

10.4 Tipos básicos de apresentação de dados na forma gráfica

Os `geoms` são os blocos fundamentais para a construção de gráficos no **ggplot2**. Grande parte dos `geoms` está associada a um tipo de gráfico específico. Sua forma básica é cartesiana bidimensional e exige parâmetros estéticos para os eixos vertical `x` e horizontal `y`. Todos eles podem receber definições estéticas de cor (`colour`

POR DENTRO DA PNAD CONTÍNUA

), de tamanho (`size`) e de preenchimento (`fill`). Isso vale para qualquer tipo de gráfico definido na função `geom` como os de: barras, linhas, pontos ou polígonos.

A seguir, são apresentados alguns tipos básicos de gráficos definidos na função `geom`:

- `geom_area()`: cria um gráfico de área, isto é, uma linha preenchida até a base do eixo y. Distintos grupos definidos são empilhados;
- `geom_bar()`: cria um gráfico de barras. O argumento `stat = "identity"` é usado para deixar os dados inalterados, podendo várias barras serem empilhadas no mesmo local;
- `geom_line()`: cria um gráfico de linha. O argumento `group` determina quais observações serão conectadas. A função `geom_line()` conecta pontos da esquerda para a direita. A função `geom_path()` é semelhante, mas conecta os pontos na ordem em que aparecem nos dados. Para ambos, pode-se definir na função `aes` o argumento `linetype`, que mapeia uma variável categórica em linhas sólidas, pontilhadas e/ou tracejadas;
- `geom_point()`: cria um gráfico de dispersão. Um dos argumentos informados na função `aes` é, por exemplo, a forma (`shape`) dos pontos;
- `geom_polygon()`: desenha polígonos, que nada mais são que caminhos preenchidos. Cada vértice do polígono requer uma linha separada nos dados. Frequentemente, faz-se necessário mesclar dados de coordenadas poligonais com outros dados antes da plotagem;
- `geom_smooth()`: ajusta uma suavização dos dados, exibindo-a como uma faixa que incorpora o erro padrão;
- `geom_boxplot()`: produz um gráfico do tipo `boxplot` para resumir a distribuição de um conjunto de pontos;
- `geom_histogram()`: produz barras que servem para apresentar a distribuição de variáveis contínuas;
- `geom_freqpoly()`: apresenta a distribuição relativa de variáveis contínuas (no formato de linha).

Esses e outros tipos de gráficos serão apresentados no presente capítulo a partir de sua melhor aplicação para os dados da PNAD Contínua. Cada um desses gráficos tem uma aplicação determinada. Isso quer dizer que os resultados de um estudo podem ser mais bem apresentados graficamente a depender da escolha correta do tipo de gráfico a ser utilizado para os dados e as variáveis por esse explorados.

Assim, sua forma básica deve assumir a seguinte estrutura:

```
ggplot(data = NULL,  
       mapping = aes()) +  
       geom_...()
```

10.5 Representação gráfica dos valores de uma única variável contínua

Existem distintos gráficos que podem ser úteis para o estudo de variáveis contínuas. Os exemplos a seguir utilizarão o subconjunto de dados do Rio Grande do Norte, ao invés da PNAD Contínua completa¹. Isso permitirá ampliar a velocidade de processamento e facilitar sua reprodução. Deve-se destacar ainda que os resultados sempre são apresentados em uma janela que fica à direita e abaixo na interface do RStudio.

Os exemplos, a seguir, são apresentados na Figura 10.1.

- `geom_area()`: cria um gráfico de linha preenchida até a base do eixo y.

Exemplo: Representação gráfica da contagem da população a partir da variável `VD5005` (Rendimento efetivo domiciliar *per capita*):

```
pnadc2019_visita1_df %>%  
  filter(UF == 24) %>%  
  ggplot(aes(VD5005, weight = V1032)) +  
  geom_area(stat = "bin")
```

¹ Para isso, todos os comando contarão com o uso da função `filter` do pacote `dplyr`, que deverá assumir a seguinte forma: `filter(UF == 24)`.

POR DENTRO DA PNAD CONTÍNUA

- `geom_density()`: calcula e desenha a estimativa da densidade de Kernel, que é uma versão suavizada do histograma. Esta é uma alternativa útil ao histograma para dados contínuos.

Exemplo: Densidade de Kernel para a variável VD5005 (Rendimento efetivo domiciliar *per capita*):

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005, weight = V1032)) +
  geom_density()
```

Os histogramas e os denominados polígonos de frequência também são gráficos utilizados para a apresentação da distribuição de uma única variável contínua.

- `geom_histogram()`: desenha a distribuição absoluta de uma variável contínua.

Exemplo: Histograma para a variável VD5005 (Rendimento efetivo domiciliar *per capita*) – deve-se reparar que, por padrão, a quantidade de faixas de rendimentos será igual a 30:

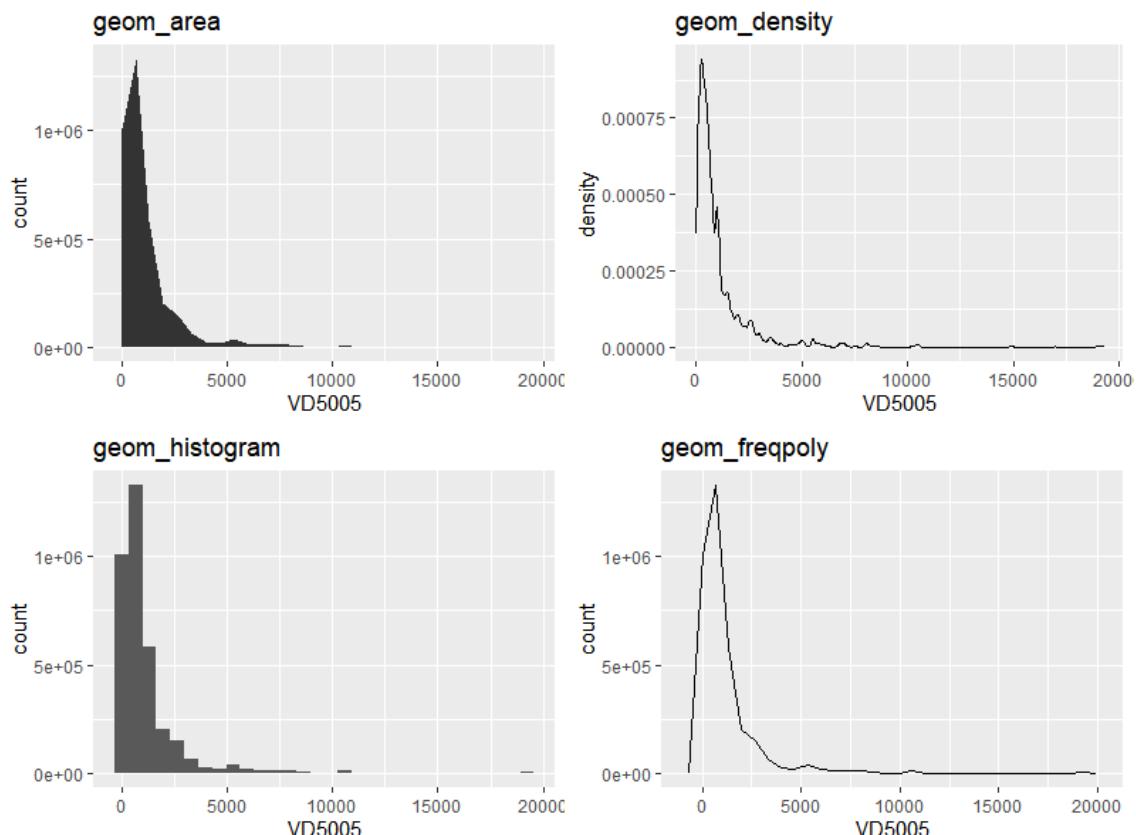
```
pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005, weight = V1032)) +
  geom_histogram()
```

- `geom_freqpoly()`: desenha a distribuição relativa de variáveis contínuas.

Exemplo: Polígono de frequência para a variável VD5005 (Rendimento efetivo domiciliar *per capita*):

```
pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005, weight = V1032)) +
  geom_freqpoly()
```

Figura 10.1: Tipos de gráficos para representar a distribuição de uma única variável contínua (VD5005).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Esses tipos de gráficos funcionam compartimentando os dados e, na sequência, contando o número de observações em cada compartimento. A diferença está na forma de exibição: os histogramas usam barras verticais; e os polígonos de frequê-

POR DENTRO DA PNAD CONTÍNUA

cia, linhas². É possível controlar a largura das barras que definem os compartimentos por meio do argumento `binwidth`. A definição padrão divide os dados em 30 compartimentos e, por isso, pode ser necessário testar outros valores para definir a melhor compartimentação para os dados específicos em estudo.

Exemplos (ver Figura 10.2):

1. Histograma para a variável VD5005 (Rendimento efetivo domiciliar *per capita*) com faixas de largura igual ao valor do salário-mínimo de 2019 (R\$ 998,00):

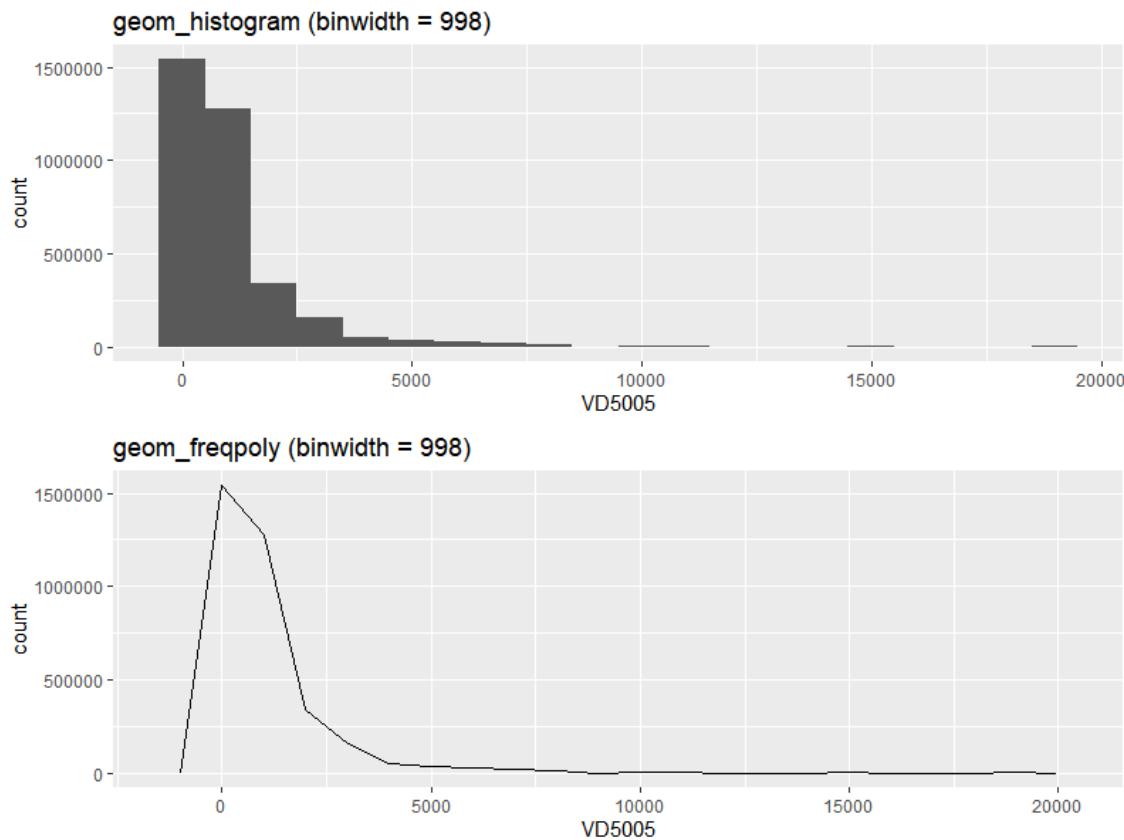
```
pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005, weight = V1032)) +
  geom_histogram(binwidth = 998)
```

2. Polígono de frequência para a variável VD5005 (Rendimento efetivo domiciliar *per capita*) com faixas de largura igual ao valor do salário-mínimo de 2019 (R\$ 998,00):

```
pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005, weight = V1032)) +
  geom_freqpoly(binwidth = 998)
```

² Alternativamente ao polígono de frequência, pode-se utilizar o gráfico de densidade (`geom_density()`).

Figura 10.2: Histograma e polígono de frequência para a variável (VD5005) com faixas de largura de tamanho igual ao valor do salário-mínimo de 2019 (R\$ 998,00).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.6 Representação gráfica dos valores de uma única variável discreta

A forma de apresentar valores de contagem para variáveis discretas pode se dar por meio de um gráfico de barras (`geom_bar()`). Os exemplos, a seguir, apresentam um gráfico simples de barras com os valores absolutos (amostrais e populacionais) para a variável sexo (v2007)³ (ver Figura 10.3).

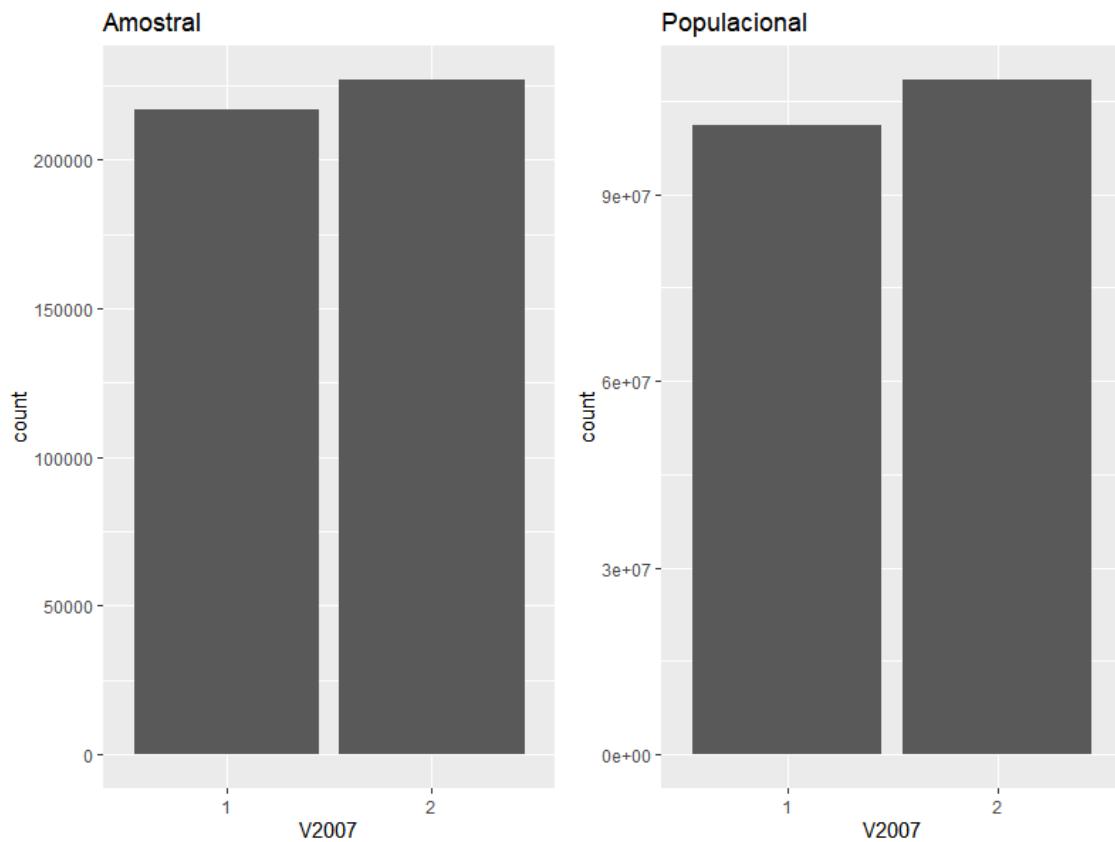
³ No dicionário da PNAD Contínua, pode-se verificar que os valores das categorias dessa variável são “Homem = 1” e “Mulher = 2”.

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df%>%
  ggplot(aes(V2007)) +
  geom_bar()

pnadc2019_visita1_df%>%
  ggplot(aes(V2007, weight = V1032)) +
  geom_bar()
```

Figura 10.3: Gráficos de barras com a contagem (amostral e populacional) para uma única variável discreta (`V2007`).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7 Representação gráfica dos valores de duas variáveis quaisquer

Boa parte das análises socioeconômicas se dá por meio de relações entre variáveis. Nesta seção, apresenta-se a potencialidade do pacote **ggplot2** para combinar variáveis e as formas de apresentação a partir de diversas opções de `geoms()`, para distintas combinações de variáveis.

10.7.1 Duas variáveis contínuas (x e y)

- `geom_point()`: cria um gráfico de dispersão. Útil para exibir a relação entre duas variáveis contínuas. Pode ser usado para comparar uma variável contínua e uma categórica, ou duas variáveis categóricas. Em alguns casos, as variações do tipo `geom_jitter`, `geom_count` ou `geom_bin2d` podem se mostrar mais apropriadas.

Exemplo: Cria um gráfico de dispersão para as variáveis de rendimento mensal no trabalho principal efetivo (VD4017) e habitual (D4016) para pessoas de 14 anos ou mais de idade:

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD4016 , y = VD4017, weight = V1032)) +
  geom_point()
```

Em gráficos de dispersão com muito ruído, pode ser difícil ver um padrão predominante. Nesses casos, é interessante adicionar uma linha suavizada ao gráfico. Isso é realizado por meio da função `geom_smooth()`, a qual adiciona uma curva suave, mostrando os intervalos de confiança das estimativas em cinza⁴. Para ocultar o intervalo de confiança, pode-se optar por usar o seguinte argumento `geom_smooth(se = FALSE)`.

⁴ Para grandes conjuntos de dados (mais de mil pontos), o **ggplot2** usa como padrão o seguinte modelo: `method = 'gam'` e `formula = 'y ~ s(x, bs = "cs")'`. O argumento `method` definido na função `geom_smooth()` permite que se escolha qual tipo de modelo é usado para ajustar a curva. Para mais detalhes sobre os distintos métodos de suavização, ver Wickham *et al.* (2016, cap. 3).

POR DENTRO DA PNAD CONTÍNUA

- `geom_smooth()`: auxilia a visualização de padrões na presença de sobreposição de pontos.

Exemplo: Cria um gráfico com curva suavizada para as variáveis de rendimento mensal no trabalho principal efetivo (VD4017) e habitual (D4016) para pessoas de 14 anos ou mais de idade⁵:

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD4016 , y = VD4017, weight = V1032)) +
  geom_smooth()
```

Exemplo: Cria um gráfico combinado de dispersão com curva suavizada para as variáveis de rendimento mensal no trabalho principal efetivo (VD4017) e habitual (D4016) para pessoas de 14 anos ou mais de idade:

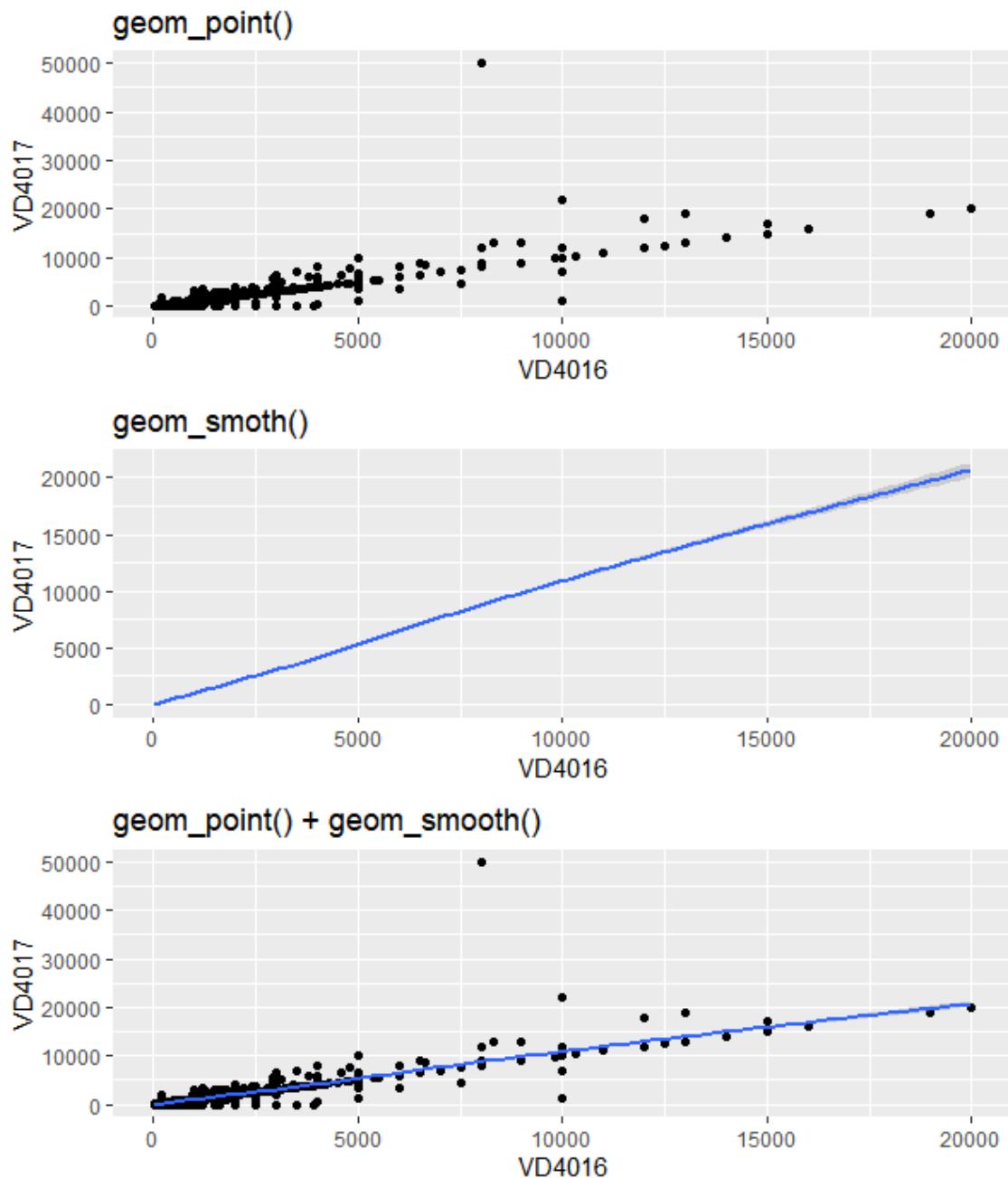
```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD4016 , y = VD4017, weight = V1032)) +
  geom_point() +
  geom_smooth()
```

Os resultados dos gráficos de pontos para a dispersão entre as variáveis VD4016 e VD4017 são exibidos na Figura 10.4.

⁵ Como o gráfico contém mais de mil pontos dispersos, talvez seja necessário habilitar a biblioteca mgcv, usada pelo `ggplot2` para definir o `method = "gam"` para grandes conjuntos de dados).

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

Figura 10.4: Gráficos de dispersão e curva suavizada para as variáveis VD4016 e VD4017.



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Deve ter ficado claro até esse ponto que os elementos básicos necessários para a construção de qualquer gráfico devem ser definidos da seguinte forma: o

argumento `data`, que recebe o nome do objeto contendo o conjunto de dados (`pnadc_2019_visita1_df`); o argumento `aes`, que recebe as variáveis e os eixos em que elas serão exibidas (`x` e `y`); e as camadas, nas quais se definem os tipos de gráfico (`geom_point()`, por exemplo).

A estrutura para a construção dos gráficos com o `ggplot2` exige, assim, que os dados e os mapeamentos estéticos sejam definidos na função `ggplot()`. Após o operador `+`, definem-se as camadas (por exemplo, a função contendo o tipo do gráfico (algum `geom_*`).

10.7.2 Uma variável discreta e uma variável contínua

É bastante comum no campo das Ciências Sociais Aplicadas que os estudos sobre renda utilizem recortes de análise para variáveis discretas como sexo, cor/raça, região geográfica, dentre outras. Para esse tipo de análise, destacam-se os seguinte tipos de gráficos (`geoms`):

- `geom_boxplot()`: exibe de forma compacta a distribuição de uma variável contínua, permitindo a visualização de cinco estatísticas resumidas (a mediana, o primeiro e o segundo quartil, além dos pontos individuais mínimo e máximo: *outlier*).

Exemplo: Cria um gráfico do tipo `boxplot` para o rendimento habitual no trabalho principal por sexo (v2007):

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = V2007, y = VD4016, weight = V1032)) +
  geom_boxplot()
```

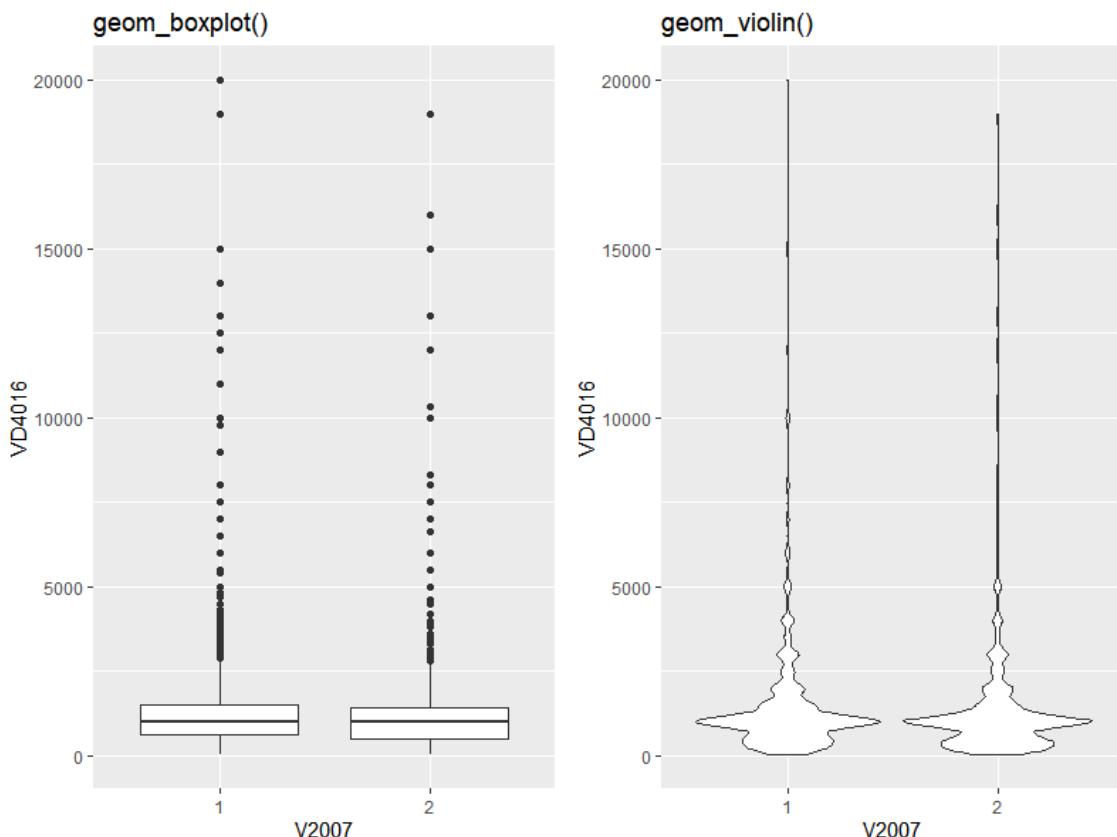
- `geom_violin()`: exibe de forma compacta uma distribuição contínua, misturando os tipos `geom_boxplot()` e `geom_density()`. O tipo violino é um gráfico de densidade espelhado, exibido da mesma forma que um `boxplot`.

Exemplo: Cria um gráfico do tipo `violin` para o rendimento habitual no trabalho principal por sexo (v2007):

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = V2007, y = VD4016, weight = V1032)) +
  geom_violin(scale = "area")
```

Os resultados desses dois exemplos (`boxplot` e `violin`) são exibidos na Figura 10.5.

Figura 10.5: Gráficos para análise de variáveis contínuas condicionadas a variáveis discreteas.



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.3 Duas variáveis discretas

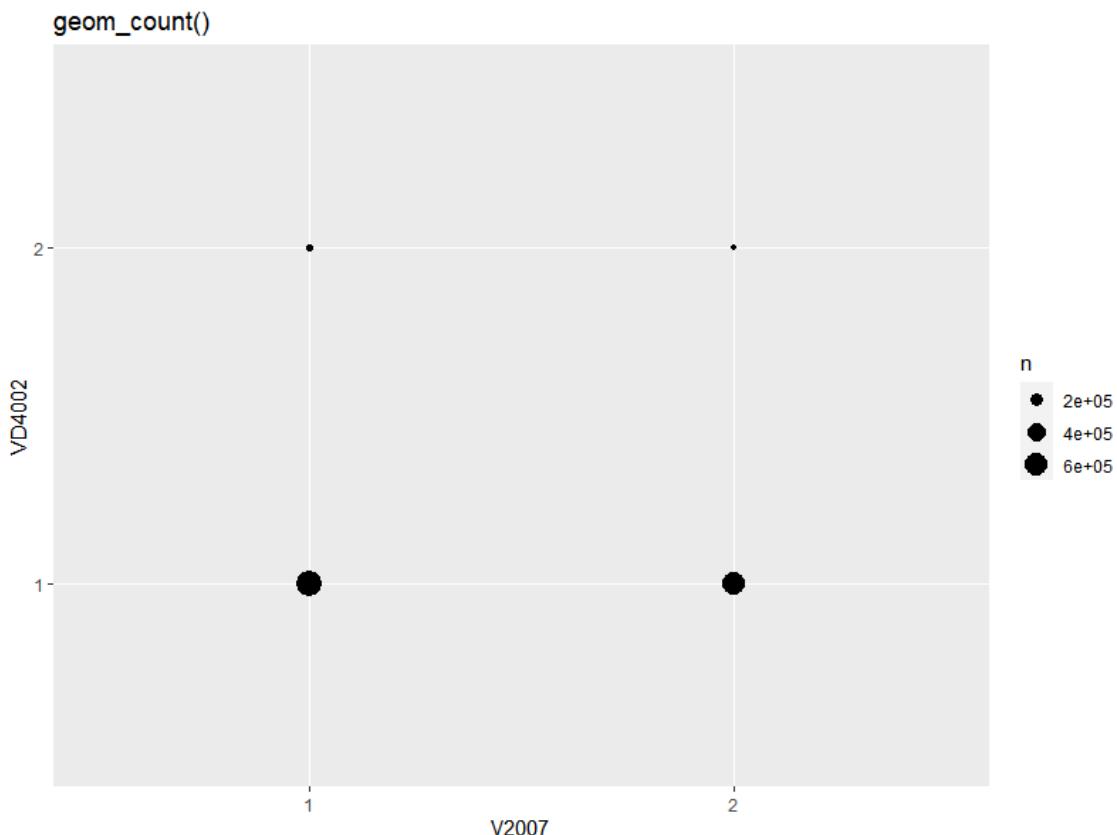
É possível, também, que se estude a relação entre duas variáveis discretas por meio da contagem de casos (ponderados ou não pelos pesos). O gráfico mais adequado para esse tipo de análise é o `geom_count`.

Exemplo: Cria um gráfico de contagem da população para as pessoas na força de trabalho (`VD4001 = 1`) segundo as variáveis `VD4002` (Pessoas ocupadas = 1 e Pessoas desocupadas = 2) e `V2007` (sexo), em que a categoria “Homem” assume o valor 1 e “Mulher” o valor 2:

```
pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4001 == 1) %>%
  ggplot(aes(x = V2007,
             y = VD4002,
             weight = V1032)) +
  geom_count()
```

Nesse tipo de gráfico (`geom_count`), o tamanho dos pontos está diretamente relacionado ao tamanho da população (ver Figura 10.6).

Figura 10.6: Gráfico de contagem para duas variáveis discretas.



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.4 Alterações estéticas e incorporação de variáveis para definição de subgrupos

O **ggplot2** permite, ainda, a incorporação de variáveis para destacar subgrupos por meio de definições estéticas como cor, forma e tamanho. Essas definições devem ser adicionadas como argumentos da função `aes()`:

Exemplos: Cria gráficos de dispersão para as variáveis de rendimento mensal no trabalho principal efetivo (`VD4017`) e habitual (`D4016`) com mapeamentos estéticos por cor, forma e tamanho de acordo com a variável “Sabe ler e escrever?” (`V3001`)⁶:

⁶ Para ampliar a didática e facilitar a visualização dos dados nos gráficos, optou-se por filtrar os dados para os rendimentos efetivos inferiores a R\$ 10.000,00, evitando alguns *outliers*.

POR DENTRO DA PNAD CONTÍNUA

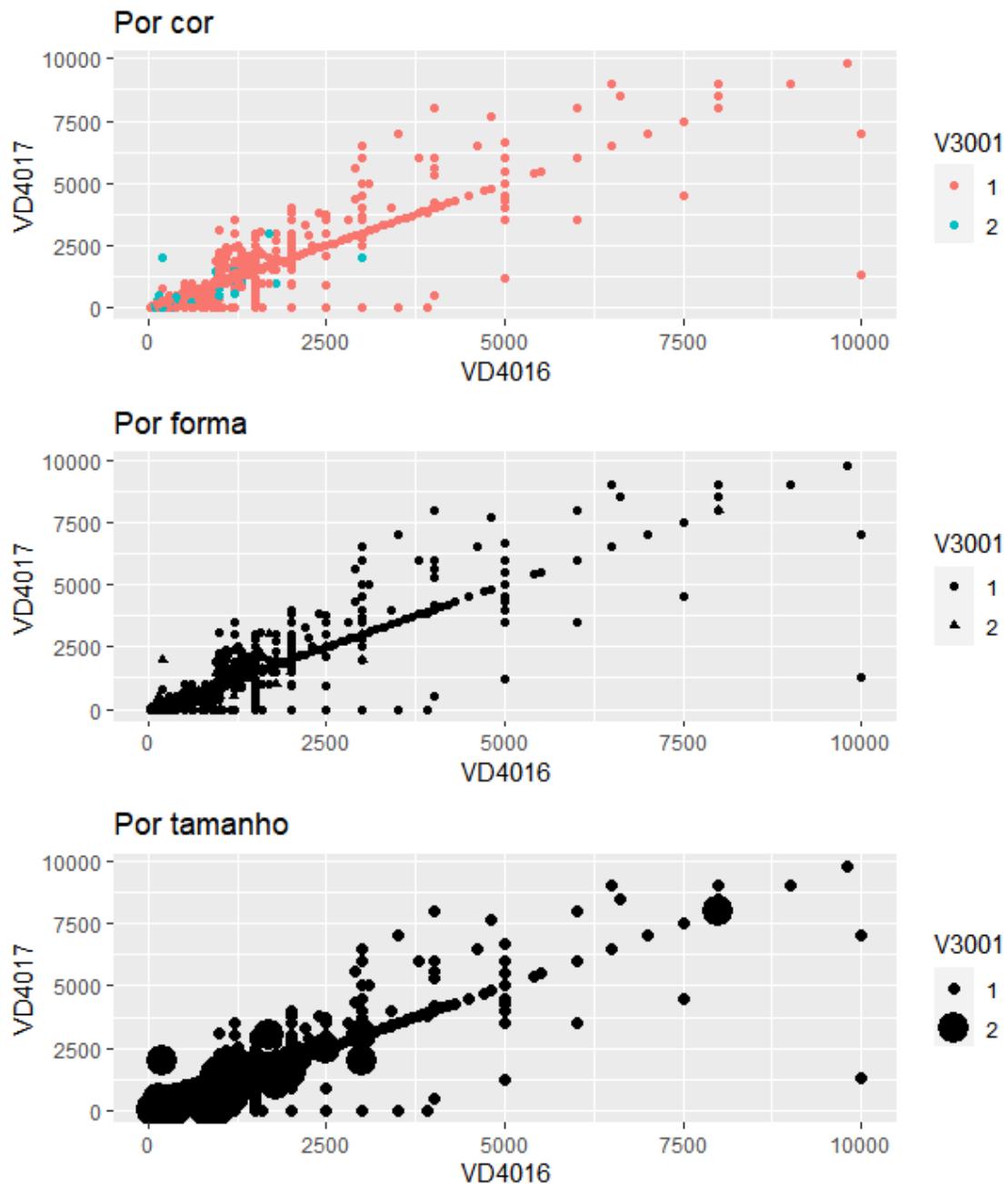
```
#Por cor
pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4017 < 10000) %>%
  ggplot(aes(x = VD4016,
              y = VD4017,
              colour = V3001,
              weight = V1032)) +
  geom_point()

#Por forma
pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4017 < 10000) %>%
  ggplot(aes(x = VD4016,
              y = VD4017,
              shape = V3001,
              weight = V1032)) +
  geom_point()

#Por tamanho
pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4017 < 10000) %>%
  ggplot(aes(x = VD4016,
              y = VD4017,
              size = V3001,
              weight = V1032)) +
  geom_point()
```

Deve-se reparar que alguns desses mapeamentos pode tornar difícil a visualização (por exemplo, por forma). Isso deve ser levado em conta na hora da escolha da melhor forma de apresentação dos dados (ver Figura 10.7).

Figura 10.7: Gráfico de dispersão para as variáveis VD4016 e VD4017, subdivididas por sexo (V2007).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

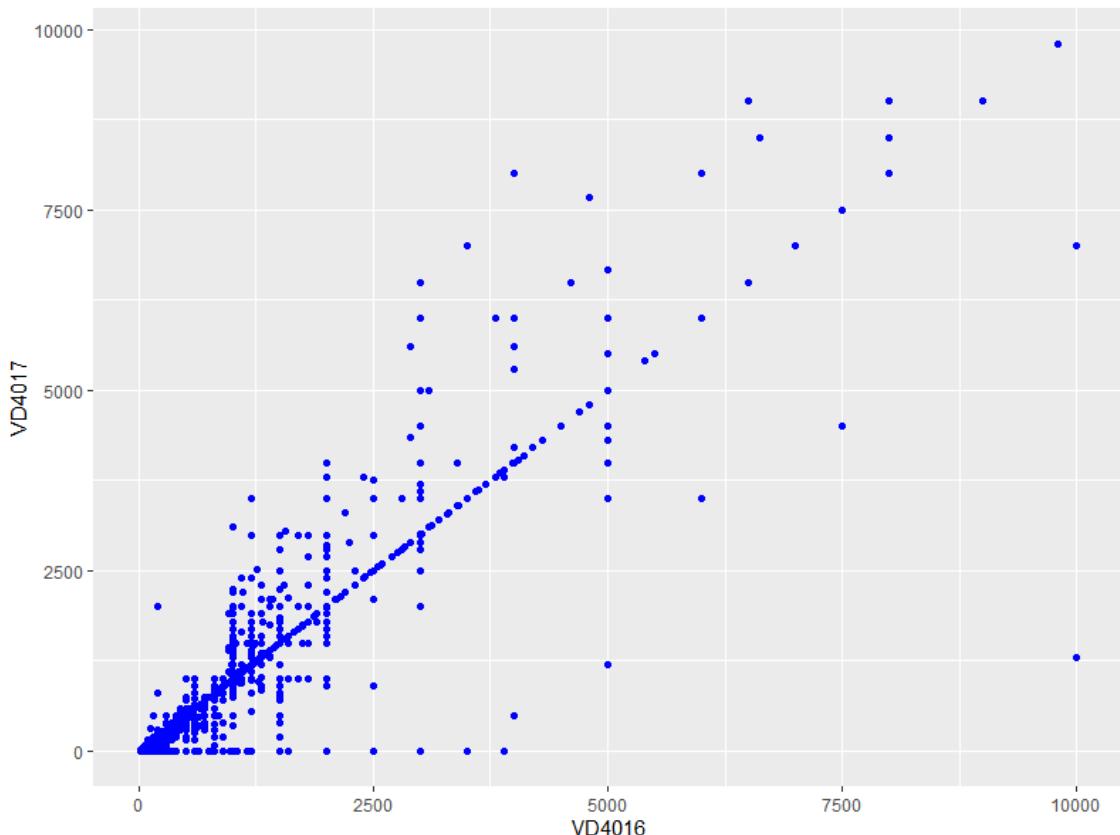
É possível, ademais, definir a cor dos elementos gráficos como, por exemplo os pontos, sem a necessidade de definir uma variável de dimensionamento. Para

POR DENTRO DA PNAD CONTÍNUA

isso, basta adicionar uma cor qualquer na função `geom_point()`, fora da função `aes`, por meio do argumento `colour` (ver Figura 10.8).

```
pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4017 < 10000) %>%
  ggplot(aes(x = VD4016,
             y = VD4017,
             weight = V1032)) +
  geom_point(colour = "blue")
```

Figura 10.8: Gráfico de dispersão para as variáveis VD4016 e VD4017 (alterando a cor dos pontos para azul).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Segundo Wickham *et al.* (2016)⁷, diferentes tipos de atributos estéticos podem funcionar melhor para diferentes variáveis.

Os atributos de cor e forma devem ser usados preferencialmente para variáveis categóricas. Já o tamanho aplica-se melhor a variáveis contínuas. Da mesma forma, gráficos de linhas são normalmente usados para explorar variáveis ao longo do tempo.

Deve-se tomar cuidado quando a quantidade dos dados é expressiva, pois pode ser difícil distinguir grupos diferentes, o que dificulta o mapeamento estético de variáveis que definem os subgrupos.

A apresentação de gráficos condicionada a valores de uma variável categórica, também, pode ser feita a partir da função `facet()`, que cria e exibe os dados separadamente por subconjuntos. Existem duas funções que possibilitam essa forma de apresentação: `facet_wrap()` e `facet_grid()`. Para realizar esse tipo de exibição, basta adicionar uma dessas duas funções como camada (após o operador `+`), informando a variável categórica que servirá para definir os subgrupos.

Exemplo: Cria gráficos de dispersão para as variáveis de rendimento mensal no trabalho principal efetivo (`VD4017`) e habitual (`D4016`) por subgrupos definidos pela variável discreta sexo (`V2007`):

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD4016,
             y = VD4017,
             weight = V1032)) +
  geom_point() +
  facet_wrap(~V2007)
```

⁷ O livro sobre o `ggplot2` pode ser acessado no seguinte endereço: <https://ggplot2-book.org/index.html>.

POR DENTRO DA PNAD CONTÍNUA

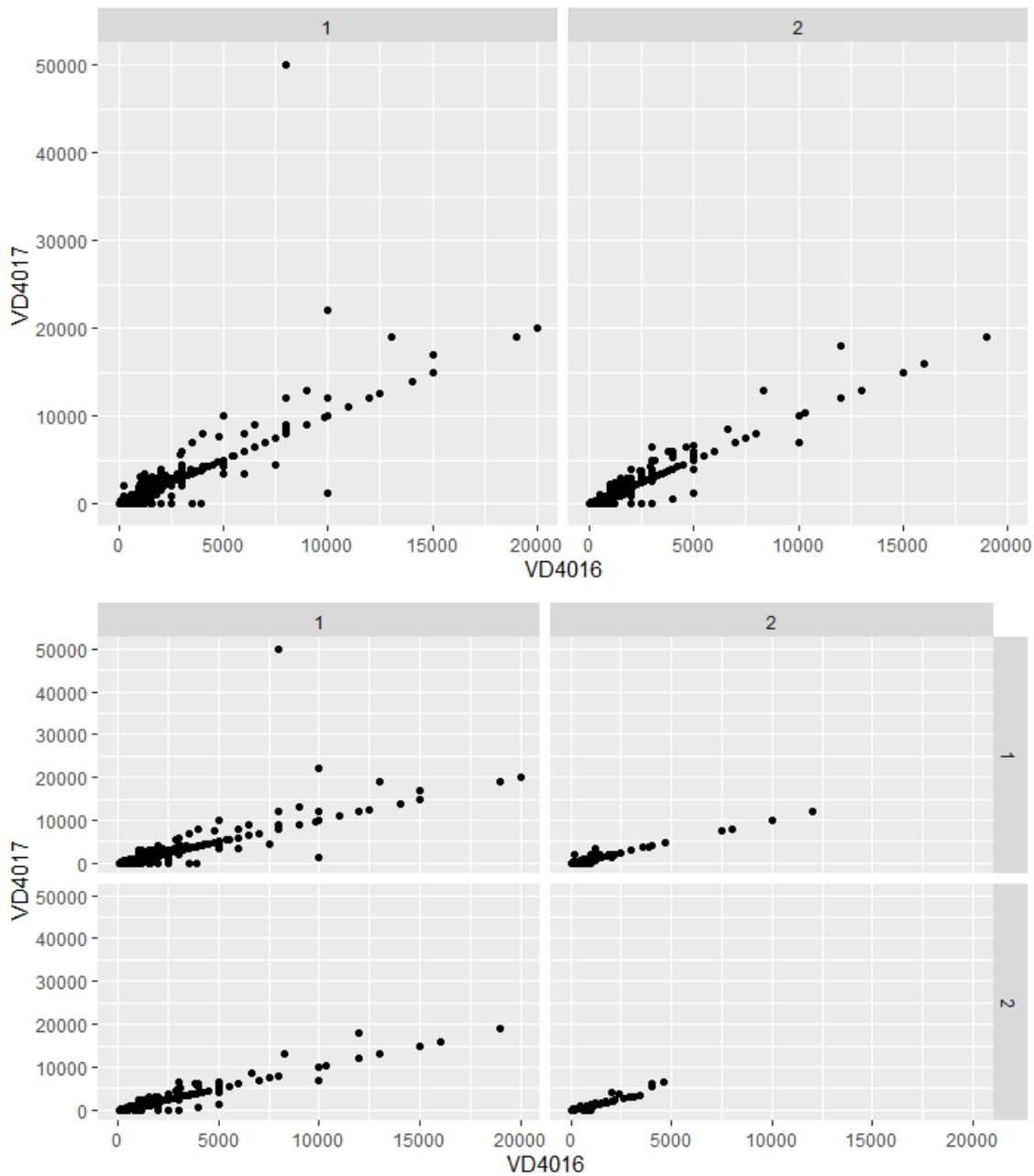
Exemplo: Cria gráficos de dispersão para as variáveis de rendimento mensal no trabalho principal efetivo (VD4017) e habitual (D4016) por subgrupos definidos pelas variáveis discretas sexo (V2007) e situação do domicílio (V1022):

```
pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD4016,
             y = VD4017,
             weight = V1032)) +
  geom_point() +
  facet_grid(V2007~V1022)
```

Os resultados dos exemplos anteriores podem ser visualizados na Figura 10.9.

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

Figura 10.9: Gráficos de dispersão para as variáveis VD4016 e VD4017 (divisão por subgrupos baseados em categorias de variáveis discretas).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.5 Apresentação de distintos tipos de gráficos na mesma imagem

O pacote **gridExtra**⁸ permite a apresentação conjunta, em uma mesma imagem/figura, de distintos gráficos, não necessariamente correlacionados, por meio da função `grid.arrange()`, a qual possibilita a organização dos gráficos em forma de vetores ou matrizes.

Para demonstrar a aplicação dessa função, serão produzidos quatro gráficos:

1. `geom_point()`: gráfico de pontos armazenado em um objeto denominado `a1`;
2. `geom_jitter()`: gráfico de pontos com adição de uma pequena quantidade de variação aleatória (ruído), armazenado no objeto `a2`⁹;
3. `geom_boxplot()`: gráfico box-plot armazenado no objeto `a3`;
4. `geom_violin()`: gráfico de violino armazenado no objeto `a4`¹⁰.

Observação: Faz-se necessária a instalação e a habilitação do pacote **gridExtra**. Para tanto, basta aplicar os seguintes comandos:

```
install.packages("gridExtra")
library(gridExtra)
```

Exemplo: Cria distintos gráficos de dispersão para o rendimento efetivo no trabalho principal (`VD4017`) por sexo (`V2007`), apresentando-os na forma de vetores ou matrizes de gráficos com o auxílio do pacote **gridExtra**:

```
a1 <- pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4017 < 10000) %>%
  ggplot(aes(x = V2007,
```

⁸ Esse pacote foi criado por Auguie (2017).

⁹ Esse tipo de `geom` permite uma melhor visualização dos pontos sobrepostos.

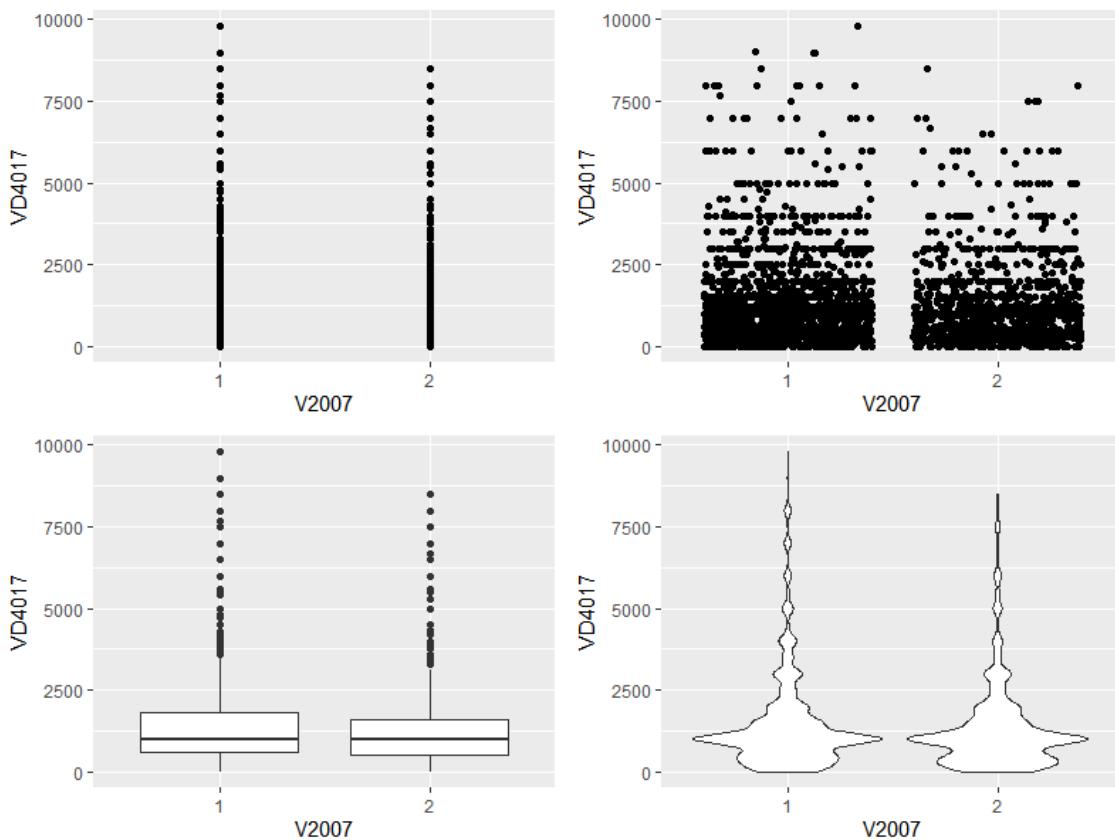
¹⁰ Para melhorar a apresentação, optou-se por filtrar os dados para considerar apenas os casos com rendimentos efetivos do trabalho (`VD4017`) inferiores a R\$10.000,00.

```
y = VD4017,  
    weight = V1032)) +  
geom_point()  
  
a2 <- pnadc2019_visita1_df %>%  
  filter(UF == 24 & VD4017 < 10000) %>%  
  ggplot(aes(x = V2007,  
             y = VD4017,  
             weight = V1032)) +  
  geom_jitter()  
  
a3 <- pnadc2019_visita1_df %>%  
  filter(UF == 24 & VD4017 < 10000) %>%  
  ggplot(aes(x = V2007,  
             y = VD4017,  
             weight = V1032)) +  
  geom_boxplot()  
  
a4 <- pnadc2019_visita1_df %>%  
  filter(UF == 24 & VD4017 < 10000) %>%  
  ggplot(aes(x = V2007,  
             y = VD4017,  
             weight = V1032)) +  
  geom_violin()  
  
# Para exibir em um vetor linha  
grid.arrange(a1 , a2 , a3, a4, ncol=4)  
  
# Para exibir em um vetor coluna  
grid.arrange(a1 , a2 , a3, a4, nrow=4)  
  
# Para exibir em uma matriz 2x2  
grid.arrange(a1, a2, a3, a4, ncol=2, nrow=2)
```

POR DENTRO DA PNAD CONTÍNUA

A apresentação conjunta de diversos gráficos na forma matricial (2x2) pode ser vista na Figura 10.10. Uma observação, ainda que superficial, pode indicar que, para essas variáveis, o gráfico mais adequado é o do tipo *boxplot*.

Figura 10.10: Gráficos apresentados na forma de uma matriz 2x2 para as variáveis sexo (`V2007`) e rendimento efetivo no trabalho principal (`VD4017`), apenas para valores inferiores a R\$ 10.000,00.



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

É possível, ainda, comparar distribuições de diferentes subgrupos em um mesmo gráfico a partir de uma variável categórica qualquer, a ser declarada na função `aes`. Para isso, basta informar a variável que definirá os grupos, no argumento `fill`. Isso provocará uma alteração no preenchimento dos elementos gráficos de um `geom`, como o histograma, por exemplo, ou no argumento `colour`, que alterará as cores dos subgrupos definidos como, por exemplo, em um gráfico do tipo polígono de frequência.

Exemplo: Cria histogramas e polígonos de frequência para o rendimento efetivo no trabalho principal (VD4017), mapeando cores e preenchimentos por meio da variável categórica sexo (V2007):

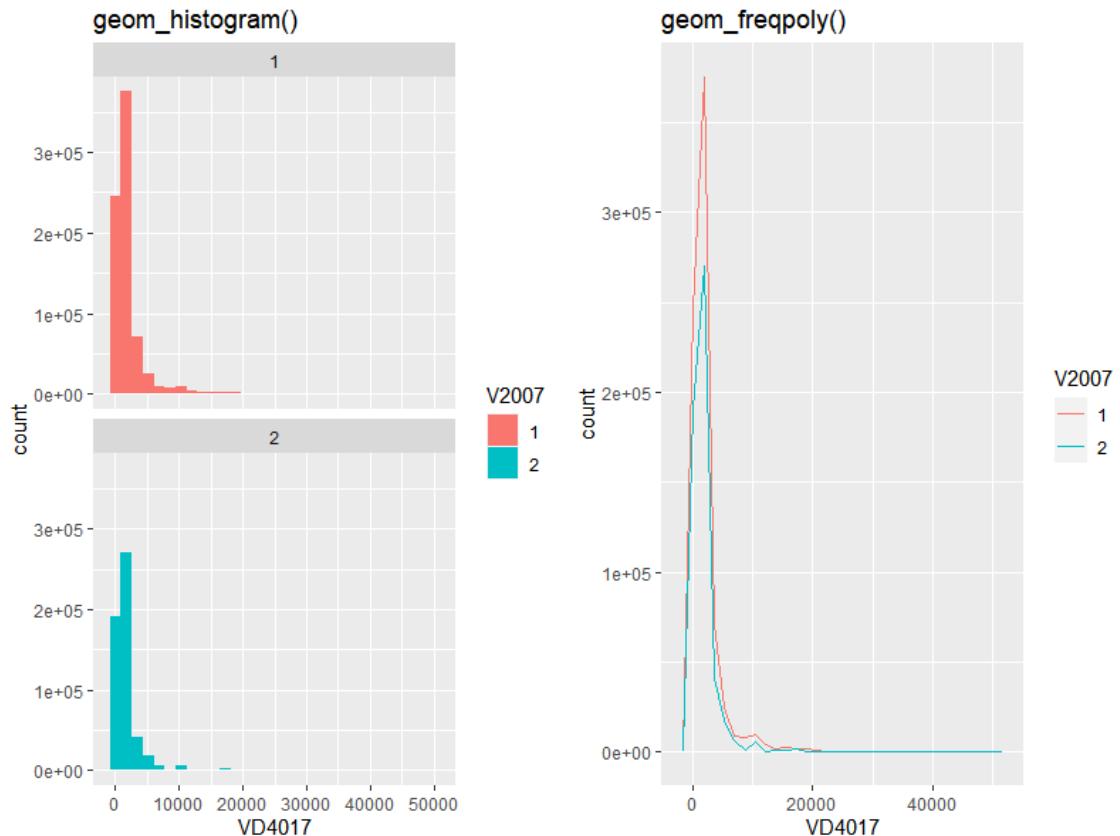
```
pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD4017, fill = V2007, weight = V1032)) +
  geom_histogram() +
  facet_wrap(~V2007, ncol = 1)

pnadc2019_visita1_df%>%
  filter(UF == 24) %>%
  ggplot(aes(VD4017, colour = V2007, weight = V1032)) +
  geom_freqpoly()
```

Os resultados desses mapeamentos estéticos com base na cor e no preenchimento podem ser vistos na Figura 10.11.

POR DENTRO DA PNAD CONTÍNUA

Figura 10.11: Histograma e polígono de frequência para a variável VD4017 com ma-peamento estético de cor e preenchimento por sexo (V2007).



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.6 Séries temporais

Ao explorar dados em séries temporais, é comum o uso de gráficos de linha (`geom_line()`), já que eles unem os pontos da esquerda para a direita, seguindo os valores do eixo x (tempo). Esse tipo de gráfico é utilizado quando se deseja apresentar alterações ocorridas nas variáveis de interesse ao longo do tempo.

O exemplo a seguir está baseado nos dados referentes à Taxa de Desocupação (TD) e à Taxa Composta de Subutilização da Força de Trabalho (TCSFT), na semana de referência, das pessoas de 14 anos ou mais de idade (%) para o período que se estende do primeiro trimestre de 2019 ao primeiro trimestre de 2021 para o total do Brasil. Para tanto, é necessário que se construa esse `data.frame` a partir do seguinte comando:

```
dados_sub <- data.frame(ano_trim =  
  as.character.Date(  
    c(2019.01, 2019.02, 2019.03, 2019.04,  
     2020.01, 2020.02, 2020.03, 2020.04, 2021.01)),  
  desocup = c(12.7, 12, 11.8, 11, 12.2, 13.3, 14.6, 13.9, 14.7),  
  subutiliza = c(25.0, 24.8, 24, 23, 24.4, 29.1, 30.3, 28.7, 29.7))
```

O resultado da criação desse `data.frame` é o seguinte:

```
# > dados_sub  
#   ano_trim   desocup   subutiliza  
# 1 2019.01     12.7      25.0  
# 2 2019.02     12.0      24.8  
# 3 2019.03     11.8      24.0  
# 4 2019.04     11.0      23.0  
# 5 2020.01     12.2      24.4  
# 6 2020.02     13.3      29.1  
# 7 2020.03     14.6      30.3  
# 8 2020.04     13.9      28.7  
# 9 2021.01     14.7      29.7
```

Exemplo: Cria um gráfico de linha para a Taxa de Desocupação:

```
dados_sub %>%  
  ggplot(aes(x = ano_trim,  
             y = desocup,  
             group = 1)) +  
  geom_line(colour="red")
```

Exemplo: Cria um gráfico de linha para Taxa Composta de Subutilização da Força de Trabalho:

POR DENTRO DA PNAD CONTÍNUA

```
dados_sub %>%
  ggplot(aes(x = ano_trim,
              y = subutiliza,
              group = 1)) +
  geom_line(colour="blue")
```

Exemplo: Cria um gráfico de linhas para a TD e para a TCSFT, no mesmo plano:

```
dados_sub %>%
  ggplot() +
  geom_line(aes(x = ano_trim,
                y = subutiliza,
                group = 1),
            colour="red")+
  geom_line(aes(x = ano_trim,
                y = desocup,
                group = 1),
            colour="blue")
```

Os três gráficos criados anteriormente podem ser vistos na Figura 10.12.

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

Figura 10.12: Gráficos de linhas para as séries temporais da taxa de desocupação e da taxa composta de subutilização da força de trabalho.



Fonte: Elaboração própria a partir da PNAD Contínua trimestral (2019-2021)

10.7.7 Apresentação gráfica de frequências relativas (participações)

Boa parte dos estudos baseados em estatísticas descritivas utiliza proporções em relação ao total, para quantificar o peso relativo dos componentes de distintas variáveis categóricas no total da população. Os dados podem ser apresentados tanto em forma de tabelas, quanto em forma gráfica. As mais comuns são os gráficos de pizza e os de barras empilhadas. No entanto, existe a possibilidade de apresentar graficamente proporções na forma de árvore por meio do pacote **treemapify**, desenvolvido por Wilkins (2021), uma extensão do **ggplot2**.

Para instalar o pacote, basta aplicar os seguintes comandos:

```
install.packages("treemapify")
library(treemapify)
```

Os exemplos, a seguir, estão baseados em dois conjuntos de dados:

1. **dados1**: frequência relativa (%) para a variável Condição de ocupação (**VD4002**);
2. **dados2**: frequência relativa para a variável Posição na ocupação no trabalho principal com subcategorias de empregados (**VD4008**).

Para preparar esses dois conjuntos de dados, pode-se utilizar os comandos do **tidyverse** apresentados no Capítulo 7, como naqueles apresentados a seguir:

```
dados1 <- pnadc2019_visita1_df %>%
  filter(UF == 24 & VD4001 == 1) %>%
  count(VD4002, wt = V1032) %>%
  mutate(freq = n/sum(n),
        VD4002 = c("Ocupados", "Desocupados")) %>%
  select("VD4002","freq")
```

```
# > dados1
# A tibble: 2 x 2
#   VD4002      freq
#   <chr>      <dbl>
# 1 Ocupados    0.871
# 2 Desocupados 0.129
```

```
dados2 <- pnadc2019_visita1_df %>%
  filter(UF == 24 & !is.na(VD4008)) %>%
  count(VD4008, wt = V1032) %>%
  mutate(freq = n/sum(n),
        VD4008 = c("Empregado no setor privado",
                   "Trabalhador doméstico",
                   "Empregado no setor público",
                   "Empregador",
                   "Conta-própria",
                   "Trabalhador familiar auxiliar")) %>%
  select("VD4008","freq")
```

```
# > dados2
# A tibble: 6 x 2
#   VD4008              freq
#   <chr>            <dbl>
# 1 Empregado no setor privado 0.422
# 2 Trabalhador doméstico     0.0677
# 3 Empregado no setor público 0.162
# 4 Empregador             0.0409
# 5 Conta-própria          0.286
# 6 Trabalhador familiar auxiliar 0.0222
```

POR DENTRO DA PNAD CONTÍNUA

Exemplos: Cria gráficos de pizza para os objetos dados¹ e dados²:

```
dados1 %>%
  ggplot(aes(x = "", y = freq, fill = VD4002)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  theme_void()

dados2 %>%
  ggplot(aes(x = "", y = freq, fill = VD4008)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  theme_void()
```

Exemplo: Cria gráficos de barras empilhadas para os objetos dados¹ e dados²:

```
dados1 %>%
  ggplot(aes(x = "", y = freq, fill = VD4002)) +
  geom_bar(stat = "identity", width = 1) +
  theme_void()

dados2 %>%
  ggplot(aes(x = "", y = freq, fill = VD4008)) +
  geom_bar(stat = "identity", width = 1) +
  theme_void()
```

Exemplo: Cria gráficos de árvore para os objetos dados¹ e dados²:

```
dados1 %>%
  ggplot(aes(area = freq, fill = VD4002)) +
  geom_treemap()

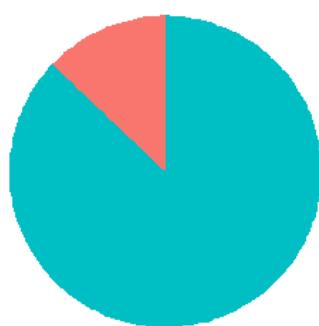
dados2 %>%
  ggplot(aes(area = freq, fill = VD4008)) +
  geom_treemap()
```

Os resultados desses seis gráficos podem ser vistos na Figura 10.13.

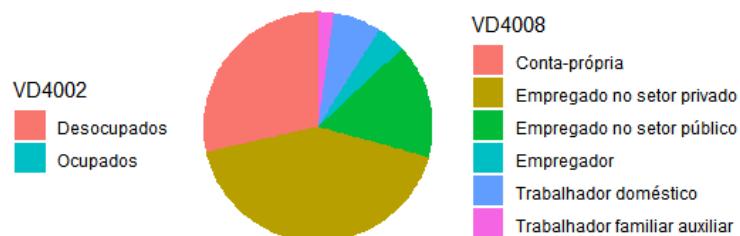
POR DENTRO DA PNAD CONTÍNUA

Figura 10.13: Gráficos de pizza, barras empilhadas e árvore para análise de frequências relativas ou participação (em %).

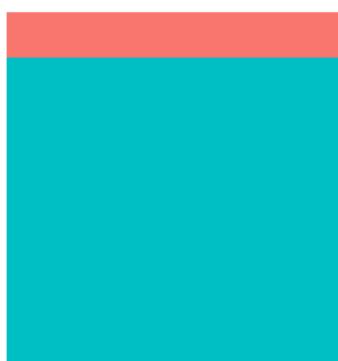
Pizza (VD4002)



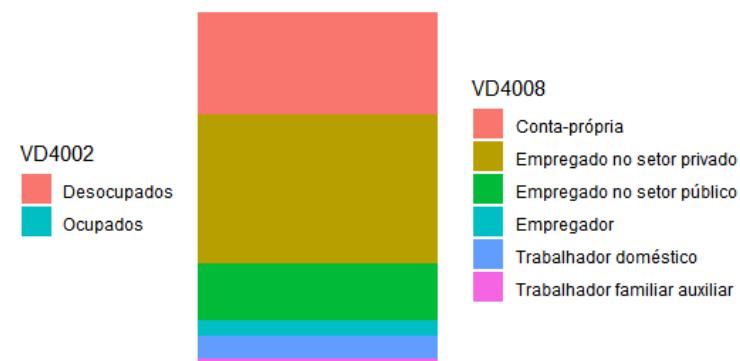
Pizza (VD4008)



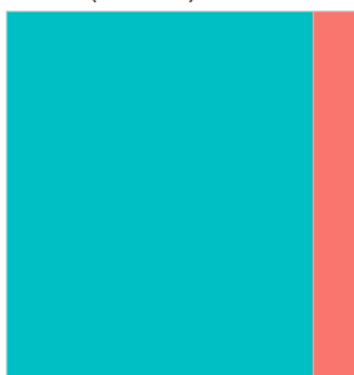
Barras empilhadas (VD4002)



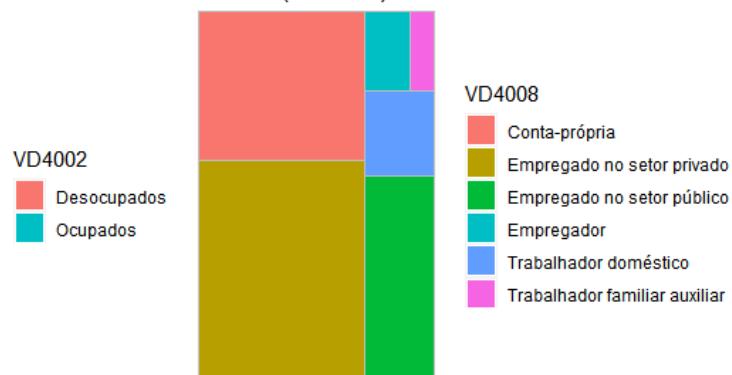
Barras empilhadas (VD4008)



Árvore (VD4002)



Árvore (VD4008)



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.8 Transformações e ajustes para melhorar a apresentação dos gráficos

As transformações e os ajustes realizados na presente seção procuram melhorar a apresentação dos gráficos. Para isso, será criado um gráfico que servirá de base e que receberá as modificações passo a passo. Após a sua criação, será armazenado em um objeto chamado `graf_base`. Ele terá a forma de um histograma e conterá a distribuição dos rendimentos efetivos domiciliares *per capita* (VD5005) para o Rio Grande do Norte.

O objeto `graf_base` pode ser obtido da seguinte forma:

```
graf_base <-
  pnadc2019_visitai1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(VD5005,
             weight = V1032)) +
  geom_histogram()
```

Uma das mais básicas transformações em um gráfico é a definição de rótulos para as variáveis apresentadas nos eixos. As funções `xlab()` e `ylab()` modificam os rótulos dos eixos x e y, respectivamente¹¹. Para fazer essas transformações, basta aplicar os seguintes comandos:

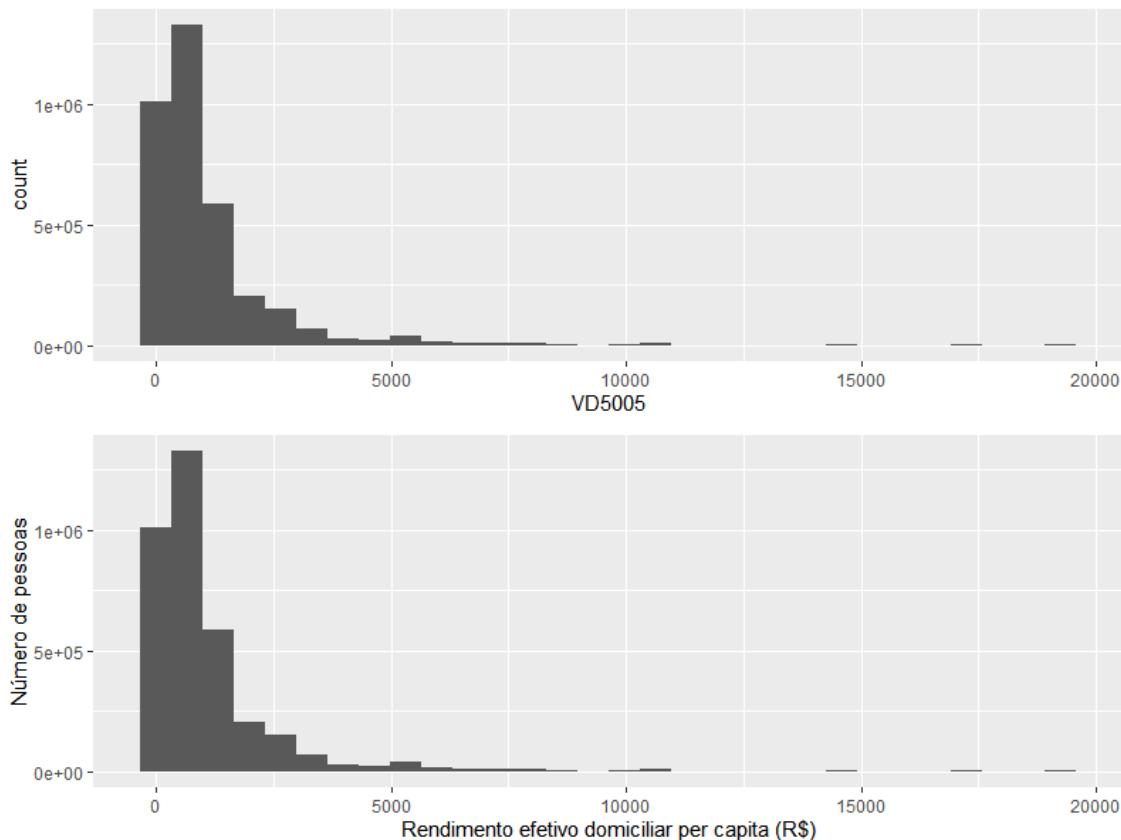
```
graf_base <-
  graf_base + xlab("Rendimento efetivo domiciliar per capita (R$)")
  + ylab("Número de pessoas")
```

Tanto o gráfico-base quanto a adição de rótulos aos eixos x e y podem ser visualizadas na Figura 10.14.

¹¹ Para excluir os eixos, basta passar como argumento nessas funções o parâmetro `NULL` da seguinte forma: `xlab(NULL)` e `ylab(NULL)`.

POR DENTRO DA PNAD CONTÍNUA

Figura 10.14: Histogramas básicos para a variável VD5005 e com alteração nos rótulos dos eixos



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Com o auxílio do pacote **scales**¹², é possível alterar a formatação das escalas dos eixos. Essas transformações podem se dar nos nomes, nas quebras, nos rótulos, nos limites, na posição etc. A título de exemplo, pode-se limitar o tamanho dos eixos, usando as funções `xlim()` e `ylim()`, ou usar o argumento `limits` nas funções `scale_y_continuous` e `scale_x_continuous`. Ambas as formas comportam valores que limitam tanto variáveis contínuas quanto discretas.

No exemplo a seguir, apresentam-se duas formas de se introduzir um limite de R\$ 2.000,00 para a variável **VD5005** no eixo x. Para limitar os eixos, basta aplicar os seguintes comandos¹³:

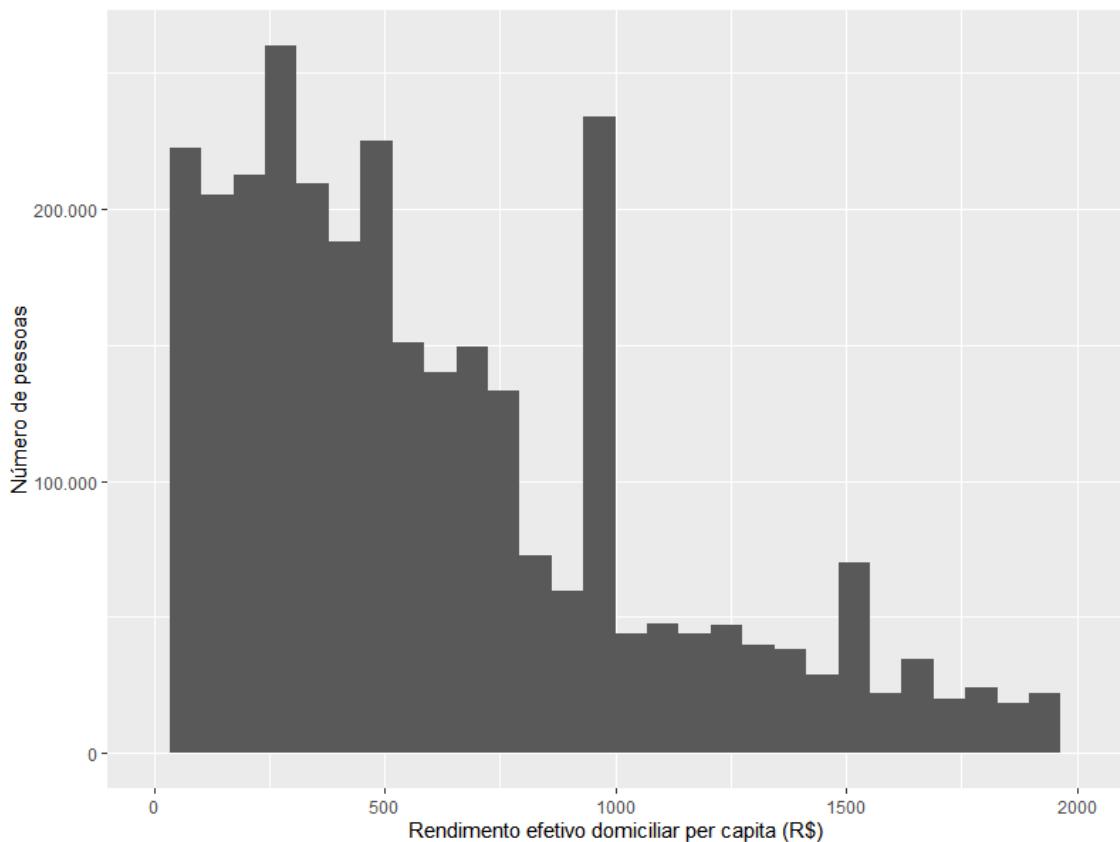
¹² Esse pacote foi criado por Wickham e Seidel (2020).

¹³ Deve-se reparar que o objeto `graf_base` não foi alterado permanentemente.

```
graf_base +  
  xlim(0,2000)  
  
#Alternativa  
library(scales)  
  
graf_base +  
  scale_x_continuous(limits = c(0,2000))
```

Os resultados podem ser visto na Figura 10.15.

Figura 10.15: Histograma básico para a variável VD5005 com o eixo x limitado a R\$ 2.000,00



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

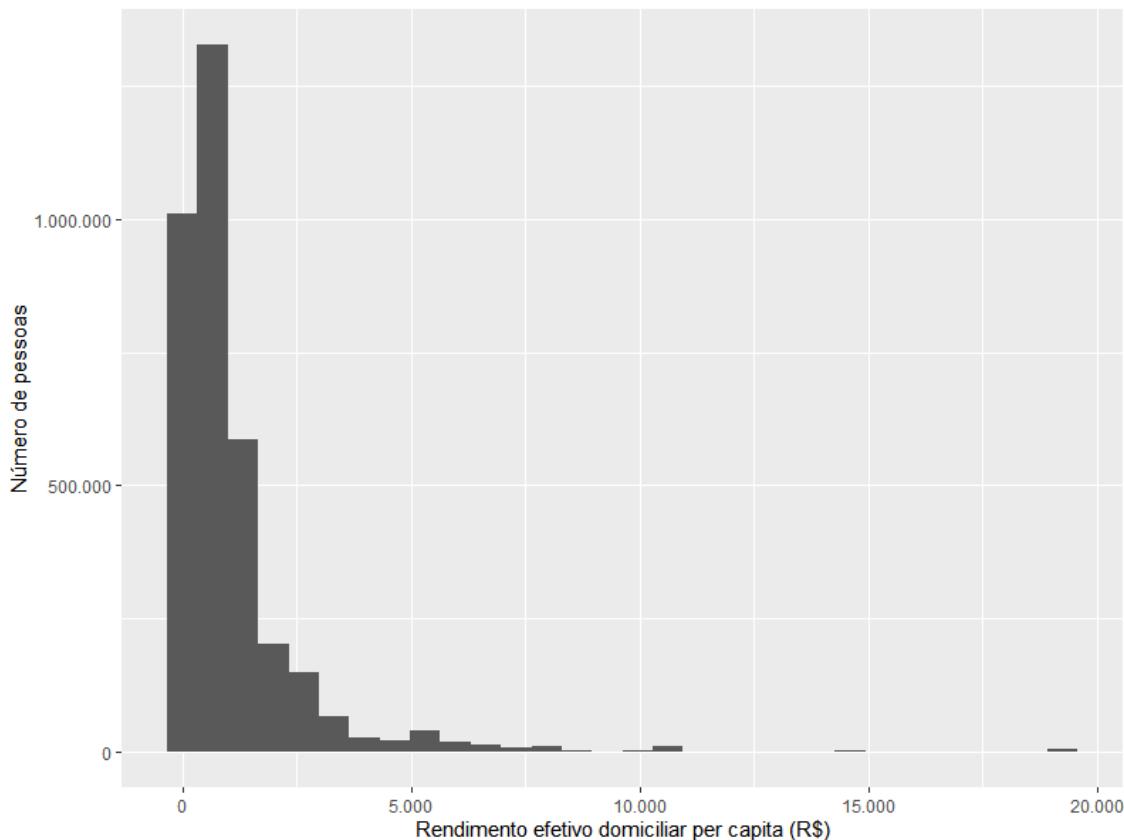
POR DENTRO DA PNAD CONTÍNUA

É possível, ainda, alterar a formatação dos valores dos rótulos dos eixos. Para tanto, basta aplicar os comandos a seguir:

```
graf_base <-
  graf_base +
  scale_y_continuous(labels =
    scales::number_format(
      accuracy = NULL,
      scale = 1,
      prefix = "",
      suffix = "",
      big.mark = ".",
      decimal.mark = ",",
      trim = TRUE)) +
  scale_x_continuous(labels =
    scales::number_format(
      accuracy = 0.01,
      scale = 1,
      prefix = "",
      suffix = "",
      big.mark = ".",
      decimal.mark = ",",
      trim = TRUE))
```

Esses resultados são apresentados na Figura 10.16.

Figura 10.16: Histograma básico para a variável VD5005 com alteração nas escalas dos eixos



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

A alteração dos rótulos dos valores das variáveis definidas nos eixos, também, pode ser aplicada a variáveis discretas/categóricas, definindo-se os nomes das categorias que aparecem no eixo x, por meio da função `scale_x_discrete`.

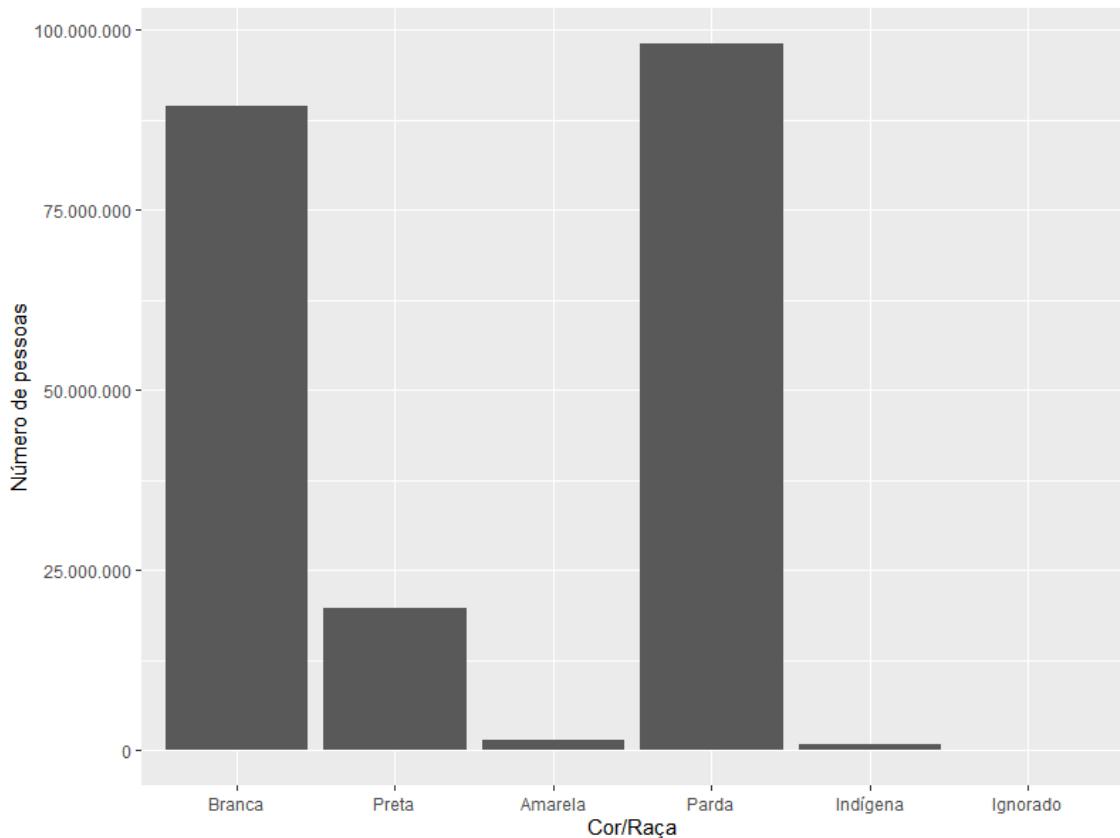
POR DENTRO DA PNAD CONTÍNUA

Exemplo: Cria um gráfico de barras com a contagem da população para a variável cor/raça (v2010) como distintas modificações estéticas:

```
pnadc2019_visita1_df %>%
  ggplot(aes(x = V2010, weight = V1032)) +
  geom_bar() +
  scale_y_continuous(labels =
    scales::number_format(
      accuracy = NULL,
      scale = 1,
      prefix = "",
      suffix = "",
      big.mark = ".",
      decimal.mark = ",",
      trim = TRUE)) +
  xlab("Cor/Raça") +
  ylab("Número de pessoas") +
  scale_x_discrete(labels =
    c("Branca",
      "Preta",
      "Amarela",
      "Parda",
      "Indígena",
      "Ignorado"))
```

O resultado desse conjunto de transformações é apresentado na Figura 10.17:

Figura 10.17: Gráfico de barras com a contagem da população segundo cor/raça (v2010)



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

10.7.9 Outras modificações estéticas

Por fim, os gráficos criados com o **ggplot2** podem receber um conjunto expressivo de modificações que contribuem para melhorar sua aparência.

Procura-se, desse modo, apresentar algumas dessas possibilidades, a partir de um gráfico de densidade de Kernel (função de densidade de probabilidade) para o rendimento efetivo domiciliar *per capita* (vD5005) segundo a variável categórica cor/raça (v2010).

POR DENTRO DA PNAD CONTÍNUA

Para isso, basta seguir os seguintes passos:

1. Criar um gráfico-base e armazená-lo em um objeto denominado `graf_1`:

```
graf_1 <- pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD5005, weight = V1032,
             colour = V2010)) +
  geom_density(aes())
```

2. Adicionar cores de preenchimento para as curvas condicionadas à variável `V2010` e armazenar o resultado no objeto `graf_2`:

```
graf_2 <- pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD5005, weight = V1032,
             colour = V2010, fill = V2010)) +
  geom_density()
```

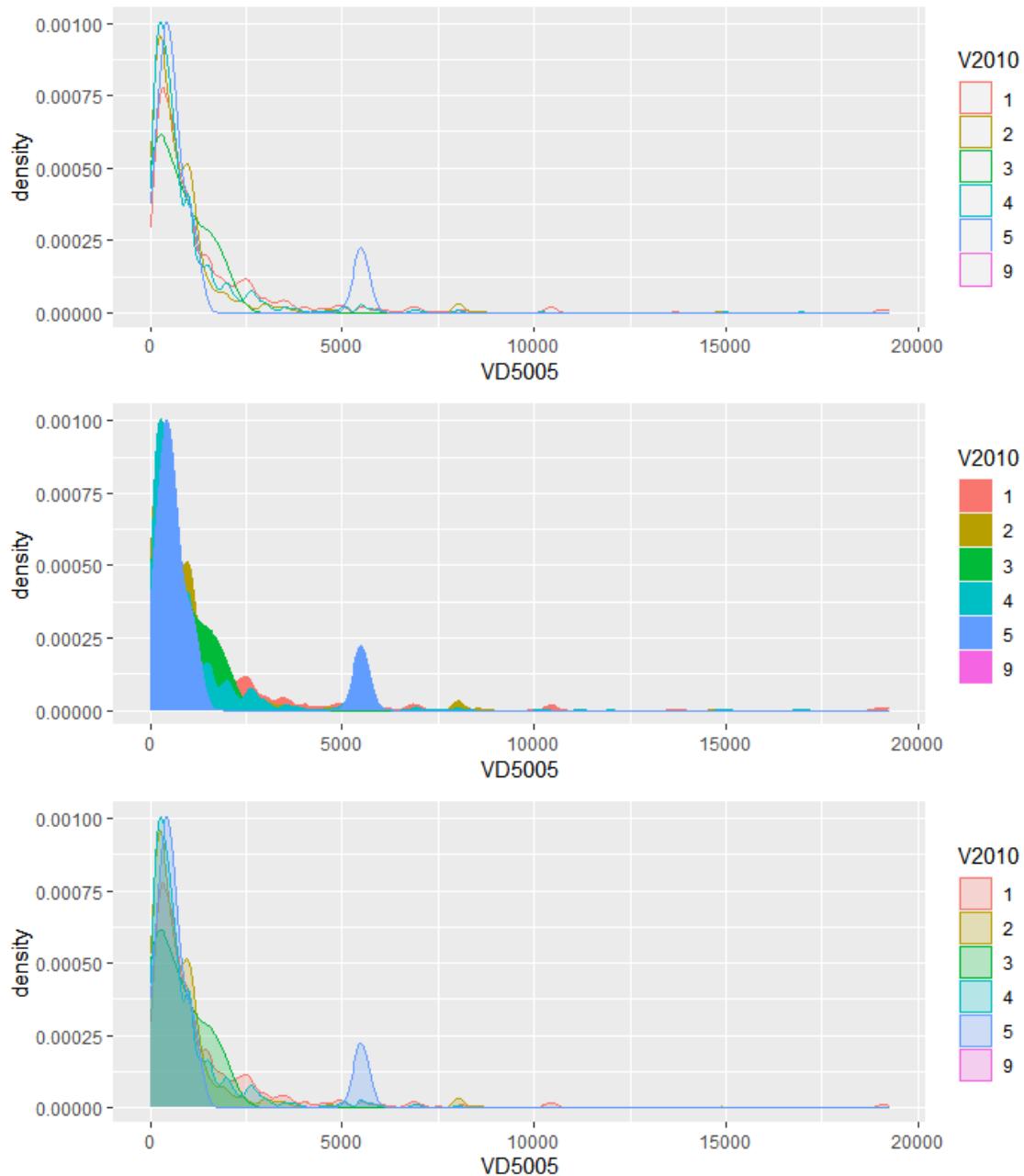
3. Adicionar transparência (25%) ao preenchimento para possibilitar a visualização das curvas em sobreposição e armazenar o resultado no objeto `graf_3`¹⁴:

```
graf_3 <- pnadc2019_visita1_df %>%
  filter(UF == 24) %>%
  ggplot(aes(x = VD5005, weight = V1032,
             colour = V2010, fill=V2010)) +
  geom_density(alpha = 0.25)
```

¹⁴ Deve-se reparar que isso pode ser feito por meio do argumento `alpha`, que deve ser informado na função `geom_density()`.

O resultado de todas essas transformações pode ser visto na Figura 10.18.

Figura 10.18: Gráficos de densidade de Kernel para as variáveis VD5005 e V2010 com modificações estéticas



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

POR DENTRO DA PNAD CONTÍNUA

Cabe destacar ainda que, além da função `grid.arrange()` do pacote **gridExtra**, existe uma forma diferente e simplificada para apresentar distintos gráficos na mesma figura. O pacote **patchwork**, criado por Pedersen (2020), permite arranjar diversos gráficos de distintas maneiras, apresentando-os de forma conjunta, por meio de operadores matemáticos.

Para instalar e habilitar esse pacote, devem-se utilizar os seguintes comandos:

```
install.packages("patchwork")
library(patchwork)
```

A lógica para definir a apresentação dos gráficos com o **patchwork** é a seguinte:

1. Os gráficos podem ser armazenados em objetos específicos, por exemplo, numerados (`g1`, `g2`, `g3` ...);
2. A disposição desses objetos deve ser definida por meio de operadores matemáticos de adição, subtração e divisão.

Segue um exemplo simples e hipotético para organização de gráficos usando o pacote **patchwork**:

```
g1 + g2
g1 + g2 + g3
g1 + g2 + g3 + g4
g1 + g2 + g3 + plot_layout(ncol = 2)
g1 + g2 / g3
g1 / g2 / g3
g1 | (g2 / (g3 | g4))
```

No caso da Figura 10.18, que traz três gráficos armazenados nos objetos `graf_1`, `graf_2` e `graf_3`, o arranjo feito por meio do **patchwork** assume a seguinte forma:

```
graf_1 / graf_2 / graf_3
```

Antes de apresentar um gráfico acabado, com diversas modificações estéticas que contribuem para aprimorar a forma de apresentação, deve-se conhecer a camada denominada `geom_text()`.

- `geom_text()`: permite rotular plotagens. Pode ser usado de forma isolada, substituindo pontos por nomes, como em diagramas de dispersão, ou combinada a outros geoms, como, por exemplo, para marcar pontos específicos ou para anotar a altura de barras ou linhas. A função `geom_text()` adiciona apenas um texto ao gráfico. Já a função `geom_label()` desenha uma caixa atrás e ao redor do texto, tornando-o mais fácil de ler, muito útil em algumas situações.

Exemplo: Rotula elementos gráficos para os objetos `dados_sub` e `dados2`, definidos nas seções anteriores:

```
dados_sub %>%
  ggplot(aes(x = ano_trim,
             y = desocup,
             group = 1,
             label = sprintf("%0.1f",
                            round(desocup, digits = 1)))) +
  geom_line(colour="red") +
  geom_text(nudge_y = 0.25)

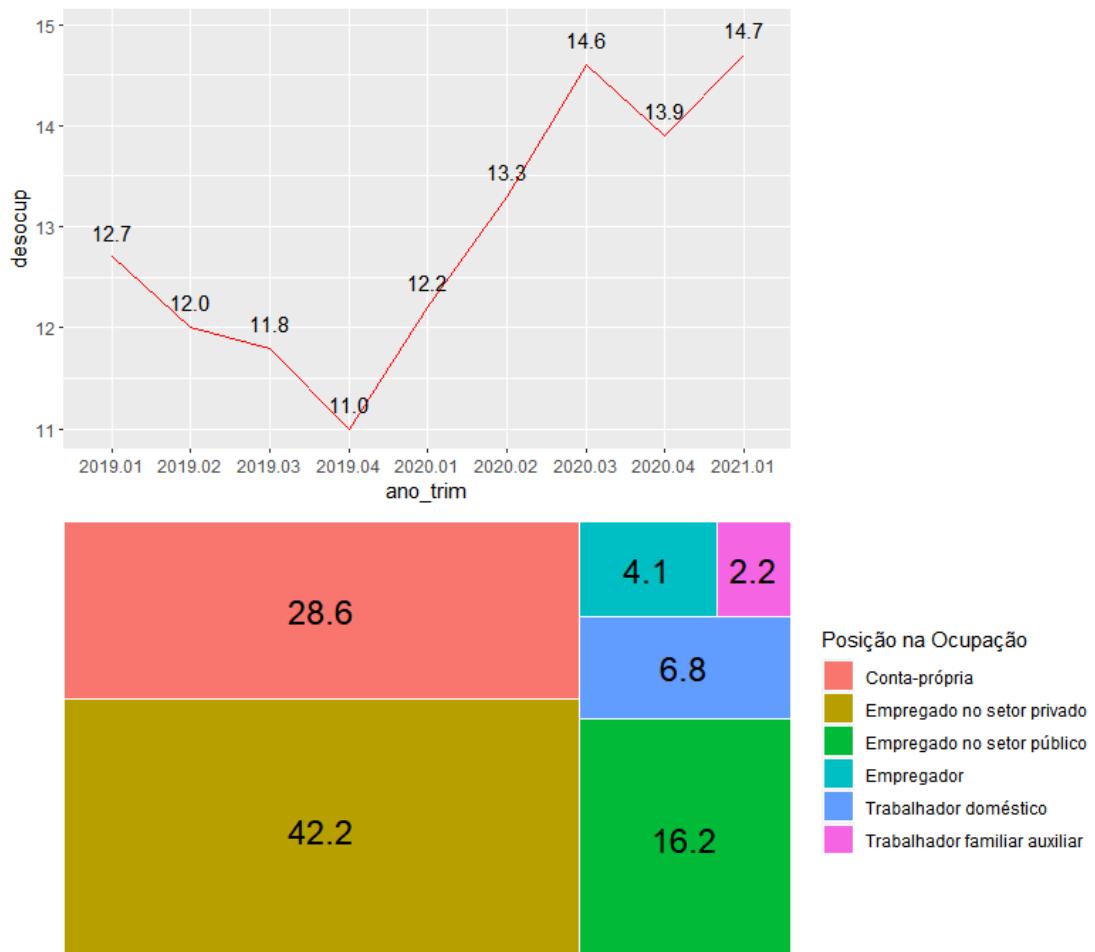
dados2 %>%
  mutate(freq = freq*100) %>%
  ggplot(aes(area = freq,
             fill = VD4008,
             label = sprintf("%0.1f", round(freq, digits = 1)))) +
  geom_treemap(colour = "white") +
```

POR DENTRO DA PNAD CONTÍNUA

```
geom_treemap_text(aes(fontface = 1),  
                   colour = "black",  
                   place = "centre") +  
guides(fill=guide_legend(title="Posição na Ocupação"))
```

Os resultados dessas transformações que rotulam alguns dos elementos gráficos, construídos em exemplos anteriores, podem ser vistos na Figura 10.19.

Figura 10.19: Gráficos com elementos gráficos de linha e área rotulados



Fonte: Elaboração própria a partir da PNAD Contínua trimestral de 2019 a 2021 (gráfico superior) e anual de 2019 – visita 1 (gráfico inferior).

A elaboração de um gráfico finalizado para compor, por exemplo, um trabalho acadêmico exige um conjunto amplo de transformações possíveis, dentre as quais se destacam:

1. Modificação dos rótulos dos eixos (`scale_x_continuous()` e `scale_y_continuous()`);
2. Definição da paleta de cores e dos rótulos das legendas (`scale_colour_brewer()`);
3. Limitação do eixo x a determinados valores (por exemplo, inferiores a R\$ 5.000,00);
4. Definição da posição e da agregação dos dados por cor e preenchimento (`colour` e/ou `fill`), segundo subgrupos que devem aparecer na legenda (`theme()` e `guides()`);
5. Adição de título, subtítulo, fonte etc. (`labs()`).

Exemplo final: Cria e apresenta um gráfico com distintas modificações estéticas:

A fim de expor todas essas alterações, será utilizado o objeto `graf_3`, definido anteriormente. Para realizá-las, basta aplicar os seguintes comandos, atentando-se para cada uma das camadas inseridas e quais suas funcionalidades:

POR DENTRO DA PNAD CONTÍNUA

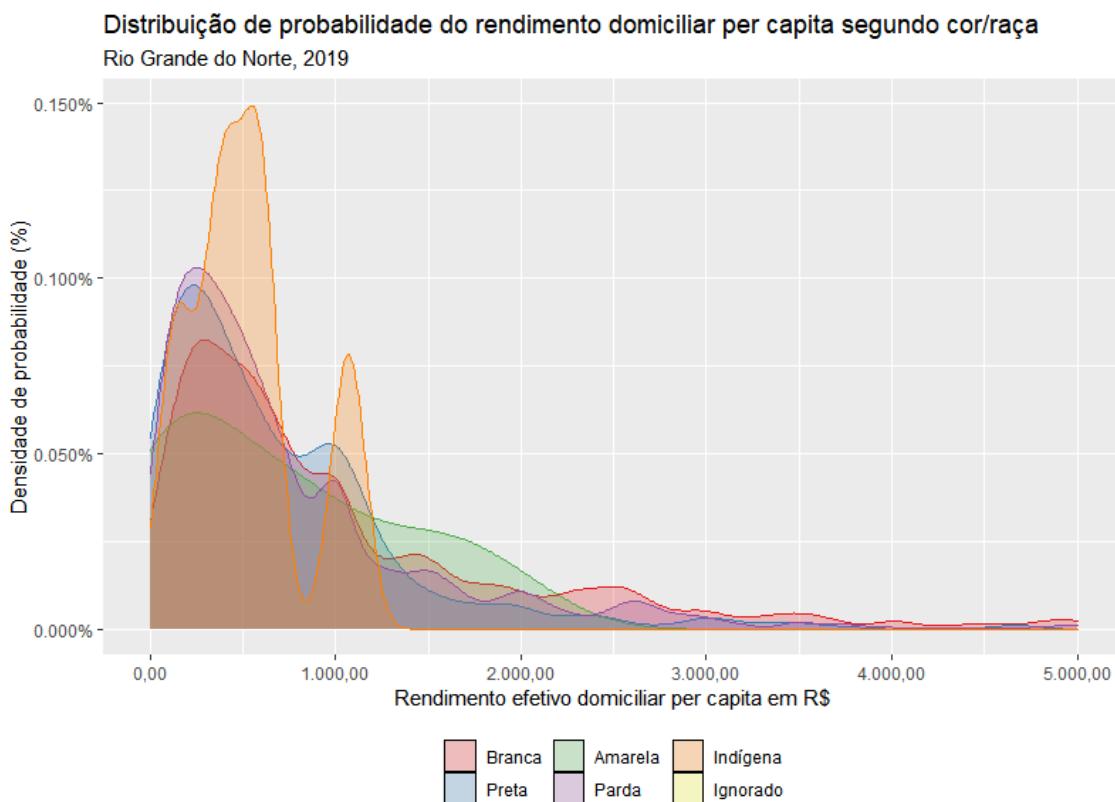
```
graf_3 +
  scale_x_continuous(labels =
    scales::number_format(
      accuracy = 0.01,
      scale = 1,
      prefix = "",
      suffix = "",
      big.mark = ".",
      decimal.mark = ",",
      trim = TRUE),
    limits = c(0,5000)) +
  scale_y_continuous(labels = scales::percent) +
  scale_colour_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1",
    labels = c("Branca", "Preta",
              "Amarela", "Parda",
              "Indígena", "Ignorado"))+
  theme(legend.position = "bottom",
        plot.caption = element_text(hjust=0)) +
  guides(colour = "none")+
  labs(
    x = "Rendimento efetivo domiciliar per capita em R$",
    y = "Densidade de probabilidade (%)",
    fill = NULL,
    title = "Distribuição de probabilidade do rendimento
              domiciliar per capita segundo cor/raça",
    subtitle = "Rio Grande do Norte, 2019",
    caption = "Fonte: IBGE, PNAD Contínua visita 1, 2019.")
```

Para verificar as paletas de cores disponíveis, pode-se utilizar o seguinte comando:

```
RColorBrewer::display.brewer.all()
```

Por fim, apresenta-se o gráfico finalizado (ver Figura 10.20), com todas as transformações realizadas anteriormente. Com base nesse gráfico, pode-se compará-lo com os três gráficos-base apresentados anteriormente, no arranjo definido com a ajuda do pacote **patchwork**.

Figura 10.20: Gráfico final de densidade de Kernel para as variáveis VD5005 e V2010



Fonte: IBGE, PNAD Contínua visita 1, 2019.

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

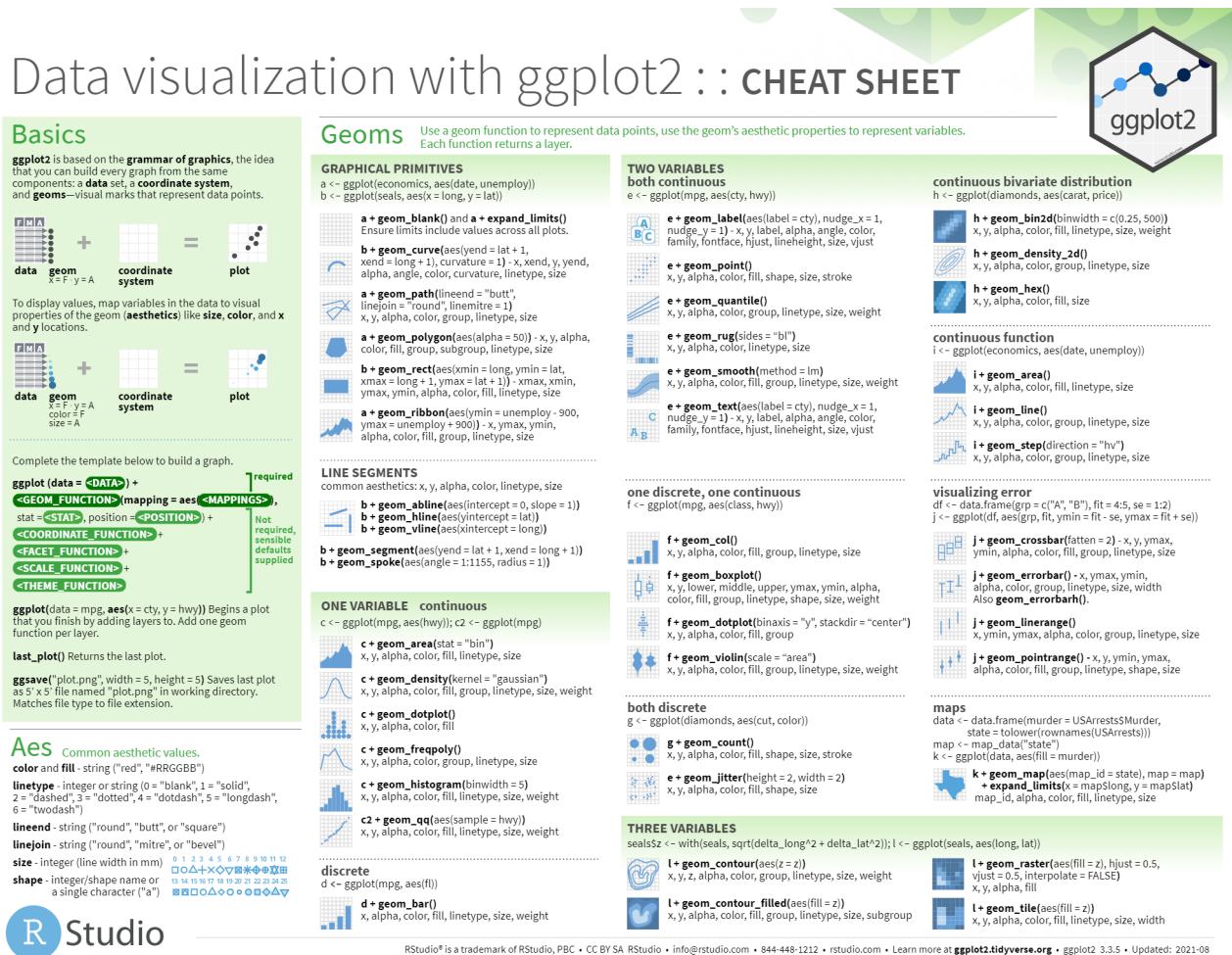
POR DENTRO DA PNAD CONTÍNUA

Assim como foi elaborado por seus desenvolvedores para o pacote **dplyr**, o **ggplot2** também conta com um material-síntese, disponibilizado no site do pacote **tidyverse**¹⁵, na forma de duas folhas de cola que resumem distintas funções e aplicações desse pacote para distintos tipos de dados. Essas folhas podem ser vistas na Figura 10.21 e na Figura 10.22¹⁶.

¹⁵ Ver: <https://ggplot2.tidyverse.org/>.

¹⁶ Para uma melhor visualização do material, recomenda-se acessar: <https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf>.

Figura 10.21: Folha de cola para gramática de gráficos – **ggplot2** (parte 1).



Fonte: RSTUDIO (2021b).

Figura 10.22: Folha de cola para gramática de gráficos – `ggplot2` (parte 2).

Stats An alternative way to build a layer.

A stat builds new variables to plot (e.g., count, prop).

```
rlang::dplyr::ctv::cvt(d, d + geom_bar(aes(fill = f1)) + coordinate_system)
```

Visualize a stat by changing the default stat of a geom function, `geom_bar(stat = "count")`, or by using a stat function, `stat_count(mapping = "count")`, which calls a default geom to make a layer (equivalent to a geom function). Use `.name_to` syntax to map stat variables to aesthetics.

geom to use **stat function** **geom mappings**

```
i + stat_density_2d(geom = "level..", geom = "polygon")
```

variable created by stat

```
c * stat_bin(binwidth = 1, boundary = 10)
x, y ..count.., ..density..
c * stat_count(width = 1) x, y ..count.., ..prop..
c * stat_density(adjust = 1, kernel = "gaussian")
x, y ..count.., ..density.., ..scaled..
e * stat_bin(bins = 30, drop = T)
x, y, fill ..count.., ..density..
e * stat_hex(x, y, fill | ..count.., ..density..
e * stat_summary_2d(contour = TRUE, n = 100)
x, y, color, size | ..level..
e * stat_ellipse(level = 0.95, segments = 51, type = "t")
l * stat_contour(aes(z = z)) x, y, z, order | ..level..
l * stat_summary_hex(aes(z = z), bins = 30, fun = max)
x, y, z, fill | ..value..
l * stat_summary_2d(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..
f * stat_boxplot(coef = 1.5)
x, y ..lower.., ..middle.., ..upper.., ..width.., ..ymin.., ..ymax..
f * stat_ydensity(kernel = "gaussian", scale = "area") x, y
| ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..
e * stat_ecdf(n = 40) x, y | ..x.., ..y..
e * stat_quantile(quantiles = c(0.1, 0.9), formula = y ~ log(x), method = "rq") x, y | ..quantile..
e * stat_smooth(method = "lm", formula = y ~ x, se = T, level = 0.95) x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..
ggplot() + xlim(-5, 5) + stat_function(fun = dnorm, n = 20, geom = "point") x | ..y..
ggplot() + stat_qq(aes(sample = 1:100))
x, y, sample | ..sample.., ..theoretical..
e * stat_sum(x, y, size | ..n.., ..prop..
e * stat_summary(fun.data = "mean_cl_boot")
h * stat_summary_bin(fun = "mean", geom = "bar")
e * stat_identity()
e * stat_unique()
```

Scales Override defaults with `scales` package.

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.

```
r <- d + geom_bar(aes(fill = f1))
```

scale_ **aesthetic to adjust** **prepackaged scale to use** **scale-specific arguments**

range of values to include in mapping **title to use in legend/aesthetics** **labels to use in legend/aesthetics** **breaks to use in legend/aesthetics**

GENERAL PURPOSE SCALES

Use with most aesthetics.

`scale_continuous()` - Map cont. values to visual ones.
`scale_discrete()` - Map discrete values to visual ones.
`scale_binned()` - Map continuous values to discrete bins.
`scale_identity()` - Use data values as visual ones.
`scale_manual(values = c())` - Map discrete values to manually chosen visual ones.
`scale_date(date_labels = "%m/%d")`, `date_breaks = "2 weeks"` - Map data values as dates.
`scale_datetime()` - Transform data values as date times. Same as `scale_date()`. See ?strptime for label formats.

X & Y LOCATION SCALES

Use with x or y aesthetics (x shown here)

`scale_x_log10()` - Plot x on log10 scale.
`scale_x_reverse()` - Reverse the direction of the x axis.
`scale_x_sqrt()` - Plot x on square root scale.

COLOR AND FILL SCALES (DISCRETE)

`n + scale_fill_brewer(palette = "Blues")`

For palette choices:
`RColorBrewer::display.brewer.all()`

`n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")`

COLOR AND FILL SCALES (CONTINUOUS)

`o <- c + geom_dotplot(aes(fill = ...))`

`o + scale_fill_distiller(palette = "Blues")`

`o + scale_fill_gradient(low = "red", high = "yellow")`

`o + scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 25)`

`o + scale_fill_gradientn(colors = topo.colors(6))`

Also: `rainbow()`, `heat.colors()`, `terrain.colors()`, `cm.colors()`, `RColorBrewer::brewer.pal()`

SHAPE AND SIZE SCALES

`p <- e + geom_point(aes(shape = f1, size = cyl))`

`p + scale_shape() + scale_size()`

`p + scale_shape_manual(values = c(3:7))`

`o + scale_radius(range = c(1,6))`

`p + scale_size(area = max_size, 6)`

Coordinate Systems

`r <- d + geom_bar()`

`r + coord_cartesian(xlim = c(0, 5))`

The default cartesian coordinate system.

`r + coord_fixed(ratio = 1/2)`

`xlim, ylim` - Cartesian coordinates with fixed aspect ratio between x and y units.

`ggplot(mpg, aes(y = f1)) + geom_bar()`

Flip cartesian coordinates by switching x and y aesthetic mappings.

`r + coord_polar(theta = "x", direction = 1)`

`theta`, `start`, `direction` - Polar coordinates.

`r + coord_trans(x = "sqrt")`

`x, y, xlim, ylim` - Transformed cartesian coordinates. Set `xtrans` and `ytrans` to the name of a window function.

`r + coord_quickmap()`

`n + coord_map(projection = "ortho", orientation = c(41, -74, 0))`

Map projections from the mapproj package (mercator (default), azequivalares, lagrange, etc.).

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

`t <- ggplot(mpg, aes(cty, hwy)) + geom_point()`

`t + facet_grid(cols = vars(f1))`

Facet into columns based on `f1`.

`t + facet_grid(rows = vars(year))`

Facet into rows based on `year`.

`t + facet_grid(rows = vars(year), cols = vars(f1))`

Facet into both rows and columns.

`t + facet_wrap(vars(f1))`

Wrap facets into a rectangular layout.

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

`s <- ggplot(mpg, aes(f1, fill = driv))`

`s + geom_bar(position = "dodge")`

Arrange elements side by side.

`s + geom_bar(position = "fill")`

Stack elements on top of one another, normalize height.

`e * geom_point(position = " jitter")`

Add random noise to x and y position of each element to avoid overplotting.

`e * geom_label(position = "nudge")`

Nudge labels away from points.

`s + geom_bar(position = "stack")`

Stack elements on top of one another.

Themes

`r + theme_bw()`

White background with grid lines.

`r + theme_classic()`

`r + theme_gray()`

Grey background (default theme).

`r + theme_linedraw()`

Minimal theme.

`r + theme_dark()`

Dark for contrast.

`r + theme_void()`

Empty theme.

`r + theme()`

Customizes aspects of the theme such as axis, legend, panel, and facet properties.

`geom`, `guide`, `Title`, `theme`, `plot.title.position` = "plot"

`r + theme(panel.background = element_rect(fill = "blue"))`

Labels and Legends

`use labs()` to label the elements of your plot.

`t + labs(x = "New x axis label", y = "New y axis label", title = "Add a title above the plot", subtitle = "Add a subtitle below title", caption = "Add a caption below plot", alt = "Add alt text to the plot", aes = "New aes legend title")`

`t + annotate(gem = "text", x = 8, y = 9, label = "A")`

Places a geom with manually selected aesthetics.

`p + guides(x = guide_axis(n.dodge = 2))`

Avoid crowded or overlapping labels with `guide_axis(n.dodge or angle)`.

`n + guides(fill = "none")`

Set legend type for each aesthetic: colorbar, legend, or none (no legend).

`n + theme(legend.position = "bottom")`

Place legend at "bottom", "top", "left", or "right".

`n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E"))`

Set legend title and labels with a scale function.

Zooming

Without clipping (preferred)

`t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))`

With clipping (removes unseen data points)

`t + xlim(0, 100) + ylim(10, 20)`

`t + scale_x_continuous(limits = c(0, 100)) + scale_y_continuous(limits = c(0, 100))`

Fonte: RSTUDIO (2021b).

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at ggplot2.tidyverse.org • ggplot2 3.3.5 • Updated: 2021-08

10.8 Salvando gráficos

Resta, agora, apresentar a função `ggsave`, que pode ser utilizada para salvar gráficos. O padrão dessa função é salvar o último gráfico exibido no RStudio Viewer, usando o tamanho do dispositivo gráfico atual. Ela também adivinha o tipo de dispositivo gráfico da extensão.

Sua forma básica é a seguinte:

```
ggsave(  
  filename,  
  plot = last_plot(),  
  device = NULL,  
  path = NULL,  
  scale = 1,  
  width = NA,  
  height = NA,  
  units = c("in", "cm", "mm", "px"),  
  dpi = 300,  
  limitsize = TRUE,  
  bg = NULL,  
  ...  
)
```

O primeiro argumento, `filename`, exige que se especifique o diretório onde a imagem deve ser salva. A extensão do arquivo deve ser informada podendo assumir as seguintes formas: `.eps`, `.pdf`, `.svg`, `.wmf`, `.png`, `.jpg`, `.bmp` e `.tiff`. Os argumentos `width` e `height` podem ser usados para definir o tamanho da saída gráfica em polegadas (se deixado em branco, a função usa como padrão o que está definido na tela). Para exportar gráficos como imagens (`.png` ou `.jpg`), o argumento `dpi` pode ser usado para definir a resolução do gráfico. O padrão é 300 dpi, podendo ser aumentado, por exemplo, para 600 dpi, caso se deseje uma saída de alta resolução.

```
ggsave("C:/Downloads/gráfico1.jpg")
```

POR DENTRO DA PNAD CONTÍNUA

10.9 Exercícios

Com base nos dados da PNAD Contínua do 1º trimestre de 2021 e utilizando os comandos do pacote **ggplot2**, defina e crie os gráficos mais adequados para representar os seguintes casos:

1. A proporção de pessoas na população por sexo e cor da pele.
2. A proporção de pessoas na população por grau de instrução para homens, pardos e com mais de 30 anos.
3. A distribuição da variável VD4020.
4. O número de horas efetivamente trabalhadas por sexo.
5. O número de horas efetivamente trabalhadas *versus* rendimento médio efetivo do trabalho principal.

11 PNAD Contínua: Indicadores sociais, de mercado de trabalho, de pobreza e de desigualdades

Nos capítulos anteriores, deve ter ficado claro que a PNAD Contínua possibilita a produção de um conjunto expressivo de indicadores sociais e de mercado de trabalho para distintos recortes de análise. O objetivo deste capítulo é apresentar alguns desses indicadores, reconhecendo as potencialidades da pesquisa, mas, também, suas limitações.

11.1 Indicadores de mercado de trabalho

A presente seção procura apresentar a forma de se calcular distintos indicadores de mercado de trabalho a partir dos conhecimentos adquiridos nos capítulos anteriores.

11.1.1 Bases de dados

Nesta seção, serão utilizados os dados da PNAD Contínua do primeiro trimestre de 2021 em seus formatos `design` e `data.frame`. Para baixar esses dados, basta realizar os seguintes comandos:

```
pnadc2021_T1_df <- get_pnadc(2021,  
                                quarter = 1,  
                                design = FALSE,  
                                labels = TRUE)  
  
pnadc2021_T1_dsg <- get_pnadc(2021,  
                                quarter = 1,  
                                design = TRUE,  
                                labels = TRUE)
```

Deve-se reparar que os dados foram atribuídos a dois objetos distintos com os sufixos `df`, para representar o formato `data.frame`, e `dsg`, para a classe de objeto do tipo `survey.design`.

11.1.2 A construção dos indicadores

Inicialmente, será apresentada uma forma de cálculo para os sete indicadores apresentados no Capítulo 3. Nesta primeira parte, serão utilizados os pacotes `expss` e `dplyr`, esse último contido no `tidyverse`. Para produzir tais indicadores, são necessários dois passos:

1. calcular o somatório de pessoas na população que se enquadram em cada uma das categorias necessárias para o cálculo das razões (numerador e denominador) que definem os indicadores;
2. calcular as relações.

No exemplo a seguir, os indicadores são calculados para as Unidades da Federação brasileiras.

Passo 1:

```
indicadores_mt <- pnadc2021_T1_df %>%
  tab_cells(UF) %>%
  tab_cols(VD4002, VD4003, VD4004A, VD4005) %>%
  tab_weight(weight = V1028) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

Os nomes das variáveis utilizadas podem ser vistos por meio da aplicação do seguinte comando:

```
names(indicadores_mt)
```

```
# > [1] "row_labels"
# [2] "VD4002|Pessoas ocupadas"
# [3] "VD4002|Pessoas desocupadas"
# [4] "VD4003|Pessoas fora da força de trabalho e na força de
#      trabalho potencial"
# [5] "VD4003|Pessoas fora da força de trabalho e fora da força de
#      trabalho potencial"
# [6] "VD4004A|Pessoas subocupadas"
# [7] "VD4005|Pessoas desalentadas"
```

Passo 2:

Para facilitar os comandos necessários aos cálculos das taxas, pode-se fazer uma transformação nos nomes das colunas (variáveis) usando a função `colnames`, como no exemplo a seguir:

Exemplo:

```
colnames(indicadores_mt) <-
  c("UF", "ocup", "desocup", "ftp", "fft", "subocup", "desalent")
```

POR DENTRO DA PNAD CONTÍNUA

Para verificar os novos nomes e se a transformação foi realizada com sucesso, basta fazer novamente:

```
names(indicadores_mt)
```

```
# > [1] "UF" "ocup" "desocup" "ftp" "fft" "subocup" "desalent"
```

Passo 3:

- Calcula taxas e relações percentuais com o pacote **dplyr**:

```
indicadores_mt <-  
  as.data.frame(indicadores_mt %>%  
    mutate(tx_desocup = desocup/(ocup+desocup)* 100,  
          tx_comb_subocup = (desocup+subocup)/(ocup+desocup)* 100,  
          tx_comb_ftp = (desocup+ftp)/(ocup+desocup+ftp)* 100,  
          tx_comp_subut = (desocup+subocup+ftp)/(ocup+desocup+ftp)* 100,  
          tx_desal_fta = (desalent)/(ocup+desocup+ftp)* 100,  
          pc_desal_fft = (desalent)/(fft + ftp)* 100,  
          pc_desal_ftp = (desalent)/(ftp)* 100))
```

- Seleciona só as variáveis referentes aos indicadores calculados:

```
indicadores_mt <-  
  as.data.frame(indicadores_mt) %>%  
    select(UF, tx_desocup, tx_comb_subocup,  
          tx_comb_ftp, tx_comp_subut, tx_desal_fta,  
          pc_desal_fft, pc_desal_ftp)  
  
#Visualiza apenas as primeiras linhas do objeto  
head(indicadores_mt)
```

A Tabela 11.1 traz a descrição dos indicadores como apresentada no Capítulo 3.

Tabela 11.1: Descrição dos indicadores de mercado de trabalho

Nome do Indicador	Descrição
tx_desocup	Taxa de desocupação
tx_comb_subocup	Taxa combinada de desocupação e de subocupação por insuficiência de horas trabalhadas
tx_comb_ftp	Taxa combinada de desocupação e força de trabalho potencial
tx_comp_subut	Taxa composta de subutilização da força de trabalho
tx_desal_fta	Taxa de desalentamento na força de trabalho ampliada
pc_desal_fft	Percentual de desalentados na população fora da força de trabalho
pc_desal_ftp	Percentual de desalentados na força de trabalho potencial

Fonte: Elaboração própria baseada em IBGE (2021).

O pacote **survey** permite o cálculo da taxa de desocupação por meio da função `svyratio`. Para saber o valor da estimativa pontual para a taxa, é preciso usar a função `coef`. Com esse pacote, pode-se calcular o erro padrão das estimativas, os coeficientes de variação e os intervalos de confiança. A lógica dessa função é definir o numerador da razão como primeiro argumento e o denominador como segundo argumento (ver código abaixo).

Exemplo:

```
tx_desocup <- svyratio(  
  ~VD4002 == "Pessoas desocupadas",  
  ~VD4001 == "Pessoas na força de trabalho",  
  pnadc2021_T1_dsg, na.rm = T)  
  
coef(tx_desocup)  
  
SE(tx_desocup)
```

POR DENTRO DA PNAD CONTÍNUA

```
cv(tx_desocup)  
  
confint(tx_desocup)
```

Ainda com relação ao **survey**, a função `svyby` permite que se calculem as taxas de desocupação por grupamentos de variáveis categóricas. No exemplo a seguir, isso é realizado por UF, para que os resultados dos dois métodos possam ser comparados.

Exemplo:

```
tx_desocup_UF<-svyby(  
  ~VD4002 == "Pessoas desocupadas",  
  by=~UF,  
  denominator=~VD4001 == "Pessoas na força de trabalho",  
  design=pnadc2021_T1_dsg,  
  svyratio,  
  na.rm = T)  
  
coef(tx_desocup_UF)
```

11.2 Indicadores sociais e de condições de vida

Nesta seção, serão explorados os dados da PNAD Contínua anual de 2019 que, como visto nos capítulos anteriores, traz um conjunto vasto de questões sobre: dimensões individuais dos moradores dos domicílios pesquisados (cor/raça, nível educacional etc.); acesso a bens de consumo individual; acesso a bens e serviços de uso coletivo; características próprias às condições de habitação; e características relativas às condições de vida da população.

Para baixar os dados novamente, basta aplicar os seguintes comandos¹:

```
pnadc2019_visita1_df <-  
  get_pnadc(2019,  
             interview = 1,  
             design = FALSE,  
             labels = TRUE)  
  
pnadc2019_visita1_dsg <-  
  get_pnadc(2019,  
             interview = 1,  
             design = TRUE,  
             labels = TRUE)
```

Do ponto de vista das dimensões que refletem elementos individuais próprios dos moradores dos domicílios brasileiros, é possível estudar a distribuição populacional por meio de indicadores de participação no total da população, definindo-se como recorte analítico variáveis como sexo e cor/raça (ver exemplos abaixo).

Exemplos:

1. Usando o pacote **expss**:

```
pnadc2019_visita1_df %>%  
  tab_cells(V2007) %>%  
  tab_cols(total()) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_cpct(total_statistic = "w_cpct",  
                total_label = "Total") %>%  
  tab_pivot()
```

¹ Deve-se reparar que os dados foram baixados, declarando-se o parâmetro **labels** = TRUE para que sejam aplicados rótulos aos conjuntos de dados.

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df %>%
  tab_cells(V2010) %>%
  tab_cols(total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Usando o pacote **dplyr**:

```
pnadc2019_visita1_df %>%
  count(V2007, wt = V1032) %>%
  mutate(prop = n / sum(n) * 100) %>%
  select("V2007", "prop")

pnadc2019_visita1_df %>%
  count(V2010, wt = V1032) %>%
  mutate(prop = n / sum(n) * 100) %>%
  select("V2010", "prop")
```

Ainda em relação a esse tipo de indicador de participação, pode ser do interesse do pesquisador agregar/recodificar algumas das categorias das variáveis de estudo. No exemplo a seguir, é feita uma reclassificação das categorias da variável **V2010** (cor/raça) para “Brancos” e “Não-brancos”, por meio da função `recode()` do pacote **expss**. O resultado desse procedimento foi armazenado em uma nova variável (`nova_cor`).

Exemplo:

```
pnadc2019_visita1_df$nova_cor <-
  recode(pnadc2019_visita1_df$V2010,
         "Branca" ~ "Brancos",
         "Preta" ~ "Não Brancos",
         "Amarela" ~ "Não Brancos",
         "Parda" ~ "Não Brancos",
         "Indígena" ~ "Não Brancos",
         "Ignorado" ~ NA)
```

O resultado das proporções para a nova variável criada (exclusive casos ignorados – transformados em omissos na variável `nova_cor`) pode ser obtido da seguinte forma (usando `dplyr`):

```
pnadc2019_visita1_df %>%
  filter(!is.na(nova_cor)) %>%
  count(nova_cor, wt = V1032) %>%
  mutate(prop = n / sum(n) * 100) %>%
  select("nova_cor", "prop")
```

Além desse tipo de análise, a PNAD Contínua permite o estudo de variáveis específicas relacionadas a características como idade e escolaridade das pessoas. Tais variáveis podem ser tratadas enquanto variáveis contínuas ou categorizadas em faixas com amplitudes próprias. No exemplo a seguir, é utilizada a variável Idade (v2009).

POR DENTRO DA PNAD CONTÍNUA

Exemplos:

1. Calcula a média de idade (V2009) por sexo (V2007):

```
#expss:  
pnadc2019_visita1_df %>%  
  tab_cells(V2009) %>%  
  tab_rows(V2007, total()) %>%  
  tab_weight(weight = V1032) %>%  
  tab_stat_fun(w_mean) %>%  
  tab_pivot()
```

				#Total	
#	-----	-----	-----	-----	-----
#	V2007	Homem	V2009	34.57	
#		Mulher	V2009	36.68	
#	#Total	V2009		35.66	

```
#dplyr  
bind_rows(  
  pnadc2019_visita1_df %>%  
    group_by(V2007) %>%  
    summarise(média = weighted.mean(V2009, w = V1032)),  
  pnadc2019_visita1_df %>%  
    filter(!is.na(V2007)) %>%  
    summarise(média = weighted.mean(V2009, w = V1032)) %>%  
    mutate(V2007 = "Total") %>%  
    select("V2007", "média"))
```

```
# > A tibble: 3 x 2
#   V2007   média
#   <chr>   <dbl>
# 1 Homem    34.6
# 2 Mulher   36.7
# 3 Total    35.7
```

Algumas dessas análises, também, podem ser feitas a partir da categorização de variáveis contínuas, como é o caso do estudo da população por faixas de idade. No exemplo a seguir, apresentam-se duas formas de se criar categorizações a partir da variável de idade em anos ([v2009](#)). O resultado desse procedimento pode ser armazenado em uma nova variável ([faixa_idade](#)).

Exemplos:

1. Cria categorias a partir de uma variável ordinal:

```
#expss
pnadc2019_visita1_df$faixa_idade <-
  recode(pnadc2019_visita1_df$V2009,
         0:10 ~ "Até 10 anos",
         11:20 ~ "De 11 a 20 anos",
         21:30 ~ "De 21 a 30 anos",
         31:40 ~ "De 31 a 40 anos",
         41:50 ~ "De 41 a 50 anos",
         51:60 ~ "De 51 a 60 anos",
         61:120 ~ "Mais de 60 anos")
```

2. Outra forma de se chegar ao mesmo resultado é por meio da função [ifelse\(\)](#), contida nas funções de base do R:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df$faixa_idade <-
  ifelse (pnadc2019_visita1_df$V2009 >= 0 &
          pnadc2019_visita1_df$V2009 <= 10,
          "Até 10 anos",
  ifelse (pnadc2019_visita1_df$V2009 >= 11 &
          pnadc2019_visita1_df$V2009 <= 20,
          "De 11 a 20 anos",
  ifelse (pnadc2019_visita1_df$V2009 >= 21 &
          pnadc2019_visita1_df$V2009 <= 30,
          "De 21 a 30 anos",
  ifelse (pnadc2019_visita1_df$V2009 >= 31 &
          pnadc2019_visita1_df$V2009 <= 40,
          "De 31 a 40 anos",
  ifelse (pnadc2019_visita1_df$V2009 >= 41 &
          pnadc2019_visita1_df$V2009 <= 50,
          "De 41 a 50 anos",
  ifelse (pnadc2019_visita1_df$V2009 >= 51 &
          pnadc2019_visita1_df$V2009 <= 60,
          "De 51 a 60 anos",
  ifelse (pnadc2019_visita1_df$V2009 > 60,
          "Mais de 60 anos", NA))))))
```

3. Calcula a distribuição populacional segundo a variável `faixa_idade` em porcentagem:

```
#expss
pnadc2019_visita1_df %>%
  tab_cells(faixa_idade) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                total_label = "Total") %>%
  tab_pivot()
```

```
#dplyr
pnadc2019_visita1_df %>%
  count(faixa_idade, wt = V1032) %>%
  mutate(prop = n / sum(n) * 100) %>%
  select("faixa_idade","prop")
```

As análises relacionadas às condições de habitação, de acesso a bens e serviços públicos e de consumo individual podem ser tratadas por meio de duas formas básicas:

1. números absolutos;
2. números relativos ao total de pessoas ou de domicílios (em porcentagem).

A seguir, serão apresentadas alguns exemplos que podem contribuir para deixar clara essa forma de abordagem. Para isso será utilizada a variável S01007 referente à seguinte questão da PNAD Contínua: “Qual é a principal forma de abastecimento de água utilizada neste domicílio?”.

Deve-se notar que esse tipo de tratamento pode ser usado para variáveis como:

- S01003: Qual é o material que predomina na cobertura (telhado) deste domicílio? (se refere à estrutura e a condição dos domicílios);
- S01012A: Para onde vai o esgoto do banheiro, sanitário ou buraco de dequeção? (referente ao acesso a bens e serviços de uso coletivo);
- S01013: Qual é o (principal) destino dado ao lixo? (referente ao acesso a bens e serviços de uso coletivo);
- S01023: Este domicílio tem geladeira? (associada à posse de bens de consumo de uso individual);
- S01028: Este domicílio tem microcomputador, considerando inclusive os portáteis, tais como: *laptop*, *notebook*, *ultrabook* ou *netbook*? (associadas à posse de bens de consumo de uso individual).

POR DENTRO DA PNAD CONTÍNUA

Tais informações podem ser exploradas das seguintes formas:

1. Total de pessoas:

```
pnadc2019_visita1_df %>%
  tab_cells(S01012A) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

2. Percentual de pessoas:

```
pnadc2019_visita1_df %>%
  tab_cells(S01007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                 total_label = "Total") %>%
  tab_pivot()
```

3. Total de domicílios²:

```
pnadc2019_visita1_df %>%
  filter(V2005 ==
         "Pessoa responsável pelo domicílio") %>%
  tab_cells(S01007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

² Note que, para fazer o cálculo do número de domicílios, pode-se recorrer à variável Condição no domicílio (V2005 == "Pessoa responsável pelo domicílio").

4. Percentual de domicílios:

```
pnadc2019_visita1_df %>%
  filter(V2005 ==
    "Pessoa responsável pelo domicílio") %>%
  tab_cells(S01007) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cpct(total_statistic = "w_cpct",
                total_label = "Total") %>%
  tab_pivot()
```

11.3 Indicadores de pobreza, desigualdades e insuficiência socioeconômica

A pobreza é um fenômeno dinâmico, complexo e multidimensional que vem sendo tratado por diversos ramos das ciências sociais ao longo do tempo. Seu conceito e sua mensuração foram sendo alterados à medida que as sociedades avançavam. Essa última tem ganhado novas formas de abordagem nas últimas décadas. Boa parte das estimativas pra as distintas medidas de pobreza tem sido calculadas com base em pesquisas domiciliares amostrais e censitárias em todos os países (DE-ATON, 1997; JACOB; DAMICO; PESSOA, 2022). No Brasil, isso não é diferente.

11.3.1 Indicadores de pobreza

O esforço de Jacob, Damico e Pessoa (2022) vai no sentido de explorar os aspectos da inferência estatística e das distintas medidas de pobreza monetária, levando em consideração os desenhos amostrais complexos das pesquisas que alimentam o conjunto de informações necessárias. A investigação das distintas estimativas de pobreza, bem como seus erros de estimativa, pode ser feita com o instrumental analítico contido no pacote **convey**, desenvolvido por esses mesmos autores³.

³ Ver Pessoa, Damico e Jacob (2021).

POR DENTRO DA PNAD CONTÍNUA

Inicialmente, é necessário fazer a instalação e a habilitação do pacote:

```
install.packages("convey")
library(convey)
```

Pobreza monetária: uma olhar a partir da renda corrente

Nesta seção, serão exploradas algumas dessas medidas para o Rendimento efetivo domiciliar *per capita* da PNAD Contínua anual de 2019. Porém, como visto no Capítulo 8, para se levar em consideração os desenhos amostrais para a estimativa de distintas estatísticas com base nos dados da PNAD Contínua, é necessário que esta esteja no formato `survey.design` (ver comando a seguir)⁴.

```
pnadc2019_visita1_dsg <-
  get_pnadc(2019,
             interview=1,
             design=TRUE,
             labels = FALSE)
```

O pacote **convey** exige a preparação do objeto que contém os dados no formato `survey.design` por meio da função `convey_prep()` (ver comando a seguir).

```
pnadc2019_visita1_dsg <-
  convey_prep(pnadc2019_visita1_dsg)
```

- Principais indicadores contidos no pacote **convey** para o tratamento da pobreza:

⁴ Para facilitar o tratamento dos dados, optou-se por desconsiderar os rótulos (argumento `labels` = FALSE).

1. **Limiar de risco de pobreza (svyarpt):** medida usada para definir o valor do limite superior que faz com que as pessoas com rendimentos inferiores a esse limite sejam consideradas em situação de baixo padrão de vida quando comparadas com as outras que possuem um padrão de vida geral médio, em termos monetários. As pessoas que se encontram abaixo desse limiar podem ser consideradas em privação relativa. Sua definição padrão é usar 60% da renda mediana.

```
svyarpt(~VD5005,  
        pnadc2019_visita1_dsg,  
        quantiles = 0.5,  
        na.rm=TRUE)  
  
#Para obter apenas o valor do limiar  
coef(svyarpt(~VD5005,  
            pnadc2019_visita1_dsg,  
            quantiles = 0.5,  
            na.rm=TRUE))
```

2. **Proporção de pessoas sob risco de pobreza (svyarpr):** proporção de pessoas com renda abaixo do limite de risco de pobreza. Embora certas pessoas abaixo do limiar de pobreza não, necessariamente, sejam pobres em seu conceito mais amplo, estas podem ser associadas a algum tipo de vulnerabilidade, ao menos do ponto de vista monetário.

```
svyarpr(~VD5005,  
        pnadc2019_visita1_dsg,  
        quantiles = 0.5,  
        percent = 0.6,  
        na.rm=TRUE)
```

POR DENTRO DA PNAD CONTÍNUA

```
#Para obter o valor em %
coef(svyarpr(~VD5005,
              pnadc2019_visita1_dsg,
              quantiles = 0.5,
              percent = 0.6,
              na.rm=TRUE))* 100
```

3. **Razão de renda mediana relativa (svyrmir):** relação entre o rendimento mediano de pessoas com idade superior a um determinado valor (no exemplo, maiores de 65) e o rendimento mediano de pessoas com idade inferior ou igual a esse mesmo valor. O resultado da função informa os erros de estimativa e os valores das medianas dos dois grupos etários, além da razão entre elas.

$$\text{Razão de renda mediana relativa} = \frac{\text{Rendimento mediano de pessoas maiores de 65 anos}}{\text{Rendimento mediano de pessoas com 65 anos ou menos}} \quad (11.1)$$

```
svyrmir(~VD5005,
        pnadc2019_visita1_dsg,
        age = ~V2009,
        agelim = 65,
        quantiles = 0.5,
        na.rm = TRUE,
        med_old = TRUE, med_young = TRUE)

coef(svyrmir(~VD5005,
            pnadc2019_visita1_dsg,
            age = ~V2009,
            agelim = 65,
            quantiles = 0.5,
            na.rm = TRUE,
            med_old = TRUE, med_young = TRUE))
```

4. **Hiato de pobreza mediana relativa** (`svyrmmpg`): estimativa da seguinte relação: no numerador, a diferença entre a renda mediana das pessoas com renda abaixo do valor limiar de risco de pobreza e o próprio valor limiar; e no denominador, o valor desse limiar (ver equação a seguir).

$$\text{Hiato de pobreza} = \frac{\text{Rendimento mediano inferior ao limiar} - \text{Rendimento limiar}}{\text{Rendimento limiar}} \quad (11.2)$$

```
svyrmmpg(~VD5005,  
         pnadc2019_visita1_dsg,  
         quantiles = 0.5,  
         percent = 0.6,  
         na.rm = TRUE,  
         thresh = TRUE,  
         poor_median = TRUE)  
  
coef(svyrmmpg(~VD5005,  
              pnadc2019_visita1_dsg,  
              quantiles = 0.5,  
              percent = 0.6,  
              na.rm = TRUE,  
              thresh = TRUE,  
              poor_median = TRUE))
```

5. **Rendimento mediano abaixo do limiar de risco de pobreza** (`svypoormed`): estimativa do valor do rendimento mediano das pessoas com rendimentos inferiores ao limiar de risco de pobreza⁵.

⁵ Repare que esse valor também é obtido no cálculo da função `svyrmmpg`.

POR DENTRO DA PNAD CONTÍNUA

```
svypoormed(~VD5005,  
           pnadc2019_visita1_dsg,  
           quantiles = 0.5,  
           percent = 0.6,  
           na.rm = TRUE)  
  
coef(svypoormed(~VD5005,  
                 pnadc2019_visita1_dsg,  
                 quantiles = 0.5,  
                 percent = 0.6,  
                 na.rm = TRUE))
```

A classe de indicadores do tipo FGT, desenvolvida por Foster, Greer e Thorbecke (may 1984), são medidas de pobreza que combinam:

1. um indicador de extensão da pobreza, mensurado pela proporção de pobres abaixo da linha de pobreza (limiar de pobreza);
 2. um indicador para o déficit médio normalizado para a renda dos pobres, o que leva em conta a insuficiência de renda;
 3. um indicador que amplifica o peso dessa insuficiência para produzir uma medida que busca explicitar a severidade da pobreza, ou a desigualdade entre os pobres.
- FGT (`svyfgt`): calcula as medidas de Foster, Greer e Thorbecke de acordo com três parâmetros gama:
 1. se gama = 0, FGT (0) traz o resultado da proporção de pessoas abaixo da linha de pobreza, indicando a extensão da pobreza;
 2. se gama = 1, FGT (1) apresenta a média do déficit de renda normalizado dos pobres, indicando a extensão combinada à intensidade da pobreza;
 3. se gama = 2, FGT(2) calcula o peso relativo dos déficits em relação à pobreza ampliado, produzindo uma medida para captar a gravidade da pobreza, ou

seja, a desigualdade entre os pobres. Indicando que a transferência da renda de uma pessoa pobre para uma pessoa ainda mais pobre tende a reduzir o indicador FGT.

A forma básica da função (svyfgt) é a seguinte:

```
svyfgt(formula,  
        design,  
        g,  
        type_thresh = "abs",  
        abs_thresh = NULL,  
        percent = 0.6,  
        quantiles = 0.5,  
        na.rm = FALSE,  
        thresh = FALSE)
```

Deve-se ter claro que, caso o argumento `type_thresh` seja igual a “abs”, o limiar é fixado pelo valor passado ao argumento `abs_thresh`. Se o usuário definir `type_thresh` como sendo igual a “relq”, a linha de pobreza (limiar de pobreza) é definida enquanto um porcentual de um determinado quantil. Já se `type_thresh` for definido como “relm”, o limiar passa a ser um percentual da renda média.

Exemplos de aplicações para os dados da PNAD Contínua:

1. Proporção de pobres – FGT(0) – usando como critério uma linha de pobreza cujo valor é 0.5 da mediana.

```
svyfgt(~VD5005,  
       pnadc2019_visita1_dsg,  
       g = 0,  
       type_thresh = "relq",  
       abs_thresh = NULL,  
       percent = 0.5,  
       quantiles = 0.5,  
       na.rm = TRUE,  
       thresh = TRUE)
```

POR DENTRO DA PNAD CONTÍNUA

2. Proporção de pobres – FGT(0) – usando como critério uma linha de pobreza cujo valor é 0.5 da média.

```
svyfgt(~VD5005,  
       pnadc2019_visita1_dsg,  
       g = 0,  
       type_thresh = "relm",  
       abs_thresh = NULL,  
       percent = 0.5,  
       quantiles = NULL,  
       na.rm = TRUE,  
       thresh = TRUE)
```

3. Proporção de pobres – FGT(0) – usando como critério uma linha de pobreza cujo valor é 0.5 do salário-mínimo⁶.

```
svyfgt(~VD5005,  
       pnadc2019_visita1_dsg,  
       g = 0,  
       type_thresh = "abs",  
       abs_thresh = 0.5*998,  
       percent = NULL,  
       quantiles = NULL,  
       na.rm = TRUE,  
       thresh = TRUE)
```

4. Intensidade combinada à extensão da pobreza – FGT(1) – usando como critério uma linha de pobreza cujo valor é 0.5 da mediana.

⁶ Deve-se atentar para o fato de o valor do salário-mínimo em 2019 ser de R\$ 998,00.

```
svyfgt(~VD5005,  
pnadc2019_visita1_dsg,  
g = 1,  
type_thresh = "relq",  
abs_thresh = NULL,  
percent = 0.5,  
quantiles = 0.5,  
na.rm = TRUE,  
thresh = TRUE)
```

5. Intensidade combinada à extensão da pobreza – FGT(1) – usando como critério uma linha de pobreza cujo valor é 0.5 da média.

```
svyfgt(~VD5005,  
pnadc2019_visita1_dsg,  
g = 1,  
type_thresh = "relm",  
abs_thresh = NULL,  
percent = 0.5,  
quantiles = NULL,  
na.rm = TRUE,  
thresh = TRUE)
```

6. Intensidade combinada à extensão da pobreza – FGT(1) – usando como critério uma linha de pobreza cujo valor é 0.5 do salário-mínimo de 2019.

```
svyfgt(~VD5005,  
pnadc2019_visita1_dsg,  
g = 1,  
type_thresh = "abs",  
abs_thresh = 0.5*998,
```

POR DENTRO DA PNAD CONTÍNUA

```
percent = NULL,  
quantiles = NULL,  
na.rm = TRUE,  
thresh = TRUE)
```

7. Intensidade combinada à extensão da pobreza com sensibilidade ampliada – FGT(2) – usando como critério uma linha de pobreza cujo valor é 0.5 da mediana.

```
svyfgt(~VD5005,  
pnadc2019_visita1_dsg,  
g = 2,  
type_thresh = "relq",  
abs_thresh = NULL,  
percent = 0.5,  
quantiles = 0.5,  
na.rm = TRUE,  
thresh = TRUE)
```

8. Intensidade combinada à extensão da pobreza com sensibilidade ampliada – FGT(2) – usando como critério uma linha de pobreza cujo valor é 0.5 da média.

```
svyfgt(~VD5005,  
pnadc2019_visita1_dsg,  
g = 2,  
type_thresh = "relm",  
abs_thresh = NULL,  
percent = 0.5,  
quantiles = NULL,  
na.rm = TRUE,  
thresh = TRUE)
```

9. Intensidade combinada à extensão da pobreza com sensibilidade ampliada – FGT(2) – usando como critério uma linha de pobreza cujo valor é 0.5 do salário-mínimo.

```
svyfgt(~VD5005,  
        pnadc2019_visita1_dsg,  
        g = 2,  
        type_thresh = "abs",  
        abs_thresh = 0.5*998,  
        percent = NULL,  
        quantiles = NULL,  
        na.rm = TRUE,  
        thresh = TRUE)
```

Pobreza multidimensional: um olhar ampliado

O objetivo desta seção é apresentar a operacionalização do cálculo do *Multidimensional Poverty Index – MPI* (Índice de Pobreza Multidimensional – IPM) do Programa das Nações Unidas para o Desenvolvimento (PNUD) e adaptado para o Brasil por Brandão (2021). Originalmente, o IPM examina as privações de cada pessoa a partir de 10 indicadores em três dimensões igualmente ponderadas – saúde, educação e padrão de vida –, oferecendo uma ferramenta para se identificar quem é pobre e quão pobres eles são.

O valor do IPM é obtido pela multiplicação da proporção de pobres multidimensionais pelo indicador de intensidade da pobreza referente a esse grupo de pessoas, o que permite captar, também, a influência da relação entre o número de privações observadas e o total de privações possíveis.

Além disso, essa implementação será explorada para os dados da PNAD Contínua anual de 2019, porém em seu formato `data.frame`. Os indicadores serão construídos para cada uma das Unidades da Federação. A fim de facilitar os cálculos e a programação, serão utilizados os microdados sem a aplicação de rótulos. Para baixar os dados novamente, basta realizar o seguinte comando:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df <-
  get_pnadc(2019,
             interview = 1,
             design = FALSE,
             labels = FALSE)
```

A implementação do IPM exige uma sequência de passos:

1. Cálculo do Rendimento real efetivo domiciliar *per capita*⁷:

```
pnadc2019_visita1_df <-
  pnadc2019_visita1_df %>%
  mutate(VD5005_def_ano = VD5005 * C03)
```

2. Cálculo dos indicadores de identificação de privação a partir das cinco dimensões definidas no trabalho de Brandão (2021). As dimensões e seus indicadores para os domicílios brasileiros são os seguintes⁸:

- (a) **Trabalho e renda:** pobreza monetária, desemprego, ausência de contribuição para a previdência;
- (b) **Consumo:** ausência de bens de consumo individual como televisão ou computador, geladeira, máquina de lavar e telefone;
- (c) **Condições de habitação:** privação de acesso a bens de uso coletivo como água encanada, luz elétrica, esgotamento sanitário adequado;

⁷ Para esse cálculo, foi utilizado o deflator disponibilizado na PNAD Contínua por meio da variável C03. Para mais detalhes, ver documentação de apoio para deflacionamento anual por visita, disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Microdados/Visita/Documentacao_Geral/PNADcIBGE_Deflator_Anual_Visita.pdf.

⁸ Para mais detalhes sobre o esquema de ponderação do IPM elaborado pelo PNUD, ver UNDP e OPHI (2020).

- (d) **Educação:** presença de adultos analfabetos, crianças e jovens fora da escola, presença de pessoas de 18 anos ou mais de idade sem ensino médio completo;
- (e) **Demográfica:** presença de idoso sem acesso à aposentadoria/pensão ou Benefício de Prestação Continuada (BPC).

```
pnadc2019_visita1_df <-
pnadc2019_visita1_df %>%
  mutate(
    H_pobre = ifs(VD5005_def_ano >= 0 &
                  VD5005_def_ano <= 998/2 ~ 1*(0.2/3),
                  TRUE ~ 0),
    H_desemp = ifs(VD4002 == 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_contprev = ifs(VD4012 == 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_TVéPC = ifs (S01025 == 4 & S01028 == 2 ~ 1*(0.2/4),
                  TRUE ~ 0),
    H_geladeira = ifs (S01023 == 3 ~ 1*(0.2/4),TRUE ~ 0),
    H_maglavar = ifs (S01024 == 2 ~ 1*(0.2/4),TRUE ~ 0),
    H_telefone = ifs (S01022 == 2 & S01021 == 0 ~ 1*(0.2/4),
                      TRUE ~ 0),
    H_aguaenc = ifs (S01007 > 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_luz = ifs (S01014 == 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_esgoto = ifs (S01012A >= 4 ~ 1*(0.2/3),TRUE ~ 0),
    H_aanalf = ifs (V3001 == 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_auxescol = ifs(V2009 >= 7 & V2009 <= 17 &
                      V3002 == 2 ~ 1*(0.2/3),TRUE ~ 0),
    H_ensmed = ifs(V2009 >= 18 & VD3004 < 5 ~ 1*(0.2/3),
                   TRUE ~ 0),
    H_idoso = ifs((V2009 >= 60 & (V5001A == 2 |
                      V5004A == 2)) ~ 1*(0.2),TRUE ~ 0))
```

3. Agrupamento dos dados por domicílios em uma base em que cada caso é um domicílio e não mais uma pessoa:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df_DOM <-
pnadc2019_visita1_df%>%
  dplyr::group_by(UPA, Estrato, V1008) %>%
  dplyr::summarise("H_pobre"=max(H_pobre),
                    "H_desemp"=max(H_desemp),
                    "H_contprev"=max(H_contprev),
                    "H_TVePC"=max(H_TVePC),
                    "H_geladeira"=max(H_geladeira),
                    "H_maqlavar"=max(H_maqlavar),
                    "H_telefone"=max(H_telefone),
                    "H_aguaenc"=max(H_aguaenc),
                    "H_luz"=max(H_luz),
                    "H_esgoto"=max(H_esgoto),
                    "H_aanalf"=max(H_aanalf),
                    "H_auxescol"=max(H_auxescol),
                    "H_ensmed"=max(H_ensmed),
                    "H_idoso"=max(H_idoso),
                    .groups="drop")
```

4. Cálculo do somatório das privações ponderadas na base agregada por domicílios:

```
pnadc2019_visita1_df_DOM <-
pnadc2019_visita1_df_DOM %>%
  mutate(H_pobre_mult = H_pobre + H_desemp +
        H_contprev + H_TVePC + H_geladeira +
        H_maqlavar + H_telefone + H_aguaenc +
        H_luz + H_esgoto + H_aanalf +
        H_auxescol + H_ensmed + H_idoso)
```

5. Imputação dos valores calculados para os domicílios para cada caso na base original, mantendo a correspondência entre o domicílio e seus moradores:

```
pnadc2019_visita1_df <-
  left_join(pnadc2019_visita1_df, pnadc2019_visita1_df_DOM,
            by=c("UPA", "Estrato", "V1008"))
```

6. Identificação dos pobres e extremamente pobres multidimensionais por meio de limites fracionários de privação (maior ou igual a 1/3 para pobres, e maior ou igual a 1/2 para extremamente pobres):

```
pnadc2019_visita1_df<-
  pnadc2019_visita1_df%>%
  mutate(
    ID_pobre_mult = ifs(H_pobre_mult >= 1/3 ~ 1,
                          TRUE ~ 0),
    ID_ext_pobre_mult = ifs(H_pobre_mult >= 1/2 ~ 1,
                           TRUE ~ 0))
```

7. Cálculo dos indicadores de proporção de pobres (Hp) e extremamente pobres (Hep) multidimensionais e da intensidade da pobreza (Ap) e da extrema pobreza (Aep), além do próprio valor do Índice de Pobreza Multidimensional (IPM) e do Índice de Extrema Pobreza Multidimensional (IEPM):

```
dados1<-
  pnadc2019_visita1_df %>%
  tab_cells(UF) %>%
  tab_cols(ID_pobre_mult,
           ID_ext_pobre_mult,
           total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_cases(total_statistic = "w_cases",
                 total_label = "Total") %>%
  tab_pivot()
```

POR DENTRO DA PNAD CONTÍNUA

```
dados2<-
  pnadc2019_visita1_df %>%
  tab_cells(H_pobre_mult) %>%
  tab_cols(ID_pobre_mult,
            ID_ext_pobre_mult) %>%
  tab_rows(UF, total()) %>%
  tab_weight(weight = V1032) %>%
  tab_stat_fun(Soma = w_sum) %>%
  tab_pivot()

dados<-cbind(dados1,dados2)
dados<-dados[,c(1,3,5,6,9,11)]
colnames(dados)<-c("UF","pm","epm","total","ext_pm","ext_epm" )
dados%>%
  mutate(Hp = pm/total,
        Ap = ext_pm/pm,
        IPM = Hp*Ap,
        Hep = epm/total,
        Aep = ext_epm/epm,
        IEPM = Hep*Aep) %>%
  select(UF,Hp, Ap, IPM, Hep, Aep, IEPM)
```

O resultado desses procedimentos será um `data.frame` que, no console, assumirá uma forma parecida com a Figura 11.1.

Figura 11.1: IPM e IEPM para as Unidades da Federação do Brasil (2019)

	UF	Hp	Ap	IPM	Hep	Aep	IEPM
1	UF 11	0.2554136	0.4140354	0.10575027	0.041236160	0.5448928	0.022469287
2	UF 12	0.3930647	0.4578916	0.17998103	0.115970754	0.5810669	0.067386770
3	UF 13	0.3434569	0.4536830	0.15582057	0.093753138	0.5919372	0.055495968
4	UF 14	0.2911759	0.4303936	0.12532027	0.055277210	0.5582325	0.030857536
5	UF 15	0.4090306	0.4553310	0.18624431	0.116597020	0.5790353	0.067513785
6	UF 16	0.3448913	0.4383906	0.15119711	0.074382249	0.5649244	0.042020348
7	UF 17	0.2950773	0.4345576	0.12822809	0.067375111	0.5477792	0.036906687
8	UF 21	0.3853926	0.4407463	0.16986036	0.092420283	0.5550325	0.051296263
9	UF 22	0.3737102	0.4374054	0.16346284	0.092904137	0.5487683	0.050982845
10	UF 23	0.3397916	0.4287430	0.14568328	0.069976803	0.5485539	0.038386050
11	UF 24	0.3674685	0.4347776	0.15976708	0.094363070	0.5514667	0.052038093
12	UF 25	0.3406083	0.4337385	0.14773492	0.079562994	0.5419686	0.043120648
13	UF 26	0.3201681	0.4324705	0.13846328	0.072719389	0.5550156	0.040360397
14	UF 27	0.3688369	0.4368122	0.16111249	0.085781744	0.5507854	0.047247328
15	UF 28	0.3062133	0.4315463	0.13214522	0.064872180	0.5543276	0.035960437
16	UF 29	0.3312062	0.4376007	0.14493604	0.084134623	0.5503516	0.046303626
17	UF 31	0.2357820	0.4098136	0.09662668	0.032909367	0.5552786	0.018273867
18	UF 32	0.1980679	0.3994863	0.07912540	0.022038459	0.5430690	0.011968403
19	UF 33	0.1731795	0.3791188	0.06565560	0.008595331	0.5373682	0.004618857
20	UF 35	0.1527603	0.3748886	0.05726810	0.007309462	0.5500958	0.004020904
21	UF 41	0.1675505	0.3858908	0.06465621	0.011592128	0.5412463	0.006274196
22	UF 42	0.1286642	0.3698170	0.04758221	0.005708938	0.5335835	0.003046195
23	UF 43	0.1767812	0.3748325	0.06626333	0.008749540	0.5499090	0.004811451
24	UF 50	0.1813515	0.3946295	0.07156666	0.013806643	0.5395220	0.007448988
25	UF 51	0.1856817	0.3955507	0.07344652	0.016122128	0.5501023	0.008868819
26	UF 52	0.2118413	0.3998298	0.08470047	0.023531747	0.5505990	0.012956557
27	UF 53	0.1478709	0.3761043	0.05561487	0.006282457	0.5320869	0.003342813
28	UF #Total	0.2379003	0.4140586	0.09850465	0.039036540	0.5563301	0.021717203

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

11.3.2 Desigualdade de renda corrente

São diversos os pacotes que permitem o cálculo de indicadores de concentração de renda no R. Nesta seção, serão apresentados alguns desses pacotes para tratar os dados da PNAD Contínua, tanto no formato `data.frame` quanto `survey.design`.

Para a apresentação da implementação das distintas funções que permitem os cálculos dos índices, será utilizada a base de dados anual de 2019 para a visita 1.

Índice de Gini

Um dos principais índices de concentração de renda, amplamente utilizado na literatura, é o chamado índice de Gini, desenvolvido por Gini (1936). Os pacotes existentes em linguagem R são diversos. Aqui serão apresentados dois: o pacote **IC2**, criado por Plat (2012), para uso com dados em formato `data.frame`; e o **convey**,

POR DENTRO DA PNAD CONTÍNUA

desenvolvido por Pessoa, Damico e Jacob (2021), para uso com o formato `survey.design`.

A instalação e a habilitação dos pacotes podem ser feitas da seguinte forma:

```
install.packages("convey")
library(convey)

install.packages("IC2")
library(IC2)
```

Os exemplos apresentados na sequência utilizarão o Rendimento real efetivo domiciliar *per capita* (`VD5005_def_ano`), variável definida anteriormente no presente capítulo. No pacote **IC2**, a função é a `calcSGini`, que calcula o coeficiente estendido de Gini para um vetor, enquanto no **convey**, a função é `svygini`. Os resultados são apresentados na Figura 11.2.

Exemplos:

1. Usando o pacote **IC2**:

```
calcSGini(pnadc2019_visita1_df$VD5005_def_ano,
           w = pnadc2019_visita1_df$V1032,
           param = 2)
```

2. Usando o pacote **convey**:

```
#Tratar os dados, caso não tenham sido preparados anteriormente
#Criar variável deflacionada
pnadc2019_visita1_dsg$variables <-
  pnadc2019_visita1_dsg$variables %>%
  mutate(VD5005_def_ano =
    VD5005 * C03)
```

```
#Preparar os dados
pnadc2019_visita1_dsg <-
  convey_prep(pnadc2019_visita1_dsg)

#Calcular o indicador
svygini(~VD5005_def_ano,
         pnadc2019_visita1_dsg,
         na.rm = TRUE)
```

O índice de Gini é calculado com base na relação entre a chamada Curva de Lorenz observada e a curva de perfeita igualdade, representada por uma reta de 45°(LORENZ, 1905). Esses pacotes, também, permitem a representação gráfica dessas curvas, como é apresentado na sequência (ver Figura 11.2).

1. Usando o pacote **IC2**:

```
curveLorenz(pnadc2019_visita1_df$VD5005_def_ano,
             w = pnadc2019_visita1_df$V1032,
             gener = FALSE,
             xlab = "Participação na população",
             ylab = "Participação na renda total",
             add = FALSE,
             grid = 0,
             col = "blue")
```

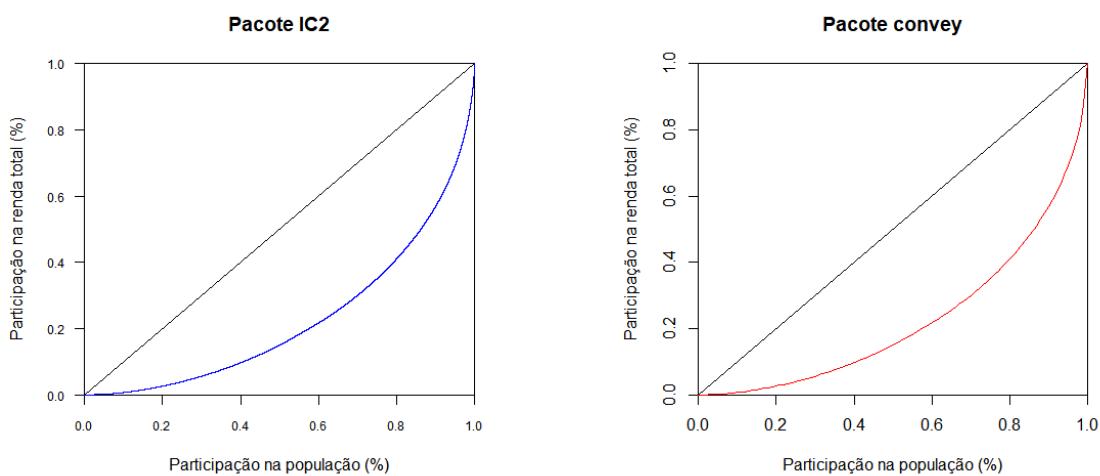
2. Usando o pacote **convey**:

```
svylorenz(
  ~VD5005_def_ano,
  pnadc2019_visita1_dsg,
  quantiles = seq(0, 1, 0.01),
```

POR DENTRO DA PNAD CONTÍNUA

```
empirical = FALSE,  
plot = TRUE,  
add = FALSE,  
xaxs = "i",  
yaxs = "i",  
type = "l",  
lwd = 1,  
main = "Pacote convey",  
xlab = "Participação na população (%)",  
ylab = "Participação na renda total (%)",  
curve.col = "red",  
ci = TRUE,  
alpha = 0.05,  
na.rm = TRUE)
```

Figura 11.2: Curvas de Lorenz criadas a partir dos pacote **IC2** e **convey**



Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Outro importante indicador de concentração de renda é o Índice de Atkinson, desenvolvido por Atkinson *et al.* (1970). Nesse indicador, é possível definir um parâmetro (“épsilon”) de aversão à desigualdade. À medida que esse parâmetro se

aproxima do infinito, maior peso relativo é atribuído às rendas na parte inferior da distribuição. Sua aplicação pode ser vista na sequência.

Exemplos:

1. Usando o pacote **IC2**⁹:

```
pnadc2019_visita1_df %>%
  filter(VD5005_def_ano > 0) %>%
  summarise(
    "epsilon = 0.5" = calcAtkinson(VD5005_def_ano,
      w = V1032, epsilon = 0.5)$ineq$index,
    "epsilon = 1" = calcAtkinson(VD5005_def_ano,
      w = V1032, epsilon = 1)$ineq$index,
    "epsilon = 1.5" = calcAtkinson(VD5005_def_ano,
      w = V1032, epsilon = 1.5)$ineq$index,
    "epsilon = 2" = calcAtkinson(VD5005_def_ano,
      w = V1032, epsilon = 2)$ineq$index)
```

```
# > A tibble: 1 x 4
# `epsilon = 0.5` `epsilon = 1` `epsilon = 1.5` `epsilon = 2`
#       <dbl>     <dbl>     <dbl>     <dbl>
# 1     0.250     0.435     0.585     0.710
```

2. Usando o pacote **convey**:

```
svyatk(~VD5005_def_ano,
  subset(pnadc2019_visita1_dsg,
    pnadc2019_visita1_df$VD5005_def_ano > 0),
    epsilon = 0.5)
```

⁹ Deve-se reparar que todos os indicadores foram calculados de uma única vez com o auxílio do pacote **dplyr**.

POR DENTRO DA PNAD CONTÍNUA

```
svyatk(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
              epsilon = 1)  
  
svyatk(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
              epsilon = 1.5)  
  
svyatk(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
              epsilon = 2)
```

O Índice de Atkinson é um caso particular dos indicadores de concentração do tipo entropia generalizada. Esse é um conceito oriundo da física, que se associa à mensuração das multiplicidades de estados de um sistema, ou do grau de “desordem” ou desconcentração de uma determinada distribuição. Segundo Medeiros (2012, p. 145):

Se toda a renda está concentrada em um único indivíduo, o grau de entropia dessa distribuição é baixo (há muita “ordem” devido à concentração ou agregação em torno de um único indivíduo). Se a renda é bem distribuída, há pouca agregação em torno de um indivíduo e, portanto, o grau de entropia é alto.

No indicador de entropia generalizada, a definição do parâmetro “épsilon” faz com que se obtenham os índices de T-Theil ($\epsilon = 0$), L-Theil ($\epsilon = 1$) e do Coeficiente de Variação ($\epsilon = -1$). Sua aplicação pode ser vista abaixo:

1. Usando o pacote **IC2**:

```
pnadc2019_visita1_df %>%
  filter(VD5005_def_ano > 0) %>%
  summarise("epsilon = -1" = calcGEI(VD5005_def_ano,
                                         w=V1032, alpha = -1)$ineq$index,
            "epsilon = 0"   = calcGEI(VD5005_def_ano,
                                         w=V1032, alpha = 0)$ineq$index,
            "epsilon = 1"   = calcGEI(VD5005_def_ano,
                                         w=V1032, alpha = 1)$ineq$index,
            "epsilon = 2"   = calcGEI(VD5005_def_ano,
                                         w=V1032, alpha = 2)$ineq$index,
            "epsilon = 3"   = calcGEI(VD5005_def_ano,
                                         w=V1032, alpha = 3)$ineq$index)
```

```
# > A tibble: 1 x 5
#   `epsilon = -1` `epsilon = 0` `epsilon = 1` `epsilon = 2` `epsilon = 3`
#   <dbl>        <dbl>        <dbl>        <dbl>
# 1 1.23         0.570       0.598       1.35
#   `epsilon = 3`
#   <dbl>
# 1 9.36
```

2. Usando o pacote **convey**:

```
svygei(~VD5005_def_ano,
       subset(pnadc2019_visita1_dsg,
              pnadc2019_visita1_df$VD5005_def_ano > 0),
       epsilon = -1)
```

POR DENTRO DA PNAD CONTÍNUA

```
svygei(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
       epsilon = 0)  
  
svygei(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
       epsilon = 1)  
  
svygei(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
       epsilon = 2)  
  
svygei(~VD5005_def_ano,  
       subset(pnadc2019_visita1_dsg,  
              pnadc2019_visita1_df$VD5005_def_ano > 0),  
       epsilon = 3)
```

11.3.3 Índice do Nível de Insuficiência Socioeconômica (INIS)

Por fim, nesta seção, pretende-se expor a implementação das metodologias desenvolvidas por Trovão e Dedecca (2017), no caso da Análise do Nível de Insuficiência Socioeconômica (ANIS), e por Trovão e Araújo (2021), para o Índice do Nível de Insuficiência Socioeconômica (INIS).

Para o cálculo desse indicador, são necessárias diversas transformações nos dados da PNAD Contínua. Por esse motivo, alguns passos devem ser seguidos:

1. Calcular os elementos de identificação das condições de insuficiência para cada um dos indicadores-chave que servirão para classificar os domicílios brasileiros em cinco dimensões, como apresentado por seus criadores:

```
pnadc2019_visita1_df <-
  pnadc2019_visita1_df %>%
  mutate(V2005 = as.numeric(V2005)) %>%
  compute(
    T_pobre = ifs(
      V2005 == 1 & VD5005_def_ano >= 0 &
      VD5005_def_ano <= 998/2 ~ 1, TRUE ~ 0),
    T_desemp = ifs(V2005 == 1 & VD4002 == 2 ~ 1,TRUE ~ 0),
    T_contprev = ifs (V2005 == 1 & VD4012 == 2 ~ 1,TRUE ~ 0),
    T_TVéPC = ifs (V2005 == 1 & S01025 == 4 &
      S01028 == 2 ~ 1, TRUE ~ 0),
    T_geladeria = ifs (V2005 == 1 & S01023 == 3 ~ 1,TRUE ~ 0),
    T_maglavar = ifs (V2005 == 1 & S01024 == 2 ~ 1,TRUE ~ 0),
    T_telefone = ifs (V2005 == 1 & S01022 == 2 &
      S01021 == 0 ~ 1, TRUE ~ 0),
    T_aguaenc = ifs (V2005 == 1 & S01007 > 2 ~ 1,TRUE ~ 0),
    T_luz = ifs (V2005 == 1 & S01014 == 2 ~ 1,TRUE ~ 0),
    T_esgoto = ifs (V2005 == 1 & S01012A >= 4 ~ 1,TRUE ~ 0),
    T_aanalf = ifs (V2005 == 1 & V3001 == 2 ~ 1,TRUE ~ 0),
    T_auxescol = ifs (V2009 >= 7 & V2009 <= 17 &
      V3002 == 2 ~ 1, TRUE ~ 0),
    T_ensmed = ifs (VD3004 >= 5 ~ 1, TRUE ~ 0),
    T_idoso = ifs ((V2009 >= 60 & (V5001A == 2 |
      V5004A == 2)) ~ 1, TRUE ~ 0),
    T_idadeativa = ifs (V2009 >= 15 ~ 1, TRUE ~ 0),
    T_crian6anos = ifs (V2009 <= 6 ~ 1, TRUE ~ 0),
    T_Numpes = 1)
```

2. Criar uma base agrupada por domicílios com os elementos de identificação e as variáveis necessárias para a definição do recorte analítico (UF, no caso do presente exemplo), além dos pesos pós-estratificação dos domicílios:

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df_DOM <- pnadc2019_visita1_df %>%
  dplyr::group_by(UPA, Estrato, V1008) %>%
  dplyr::summarise(
    "T_pobre_sum"=sum(T_pobre),
    "T_desemp_sum"=sum(T_desemp),
    "T_contprev_sum"=sum(T_contprev),
    "T_TVéPC_sum"=sum(T_TVéPC),
    "T_geladeria_sum"=sum(T_geladeria),
    "T_maqlavar_sum"=sum(T_maqlavar),
    "T_telefone_sum"=sum(T_telefone),
    "T_aguaenc_sum"=sum(T_aguaenc),
    "T_luz_sum"=sum(T_luz),
    "T_esgoto_sum"=sum(T_esgoto),
    "T_aanalf_sum"=sum(T_aanalf),
    "T_auxescol_sum"=sum(T_auxescol),
    "T_ensmed_sum"=sum(T_ensmed),
    "T_idoso_sum"=sum(T_idoso),
    "T_idadeativa_sum"=sum(T_idadeativa),
    "T_crian6anos_sum"=sum(T_crian6anos),
    "T_Numpes_sum"=sum(T_Numpes),
    "UF_first"=first(UF),
    "V1032_first"=first(V1032),
    "VD5005_def_ano_first"=first(VD5005_def_ano),
    .groups="drop")
```

3. Realizar algumas transformações necessárias para a construção dos indicadores a partir das variáveis criadas na base por domicílios¹⁰:

¹⁰ Basicamente o que se está fazendo é: definir por exclusão os casos em que há presença de pessoas com ensino médio para se chegar no indicador, definido por Trovão e Araújo (2021), “ausência de pessoas com ensino médio completo no domicílio”; e calcular a taxa de atividade dentro dos domicílios e identificar aqueles em que esse valor é inferior a 0.5.

```
pnadc2019_visita1_df_DOM <-
  pnadc2019_visita1_df_DOM %>%
  compute(
    T_ensmed_mod = ifs(T_ensmed_sum > 0 ~ 0,
                        TRUE ~ 1),
    T_atividade = T_idadeativa_sum / T_Numpes_sum,
    T_ativ = ifs(T_atividade < 0.5 ~ 1,
                  TRUE ~ 0))
```

4. Identificar, a partir dos indicadores individualizados por domicílio, em quais dimensões os domicílios apresentam algum nível de insuficiência:

```
pnadc2019_visita1_df_DOM <-
  pnadc2019_visita1_df_DOM %>%
  compute(T_pobre_sum =
  ifs(T_pobre_sum > 0 ~ 1, TRUE ~ 0),
          T_desemp_sum = ifs (T_desemp_sum > 0 ~ 1,TRUE ~ 0),
          T_contprev_sum = ifs (T_contprev_sum > 0 ~ 1,TRUE ~ 0),
          T_TVéPC_sum = ifs (T_TVéPC_sum > 0 ~ 1,TRUE ~ 0),
          T_geladeria_sum = ifs (T_geladeria_sum > 0 ~ 1,TRUE ~ 0),
          T_maqlavar_sum = ifs (T_maqlavar_sum > 0 ~ 1,TRUE ~ 0),
          T_telefone_sum = ifs (T_telefone_sum > 0 ~ 1,TRUE ~ 0),
          T_aguaenc_sum = ifs (T_aguaenc_sum > 0 ~ 1,TRUE ~ 0),
          T_luz_sum = ifs (T_luz_sum > 0 ~ 1, TRUE ~ 0),
          T_esgoto_sum = ifs (T_esgoto_sum > 0 ~ 1,TRUE ~ 0),
          T_aanalf_sum = ifs (T_aanalf_sum > 0 ~ 1,TRUE ~ 0),
          T_auxescol_sum = ifs (T_auxescol_sum > 0 ~ 1,TRUE ~ 0),
          T_ensmed_sum = ifs (T_ensmed_sum > 0 ~ 1,TRUE ~ 0),
          T_idoso_sum = ifs (T_idoso_sum > 0 ~ 1,TRUE ~ 0),
          T_idadeativa_sum = coalesce(T_idadeativa_sum, 0),
          T_Numpes_sum = coalesce(T_Numpes_sum, 0),
          T_crian6anos_sum = ifs (T_crian6anos_sum > 0 ~ 1,TRUE ~ 0))
```

POR DENTRO DA PNAD CONTÍNUA

```
pnadc2019_visita1_df_DOM <-
  pnadc2019_visita1_df_DOM %>%
  compute(D_MTeY = T_pobre_sum + T_desemp_sum + T_contprev_sum,
         D_consumo = T_TVéPC_sum + T_geladeria_sum +
                     T_maqlavar_sum + T_telefone_sum,
         D_condhab = T_aguaenc_sum + T_luz_sum + T_esgoto_sum,
         D_educ = T_aanalf_sum + T_auxescol_sum + T_ensmed_mod,
         D_demogr = T_idoso_sum + T_crian6anos_sum + T_ativ)

pnadc2019_visita1_df_DOM <-
  pnadc2019_visita1_df_DOM %>%
  compute(
    D_MTeY = ifs (D_MTeY > 0 ~ 1, TRUE ~ 0),
    D_consumo = ifs (D_consumo > 0 ~ 1, TRUE ~ 0),
    D_condhab = ifs (D_condhab > 0 ~ 1, TRUE ~ 0),
    D_educ = ifs (D_educ > 0 ~ 1, TRUE ~ 0),
    D_demogr = ifs (D_demogr > 0 ~ 1, TRUE ~ 0))
```

5. Calcular os valores da ANIS (TROVÃO; DEDECCA, 2017), isto é, os valores domiciliares do nível de insuficiência que varia de 0 a 5, em que zero indica o menor nível possível, e 5, o maior nível de insuficiência; e realizar sua transformação em um indicador sintético, definido por Trovão e Araújo (2021), para cada domicílio, ou seja, calcular o INIS domiciliar, que assume valores que variam de 0 a 1¹¹:

```
pnadc2019_visita1_df_DOM <-
  pnadc2019_visita1_df_DOM %>%
  compute(ANIS = D_MTeY + D_consumo + D_condhab +
          D_educ + D_demogr, INIS = ANIS / 5)
```

¹¹ O INIS agregado para subgrupos populacionais, como é o caso do recorte analítico por Unidades da Federação, nada mais é do que a média ponderada pelos pesos amostrais dos INIS domiciliares.

6. Calcular as proporções de domicílios segundo níveis de insuficiência socioeconômica da ANIS:

```
pnadc2019_visita1_df_DOM %>%
  tab_cells(UF_first) %>%
  tab_cols(ANIS, total()) %>%
  tab_weight(weight = V1032_first) %>%
  tab_stat_rpct( total_statistic = "w_rpct",
                 total_label = "Total") %>%
  tab_pivot()
```

7. Calcular o INIS pela média ponderada do INIS domiciliar:

```
expss_digits(digits = 3)

pnadc2019_visita1_df_DOM %>%
  tab_cells(INIS) %>%
  tab_rows(UF_first, total()) %>%
  tab_weight(weight = V1032_first) %>%
  tab_stat_fun(INIS = w_mean) %>%
  tab_pivot()
```

A apresentação dos resultados da proporção de domicílios segundo níveis de insuficiência socioeconômica da ANIS, por Unidades da Federação, pode ser vista na Figura 11.3.

POR DENTRO DA PNAD CONTÍNUA

Figura 11.3: Proporção de domicílios segundo níveis de insuficiência socieconômica por Unidades da Federação do Brasil (2019)

		ANIS	0	1	2	3	4	5	#Total
UF_first	11	9.9	21.7	25.6	23.1	16.1	3.6	100	
	12	5.0	15.9	21.3	24.2	22.6	11.0	100	
	13	11.5	26.5	23.4	17.6	14.0	7.0	100	
	14	11.3	23.6	23.1	23.5	13.5	4.9	100	
	15	5.1	14.8	22.8	25.8	21.9	9.6	100	
	16	9.5	20.7	25.8	21.6	15.2	7.2	100	
	17	10.5	19.1	21.9	24.9	18.2	5.3	100	
	21	4.4	13.7	21.2	30.5	22.7	7.5	100	
	22	5.3	16.4	22.6	32.4	18.8	4.6	100	
	23	8.0	20.2	22.1	26.7	18.1	4.9	100	
	24	7.2	16.1	21.0	23.9	23.9	7.8	100	
	25	7.4	18.9	20.1	26.7	20.8	6.2	100	
	26	8.2	21.4	23.3	25.0	17.0	5.1	100	
	27	7.2	16.0	19.9	27.3	21.6	8.0	100	
	28	9.2	18.9	19.8	25.2	20.1	6.7	100	
	29	8.1	18.3	22.4	25.4	20.8	5.0	100	
	31	17.1	28.7	24.0	18.2	9.8	2.2	100	
	32	18.3	29.6	23.5	18.1	8.7	1.7	100	
	33	20.4	38.0	25.3	12.4	3.6	0.3	100	
	35	24.9	37.1	24.4	10.7	2.6	0.4	100	
	41	20.7	33.3	24.9	14.7	5.8	0.7	100	
	42	26.7	35.1	24.6	10.6	2.7	0.3	100	
	43	22.4	33.3	25.5	13.6	4.5	0.8	100	
	50	16.7	30.7	26.5	17.3	7.8	1.0	100	
	51	15.9	29.9	26.8	17.4	8.5	1.6	100	
	52	16.8	27.2	27.5	18.4	8.4	1.7	100	
	53	30.0	39.0	18.7	9.4	2.5	0.4	100	
	#Total	16.9	28.9	23.9	17.7	10.0	2.7	100	

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

Já os resultados das estimativas pontuais do INIS, por Unidades da Federação, podem ser vistos na Figura 11.4.

Figura 11.4: INIS segundo Unidades da Federação do Brasil (2019)

				#Total
UF_first	11	INIS	INIS	0.449
	12	INIS	INIS	0.553
	13	INIS	INIS	0.434
	14	INIS	INIS	0.438
	15	INIS	INIS	0.547
	16	INIS	INIS	0.468
	17	INIS	INIS	0.474
	21	INIS	INIS	0.552
	22	INIS	INIS	0.513
	23	INIS	INIS	0.483
	24	INIS	INIS	0.529
	25	INIS	INIS	0.506
	26	INIS	INIS	0.473
	27	INIS	INIS	0.529
	28	INIS	INIS	0.496
	29	INIS	INIS	0.495
	31	INIS	INIS	0.363
	32	INIS	INIS	0.349
	33	INIS	INIS	0.284
	35	INIS	INIS	0.260
	41	INIS	INIS	0.307
	42	INIS	INIS	0.257
	43	INIS	INIS	0.294
	50	INIS	INIS	0.343
	51	INIS	INIS	0.355
	52	INIS	INIS	0.359
	53	INIS	INIS	0.233
	#Total	INIS	INIS	0.366

Fonte: Elaboração própria a partir da PNAD Contínua anual de 2019 – visita 1

11.4 Exercícios

1. Calcule todas as medidas de subutilização da força de trabalho por Regiões Geográficas montando uma série de dados trimestrais para os anos de 2019 e 2020.
2. Calcule a proporção de domicílios, com base nos dados anuais da primeira entrevista, segundo as características da habitação (água encanada, coleta de lixo, energia elétrica), dimensões individuais da pessoa de referência do domicílio associadas ao sexo, à cor/raça e à faixa de idade, e a renda (faixas de renda domiciliar *per capita*).

POR DENTRO DA PNAD CONTÍNUA

3. Calcule todas as medidas de pobreza monetária para os dados da PNAD Contínua trimestral ao longo dos anos 2019 e 2020, para verificar o impacto da pandemia da COVID-19. Vale destacar que essas medidas ficarão restritas aos rendimentos oriundos do trabalho.
4. Calcule os índices IPM e INIS para as Regiões Geográficas para os anos de 2016 a 2020.
5. Calcule os índices de concentração de renda, discutidos no presente capítulo, para o rendimento real efetivo domiciliar *per capita* por Unidades da Federação para os anos de 2016 a 2020.

Referências Bibliográficas

AQUINO, J. *Descr*: Descriptive statistics. CRAN, 2021. R package version 1.1.5. Disponível em: <https://cran.r-project.org/web/packages/descr/index.html>. Acesso em: 15 jun. 2022.

ATKINSON, A. B. *et al.* On the measurement of inequality. *Journal of economic theory*, v. 2, n. 3, p. 244–263, 1970. Disponível em: [https://doi.org/10.1016/0022-0531\(70\)90039-6](https://doi.org/10.1016/0022-0531(70)90039-6). Acesso em: 23 maio 2022.

AUGUIE, B. *GridExtra*: Miscellaneous functions for “Grid” Graphics. CRAN, 2017. R package version 2.3. Disponível em: <https://cran.r-project.org/web/packages/gridExtra/index.html>. Acesso em: 23 maio 2022.

BACHE, S. M.; WICKHAM, H. *magrittr*: A forward-pipe operator for R. CRAN, 2020. R package version 2.0.1. Disponível em: <https://cran.r-project.org/web/packages/magrittr/index.html>. Acesso em: 23 maio 2022.

BADDELEY, A.; TURNER, R. Spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, v. 12, n. 6, p. 1–42, 2005. Disponível em: <https://www.jstatsoft.org/v12/i06/>. Acesso em: 07 jul. 2022.

BAUER, R. A. Social indicators and sample surveys. *Public Opinion Quarterly*, v. 30, n. 3, p. 339–352, jan. 1966. ISSN 0033–362X. Disponível em: <https://doi.org/10.1086/267428>. Acesso em: 07 jul. 2022.

BRAGA, D.; ASSUNCAO, G. *PNADCIBGE*: Downloading, reading and analysing PNADC microdata. CRAN, 2020. R package version 0.6.0. Disponível em: <https://cran.r-project.org/web/packages/PNADCIBGE/index.html>. Acesso em: 23 maio 2022.

BRANDÃO, G. S. *Pobreza multidimensional no Nordeste*: uma análise a partir dos dados da PNAD Contínua (2016–2019). Dissertação (Mestrado em Economia) — Centro de Ciências Sociais Aplicadas, Universidade Federal do Rio Grande do Norte, 2021. Disponível em: <https://repositorio.ufrn.br/handle/123456789/33022>. Acesso em: 28 abr. 2022.

POR DENTRO DA PNAD CONTÍNUA

DAMICO, A. J. *SASci*: Import ASCII files directly into R using only a SAS input script. CRAN, 2012. R package version 1.0. Disponível em: <https://cran.r-project.org/web/packages/SASci/index.html>. Acesso em: 23 maio 2022.

DAMICO, A. J. *Analyze Survey Data for Free*: Step by step instructions to explore public microdata from an easy to type website. E-Book, 2019. Disponível em: <http://asdfree.com/>. Acesso em: 23 maio 2022.

DAMICO, A. J. *Lodown*: locally download and prepare publicly-available microdata. GitHub, 2022. R package version 0.1.0. Disponível em: <https://github.com/ajdamico/lodown>. Acesso em: 23 maio 2022.

DEATON, A. *The analysis of household surveys*: a microeconometric approach to development policy. Washington, DC: World Bank Publications, 1997. Disponível em: <https://documents1.worldbank.org/curated/en/593871468777303124/pdf/17140-PUB-revised-PUBLIC-9781464813313-Updated.pdf>. Acesso em: 15 jun. 2022.

DEDECCA, C. S. *Políticas públicas e trabalho*: textos para estudo dirigido. Campinas: Instituto de Economia – Unicamp, 2006.

DEMIN, G. *Expss*: Tables, labels and some useful functions from spreadsheets and 'SPSS' Statistics. CRAN, 2020. R package version 0.10.7. Disponível em: <https://cran.r-project.org/web/packages/expss/index.html>. Acesso em: 23 maio 2022.

DOWLE, M.; SRINIVASAN, A. *Data.table*: Extension of 'data.frame'. CRAN, 2021. R package version 1.14.0. Disponível em: <https://cran.r-project.org/web/packages/data.table/index.html>. Acesso em: 23 maio 2022.

FOSTER, J.; GREER, J.; THORBECKE, E. A class of decomposable poverty measures. *Econometrica: journal of the econometric society*, v. 52, n. 3, p. 761–766, may 1984. Disponível em: <https://www.jstor.org/stable/1913475>. Acesso em: 23 maio 2022.

GINI, C. On the measure of concentration with special reference to income and statistics. *Colorado College Publication*, Colorado Springs, v. 208, n. 1, p. 73–79, 1936.

HUGH-JONES, D. *Huxtable*: Easily create and style tables for LaTeX, HTML and other formats. CRAN, 2021. R package version 5.4.0. Disponível em: <https://cran.r-project.org/web/packages/huxtable/index.html>. Acesso em: 23 maio 2022.

IBGE. *Desemprego*. Portal do IBGE, 2010. Disponível em: <https://www.ibge.gov.br/explica/desemprego.php>. Acesso em: 02 maio 2022.

IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua*: Nota técnica 02/2016: Medidas de subutilização da força de trabalho. 4. ed. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2016. Disponível em: https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Nota_Tecnica/Nota_Tecnica_022016.pdf. Acesso em: 30 set. 2020.

IBGE. *Matriz de insumo–produto*: Brasil: 2015. 62. ed. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, Coordenação de Contas Nacionais, 2018. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101604.pdf>. Acesso em: 31 out. 2021.

IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua*: Notas técnicas – versão 1.5. 4. ed. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2019. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101651_notas_tecnicas.pdf. Acesso em: 15 jul. 2020.

IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua*: Quarto trimestre de 2020 – indicadores IBGE. 4. ed. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2020a. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/2421/pnact_2020_4tri.pdf. Acesso em: 28 out. 2021.

IBGE. *Síntese de Indicadores Sociais*: Uma análise das condições de vida da população brasileira. 4. ed. Rio de Janeiro: Coordenação de População e Indicadores Sociais, 2020b. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101760.pdf>. Acesso em: 28 out. 2021.

IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua*: Notas técnicas – versão 1.8. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2021. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101733_notas_tecnicas.pdf. Acesso em: 28 out. 2021.

IBGE. *Pesquisa Nacional por Amostra de Domicílios Contínua*: O que é. Rio de Janeiro: Centro de Documentação e Disseminação de Informações. Fundação Instituto Brasileiro de Geografia e Estatística, 2021a. Disponível em: <https://www.ibge.gov.br/estatisticas/sociais/populacao/9173-pesquisa-nacional-por-amostra-de-domicilios-continua-trimestral.html?=&t=o-que-e>. Acesso em: 26 out. 2021.

ILO. Resolution I: Resolution concerning statistics of work, employment and labour underutilization. In: INTERNATIONAL CONFERENCE OF LABOUR STATISTICIANS, 19., Geneve, 2013. *Proceedings* [...]. Geneve: ICLS, 2013. Disponível em:

POR DENTRO DA PNAD CONTÍNUA

https://www.ilo.org/wcmsp5/groups/public/---dgreports/---stat/documents/normativeinstrument/wcms_230304.pdf. Acesso em: 20 maio 2020.

JACOB, G.; DAMICO, G.; PESSOA, D. *Poverty and Inequality with Complex Survey Data*: Auxiliary information about the usage of the R convey package. 2022. Disponível em: <https://guilhermejacob.github.io/context/>. Acesso em: 23 maio 2022.

LORENZ, M. O. Methods of measuring the concentration of wealth. *Publications of the American statistical association*, Taylor and Francis, v. 9, n. 70, p. 209–219, 1905. Disponível em: <https://www.jstor.org/stable/2276207>. Acesso em: 23 maio 2022.

LUMLEY, T. Analysis of complex survey samples. *Journal of Statistical Software*, v. 9, n. 1, p. 1–19, 2004. Disponível em: <https://doi.org/10.18637/jss.v009.i08>. Acesso em: 15 jun. 2022.

LUMLEY, T. *Survey*: analysis of complex survey samples. Auckland, NZ, 2019. R package version 3.35–1. Disponível em: <http://r-survey.r-forge.r-project.org/survey/>. Acesso em: 23 maio 2022.

MEDEIROS, M. *Medidas de desigualdade e pobreza*. Brasília, DF: Editora Universidade de Brasília, 2012.

MORETTIN, P. A.; BUSSAB, W. O. *Estatística básica*. 6. ed. São Paulo: Saraiva Educação SA, 2017.

NIJS, V. *radiant*: Business analytics using R and Shiny. CRAN, 2021. R package version 1.4.0. Disponível em: <https://cran.r-project.org/web/packages/radiant/index.html>. Acesso em: 23 maio 2022.

OSBERG, L.; SHARPE, A. An index of economic well-being for selected OECD countries. *Review of Income and Wealth*, v. 48, n. 3, p. 291–316, 2002. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1475-4991.00056>. Acesso em: 23 maio 2022.

OSBERG, L.; SHARPE, A. Changing Trends in Economic Well-being in OECD Countries: What Measure is Most Relevant for Health? In: HEYMANN, J. et al. (ed.) *Healthier Societies*: From Analysis to Action. New York: Oxford University Press, 2006. Disponível em: <https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780195179200.001.0001/acprof-9780195179200-chapter-12>. Acesso em: 23 maio 2022.

OSBERG, L.; SHARPE, A. Moving from a GDP-Based to a Well-Being Based Metric of Economic Performance and Social Progress: Results from the Index of

Economic Well-Being for OECD Countries, 1980–2009. *CSLS Research Reports*, 2011. Disponível em: <https://ideas.repec.org/p/sls/resrep/1112.html>. Acesso em: 23 maio 2022.

PEDERSEN, T. L. *Patchwork*: The composer of plots. CRAN, 2020. R package version 1.1.1. Disponível em: <https://cran.r-project.org/web/packages/patchwork/index.html>. Acesso em: 23 maio 2022.

PESSOA, D.; DAMICO, A.; JACOB, G. *Convey*: Estimation of indicators on social exclusion and poverty and its linearization, variance estimation. GitHub: [s.n.], 2021. R package version 0.2.4. Disponível em: <https://github.com/ajdamico/convey/>. Acesso em: 23 maio 2022.

PLAT, D. *IC2*: Inequality and concentration indices and curves. CRAN, 2012. R package version 1.0–1. Disponível em: <https://CRAN.R-project.org/package=IC2>. Acesso em: 1 dez. 2020.

RSTUDIO. *Data transformation with dplyr*: Cheet sheet. [S.l.], 2021a. Disponível em: <https://github.com/rstudio/cheatsheets/blob/main/data-transformation.pdf>. Acesso em: 27 jun. 2022.

RSTUDIO. *Data visualization with ggplot2*: Cheet sheet. [S.l.], 2021b. Disponível em: <https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf>. Acesso em: 27 jun. 2022.

RStudio Team. *RStudio*: Integrated development environment for R. Boston, MA, 2021. Disponível em: <http://www.rstudio.com/>. Acesso em: 27 maio 2022.

SCHAUBERGER, P.; WALKER, A. *Openxlsx*: Read, write and edit xlsx files. CRAN, 2021. R package version 4.2.4. Disponível em: <https://cran.r-project.org/web/packages/openxlsx/index.html>. Acesso em: 23 maio 2022.

SCHULENBERG, R. *Dineq*: Decomposition of (income) inequality. CRAN, 2018. R package version 0.1.0. Disponível em: <https://cran.r-project.org/web/packages/dineq/index.html>. Acesso em: 23 maio 2022.

SEN, A. *Development as Freedom*. New York: Alfred Knopf, 1999.

STERN, J. M. *et al.* Otimização e processos estocásticos aplicados à economia e finanças. *CoRR*, arXiv:2005.13459, 2020. Disponível em: <https://arxiv.org/abs/2005.13459>. Acesso em: 23 maio 2022.

TROVÃO, C. J. B. M. *Desigualdade multidimensional*: uma abordagem keynesiana para o seu enfrentamento. Tese (Doutorado em Desenvolvimento Econômico) — Instituto de Economia – IE, Universidade Estadual de Campinas, Campinas, São

POR DENTRO DA PNAD CONTÍNUA

Paulo, 2015. Disponível em: <https://hdl.handle.net/20.500.12733/1627579>. Acesso em: 16 jun. 2022.

TROVÃO, C. J. B. M.; ARAÚJO, J. B. de. Desigualdade multidimensional, insuficiência socioeconômica e concentração de renda no Brasil a partir de um olhar macrorregional. *Desenvolvimento em Debate*, v. 9, n. 1, p. 121–157, 2021. Disponível em: <https://revistas.ufrj.br/index.php/dd/article/view/39632>. Acesso em: 15 jun. 2022.

TROVÃO, C. J. B. M.; DEDECCA, C. S. Análise do Nível de Insuficiência Socioeconômica (ANIS): uma avaliação do Brasil entre 2000 e 2010. *Revista Argumentos*, v. 14, n. 1, p. 217–248, 2017. Disponível em: <https://www.periodicos.unimontes.br/index.php/argumentos/article/view/1164>. Acesso em: 15 jun. 2022.

UNDP. *Human Development Report 1994*. New York: United Nations Development Programme (UNDP), 1994. 137 p. Disponível em: https://hdr.undp.org/sites/default/files/reports/255/hdr_1994_en_complete_nostats.pdf. Acesso em: 23 maio 2022.

UNDP; OPHI. *Global Multidimensional Poverty index 2020 – Charting Pathways out of Multidimensional Poverty*: Achieving the sdgs. Oxford, UK: UNDP: OPHI, 2020. Acesso em: 23 maio 2022.

VENABLES, W. N. *et al.* *An introduction to R*: Notes on r: A programming environment for data analysis and graphics. Vienna, Austria: Citeseer, 2009. Disponível em: <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>. Acesso em: 23 maio 2022.

WICKHAM, H. *Stringr*: Simple, consistent wrappers for common string operations. CRAN, 2019. R package version 1.4.0. Disponível em: <https://cran.r-project.org/web/packages/stringr/index.html>. Acesso em: 23 maio 2022.

WICKHAM, H. *Tidyr*: Tidy messy data. CRAN, 2021. R package version 1.1.3. Disponível em: <https://cran.r-project.org/web/packages/tidyr/index.html>. Acesso em: 23 maio 2022.

WICKHAM, H. *et al.* Welcome to the tidyverse. *Journal of Open Source Software*, v. 4, n. 43, p. 1686, 2019.

WICKHAM, H. *et al.* *Dplyr*: A grammar of data manipulation. CRAN, 2021. R package version 1.0.7. Disponível em: <https://cran.r-project.org/web/packages/dplyr/index.html>. Acesso em: 23 maio 2022.

WICKHAM, H.; GROLEMUND, G. *R for data science*: import, tidy, transform, visualize, and model data. Newton, MA: "O'Reilly Media", 2016.

Cassiano José Bezerra Marques Trovão
Antonio Hermes Marques da Silva Júnior

WICKHAM, H.; HESTER, J. *Readr*: Read rectangular text data. CRAN, 2020. R package version 1.4.0. Disponível em: <https://cran.r-project.org/web/packages/readr/index.html>. Acesso em: 23 maio 2022.

WICKHAM, H. *et al.* *Ggplot2*: Elegant graphics for data analysis. New York: Springer–Verlag, 2016. ISBN 978-3-319-24277-4. Disponível em: <https://ggplot2.tidyverse.org>. Acesso em: 23 maio 2022.

WICKHAM, H.; SEIDEL, D. *Scales*: Scale functions for visualization. CRAN, 2020. R package version 1.1.1. Disponível em: <https://cran.r-project.org/web/packages/scales/index.html>. Acesso em: 23 maio 2022.

WILKINS, D. *Treemapify*: Draw treemaps in 'ggplot2'. CRAN, 2021. R package version 2.5.5. Disponível em: <https://cran.r-project.org/web/packages/treemapify/index.html>. Acesso em: 23 maio 2022.

WILKINSON, L. *The grammar of graphics*. Berlin: Springer Science & Business Media, 2013. Disponível em: <https://doi.org/10.1007/0-387-28695-0>. Acesso em: 20 maio 2021.

XIE, Y. knitr: A comprehensive tool for reproducible research in R. In: STODDEN, V.; LEISCH, F.; PENG, R. D. (ed.) *Implementing Reproducible Computational Research*. London: Chapman and Hall/CRC, 2014. ISBN 978-1466561595. Disponível em: <https://www.routledge.com/Implementing-Reproducible-Research/Stodden-Leisch-Peng/p/book/9781466561595>. Acesso em: 23 maio 2022.

Listas de Figuras

3.1	Divisões da população e da força de trabalho	37
5.1	Tela inicial do R “dentro” do <i>RStudio</i>	58
5.2	Identificação dos painéis do R “dentro” do <i>RStudio</i>	64
6.1	Dicionário original .xls (à esquerda) e arquivo .csv alterado (à direita), necessário para a abertura das bases da PNAD anual que apresentem sobreposição de colunas/variáveis.	131
7.1	Folha de cola para o tratamento de dados – dplyr (parte 1).	162
7.2	Folha de cola para o tratamento de dados – dplyr (parte 2).	163
9.1	Distintas opções de visualização de tabelas com o pacote expss	221
9.2	Apresentação da tabela armazenada no objeto <code>tabela_exemplo</code> por meio do openxlsx	224
10.1	Tipos de gráficos para representar a distribuição de uma única variável contínua (VD5005).	234
10.2	Histograma e polígono de frequência para a variável (VD5005) com faixas de largura de tamanho igual ao valor do salário-mínimo de 2019 (R\$ 998,00).	236
10.3	Gráficos de barras com a contagem (amostral e populacional) para uma única variável discreta (V2007).	237
10.4	Gráficos de dispersão e curva suavizada para as variáveis VD401 e VD4017	240
10.5	Gráficos para análise de variáveis contínuas condicionadas a variáveis discretas.	242
10.6	Gráfico de contagem para duas variáveis discretas.	244

10.7	Gráfico de dispersão para as variáveis VD4016 e VD4017, subdivididas por sexo (V2007)	246
10.8	Gráfico de dispersão para as variáveis VD4016 e VD4017 (alterando a cor dos pontos para azul)	247
10.9	Gráficos de dispersão para as variáveis VD4016 e VD4017 (divisão por subgrupos baseados em categorias de variáveis discretas)	250
10.10	Gráficos apresentados na forma de uma matriz 2x2 para as variáveis sexo (V2007) e rendimento efetivo no trabalho principal (VD4017), apenas para valores inferiores a R\$ 10.000,00.	253
10.11	Histograma e polígono de frequência para a variável VD4017 com mapeamento estético de cor e preenchimento por sexo (V2007)	255
10.12	Gráficos de linhas para as séries temporais da taxa de desocupação e da taxa composta de subutilização da força de trabalho.	258
10.13	Gráficos de pizza, barras empilhadas e árvore para análise de frequências relativas ou participação (em %)	263
10.14	Histogramas básicos para a variável VD5005 e com alteração nos rótulos dos eixos	265
10.15	Histograma básico para a variável VD5005 com o eixo x limitado a R\$ 2.000,00	266
10.16	Histograma básico para a variável VD5005 com alteração nas escalas dos eixos	268
10.17	Gráfico de barras com a contagem da população segundo cor/raça (V2010)	270
10.18	Gráficos de densidade de Kernel para as variáveis VD5005 e V2010 com modificações estéticas	272
10.19	Gráficos com elementos gráficos de linha e área rotulados	275
10.20	Gráfico final de densidade de Kernel para as variáveis VD5005 e V2010	278
10.21	Folha de cola para gramática de gráficos – ggplot2 (parte 1)	280
10.22	Folha de cola para gramática de gráficos – ggplot2 (parte 2)	281
11.1	IPM e IEPM para as Unidades da Federação do Brasil (2019)	314
11.2	Curvas de Lorenz criadas a partir dos pacote IC2 e convey	317
11.3	Proporção de domicílios segundo níveis de insuficiência socieconômica por Unidades da Federação do Brasil (2019)	327
11.4	INIS segundo Unidades da Federação do Brasil (2019)	328

Lista de Tabelas

7.1 Descrição de variáveis da PNAD Contínua trimestral.	138
9.1 Rendimento médio efetivo domiciliar <i>per capita</i> por Região Geográfica e sexo no Brasil (2019)	227
11.1 Descrição dos indicadores de mercado de trabalho	288