



SÉRIE DE NOTAS TÉCNICAS EM ECONOMIA DA UFG
NT N. 13

NOTA TÉCNICA EM ECONOMIA
n. 13

Publicação cujo objetivo é auxiliar na elaboração de aulas e de pesquisas do Curso de Ciências Econômicas da UFG, divulgando demonstrações técnicas e metodológicas ou rotinas computacionais voltadas para a economia.

As opiniões contidas nesta publicação não representam o ponto de vista do Curso ou da FACE/UFG, sendo de inteira responsabilidade de seu(s) autor(es).

Reprodução permitida, desde que citada a fonte.

FACE – Faculdade de Administração, Ciências Contábeis e Ciências Econômicas
Curso de Ciências Econômicas
Direção FACE
Profª. Andrea Freire de Lucena
Vice-Direção FACE
Profª. Daiana Paula Pimenta
Coordenação do Curso de Ciências Econômicas
Prof. Everton Sotto Tibiriçá Rosa

Introdução aos pacotes
“PNADcIBGE”, “Convey” e
“Survey” no Software R

Guilherme Brito Brandão Ferreira de
Deus
PPGECON/FACE/UFG
Heris Coutinho Vieira
PPGECON/FACE/UFG
Tatiane Ferreira do Nascimento Melo
PPGECON e IME/UFG

FACE/UFG
Goiânia – Dezembro de 2023
Versão 1.0

Endereço

Campus Samambaia, Prédio da FACE – Rodovia
Goiânia/Nova Veneza, km. 0, CEP 74690-900,
Goiânia – GO. Tel. (62) 3521 – 1390

URL

<http://www.eco.face.ufg.br>

CIÊNCIAS
ECONÔMICAS

FACE
FACULDADE DE
ADMINISTRAÇÃO, CIÊNCIAS CONTÁBEIS E
CIÊNCIAS ECONÔMICAS



Dados Internacionais de Catalogação na Publicação (CIP)
GPT/BC/UFG

Ferreira de Deus, Guilherme Brito Brandão. Introdução aos pacotes “PNADcIBGE”, “Convey” e “Survey” no Software R / Vieira, Heris Coutinho. Melo, Tatiane Ferreira do Nascimento. - 2023.

13 f.: (Série de Notas Técnicas em Economia, 013)

Universidade Federal de Goiás, Goiânia, 2023. Bibliografia.

Anexos. 1. Software R. 2. Convey. 3. Dados. I. Título.

Introdução aos pacotes “PNADcIBGE”, “Convey” e “Survey” no Software R

Guilherme Brito Brandão Ferreira de Deus¹

PPGECON/FACE/UFG

Heris Coutinho Vieira²

PPGECON/FACE/UFG

Tatiane Ferreira do Nascimento Melo³

PPGECON e IME/UFG

Resumo: O objetivo, nesta nota técnica, é apresentar e aplicar os pacotes PNADcIBGE, convey e survey no software estatístico R (R Core Team, 2023). Primeiramente, conjuntos de dados da Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD contínua) do Instituto Brasileiro de Geografia e Estatística (IBGE) são importados, usando o pacote PNADcIBGE e depois esses dados são processados e aplicados com os pacotes convey e survey. Para ilustrar o uso destes pacotes serão utilizados dois conjuntos de dados, a saber: Índice de Gini e Taxa de desemprego por Unidade da Federação (UF) no quarto trimestre do ano de 2022.

Palavras chave: Índice de Gini, pacote PNADcIBGE, PNAD contínua, Taxa de Desemprego

Abstract: *This technical note aims to explain how to import data from the Continuous National Household Sample Survey (Continuous PNAD) from the Brazilian Institute of Geography and Statistics (IBGE) with the “PNADcIBGE” package into the R statistical software. will be about the Gini index and the general unemployment rate in Brazil in the fourth quarter of 2022.*

Key words: Gini Index, PNADcIBGE package, continuous PNAD, Unemployment Rate

¹ guilhermebritobrandao@gmail.com

² heris.coutim@hotmail.com

³ tmelo@ufg.br

1. Introdução

Esta nota técnica tem a finalidade de importar conjuntos de dados da PNAD Contínua usando o pacote PNADcIBGE e como utilizar os mesmos com os pacotes survey e convey. Dessa forma, será realizada uma análise de regressão linear simples, explicando os comandos de importação dos microdados, disponíveis na pesquisa, para o software *R*. A PNAD Contínua tem microdados anuais e trimestrais e o pacote PNADcIBGE permite importar os dois tipos, contudo, esta nota terá como foco os dados trimestrais da base. O exemplo considerado, nesta nota, é de caráter ilustrativo e serve para mostrar como utilizar as funções e pacotes. Usando os pacotes e seus respectivos comandos, extraímos da base de dados da PNAD contínua uma estimativa do Índice de Gini e da taxa de desemprego por UF do quarto trimestre de 2022, para a análise de regressão supracitada.

Os pacotes utilizados possuem diversas funções, nesta nota técnica destacamos algumas. A principal função do pacote **PNADcIBGE** é: (i) **get_pnadc()**: importa os dados da PNAD contínua. As mais importantes do **convey** são: (ii) **prep_convey()**: prepara os dados importados e (iii) **svygini()**: estima o coeficiente de Gini. Por último, a função do pacote **survey** que será trabalhada é: (iv) **svyby()**: calcula estatísticas amostrais de uma base de dados e (v) **svytotal()**: computa quantidades totais de dados. No final desta nota, faremos uma regressão linear simples com os dados importados. Mais detalhes sobre o pacote PNADcIBGE e seus respectivos comandos, veja Pedroza Jr. (2021), Trovão e Silva Jr. (2022) e Assunção (2023).

2. Passos Iniciais

2.1. Instalar o pacote PNADcIBGE

Primeiramente, utilizamos o comando `install.packages()` para instalar o pacote PNADcIBGE, conforme script abaixo:

```
# Instalar pacotes
rm(list=ls()) # remover objetos
install.packages("PNADcIBGE") # Instalação do pacote PNADcIBGE
```

2.2. Carregar o pacote PNADcIBGE

Após instalar um pacote, o usuário deve carregá-lo para executar os seus comandos. Ele não fica automaticamente na memória e deve ser carregado toda vez que os comandos que pertencem ao

pacote precisarem ser executados. Utilize o comando `library()` e indique o pacote desejado conforme abaixo:

```
#Carregar o pacote  
library(PNADcIBGE)
```

Se o pacote não for carregado, não será possível refazer os exemplos nesta nota técnica.

3. Importação dos dados online através do comando “get_pnadc”

Nesta seção, será apresentado como importar os dados da PNAD contínua usando o comando `get_pnadc`. Nenhum *download* prévio da base de dados é necessário, o comando faz a importação online, ou seja, não funcionará sem conexão com a internet.

3.1. Importando dados trimestrais (Argumento “year”, “quarter”)

Para importar os dados do ano e trimestre desejados, usamos o comando indicando as opções `year` e `quarter`. Neste caso, se o objetivo for importar dados do quarto trimestre de 2022, temos

```
#Importando os dados trimestrais  
dadosPNADc<- get_pnadc(year = 2022, quarter = 4)
```

Dessa maneira, o comando baixa todas as variáveis da PNAD contínua daquele ano e trimestre. Atente-se que o *download* pode ser demorado, pois a base de dados é muito grande. Se o usuário não for trabalhar com todas as variáveis, seria interessante selecionar aquelas que lhes são importantes.

3.1.1. Selecionando variáveis desejadas (Argumento “vars”)

Como exemplo, selecionamos três variáveis principais, a saber: (i) Condição de ocupação na semana de referência para pessoas de 14 anos ou mais de idade (VD4002), (ii) Rendimento efetivamente recebido em 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) e (iii) Condição em relação à força de trabalho na semana de referência para pessoas de 14 anos ou mais de idade (VD4001). Entre parênteses indicamos os códigos das variáveis. Aqui, também incluímos a variável Unidade de Federação (UF), já que, para ilustrar o uso dos

pacotes, queremos dados estaduais. Além disso, para a análise, serão usados dados do quarto trimestre de 2022. As opções `year` e `quarter` são mantidas, mas também adicionamos a opção `vars` e indicamos os códigos das respectivas variáveis. Observe o script abaixo:

```
#Importando as variáveis desejadas
dadosPNADc<-
get_pnadc(year=2022,quarter=4,vars=c("UF","VD4002","VD4001","VD4020"))
```

Assim, o comando só baixará as variáveis que foram indicadas pelo usuário. Observe que é importante que o usuário saiba o código das variáveis, isso pode ser encontrado no site do IBGE na seção de microdados da PNAD Contínua. Acesse o link (<https://www.ibge.gov.br/estatisticas/sociais/trabalho/9173-pesquisa-nacional-por-amostra-de-domicilios-continua-trimestral.html?edicao=36938&t=microdados>), selecione a categoria desejada (microdados anuais ou trimestrais) e faça o download do documento que define as variáveis.

4. Análise com os pacotes `survey` e `convey`

O objeto criado pela função `PNADcIBGE` será um `"survey.design"` ou `"svyrep.design"`, eles são objetos usados para análise com o pacote `survey`. O comando `class()` nos retorna o objeto criado pela função:

```
#Ver a classe dos dados
class(dadosPNADc)
```

O resultado exibido:

```
[1] "svyrep.design"
```

Portanto, o `PNADcIBGE` no *R* é utilizado para importar os dados e a análise é realizada usando outros pacotes. Inicialmente, mostraremos como instalar e carregar os pacotes `survey` e `convey`. Em seguida, usaremos o pacote `convey`, pois ele tem a função `svygini()` que estima o índice de Gini e o pacote `survey` para obtermos os dados relacionados ao desempregos, a nível de estado.

4.1. Instalar e carregar os pacotes Survey e Convey

As funções utilizadas para instalar e carregar serão as mesmas, usaremos o comando `install.packages()` para instalar os pacotes e o `library()` para carregar o pacote, conforme o seguinte script:

```
#Instalando e carregando os pacotes survey e convey
install.packages("survey")
install.packages("convey")
library(survey)
library(convey)
```

4.2. Preparando os dados obtidos

Na seção três, obtivemos os dados com o pacote PNADcIBGE, porém, é necessário prepará-los para que possam ser aplicadas as funções do pacote convey. A preparação é simples, basta usar o comando `convey_prep()` e informar qual a base de dados que será preparada como de acordo com o script abaixo:

```
#Preparando os dados
dadosPNADc <- convey_prep(design=dadosPNADc)
```

Se os dados não sejam preparados, não será possível usar as funções do pacote convey.

4.3. Índice de Gini

O índice de Gini será estimado com as funções `svyby()` e `svygini()` do pacote `survey` e `convey`, respectivamente. A função `svygini()` é a que estima o coeficiente de Gini, porém, só serviria para calcular o coeficiente geral do Brasil. Como queremos calcular por UF, devemos usar a `svyby()` e indicar no argumento `FUN` qual função ela deve aplicar. Além da função `svygini()`, também devemos indicar uma variável de renda, neste caso, consideramos a VD4020. Adicionalmente, devemos apontar que será por UF e informar qual será o conjunto de dados usado pela função. Usaremos o conjunto de dados que foi preparado na subseção anterior e o argumento `na.rm=TRUE` omite os valores ausentes da variável. Por fim, vamos renomear a segunda e a terceira colunas para facilitar o entendimento do objeto criado. Segue o script abaixo:

```
#Estimando o índice de Gini
giniUF <- svyby(formula=~VD4020, by=~UF, design=dadosPNADc, FUN=svygini,
na.rm=TRUE) #Estimando o índice
colnames(giniUF)[2] <- "gini" #Renomeando a segunda coluna
colnames(giniUF)[3] <- "erro padrão" #Renomeando a terceira coluna
giniuf #Exibindo o objeto criado
```

O *data frame* criado será exibido conforme abaixo:

	UF	gini	erro padrão
Rondônia	Rondônia	0.4379337	0.012049294
Acre	Acre	0.4635926	0.010672074
Amazonas	Amazonas	0.5050460	0.012376445
Roraima	Roraima	0.5212136	0.020711239
Pará	Pará	0.5469579	0.011739516
Amapá	Amapá	0.5010884	0.014483361
Tocantins	Tocantins	0.4996150	0.011144626
Maranhão	Maranhão	0.5123910	0.012036913
Piauí	Piauí	0.5813827	0.022906859
Ceará	Ceará	0.5209561	0.016758802
Rio Grande do Norte	Rio Grande do Norte	0.5346413	0.021572553
Paraíba	Paraíba	0.5508491	0.020110133
Pernambuco	Pernambuco	0.5036104	0.012651360
Alagoas	Alagoas	0.4644419	0.015471983
Sergipe	Sergipe	0.5569250	0.021540649
Bahia	Bahia	0.5279421	0.012804452
Minas Gerais	Minas Gerais	0.4673820	0.008019685
Espírito Santo	Espírito Santo	0.4957296	0.011235141
Rio de Janeiro	Rio de Janeiro	0.5264051	0.006729311
São Paulo	São Paulo	0.4957087	0.008558828
Paraná	Paraná	0.4489797	0.007478786
Santa Catarina	Santa Catarina	0.4049662	0.005869953
Rio Grande do Sul	Rio Grande do Sul	0.4893804	0.007471994
Mato Grosso do Sul	Mato Grosso do Sul	0.4928539	0.020197697
Mato Grosso	Mato Grosso	0.4440160	0.014103796
Goiás	Goiás	0.4709154	0.014012555
Distrito Federal	Distrito Federal	0.5701907	0.011956530

4.4. Taxa de desemprego

A taxa de desemprego por UF será obtida com a função `svyby()` do pacote `survey`. Porém, será necessário obter a quantidade de pessoas desocupadas e a quantidade na força de trabalho em cada estado. Em seguida, dividiremos o primeiro pelo segundo para encontrar a taxa de desemprego estadual. Para obter o total de desempregados usaremos a variável Condição de ocupação na semana de referência para pessoas de 14 anos ou mais de idade (VD4002). Esta variável é dividida em duas categorias: pessoas ocupadas e pessoas desocupadas. Portanto, indicaremos no argumento que buscamos as observações de pessoas desocupadas. Ademais, denotamos o ordenamento por UF, a base de dados usada, a função aplicada para obtermos as quantidades totais, que será a `svytotal()`, e novamente removendo observações ausentes.

Assim, como o comando `get_pnadc()`, o objeto retornado é um “`svyrep.design`” e vamos converter para um *data frame*. Segue o comando abaixo:

```
#Quantidade de pessoas desocupadas por UF
desocupUF <- svyby(~VD4002 == "Pessoas desocupadas", ~UF , dadosPNADc,
svytotal, na.rm = T) #Pessoas desocupadas por UF
desuf<- as.data.frame(desocupUF) #Transformando em dataframe
```

A coluna com a quantidade de pessoas desocupadas é a terceira:

```
#Mostrando a terceira coluna
desuf[,3]
```

```
[1] 26131.20 35272.07 196249.68 12320.26 347082.76 57403.30
[7] 41069.32 235890.39 134026.41 313248.43 152520.30 173669.64
[13] 524951.53 128152.69 130826.17 945131.07 648036.18 155823.78
[19] 1027361.25 1985616.18 318451.92 132704.53 289399.29 49808.45
[25] 64345.97 260326.72 186015.10
```

Por enquanto, separamos o *data frame* criado e a coluna desejada. Em seguida, encontramos a quantidade de pessoas na força de trabalho em cada UF usando a variável VD4001. Faremos um processo semelhante ao anterior:

```
#Quantidade de pessoas na força de trabalho por UF
pessoasftUF <- svyby(~VD4001 == "Pessoas na força de trabalho",~UF ,
dadosPNADc, svytotal, na.rm = T) #Pessoas na força de trabalho por UF
pftuf<-as.data.frame(pessoasftUF) #Transformando em data frame
```

A coluna com a quantidade de pessoas na força de trabalho também é a terceira:

```
#Mostrando a terceira coluna
pftuf[,3]
```

```
[1] 855649.3 353637.3 1956929.3 266522.3 4214802.2 430903.3
[7] 793723.2 2825086.6 1417975.4 4019783.7 1536166.7 1688944.6
[13] 4262995.6 1376117.6 1095877.1 6996906.2 11217978.6 2152476.1
[19] 8983494.0 25891540.5 6197790.8 4115742.5 6240473.4 1493720.0
[25] 1825650.5 3916781.1 1813938.1
```

Após calcular as quantidades e obter as tabelas, vamos dividir a terceira coluna do primeiro *data frame* pela terceira coluna do segundo, para encontrar a taxa de desemprego por UF. O comando será o seguinte:

```
#Estimando a taxa de desemprego
txdesuf=desuf[,3]/pftuf[,3]*100 #Estimando a taxa de desocupação por UF
txdesuf                          #Retornando os valores encontrados
```

```
[1] 3.053961 9.974079 10.028450 4.622600 8.234853 13.321621
[7] 5.174262 8.349846 9.451956 7.792669 9.928629 10.282731
[13] 12.314147 9.312626 11.938033 13.507842 5.776764 7.239280
[19] 11.436099 7.668977 5.138152 3.224316 4.637457 3.334524
[25] 3.524550 6.646445 10.254765
```

Vamos montar um *data frame* completo com os valores calculados e as UFs correspondentes. Para isso, nomeamos a coluna com o nome dos estados, as juntamos e, por fim, nomeamos a coluna com o nome do desemprego.

```
#Montando a dataframe final
UF<- desuf[,1] #Coluna com os nomes dos estados.
txu = data.frame(UF, txdesuf) #Definindo o data frame completo
colnames(txu)[2] <- "desemprego" #Nomeando a coluna
txu #Exibindo o data frame completo
```

```
UF desemprego
1 Rondônia 3.053961
2 Acre 9.974079
3 Amazonas 10.028450
4 Roraima 4.622600
5 Pará 8.234853
6 Amapá 13.321621
7 Tocantins 5.174262
8 Maranhão 8.349846
9 Piauí 9.451956
10 Ceará 7.792669
11 Rio Grande do Norte 9.928629
12 Paraíba 10.282731
13 Pernambuco 12.314147
14 Alagoas 9.312626
15 Sergipe 11.938033
16 Bahia 13.507842
17 Minas Gerais 5.776764
18 Espírito Santo 7.239280
19 Rio de Janeiro 11.436099
20 São Paulo 7.668977
21 Paraná 5.138152
22 Santa Catarina 3.224316
23 Rio Grande do Sul 4.637457
24 Mato Grosso do Sul 3.334524
25 Mato Grosso 3.524550
26 Goiás 6.646445
27 Distrito Federal 10.254765
```

Assim, chegamos a dois *data frames*, um contendo o índice de Gini e o outro a taxa de desemprego por UF's.

4.5. Unindo os *data frames*

Usaremos o comando `cbind()` para colocar o índice de Gini e a taxa de desemprego em um mesmo *data frame*. Não é necessário instalar ou carregar nenhum pacote. Conforme abaixo:

```
#Unindo o data frame do índice de GINI e tx de desemprego por UF
gini_desempregoUF <- cbind(giniUF,txu)
```

	UF	gini	erro	padrão	UF	desemprego
Rondônia	Rondônia	0.4379337	0.012049294		Rondônia	3.053961
Acre	Acre	0.4635926	0.010672074		Acre	9.974079
Amazonas	Amazonas	0.5050460	0.012376445		Amazonas	10.028450
Roraima	Roraima	0.5212136	0.020711239		Roraima	4.622600
Pará	Pará	0.5469579	0.011739516		Pará	8.234853
Amapá	Amapá	0.5010884	0.014483361		Amapá	13.321621
Tocantins	Tocantins	0.4996150	0.011144626		Tocantins	5.174262
Maranhão	Maranhão	0.5123910	0.012036913		Maranhão	8.349846
Piauí	Piauí	0.5813827	0.022906859		Piauí	9.451956
Ceará	Ceará	0.5209561	0.016758802		Ceará	7.792669
Rio Grande do Norte	Rio Grande do Norte	0.5346413	0.021572553		Rio Grande do Norte	9.928629
Paraíba	Paraíba	0.5508491	0.020110133		Paraíba	10.282731
Pernambuco	Pernambuco	0.5036104	0.012651360		Pernambuco	12.314147
Alagoas	Alagoas	0.4644419	0.015471983		Alagoas	9.312626
Sergipe	Sergipe	0.5569250	0.021540649		Sergipe	11.938033
Bahia	Bahia	0.5279421	0.012804452		Bahia	13.507842
Minas Gerais	Minas Gerais	0.4673820	0.008019685		Minas Gerais	5.776764
Espírito Santo	Espírito Santo	0.4957296	0.011235141		Espírito Santo	7.239280
Rio de Janeiro	Rio de Janeiro	0.5264051	0.006729311		Rio de Janeiro	11.436099
São Paulo	São Paulo	0.4957087	0.008558828		São Paulo	7.668977
Paraná	Paraná	0.4489797	0.007478786		Paraná	5.138152
Santa Catarina	Santa Catarina	0.4049662	0.005869953		Santa Catarina	3.224316
Rio Grande do Sul	Rio Grande do Sul	0.4893804	0.007471994		Rio Grande do Sul	4.637457
Mato Grosso do Sul	Mato Grosso do Sul	0.4928539	0.020197697		Mato Grosso do Sul	3.334524
Mato Grosso	Mato Grosso	0.4440160	0.014103796		Mato Grosso	3.524550
Goiás	Goiás	0.4709154	0.014012555		Goiás	6.646445
Distrito Federal	Distrito Federal	0.5701907	0.011956530		Distrito Federal	10.254765

Note que, o *data frame* fica muito extenso e redundante. Então, excluiremos a terceira e a quarta coluna para resolver estes problemas e manter só as variáveis importantes para um *data frame* final

```
#Excluindo a terceira e quarta colunas
gini_desempregoUF <- gini_desempregoUF[, -c(3, 4)] #Excluindo as colunas
gini_desempregoUF                                     #Exibindo o data frame
final
```

	UF	gini	desemprego
Rondônia	Rondônia	0.4379337	3.053961
Acre	Acre	0.4635926	9.974079
Amazonas	Amazonas	0.5050460	10.028450
Roraima	Roraima	0.5212136	4.622600
Pará	Pará	0.5469579	8.234853
Amapá	Amapá	0.5010884	13.321621
Tocantins	Tocantins	0.4996150	5.174262
Maranhão	Maranhão	0.5123910	8.349846
Piauí	Piauí	0.5813827	9.451956
Ceará	Ceará	0.5209561	7.792669
Rio Grande do Norte	Rio Grande do Norte	0.5346413	9.928629
Paraíba	Paraíba	0.5508491	10.282731
Pernambuco	Pernambuco	0.5036104	12.314147
Alagoas	Alagoas	0.4644419	9.312626
Sergipe	Sergipe	0.5569250	11.938033
Bahia	Bahia	0.5279421	13.507842
Minas Gerais	Minas Gerais	0.4673820	5.776764
Espírito Santo	Espírito Santo	0.4957296	7.239280
Rio de Janeiro	Rio de Janeiro	0.5264051	11.436099
São Paulo	São Paulo	0.4957087	7.668977
Paraná	Paraná	0.4489797	5.138152
Santa Catarina	Santa Catarina	0.4049662	3.224316
Rio Grande do Sul	Rio Grande do Sul	0.4893804	4.637457
Mato Grosso do Sul	Mato Grosso do Sul	0.4928539	3.334524
Mato Grosso	Mato Grosso	0.4440160	3.524550
Goiás	Goiás	0.4709154	6.646445
Distrito Federal	Distrito Federal	0.5701907	10.254765

Com um *data frame* final contendo os dados de interesse, é possível realizar a regressão com as variáveis.

4.6. Regressão linear simples

De modo a ilustrar como os dados importados pelo pacote `PNADcIBGE` e formatados com os pacotes `survey` e `convey` podem ser aplicados, realizaremos uma regressão linear simples, na qual o índice de Gini é a variável explicativa e a taxa de desemprego é a variável resposta. Uma regressão linear no R é feita pela função `lm()`, o usuário não precisa carregar ou instalar nenhum pacote. Na função deve ser informada a variável dependente (resposta) seguida da variável explicativa e também o conjunto de dados que contém as variáveis. Ou seja,

```
#Regressão linear simples
lmgini_desempregoUF <- lm(desemprego~gini, data=gini_desempregoUF)
#Comando para realizar a regressão
Summary(lmgini_desempregoUF) #Exibindo o resultado
```

```

Call:
lm(formula = desemprego ~ gini, data = gini_desempregoUF)

Residuals:
    Min       1Q   Median       3Q      Max
-4.2912 -1.8616 -0.3919  1.6242  5.3250

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -14.839      6.062  -2.448 0.021738 *
gini          45.572     12.051   3.781 0.000867 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.602 on 25 degrees of freedom
Multiple R-squared:  0.3639,    Adjusted R-squared:  0.3384
F-statistic: 14.3 on 1 and 25 DF,  p-value: 0.0008667

```

5. Considerações finais

Nesta nota, foram mostrados alguns comandos do pacote PNADcIBGE, convey e survey no software *R* através de dados da PNAD Contínua. O pacote PNADcIBGE permite importar os dados e os outros dois pacotes permitem que o usuário os manipule com muita flexibilidade. Porém, esta nota serve como uma introdução aos pacotes e não esgota todos os seus aspectos. Caso o usuário deseje se aprofundar na temática, recomenda-se que pesquise, por exemplo, nas bibliografias citadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Assunção, G. (2023). *Análise de microdados da PNAD Contínua: com os pacotes PNADcIBGE e survey*. Rpubs. Disponível em: <https://rpubs.com/gabriel-assuncao-ibge/pnadc>. Acesso em: 27 de setembro de 2023.
- [2] Pedroza Jr., D. (2021). *PNADc Trimestral*. Rpubs. Disponível em: <https://rpubs.com/dinilson/844332>. Acesso em: 27 de setembro de 2023.
- [3] R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- [4] Trovão, C. J. B. M. e Silva Jr, A. H. M (2022). *Por dentro da PNAD contínua: uma introdução ao tratamento de dados usando R*. Natal: EDUFRN.