

# Introdução ao R para análise de dados de imigração

Felipe Quintino

03-11-2025

## Contents

<b>Introdução</b>	<b>2</b>
Apresentação do minicurso . . . . .	2
Equipe do minicurso . . . . .	2
Microdados utilizados . . . . .	2
<b>Aula 1</b>	<b>3</b>
Conceitos básicos do R . . . . .	3
Usando R como uma calculadora . . . . .	3
Salvando resultados em objetos . . . . .	3
Letras maiúsculas e minúsculas . . . . .	3
Help . . . . .	3
Diretório de trabalho . . . . .	4
Leitura dos microdados (formato .csv) . . . . .	4
Selecionando linhas e colunas específicas nos microdados . . . . .	4
Tabelas de frequências . . . . .	5
Aplicando os filtros de interesse . . . . .	5
Agregando dados de 2024 e 2025 . . . . .	5
Manipulação das variáveis de interesse e geração das tabelas . . . . .	6
<b>Aula 2</b>	<b>21</b>
Instalação de pacotes e finalidade de cada função utilizada . . . . .	21
pacman . . . . .	21
tidyverse . . . . .	21
geobr . . . . .	21
sf (Simple Features) . . . . .	21
stringi . . . . .	21
scales . . . . .	21
Leitura dos dados . . . . .	22
Gráfico de Barras . . . . .	23
Análise de gênero . . . . .	23
Análise da faixa etária . . . . .	26
Análises bivariadas . . . . .	28
Relação entre ocupação e escolaridade . . . . .	28
Gênero vs Andamento . . . . .	30
Análise de países de origem . . . . .	32
Gráficos de Linhas . . . . .	35
Mapas de calor . . . . .	37
Visualização geográfica - Distribuição por UF . . . . .	37

# Introdução

## Apresentação do minicurso

Esse material é parte do minicurso de 8h oferecido como atividade da Semana Universitária (SEMUNI) de 2025 da Universidade de Brasília (UnB) e tem como objetivo introduzir os conceitos básicos de R na análise de dados de imigração. Esse minicurso contou com o apoio do Instituto de Ciências Exatas da UnB, do Departamento de Estatística da UnB, da Empresa Júnior ESTAT Consultoria e do Observatório das Migrações Internacionais (OBMIgra).

## Equipe do minicurso

O minicurso foi ministrado por

- Felipe Quintino, Bacharel em Estatística, Doutor em Matemática Aplicada pela UnB, professor do Departamento de Estatística da UnB, pesquisador do OBMIgra e coordenador do minicurso. <<http://lattes.cnpq.br/2290992968851891>>
- Sabrina França, Estatística e Mestranda no Programa de Pós-graduação em Estatística da UnB, atua como analista de dados na Coordenação-Geral de Monitoramento, Avaliação e Inteligência Analítica da Atenção Primária à Saúde no Ministério da Saúde. <<http://lattes.cnpq.br/3439903933557735>>

Além disso, o minicurso contou com o apoio dos alunos de graduação em estatística Felipe Adriano de Castro, Gabriel Leonardo de Oliveira, Leonardo Santos e Pedro Tepedino.

## Microdados utilizados

Os microdados da Coordenação Geral de Imigração Laboral (CGIL) estão sob a responsabilidade do Ministério da Justiça e Segurança Pública e informam sobre os imigrantes que entraram com pedido de autorização de residência para fins laborais. O banco de dados contém informações sobre o país de nascimento, sexo, idade, amparo legal, ocupação, entre outros aspectos trabalhistas relacionados ao solicitante da autorização. Os microdados estão disponíveis no site <https://portaldeimigracao.mj.gov.br/pt/base-de-dados/datamigra> (acesso em 03-11-2025).

# Aula 1

## Conceitos básicos do R

### Usando R como uma calculadora

Podemos utilizar o R para fazer operações matemáticas simples.

```
2 + 2
```

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

```
5 * 3
```

```
## [1] 15
```

```
10 / 2
```

```
## [1] 5
```

```
4^2
```

```
## [1] 16
```

### Salvando resultados em objetos

Podemos utilizar as setas '<-' para atribuir valores a objetos.

Caso necessite fazer comentários, utilizaremos '#' para fazer comentários no código.

```
#Atribuindo valores a objetos
```

```
a <- 10
```

```
b <- 5
```

```
c <- a + b
```

```
c#printar o valor de 'c'
```

```
## [1] 15
```

```
#Obs: tambem e possivel utilizar '=' para atribuicao
```

```
c2 = a * b
```

```
c2
```

```
## [1] 50
```

### Letras maiúsculas e minúsculas

O R diferencia letras maiúsculas e minúsculas. Portanto, 'A' e 'a' são objetos diferentes.

```
A <- 3
```

```
a <- 5
```

```
A; a
```

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

```
## [1] 5
```

### Help

É possível utilizar o 'help' do R para pesquisar funções, objetos e/ou palavras que identifiquem o que deseja buscar

```
#help: pesquise por palavra chave ou o nome de alguma funcao
```

```
#help(sum)
```

## Diretório de trabalho

Utilize as funções ‘getwd()’ e ‘setwd()’ para verificar e alterar o diretório de trabalho do R, respectivamente.

## Leitura dos microdados (formato .csv)

Fazer uma **pré-leitura** dos microdados auxilia na escolha dos parâmetros da função ‘read.csv2()’. Dica: verifique possível acentuação no arquivo antes de fazer a leitura da base. Isso influenciará a escolha do Encoding como “UTF-8” ou “latin1”,...

Outras funções de leitura de arquivos são: read.csv(), read.table(), fread() (pacote data.table), read\_excel() (pacote readxl) entre outras.

```
##identificar o diretorio de trabalho
#getwd()

##ver quais arquivos e pastas estao no diretorio
#dir()

##pre-leitura dos microdados
#readLines("CGIL_CNIg_2024.csv",10)

#lendo o arquivo com a função read.csv2 (ponto e vírgula como separador)
dados <- read.csv2("CGIL_CNIg_2024.csv", fileEncoding = "UTF-8")
#Outra possibilidade de Encoding: "latin1"
```

Para ver as primeiras linhas do banco de dados, utilize a função head() ou selecione as linhas e colunas desejadas. Para isso, utilize o nome do objeto seguido de [,]. Os valores antes da vírgula são referentes às linhas e após a vírgula referentes às colunas desejadas.

```
##verificando as primeiras linhas do banco de dados
#head(dados)
dados[1:5,2:4]#peimeiras 5 linhas, colunas 2, 3 e 4
```

	andamento	amparo_legal
## 1	INDEFERIDO	PORTARIA INTERMINISTERIAL MJSP/MRE Nº 38/2023
## 2	DEFERIDO	RESOLUÇÃO NORMATIVA 03/2017 (Art. 2º)
## 3	DEFERIDO	RESOLUÇÃO NORMATIVA 03/2017 (Art. 2º)
## 4	DEFERIDO	RESOLUÇÃO NORMATIVA 03/2017 (Art. 2º)
## 5	DEFERIDO	RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 02/2017)
	modalidade	
## 1	CNIg	
## 2	CGIL	
## 3	CGIL	
## 4	CGIL	
## 5	CGIL	

## Selecionando linhas e colunas específicas nos microdados

```
names(dados) #ver nomes das variaveis
```

## [1]	"tipo_visto"	"andamento"
## [3]	"amparo_legal"	"modalidade"
## [5]	"mes"	"ano"
## [7]	"valor_investimento_ajustado"	"uf_estrangeiro"
## [9]	"data_nascimento"	"genero"
## [11]	"escolaridade"	"pais"

```
## [13] "codigo_cbo"          "descricao_cbo"
## [15] "codigo_cnae"         "descricao_cnae"
## [17] "estado_civil"

dim(dados) #ver dimensões do banco de dados (linhas e colunas)

## [1] 56898      17

table(dados$andamento) #tabela de frequencias da variavel andamento

##
## CANCELAMENTO DE ESTRANGEIRO    CANCELAMENTO DE PROCESSO
##                               3                               4
##          DEFERIDO              INDEFERIDO
##          47085                  9806

#selecionando apenas linhas com andamento 'DEFERIDO'
dados_deferidos <- dados[dados$andamento == "DEFERIDO", ]

#selecionando apenas as colunas 'ano' e 'andamento'
dados_ano_andamento <- dados[, c("ano", "andamento")]
```

## Tabelas de frequências

### Aplicando os filtros de interesse

Para gerar Tabelas de frequências, serão aplicados os filtros

- andamento: DEFERIDO
- mês: 09
- modalidade: CGIL
- ano: 2024

As variáveis de interesse são: tipo de visto, país de origem, faixa etária, sexo, amparo, escolaridade, ocupação, estado de destino, valor do investimento.

```
#filtrando os dados conforme os critérios acima
##usar o padrao dados[linhas, colunas]
## & representa o operador "E" logico
## | representa o operador "OU" logico
dados_filtrados <- dados[
  dados$andamento == "DEFERIDO" &
  dados$mes == 9 &
  dados$modalidade == "CGIL" &
  dados$ano == "2024",
  c("tipo_visto", "andamento", "amparo_legal", "modalidade",
    "mes", "ano", "valor_investimento_ajustado", "uf_estrangeiro",
    "data_nascimento", "genero", "escolaridade", "pais",
    "codigo_cbo")
]
```

### Agregando dados de 2024 e 2025

Para o relatório tabular, desejamos agregar as informações de 2025. Para isso, será necessário utilizar uma agregação entre os dados de 2024 e de 2025.

```
dados25 <- read.csv2("CGIL_CNig_jan-set2025.csv", fileEncoding = "UTF-8")
dados25_filtrados <- dados25[
  dados25$andamento == "DEFERIDO" &
  dados25$mes %in% c(8,9) &
  dados25$modalidade == "CGIL" &
  dados25$ano == "2025",
  c("tipo_visto", "andamento", "amparo_legal", "modalidade",
    "mes", "ano", "valor_investimento_ajustado", "uf_estrangeiro",
    "data_nascimento", "genero", "escolaridade", "pais",
    "codigo_cbo")
]
```

*#verificando se as colunas estao iguais*

```
names(dados_filtrados)
```

```
## [1] "tipo_visto"          "andamento"
## [3] "amparo_legal"        "modalidade"
## [5] "mes"                 "ano"
## [7] "valor_investimento_ajustado" "uf_estrangeiro"
## [9] "data_nascimento"     "genero"
## [11] "escolaridade"        "pais"
## [13] "codigo_cbo"
```

```
names(dados25_filtrados)
```

```
## [1] "tipo_visto"          "andamento"
## [3] "amparo_legal"        "modalidade"
## [5] "mes"                 "ano"
## [7] "valor_investimento_ajustado" "uf_estrangeiro"
## [9] "data_nascimento"     "genero"
## [11] "escolaridade"        "pais"
## [13] "codigo_cbo"
```

```
names(dados_filtrados)==names(dados25_filtrados)#teste logico de igualdade
```

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

*## juntando dados de 2024 e 2025*

```
dados_final <- rbind.data.frame(dados_filtrados, dados25_filtrados)
```

## Manipulação das variáveis de interesse e geração das tabelas

Para geração das informações, pode ser necessário a criação de novas variáveis ou a modificação das variáveis existentes. Para salvar a nova variável, utilizamos o operador '\$' para indicar o nome do objeto e o nome da nova variável.

*# criando variavel 'ano\_mes' a partir das variaveis 'ano' e 'mes'*

```
dados_final$ano_mes <- paste(dados_final$ano, dados_final$mes, sep="_")
```

```
table(dados_final$ano_mes)
```

```
##
## 2024_9 2025_8 2025_9
##   2589   4015   4672
```

*#sexo*

```
table(dados_final$genero)
```

```
##
```

```
##      F      M
## 1191 10085

dados_final$genero <- factor(dados_final$genero,
                             levels = c("M", "F"),
                             labels = c("Masculino", "Feminino"))

table(dados_final$genero)
```

```
##
## Masculino Feminino
##      10085      1191

table(dados_final$genero, dados_final$ano_mes)
```

```
##
##           2024_9 2025_8 2025_9
## Masculino    2298    3703    4084
## Feminino      291     312     588
```

```
#add o total das linhas
addmargins(table(dados_final$genero, dados_final$ano_mes),1)
```

```
##
##           2024_9 2025_8 2025_9
## Masculino    2298    3703    4084
## Feminino      291     312     588
## Sum           2589    4015    4672
```

```
#atribuir a um objeto e exportar para o diretorio
tab=addmargins(table(dados_final$genero, dados_final$ano_mes),1)
write.csv2(tab, "tabela1.csv", row.names = TRUE)
```

```
#tipo de visto
table(dados_final$tipo_visto)
```

```
##
##      Residência Residência Prévia
##      3380           7896
```

```
(tab2=addmargins(table(dados_final$tipo_visto, dados_final$ano_mes),1))
```

```
##
##           2024_9 2025_8 2025_9
## Residência      790    1131    1459
## Residência Prévia 1799    2884    3213
## Sum             2589    4015    4672
```

```
#write.csv2(tab2, "tabela2.csv", row.names = TRUE)
```

```
#países
table(dados_final$pais)
```

```
##
##      ALBÂNIA      ALEMANHA      ALGÉRIA
##      4           310           1
##      ANGOLA      ARÁBIA SAUDITA      ARGÉLIA
##      19           1           2
##      ARGENTINA      AUSTRÁLIA      ÁUSTRIA
##      37           24           62
```

##	AZERBAIJÃO	BANGLADESH	BELARUS
##	7	320	5
##	BÉLGICA	BENIN	BOLÍVIA
##	36	3	14
##	BÓSNIA-HERZEGOVINA	BÓSNIA E HERZEGOVINA	BRASIL
##	1	8	3
##	BULGÁRIA	CABO VERDE	CAMARÕES
##	32	5	8
##	CANADÁ	CAZAQUISTÃO	CHILE
##	65	3	27
##	CHINA	CHIPRE	COLÔMBIA
##	3144	6	98
##	CONGO	CORÉIA DO SUL	COSTA DO MARFIM
##	6	132	4
##	COSTA RICA	CROÁCIA	CUBA
##	12	93	20
##	DINAMARCA	DOMINICA	EGITO
##	42	1	29
##	EL SALVADOR	EMIRADOS ÁRABES UNIDOS	EQUADOR
##	28	2	12
##	ESCÓCIA	ESLOVÁQUIA	ESLOVÊNIA
##	1	5	13
##	ESPANHA	ESTADOS UNIDOS	ESTÔNIA
##	179	620	3
##	ETIÓPIA	FILIPINAS	FINLÂNDIA
##	3	850	51
##	FRANÇA	GANÁ	GEÓRGIA
##	262	9	6
##	GRÃ-BRETANHA	GRÉCIA	GUATEMALA
##	323	27	14
##	GUIANA	GUINÉ-BISSAU	HAITI
##	3	1	3
##	HOLANDA	HONDURAS	HUNGRIA
##	138	123	17
##	ÍNDIA	INDONÉSIA	INGLATERRA
##	584	129	18
##	IRÃ	IRLANDA	ISLÂNDIA
##	15	23	1
##	ISRAEL	ITÁLIA	JAMAICA
##	14	546	1
##	JAPÃO	LESOTO	LETÔNIA
##	241	2	35
##	LÍBANO	LÍBIA	LITUÂNIA
##	11	1	11
##	LUXEMBURGO	MACEDÔNIA	MADAGASCAR
##	1	1	67
##	MALÁSIA	MALGAXE	MALI
##	127	3	3
##	MALTA	MARROCOS	MAURÍCIO
##	5	6	18
##	MÉXICO	MIANMAR	MOÇAMBIQUE
##	293	5	21
##	MOLDOVA	MONTENEGRO	NAMÍBIA
##	1	6	4

##	NEPAL	NICARÁGUA	NIGÉRIA
##	5	7	29
##	NORUEGA	NOVA ZELÂNDIA	PALESTINA
##	174	6	3
##	PANAMÁ	PAQUISTÃO	PARAGUAI
##	6	9	8
##	PERU	POLÔNIA	PORTUGAL
##	99	181	99
##	QUÊNIA	QUIRGUISTÃO	REINO UNIDO
##	45	7	173
##	REPÚBL. DA ÁFRICA DO SUL	REPÚBLICA DO CONGO	REPÚBLICA DOMINICANA
##	130	2	22
##	REPÚBLICA THECA	ROMÊNIA	RUANDA
##	16	124	3
##	RÚSSIA	SAMOA OCIDENTAL	SENEGAL
##	121	7	12
##	SÉRVIA	SINGAPURA	SRI LANKA
##	14	44	11
##	SUDÃO	SUÉCIA	SUIÇA
##	1	70	30
##	TAILÂNDIA	TAIWAN	TANZÂNIA
##	50	4	2
##	TIMOR	TOGO	TRINIDAD E TOBAGO
##	5	2	14
##	TUNÍSIA	TURCOMENISTÃO	TURQUIA
##	11	1	75
##	UCRÂNIA	UGANDA	UNIÃO SOVIÉTICA
##	106	3	1
##	VENEZUELA	VIETNÃ	ZÂMBIA
##	41	11	1
##	ZIMBÁBUE		
##	21		

```
(tab3=addmargins(table(dados_final$país, dados_final$ano_mes),1))
```

##		2024_9	2025_8	2025_9
##	ALBÂNIA	0	2	2
##	ALEMANHA	80	117	113
##	ALGÉRIA	0	0	1
##	ANGOLA	4	11	4
##	ARÁBIA SAUDITA	0	0	1
##	ARGÉLIA	0	1	1
##	ARGENTINA	11	4	22
##	AUSTRÁLIA	6	4	14
##	ÁUSTRIA	11	21	30
##	AZERBAIJÃO	2	1	4
##	BANGLADESH	6	249	65
##	BELARUS	3	0	2
##	BÉLGICA	2	15	19
##	BENIN	1	0	2
##	BOLÍVIA	4	5	5
##	BÓSNIA-HERZEGOVINA	0	1	0
##	BÓSNIA E HERZEGOVINA	5	2	1
##	BRASIL	1	1	1

##	BULGÁRIA	8	9	15
##	CABO VERDE	3	1	1
##	CAMARÕES	1	3	4
##	CANADÁ	16	20	29
##	CAZAQUISTÃO	0	2	1
##	CHILE	7	5	15
##	CHINA	695	1291	1158
##	CHIPRE	1	1	4
##	COLÔMBIA	22	30	46
##	CONGO	1	4	1
##	CORÉIA DO SUL	34	62	36
##	COSTA DO MARFIM	2	0	2
##	COSTA RICA	4	2	6
##	CROÁCIA	15	38	40
##	CUBA	8	7	5
##	DINAMARCA	16	17	9
##	DOMINICA	0	0	1
##	EGITO	2	2	25
##	EL SALVADOR	10	1	17
##	EMIRADOS ÁRABES UNIDOS	1	0	1
##	EQUADOR	2	2	8
##	ESCÓCIA	0	0	1
##	ESLOVÁQUIA	0	2	3
##	ESLOVÊNIA	2	7	4
##	ESPANHA	54	56	69
##	ESTADOS UNIDOS	181	227	212
##	ESTÔNIA	0	0	3
##	ETIÓPIA	3	0	0
##	FILIPINAS	193	261	396
##	FINLÂNDIA	17	15	19
##	FRANÇA	61	102	99
##	GANÁ	1	3	5
##	GEÓRGIA	3	0	3
##	GRÃ-BRETANHA	83	138	102
##	GRÉCIA	5	14	8
##	GUATEMALA	1	5	8
##	GUIANA	2	0	1
##	GUINÉ-BISSAU	1	0	0
##	HAITI	1	0	2
##	HOLANDA	20	67	51
##	HONDURAS	47	2	74
##	HUNGRIA	5	2	10
##	ÍNDIA	151	124	309
##	INDONÉSIA	15	33	81
##	INGLATERRA	6	4	8
##	IRÃ	3	5	7
##	IRLANDA	7	7	9
##	ISLÂNDIA	0	1	0
##	ISRAEL	4	3	7
##	ITÁLIA	111	126	309
##	JAMAICA	0	1	0
##	JAPÃO	74	83	84
##	LESOTO	1	0	1
##	LETÔNIA	8	16	11

##	LÍBANO	4	3	4
##	LÍBIA	0	0	1
##	LITUÂNIA	1	4	6
##	LUXEMBURGO	0	1	0
##	MACEDÔNIA	0	0	1
##	MADAGASCAR	32	0	35
##	MALÁSIA	26	46	55
##	MALGAXE	0	0	3
##	MALI	2	1	0
##	MALTA	2	0	3
##	MARROCOS	2	2	2
##	MAURÍCIO	5	0	13
##	MÉXICO	58	126	109
##	MIANMAR	1	1	3
##	MOÇAMBIQUE	5	9	7
##	MOLDOVA	0	1	0
##	MONTENEGRO	5	0	1
##	NAMÍBIA	1	1	2
##	NEPAL	1	0	4
##	NICARÁGUA	2	3	2
##	NIGÉRIA	2	13	14
##	NORUEGA	47	62	65
##	NOVA ZELÂNDIA	0	1	5
##	PALESTINA	1	0	2
##	PANAMÁ	1	2	3
##	PAQUISTÃO	2	4	3
##	PARAGUAI	1	0	7
##	PERU	14	41	44
##	POLÔNIA	32	68	81
##	PORTUGAL	32	31	36
##	QUÊNIA	9	0	36
##	QUIRGUISTÃO	3	0	4
##	REINO UNIDO	49	52	72
##	REPÚBL. DA ÁFRICA DO SUL	30	28	72
##	REPÚBLICA DO CONGO	0	2	0
##	REPÚBLICA DOMINICANA	14	2	6
##	REPÚBLICA THECA	10	3	3
##	ROMÊNIA	32	33	59
##	RUANDA	1	0	2
##	RÚSSIA	29	45	47
##	SAMOA OCIDENTAL	3	0	4
##	SENEGAL	2	8	2
##	SÉRVIA	6	4	4
##	SINGAPURA	5	15	24
##	SRI LANKA	3	2	6
##	SUDÃO	0	1	0
##	SUÉCIA	15	23	32
##	SUÍÇA	5	11	14
##	TAILÂNDIA	0	29	21
##	TAIWAN	1	0	3
##	TANZÂNIA	1	1	0
##	TIMOR	1	0	4
##	TOGO	1	0	1
##	TRINIDAD E TOBAGO	5	2	7

```
## TUNÍSIA 3 3 5
## TURCOMENISTÃO 0 0 1
## TURQUIA 9 36 30
## UCRÂNIA 16 44 46
## UGANDA 0 0 3
## UNIÃO SOVIÉTICA 0 0 1
## VENEZUELA 8 13 20
## VIETNÃ 0 2 9
## ZÂMBIA 0 1 0
## ZIMBÁBUE 5 0 16
## Sum 2589 4015 4672
```

```
#write.csv2(tab3, "tabela3.csv", row.names = TRUE)
```

```
#escolaridade
```

```
table(dados_final$escolaridade)
```

```
##
## 2_Fundamental Incompleto 3_Fundamental 4_Médio
## 1 12 1019
## 5_Superior 6_Pós-Graduação 7_Mestrado
## 1343 37 152
## 8_Doutorado Doutorado Especialização
## 25 42 242
## Fundamental Completo Fundamental Incompleto Médio Completo
## 26 3 3403
## Médio Incompleto Mestrado Pós-doutorado
## 50 483 24
## Superior Completo Superior Incompleto
## 4390 24
```

```
levels(as.factor(dados_final$escolaridade))
```

```
## [1] "2_Fundamental Incompleto" "3_Fundamental"
## [3] "4_Médio" "5_Superior"
## [5] "6_Pós-Graduação" "7_Mestrado"
## [7] "8_Doutorado" "Doutorado"
## [9] "Especialização" "Fundamental Completo"
## [11] "Fundamental Incompleto" "Médio Completo"
## [13] "Médio Incompleto" "Mestrado"
## [15] "Pós-doutorado" "Superior Completo"
## [17] "Superior Incompleto"
```

```
#transformar e padronizar os níveis da variável escolaridade
```

```
dados_final$escolaridade[dados_final$escolaridade=="Doutorado"]
```

```
## [1] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [7] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [13] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [19] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [25] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [31] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
## [37] "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado" "Doutorado"
```

```
dados_final$escolaridade[dados_final$escolaridade=="Doutorado"]="8_Doutorado"
```

```
dados_final$escolaridade[dados_final$escolaridade=="Mestrado"]="7_Mestrado"
```

```
dados_final$escolaridade[dados_final$escolaridade=="Especialização"]="6_Pós-Graduação"
```

```

dados_final$escolaridade[dados_final$escolaridade=="Fundamental Completo"]="3_Fundamental"
dados_final$escolaridade[dados_final$escolaridade=="Fundamental Incompleto"]="2_Fundamental Incompleto"
dados_final$escolaridade[dados_final$escolaridade=="Médio Completo"]="4_Médio"
dados_final$escolaridade[dados_final$escolaridade=="Médio Incompleto"]="3_Fundamental"
dados_final$escolaridade[dados_final$escolaridade=="Pós-doutorado"]="8_Doutorado"
dados_final$escolaridade[dados_final$escolaridade=="Superior Completo"]="5_Superior"
dados_final$escolaridade[dados_final$escolaridade=="Superior Incompleto"]="4_Médio"
table(dados_final$escolaridade)

```

```

##
## 2_Fundamental Incompleto      3_Fundamental      4_Médio
##              4              88              4446
##          5_Superior      6_Pós-Graduação      7_Mestrado
##              5733              279              635
##          8_Doutorado
##              91

```

```

(tab4=addmargins(table(dados_final$escolaridade, dados_final$ano_mes),1))

```

```

##
##              2024_9 2025_8 2025_9
## 2_Fundamental Incompleto      1      2      1
## 3_Fundamental      12      38      38
## 4_Médio      1019     1495     1932
## 5_Superior      1343     2070     2320
## 6_Pós-Graduação      37      148      94
## 7_Mestrado      152      235      248
## 8_Doutorado      25      27      39
## Sum      2589     4015     4672

```

```

#write.csv2(tab4, "tabela4.csv", row.names = TRUE)

```

```

#Unidade da Federacao
table(dados_final$uf_estrangerio)

```

```

##
##          Acre          Alagoas          Amapá          Amazonas
##              1              20              2              142
##          Bahia          Ceará          Distrito Federal          Espírito Santo
##          1167              113              105              106
##          Goiás          Maranhão          Mato Grosso          Mato Grosso do Sul
##              45              21              25              21
##          Minas Gerais          Pará          Paraíba          Paraná
##              474              32              12              601
##          Pernambuco          Piauí          Rio de Janeiro          Rio Grande do Norte
##              72              1              4039              50
##          Rio Grande do Sul          Rondônia          Roraima          Santa Catarina
##              95              6              4              221
##          São Paulo          Sergipe          Tocantins
##              3889              6              6

```

```

(tab5=addmargins(table(dados_final$uf_estrangerio, dados_final$ano_mes),1))

```

```

##
##              2024_9 2025_8 2025_9

```

##	Acre	1	0	0
##	Alagoas	3	7	10
##	Amapá	0	2	0
##	Amazonas	41	57	44
##	Bahia	221	580	366
##	Ceará	37	41	35
##	Distrito Federal	23	43	39
##	Espírito Santo	34	17	55
##	Goiás	16	14	15
##	Maranhão	6	7	8
##	Mato Grosso	5	10	10
##	Mato Grosso do Sul	2	11	8
##	Minas Gerais	174	157	143
##	Pará	9	10	13
##	Paraíba	4	4	4
##	Paraná	64	347	190
##	Pernambuco	22	26	24
##	Piauí	0	0	1
##	Rio de Janeiro	943	1510	1586
##	Rio Grande do Norte	11	23	16
##	Rio Grande do Sul	32	31	32
##	Rondônia	0	1	5
##	Roraima	1	1	2
##	Santa Catarina	83	56	82
##	São Paulo	852	1058	1979
##	Sergipe	3	1	2
##	Tocantins	2	1	3
##	Sum	2589	4015	4672

```
#write.csv2(tab5, "tabela5.csv", row.names = TRUE)
```

```
#faixa etaria
```

```
#primeiro, calcular a idade a partir da data de nascimento
```

```
dados_final$data_nascimento <- as.Date(dados_final$data_nascimento)
```

```
#utilizar uma data de referencia para o calculo da idade
```

```
#para fins de exemplificacao, utilizaremos a data do mesmo mes em que foi concedida a autorizacao
```

```
dados_final$ano
```

```
dados_final$mes
```

```
paste(dados_final$ano, dados_final$mes, "01", sep = "-")
```

```
data_ref <- as.Date(paste(dados_final$ano, dados_final$mes, "01", sep = "-"))
```

```
#dias entre o nascimento e a data de referencia
```

```
difftime(data_ref,dados_final$data_nascimento,unit="days")
```

```
#conversao para anos, levando em consideracao anos bisestos
```

```
difftime(data_ref,dados_final$data_nascimento,unit="days")/365.25
```

```
#escolher apenas a parte inteira do numero
```

```
floor(difftime(data_ref,dados_final$data_nascimento,unit="days")/365.25)
```

```
#atribuir a variavel idade
```

```
dados_final$idade <- floor(difftime(data_ref,dados_final$data_nascimento,unit="days")/365.25)
```

```
#fazer uma conferencia da variavel idade
```

```
dados_final$idade[1:10]#10 primeiras idades
```

```
## Time differences in days
```

```
## [1] 32 43 22 71 55 64 28 30 54 29
```

```
dados_final$data_nascimento[1:10]#10 primeiras datas de nascimento
```

```
## [1] "1992-01-12" "1980-12-26" "2002-07-06" "1952-11-09" "1969-07-30"
## [6] "1960-05-02" "1996-03-10" "1994-06-12" "1970-02-11" "1994-12-30"
```

```
data_ref[1:10]#10 primeiras datas de referencia
```

```
## [1] "2024-09-01" "2024-09-01" "2024-09-01" "2024-09-01" "2024-09-01"
## [6] "2024-09-01" "2024-09-01" "2024-09-01" "2024-09-01" "2024-09-01"
```

```
table(dados_final$idade)
```

```
##
## -1  0  1  4  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
##  1  1  1  2  1  10  50  68  95 120 182 268 320 325 352 366 343 343 332 348
## 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
## 373 360 418 418 410 418 386 363 345 306 313 281 306 268 251 266 223 216 244 171
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## 173 164 161 126 138 102 84 86 72 56 48 48 28 18 22 26 10 15 9 3
## 73 74 75 76 77 78 79 80 81 82 95
## 6 5 1 2 1 1 2 2 1 1 1
```

```
#temos um problema em idade = -1, precisamos investigar essa idade
```

```
dados_final[dados_final$idade==-1,]
```

```
##      tipo_visto andamento      amparo_legal modalidade mes  ano
## 34840 Residência DEFERIDO RESOLUÇÃO 50/2024      CGIL    9 2024
##      valor_investimento_ajustado uf_estrangeiro data_nascimento  genero
## 34840                          NA Rio de Janeiro      2024-09-24 Masculino
##      escolaridade  pais codigo_cbo ano_mes  idade
## 34840    5_Superior BRASIL      716610 2024_9 -1 days
```

```
#criando faixas de idade
```

```
#dados_final$faixa_etaria <- NA
```

```
#para fins de exemplo didatico, vamos adotar que idades menores que 15 serao consideradas NA
```

```
dados_final$faixa_etaria[dados_final$idade < 15 ] <- "Não informado"
```

```
dados_final$faixa_etaria[dados_final$idade >= 15 & dados_final$idade < 20 ] <- "Menor que 20"
```

```
dados_final$faixa_etaria[dados_final$idade >= 20 & dados_final$idade <= 34 ] <- "20-34"
```

```
dados_final$faixa_etaria[dados_final$idade >= 35 & dados_final$idade <= 49 ] <- "35-49"
```

```
dados_final$faixa_etaria[dados_final$idade >= 50 & dados_final$idade <= 64 ] <- "50-64"
```

```
dados_final$faixa_etaria[dados_final$idade >= 65] <- "65 ou mais"
```

```
table(dados_final$faixa_etaria)
```

```
##
##      20-34      35-49      50-64      65 ou mais      Menor que 20
##      4195      4972      1889      154      61
## Não informado
##      5
```

```
(tab6=addmargins(table(dados_final$faixa_etaria, dados_final$ano_mes),1))
```

```
##
##      2024_9 2025_8 2025_9
## 20-34      977    1481    1737
## 35-49     1091    1832    2049
## 50-64      459     633     797
## 65 ou mais     44      49      61
```

##	Menor que 20	16	20	25
##	Não informado	2	0	3
##	Sum	2589	4015	4672

```
#write.csv2(tab6, "tabela6.csv", row.names = TRUE)
```

Além do método básico de criação/manipulação de variáveis, é possível utilizarmos bases externas para importação de informações via função ‘merge’. Exemplificaremos isso transformando a variável ‘amparo’ em ‘RN’.

```
#Resolucao Normativa
table(dados_final$amparo_legal)
```

```
##
## MUDANÇA DE EMPREGADOR - Dec. 9.199/2017 (Artigo 147, Parágrafo 8)
## 4
## PORTARIA MJSP Nº 885/2025
## 3
## RESOLUÇÃO 47/2022 - Alteração do Prazo
## 1
## RESOLUÇÃO 47/2022 - Renovação (Art. 5º)
## 4
## RESOLUÇÃO 47/2022 - Renovação (Art. 6º)
## 1
## RESOLUÇÃO 47/2022 (Art. 2º)
## 16
## RESOLUÇÃO 47/2022 (Art. 3º)
## 5
## RESOLUÇÃO 47/2022 (Art. 4º c/c 2º)
## 39
## RESOLUÇÃO 47/2022 (Art. 4º c/c 3º)
## 2
## RESOLUÇÃO 50/2024
## 9
## RESOLUÇÃO NORMATIVA 02/2017
## 1578
## RESOLUÇÃO NORMATIVA 03/2017 (Art. 2º)
## 3390
## RESOLUÇÃO NORMATIVA 03/2017 (Art. 3º)
## 39
## RESOLUÇÃO NORMATIVA 03/2017 (Art. 4º, §1º)
## 83
## RESOLUÇÃO NORMATIVA 03/2017 (Art. 4º, Caput)
## 361
## RESOLUÇÃO NORMATIVA 04/2017
## 119
## RESOLUÇÃO NORMATIVA 05/2017
## 1068
## RESOLUÇÃO NORMATIVA 06/2017
## 2052
## RESOLUÇÃO NORMATIVA 07/2017
## 5
## RESOLUÇÃO NORMATIVA 10/2017
## 3
## RESOLUÇÃO NORMATIVA 11/2017 (Art. 1º, §3º)
```

##	1
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 2º, Inc I)
##	79
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 2º, Inc II)
##	3
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 4º, §1º)
##	30
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 1º, §3º)
##	1
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 2º, Inc I)
##	74
##	RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 2º, Inc II)
##	7
##	RESOLUÇÃO NORMATIVA 12/2017
##	1
##	RESOLUÇÃO NORMATIVA 13/2017
##	62
##	RESOLUÇÃO NORMATIVA 13/2017 (Art. 3º)
##	1
##	RESOLUÇÃO NORMATIVA 14/2017
##	95
##	RESOLUÇÃO NORMATIVA 15/2017
##	19
##	RESOLUÇÃO NORMATIVA 16/2017
##	1
##	RESOLUÇÃO NORMATIVA 19/2017
##	80
##	RESOLUÇÃO NORMATIVA 20/2017
##	36
##	RESOLUÇÃO NORMATIVA 24/2018
##	23
##	RESOLUÇÃO NORMATIVA 26/2018 (Art. 2º)
##	3
##	RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 02/2017)
##	306
##	RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 14/2017)
##	30
##	RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 24/2018)
##	2
##	RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 40/2019)
##	20
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 02/2017)
##	153
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 03/2017)
##	440
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 04/2017)
##	20
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 06/2017)
##	484
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 07/2017)
##	7
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 08/2017)
##	2
##	RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 09/2017)

```

## 1
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 11/2017)
## 3
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 13/2017)
## 3
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 14/2017)
## 79
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 15/2017)
## 14
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 16/2017)
## 1
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 17/2017)
## 9
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 20/2017)
## 19
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 24/2018)
## 7
## RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 40/2019)
## 10
## RESOLUÇÃO NORMATIVA 36/2018 - Alteração de prazo
## 1
## RESOLUÇÃO NORMATIVA 36/2018 (Art. 2º, Alínea a)
## 16
## RESOLUÇÃO NORMATIVA 36/2018 (Art. 2º, Alínea b)
## 7
## RESOLUÇÃO NORMATIVA 36/2018 (Art. 4º c/c 2º, Alínea a)
## 47
## RESOLUÇÃO NORMATIVA 36/2018 (Art. 4º c/c 2º, Alínea b)
## 4
## RESOLUÇÃO NORMATIVA 40/2019 (Art. 3º c/c 4º, Inciso I)
## 22
## RESOLUÇÃO NORMATIVA 40/2019 (Art. 4º c/c 2º, Inciso II)
## 17
## RESOLUÇÃO 45/2021 - Renovação de Residência
## 111
## RESOLUÇÃO 45/2021
## 143

```

```

#transformar dos dados em factor
levels(as.factor(dados_final$amparo_legal))

```

```

## [1] "MUDANÇA DE EMPREGADOR - Dec. 9.199/2017 (Artigo 147, Parágrafo 8)"
## [2] "PORTARIA MJSP Nº 885/2025"
## [3] "RESOLUÇÃO 47/2022 - Alteração do Prazo"
## [4] "RESOLUÇÃO 47/2022 - Renovação (Art. 5º)"
## [5] "RESOLUÇÃO 47/2022 - Renovação (Art. 6º)"
## [6] "RESOLUÇÃO 47/2022 (Art. 2º)"
## [7] "RESOLUÇÃO 47/2022 (Art. 3º)"
## [8] "RESOLUÇÃO 47/2022 (Art. 4º c/c 2º)"
## [9] "RESOLUÇÃO 47/2022 (Art. 4º c/c 3º)"
## [10] "RESOLUÇÃO 50/2024"
## [11] "RESOLUÇÃO NORMATIVA 02/2017 "
## [12] "RESOLUÇÃO NORMATIVA 03/2017 (Art. 2º)"
## [13] "RESOLUÇÃO NORMATIVA 03/2017 (Art. 3º)"
## [14] "RESOLUÇÃO NORMATIVA 03/2017 (Art. 4º, §1º)"

```

## [15] "RESOLUÇÃO NORMATIVA 03/2017 (Art. 4º, Caput)"  
 ## [16] "RESOLUÇÃO NORMATIVA 04/2017"  
 ## [17] "RESOLUÇÃO NORMATIVA 05/2017"  
 ## [18] "RESOLUÇÃO NORMATIVA 06/2017"  
 ## [19] "RESOLUÇÃO NORMATIVA 07/2017"  
 ## [20] "RESOLUÇÃO NORMATIVA 10/2017"  
 ## [21] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 1º, §3º)"  
 ## [22] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 2º, Inc I)"  
 ## [23] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 2º, Inc II)"  
 ## [24] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 4º, §1º)"  
 ## [25] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 1º, §3º)"  
 ## [26] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 2º, Inc I)"  
 ## [27] "RESOLUÇÃO NORMATIVA 11/2017 (Art. 6º c/c 2º, Inc II)"  
 ## [28] "RESOLUÇÃO NORMATIVA 12/2017"  
 ## [29] "RESOLUÇÃO NORMATIVA 13/2017"  
 ## [30] "RESOLUÇÃO NORMATIVA 13/2017 (Art. 3º)"  
 ## [31] "RESOLUÇÃO NORMATIVA 14/2017"  
 ## [32] "RESOLUÇÃO NORMATIVA 15/2017"  
 ## [33] "RESOLUÇÃO NORMATIVA 16/2017"  
 ## [34] "RESOLUÇÃO NORMATIVA 19/2017"  
 ## [35] "RESOLUÇÃO NORMATIVA 20/2017"  
 ## [36] "RESOLUÇÃO NORMATIVA 24/2018"  
 ## [37] "RESOLUÇÃO NORMATIVA 26/2018 (Art. 2º)"  
 ## [38] "RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 02/2017)"  
 ## [39] "RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 14/2017)"  
 ## [40] "RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 24/2018)"  
 ## [41] "RESOLUÇÃO NORMATIVA 30/2018 - Alteração de Prazo (RN 40/2019)"  
 ## [42] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 02/2017)"  
 ## [43] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 03/2017)"  
 ## [44] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 04/2017)"  
 ## [45] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 06/2017)"  
 ## [46] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 07/2017)"  
 ## [47] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 08/2017)"  
 ## [48] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 09/2017)"  
 ## [49] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 11/2017)"  
 ## [50] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 13/2017)"  
 ## [51] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 14/2017)"  
 ## [52] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 15/2017)"  
 ## [53] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 16/2017)"  
 ## [54] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 17/2017)"  
 ## [55] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 20/2017)"  
 ## [56] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 24/2018)"  
 ## [57] "RESOLUÇÃO NORMATIVA 30/2018 - Renovação de Residência (RN 40/2019)"  
 ## [58] "RESOLUÇÃO NORMATIVA 36/2018 - Alteração de prazo"  
 ## [59] "RESOLUÇÃO NORMATIVA 36/2018 (Art. 2º, Alínea a)"  
 ## [60] "RESOLUÇÃO NORMATIVA 36/2018 (Art. 2º, Alínea b)"  
 ## [61] "RESOLUÇÃO NORMATIVA 36/2018 (Art. 4º c/c 2º, Alínea a)"  
 ## [62] "RESOLUÇÃO NORMATIVA 36/2018 (Art. 4º c/c 2º, Alínea b)"  
 ## [63] "RESOLUÇÃO NORMATIVA 40/2019 (Art. 3º c/c 4º, Inciso I)"  
 ## [64] "RESOLUÇÃO NORMATIVA 40/2019 (Art. 4º c/c 2º, Inciso II)"  
 ## [65] "RESOLUÇÃO 45/2021 - Renovação de Residência"  
 ## [66] "RESOLUÇÃO 45/2021"

```

#leitura
#readLines("RNs_geral_17052025.csv",10)
rns <- read.csv2("RNs_geral_17052025.csv", fileEncoding = "UTF-8")

#verificar se todos os amparos estao na base de RNs
dados_final$amparo_legal%in%rns$amparo_legal
dados_final$amparo_legal[!dados_final$amparo_legal%in%rns$amparo_legal]

#juntar a base de dados com as informacoes de RNs por meio de um 'merge'
dados_final2 <- merge.data.frame(dados_final, rns,
                                by.x = "amparo_legal",
                                by.y = "amparo_legal",
                                all.x = TRUE)

#table(dados_final2$RN)
(tab7=addmargins(table(dados_final2$RN, dados_final2$ano_mes),1))

##
##          2024_9 2025_8 2025_9
## MUDANÇA DE EMPREGADOR      4      0      0
## PORTARIA MJSP Nº 885/2025    0      2      1
## R 50                        1      3      5
## RN 02                      313     620     645
## RN 03                      984    1769    1120
## RN 04                       30      20      69
## RN 05                      226      0     842
## RN 06                      431     713     908
## RN 07                       1       2       2
## RN 10                       2       1       0
## RN 11                       57      50      88
## RN 12                       1       0       0
## RN 13                       12      12      39
## RN 14                       30      27      38
## RN 15                       4       6       9
## RN 16                       0       1       0
## RN 19                       20      27      33
## RN 20                       9      10      17
## RN 24                       8       4      11
## RN 26                       1       2       0
## RN 30                      341     600     669
## RN 36                       21      22      32
## RN 40                       13      12      14
## RN 45                       60      86     108
## RN 47                       20      26      22
## Sum                      2589    4015    4672

#write.csv2(tab7, "tabela7.csv", row.names = TRUE)

```

## Aula 2

### Instalação de pacotes e finalidade de cada função utilizada

```
#instalando pacotes necessarios
install.packages("tinytex")
install.packages("tidyverse")
install.packages("stringr")
library(tinytex)
library(tidyverse)
library(stringr)

# Verifica e instala pacman se necessário, depois carrega
if (!require("pacman")) install.packages("pacman")

## Carregando pacotes exigidos: pacman
library(pacman)

# Carrega todos os pacotes e se necessário, instala
p_load(tidyverse, gt, geobr, stringi, sf, scales)
```

#### **pacman**

- Gerenciador de pacotes que simplifica o carregamento e instalação de múltiplos pacotes em uma única função, evitando a necessidade de usar `library()` para cada pacote individualmente.

#### **tidyverse**

- Conjunto integrado de pacotes para ciência de dados que segue a filosofia de “**dados organizados**” (**tidy data**).
- Inclui, entre outros:
  - **dplyr**: Manipulação de dados — filtragem, seleção, agregação, criação de novas colunas e ordenação.
  - **ggplot2**: Criação de gráficos sofisticados e personalizáveis usando a gramática de gráficos.
  - **tidyr**: Organização e transformação da estrutura dos dados.

#### **geobr**

- Fornece shapefiles e limites territoriais do Brasil (municípios, estados, regiões) para análise espacial e criação de mapas.

#### **sf (Simple Features)**

- Manipulação e análise de dados geoespaciais — trabalha com shapefiles, coordenadas, projeções e operações espaciais.

#### **stringi**

- Manipulação avançada de textos — inclui funções para busca, substituição, padronização, conversão de caracteres e tratamento de acentos.

#### **scales**

- Ferramentas para formatar e controlar escalas em gráficos — conversão para percentuais, separadores de milhares, ajuste de cores e tamanhos.

## Leitura dos dados

```
#Tente ler como UTF-8 ou latin1, mas usando fileEncoding e encoding
dados_24 <- read.csv2("CGIL_CNig_2024.csv", sep = ";",
  fileEncoding = "UTF-8", encoding = "UTF-8",
  stringsAsFactors = FALSE)
dados_25 <- read.csv2("CGIL_CNig_jan-set2025.csv", sep = ";",
  fileEncoding = "UTF-8", encoding = "UTF-8",
  stringsAsFactors = FALSE)

# Juntandos os dados
df <- rbind(dados_24, dados_25)
```

```
#Funções para explorar o Data Frame:
```

```
## Abre o dataframe em uma aba visual (planilha)
##no RStudio para inspeção manual dos dados.
#View(df)
```

```
## Mostra as primeiras linhas
##(por padrão, 6) do dataframe.
#head(df)
```

```
## Exibe um resumo estatístico das variáveis do dataframe
## (média, mediana, mínimo, máximo etc.).
#summary(df)
```

```
## Mostra a estrutura do dataframe:
#tipo de objeto, número de observações, variáveis e seus tipos.
#str(df)
```

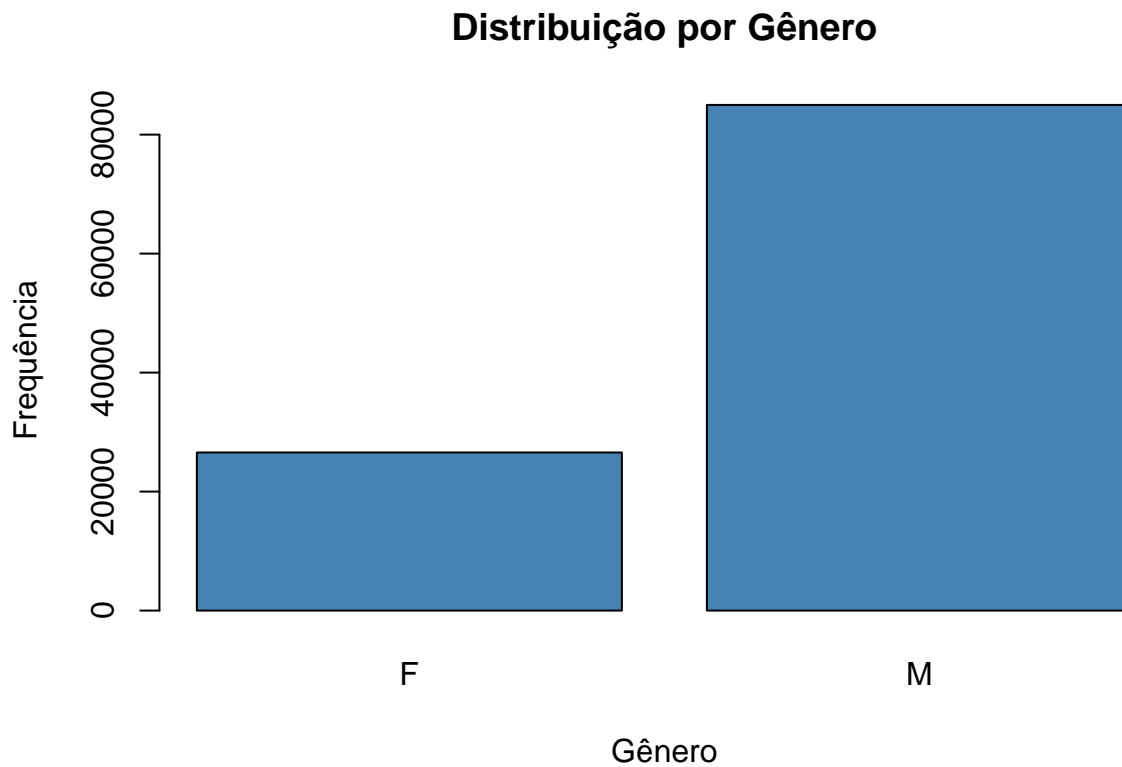
## Gráfico de Barras

### Análise de gênero

```
# Verificando a frequência da variável gênero

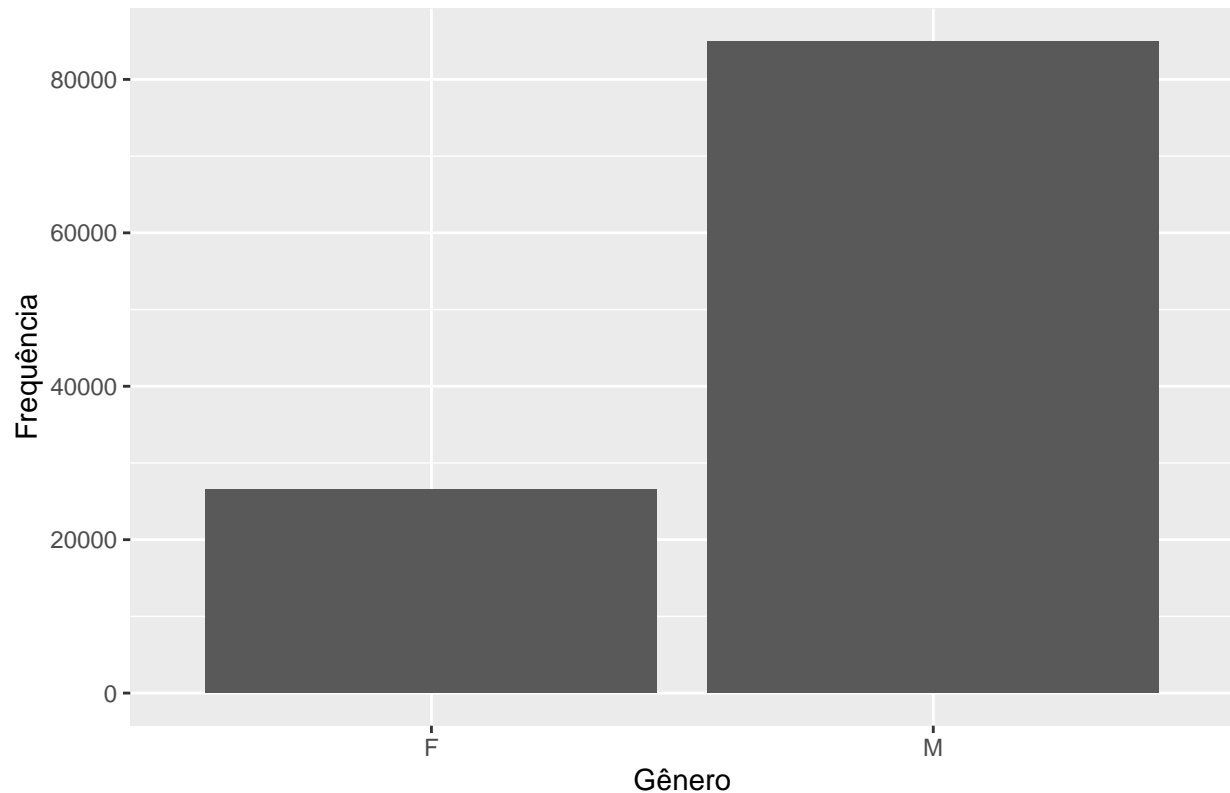
# 1. Versão R Base

# Gráfico de barras simples
# table(df$genero) cria uma tabela de frequência da variável gênero
barplot(table(df$genero), # Dados: frequência de cada categoria de gênero
        main = "Distribuição por Gênero", # Título do gráfico
        xlab = "Gênero", # Rótulo do eixo X
        ylab = "Frequência", # Rótulo do eixo Y
        col = "steelblue") # Cor das barras (azul aço)
```



```
# 2. Versão ggplot2(simples)
```

```
ggplot(df, aes(x = genero)) + # aes = aesthetics, define variável no eixo X
  geom_bar() +                # geom_bar() cria gráfico de barras para variáveis categóricas
  labs(title = "",            # Título (vazio neste caso)
        x = "Gênero",         # Rótulo eixo X
        y = "Frequência")     # Rótulo eixo
```



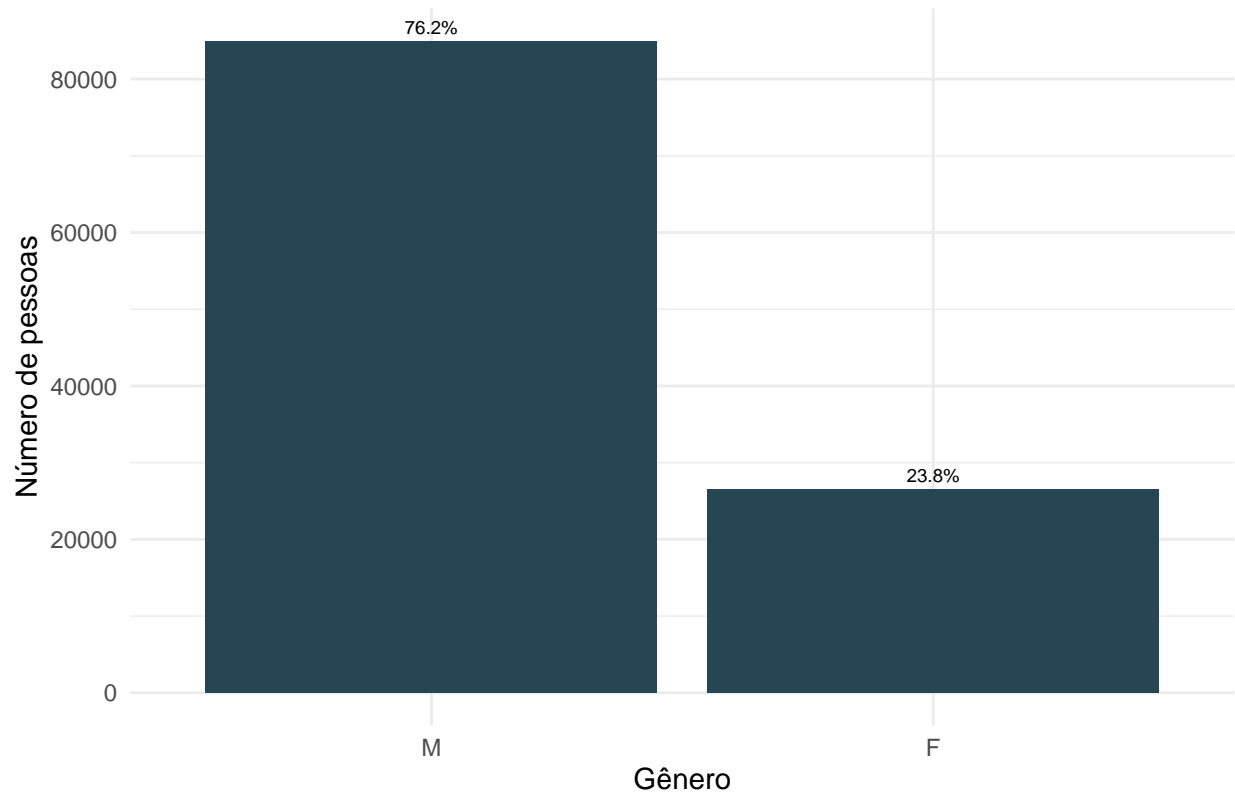
```
#3. Versão ggplot personalizado (mais detalhes)
```

```
## - Acrescentando rótulos de dados
## - Mudando as cores das colunas
## - Mudando nome dos eixos
## - Mudando o tema
## - Reordenando a coluna pela maior frequência
```

```
# Primeiro calculamos estatísticas antes de plotar para maior eficiência
df_summary_gen <- df %>%                # Usando pipe (%>%) para encadear operações
  count(genero) %>%                     # Conta frequência de cada categoria de gênero
  mutate(porcentagem = n / sum(n) * 100, # Calcula porcentagem de cada categoria
        label = paste0(round(porcentagem, 1), "%")) # Cria rótulo formatado (ex: "45.5%")
```

```
ggplot(df_summary_gen, aes(x = reorder(genero, -n), y = n)) +
  # reorder(genero, -n) ordena as categorias pela frequência (decrecente)
  geom_col(fill = "#264653") +
```

```
# fill = "#264653" define cor específica
geom_text(aes(label = label), # Adiciona rótulos com as porcentagens
  vjust = -0.5, # Posiciona texto acima das barras (vertical)
  size = 2.5) + # Tamanho da fonte dos rótulos
labs(title = "", # Título do gráfico
  x = "Gênero", # Rótulo eixo X
  y = "Número de pessoas") + # Rótulo eixo Y
theme_minimal() # Tema
```



## Análise da faixa etária

```
# Primeiro, calcular a idade a partir da data de nascimento
df$data_nascimento <- as.Date(df$data_nascimento)

# Utilizar uma data de referência para o cálculo da idade
# Para fins de exemplificação, utilizaremos a data do mesmo mês em que
# foi concedida a autorização
data_ref <- as.Date(paste(df$ano, df$mes, "01", sep = "-"))

# Atribuir a variável idade
# Calcula a diferença em dias e converte para anos
# (considerando anos bissextos com 365.25)
# floor() arredonda para baixo, pegando apenas a parte inteira (anos completos)
df$idade <- floor(difftime(data_ref, df$data_nascimento, unit = "days")/365.25)

# CRIANDO FAIXAS DE IDADE
# Para fins de exemplo didático, vamos adotar que idades menores que
# 15 serão consideradas NA
df$faixa_etaria[df$idade < 15] <- NA
df$faixa_etaria[df$idade >= 15 & df$idade < 20] <- "Menor que 20"
df$faixa_etaria[df$idade >= 20 & df$idade <= 34] <- "20-34"
df$faixa_etaria[df$idade >= 35 & df$idade <= 49] <- "35-49"
df$faixa_etaria[df$idade >= 50 & df$idade <= 64] <- "50-64"
df$faixa_etaria[df$idade >= 65] <- "65 ou mais"

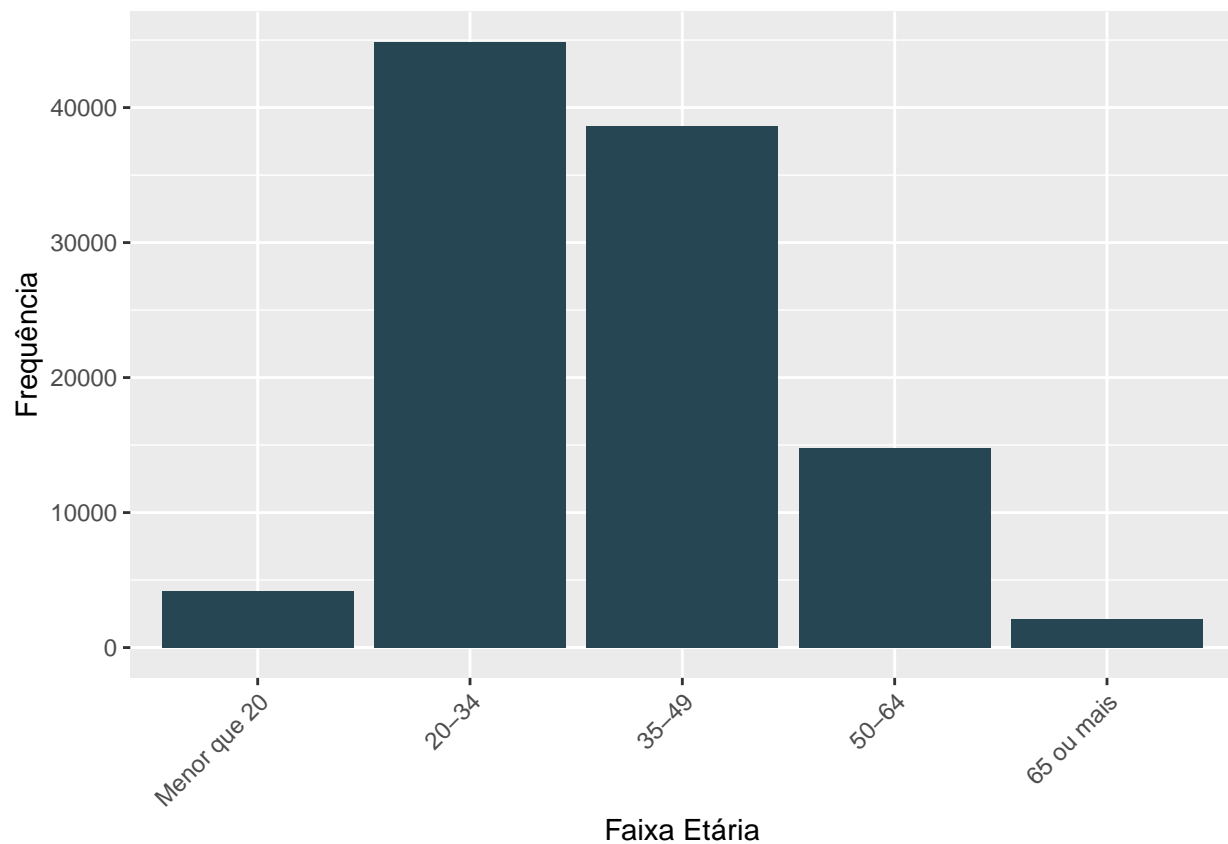
# DEFINIR A ORDEM ESPECÍFICA DAS FAIXAS ETÁRIAS
# Criar vetor com a ordem desejada para as categorias
ordem_faixas <- c("Menor que 20", "20-34", "35-49", "50-64", "65 ou mais")

# Converter para factor com a ordem especificada
# Isso garante que as faixas apareçam na ordem correta nos gráficos e análises
df$faixa_etaria <- factor(df$faixa_etaria, levels = ordem_faixas)
```

```

# CRIAR GRÁFICO DE BARRAS
# Filtrar dados removendo valores NA (idades < 15)
# Criar gráfico de barras da distribuição por faixa etária
df %>%
  filter(!is.na(faixa_etaria)) %>% # Remove observações com faixa etária NA
  ggplot(aes(x = faixa_etaria)) +
  geom_bar(fill = "#264653") + # Cor das barras em azul escuro
  labs(x = "Faixa Etária",      # Rótulo do eixo X
       y = "Frequência") +     # Rótulo do eixo Y
  # Inclina textos do eixo X para melhor legibilidade
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



## Análises bivariadas

### Relação entre ocupação e escolaridade

```
# AJUSTANDO A VARIÁVEL ESCOLARIDADE
# Padronizar e categorizar os níveis de escolaridade para facilitar a análise
# Adiciona prefixos numéricos para ordenação correta

df$escolaridade <- iconv(df$escolaridade, from = "", to = "UTF-8")

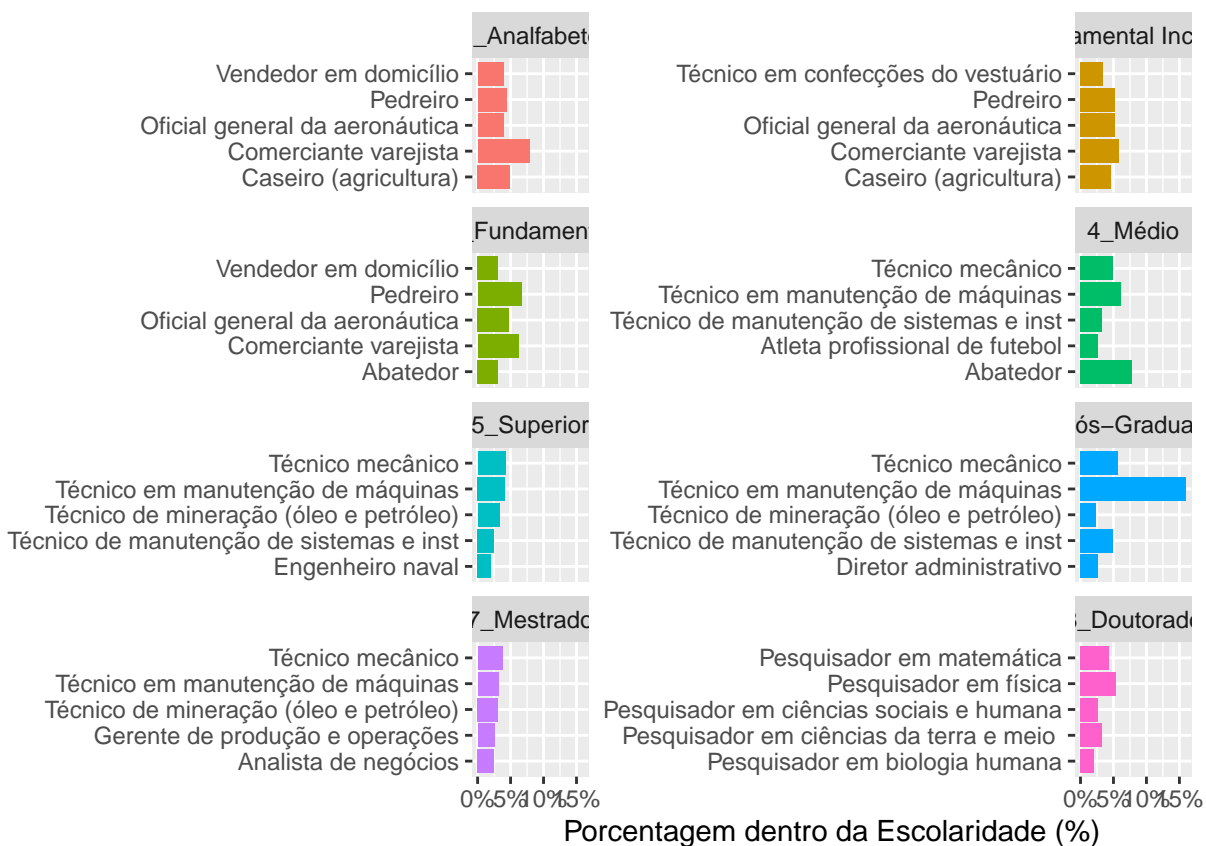
# --- Ajuste dos níveis de escolaridade ---
df$escolaridade[df$escolaridade == "Analfabeto"] = "1_Analfabeto"
df$escolaridade[df$escolaridade == "Doutorado"] = "8_Doutorado"
df$escolaridade[df$escolaridade == "Mestrado"] = "7_Mestrado"
df$escolaridade[df$escolaridade == "Especialização"] = "6_Pós-Graduação"
df$escolaridade[df$escolaridade == "Fundamental Completo"] = "3_Fundamental"
df$escolaridade[df$escolaridade == "Fundamental Incompleto"] = "2_Fundamental Incompleto"
df$escolaridade[df$escolaridade == "Médio Completo"] = "4_Médio"
df$escolaridade[df$escolaridade == "Médio Incompleto"] = "3_Fundamental"
df$escolaridade[df$escolaridade == "Pós-doutorado"] = "8_Doutorado"
df$escolaridade[df$escolaridade == "Superior Completo"] = "5_Superior"
df$escolaridade[df$escolaridade == "Superior Incompleto"] = "4_Médio"

# IDENTIFICAR AS PRINCIPAIS OCUPAÇÕES POR NÍVEL DE ESCOLARIDADE
# - Calcular frequências e porcentagens das ocupações dentro de cada nível
# educacional
# - Selecionar as 5 ocupações mais frequentes por escolaridade

top_ocupacoes_pct <- df %>%
  count(escolaridade, codigo_cbo, descricao_cbo) %>%
  group_by(escolaridade) %>%
  mutate(
    total_escolaridade = sum(n),
    porcentagem = n / total_escolaridade * 100
  ) %>%
  slice_max(n, n = 5) %>%
  ungroup() %>%
  mutate(
    descricao_curta = str_sub(descricao_cbo, 1, 40),
    descricao_curta = iconv(descricao_curta, from = "", to = "UTF-8"),
    escolaridade = iconv(escolaridade, from = "", to = "UTF-8") # <- chave
  )
```

```
# CRIAR GRÁFICO DE BARRAS FACETADO
# Mostra as ocupações mais comuns para cada nível de escolaridade

ggplot(top_ocupacoes_pct, aes(x = porcentagem, y = descricao_curta)) +
  geom_col(aes(fill = escolaridade)) + # Barras coloridas por escolaridade
  # Cria painéis separados por escolaridade
  facet_wrap(~ escolaridade, scales = "free_y", ncol = 2) +
  labs(
    x = "Porcentagem dentro da Escolaridade (%)",      # Rótulo do eixo X
    y = ""                                              # Rótulo do eixo Y vazio
  ) +
  # Formata eixo X como porcentagem
  scale_x_continuous(labels = scales::percent_format(scale = 1)) +
  theme(legend.position = "none")
```

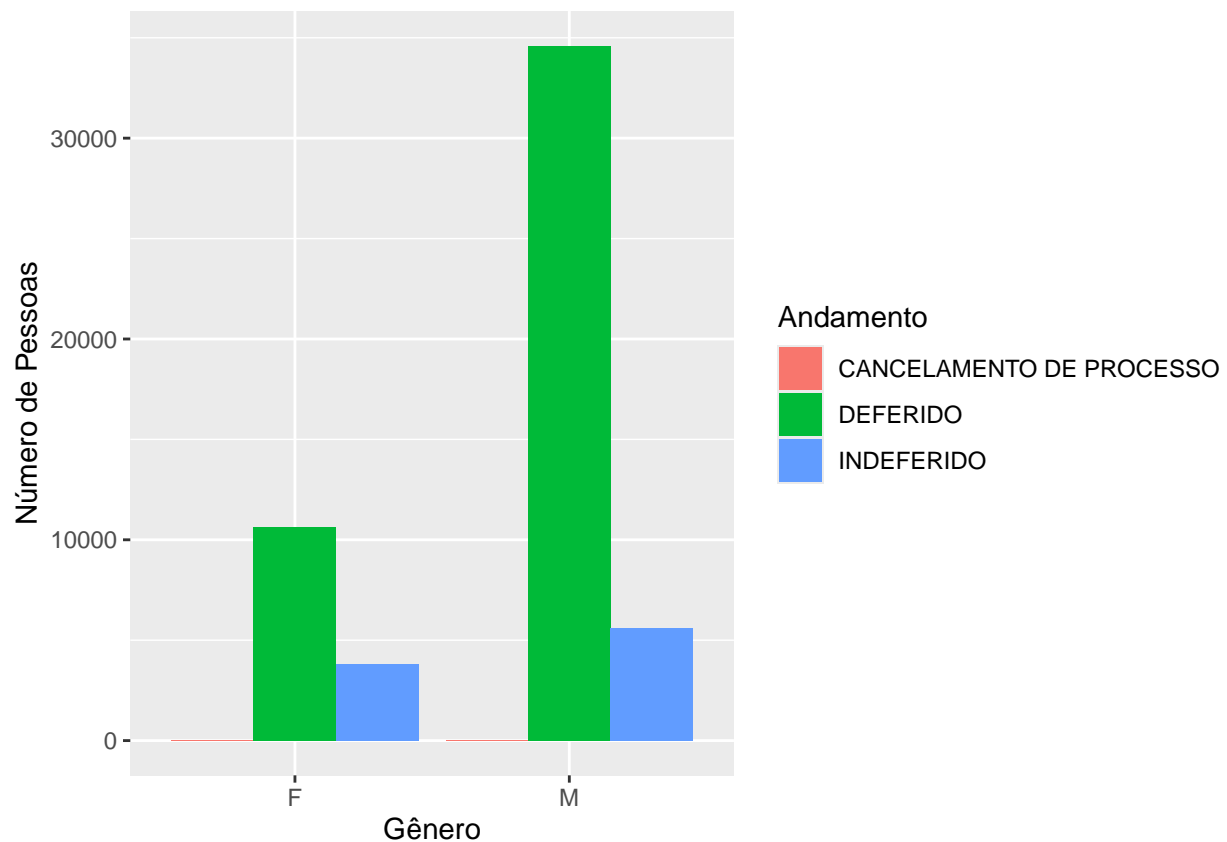


## Gênero vs Andamento

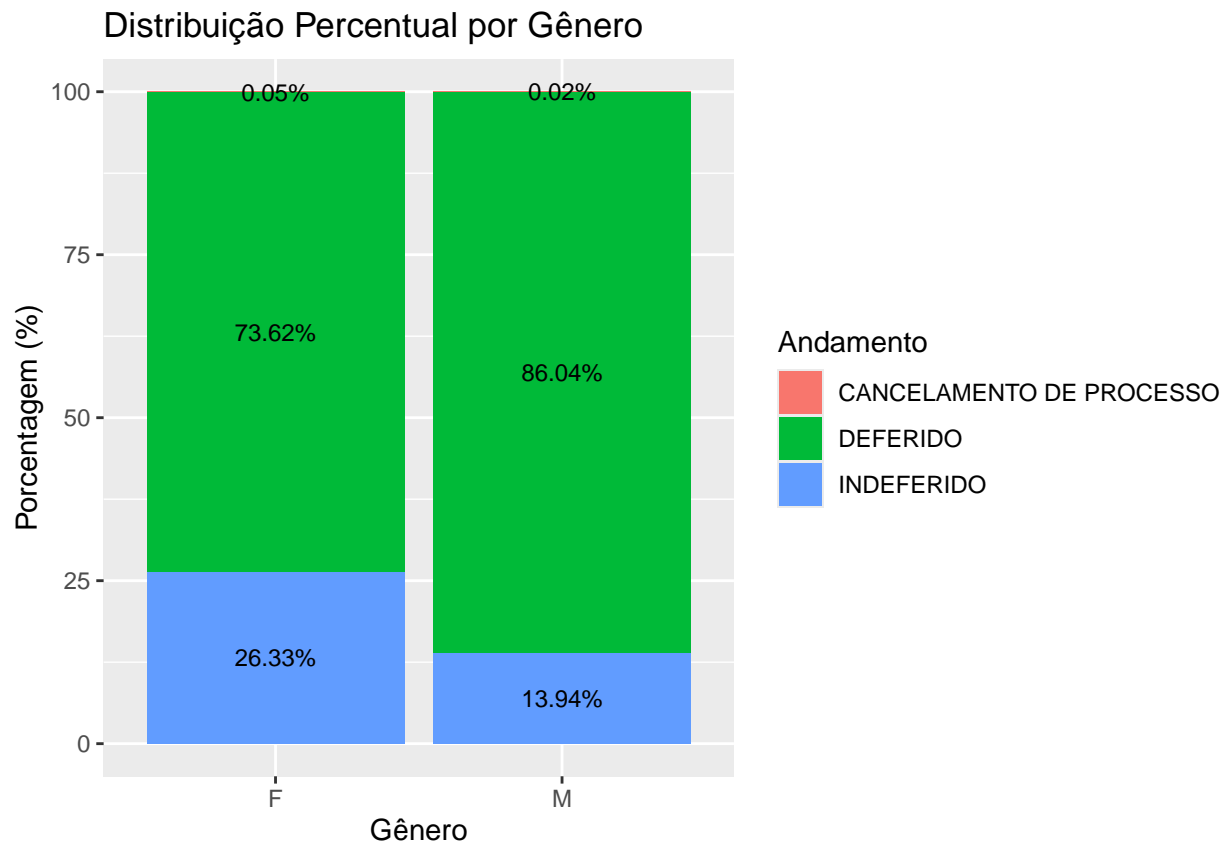
```
## - Inclusão da 2ª variável (análise bivariada)

# PREPARA DADOS AGRUPADOS
# Filtra apenas dados de 2025 e calcula estatísticas por gênero e andamento
df_andamento_genero <- df %>%
  filter(ano == 2025) %>% # Filtra apenas o ano de 2025
  group_by(andamento, genero) %>% # Agrupa por tipo de andamento e gênero
  summarise(n = n()) %>% # Conta número de casos em cada combinação
  group_by(genero) %>% # Reagrupa apenas por gênero
  mutate(
    porcentagem = n / sum(n) * 100, # Calcula porcentagem dentro de cada gênero
    label = paste0(round(porcentagem,2), "%") # Cria rótulo formatado para o gráfico
  )

# OPÇÃO 1: BARRAS LADO A LADO (position = "dodge")
# Mostra comparação direta dos valores absolutos entre gêneros
ggplot(df_andamento_genero, aes(x = genero, y = n, fill = andamento)) +
  geom_bar(stat = "identity", position = "dodge") + # Barras lado a lado (dodge)
  labs( y = "Número de Pessoas", # Eixo Y: contagem absoluta
        x = "Gênero", # Eixo X: categorias de gênero
        fill = "Andamento") # Legenda: tipos de andamento
```



```
# OPÇÃO 2: BARRAS EMPILHADAS COM PORCENTAGEM
# Mostra distribuição percentual dentro de cada gênero
ggplot(df_andamento_genero, aes(x = genero, y = porcentagem, fill = andamento)) +
  geom_bar(stat = "identity", position = "stack") + # Barras empilhadas (stack)
  geom_text(aes(label = label), # Adiciona rótulos com porcentagens
            position = position_stack(vjust = 0.5), # Posiciona texto no meio das barras
            size = 3) + # Tamanho do texto
  labs(title = "Distribuição Percentual por Gênero",
        y = "Porcentagem (%)", # Eixo Y: porcentagem
        x = "Gênero", # Eixo X: categorias de gênero
        fill = "Andamento")
```

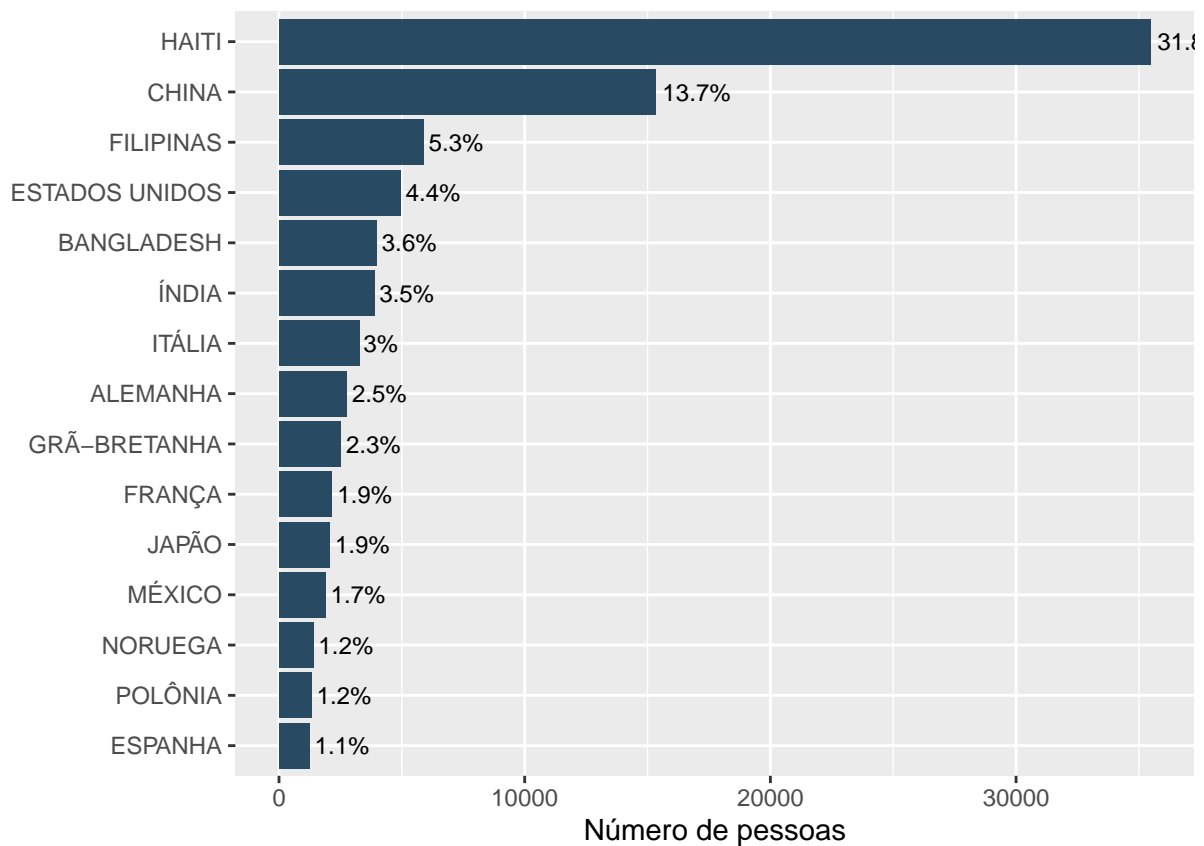


## Análise de países de origem

```
# TOP 15 PAÍSES MAIS FREQUENTES
# Calcula a frequência e porcentagem de cada país no dataset
df_summary_pais <- df %>%
  count(pais) %>% # Conta o número de ocorrências por país
  mutate(
    porcentagem = n / sum(n) * 100, # Calcula a % em relação ao total
    label = paste0(round(porcentagem, 1), "%") # Cria rótulo formatado com 1 casa decimal
  ) %>%
  arrange(desc(n)) %>% # Ordena do maior para o menor número de ocorrências
  head(15) # Seleciona apenas os 15 países mais frequentes

df_summary_pais$pais <- iconv(df_summary_pais$pais, from = "", to = "UTF-8")

# GRÁFICO DE BARRAS HORIZONTAIS
# Usa gráfico horizontal para melhor visualização dos nomes dos países
ggplot(df_summary_pais, aes(x = reorder(pais, n), y = n)) +
  geom_col(fill = "#284b63") + # Barras na cor azul escuro
  coord_flip() + # Inverte eixos para deixar barras horizontais
  geom_text(aes(label = label), # Adiciona rótulos com as porcentagens
    hjust = -0.1, # Posiciona texto à direita das barras
    size = 3) + # Define tamanho da fonte
  labs(x = "", # Eixo X vazio (países ficam no eixo Y após coord_flip)
    y = "Número de pessoas")
```



**Análise extra:** Distribuição das taxas de deferimento dos países de origem por faixas de num. de solicitação

```

# CALCULAR TAXAS DE DEFERIMENTO POR PAÍS
df_taxa_pais <- df %>% filter(ano == 2025) %>%
  group_by(pais, andamento) %>% # Agrupa por país e tipo de andamento
  summarise(n = n(), .groups = "drop") %>% # Conta casos, remove agrupamento
  pivot_wider(
    names_from = andamento, # Transforma andamento em colunas
    values_from = n,        # Valores vêm da contagem
    values_fill = 0         # Preenche NAs com 0
  ) %>%
  mutate(
    solicitacoes = rowSums(across(where(is.numeric))), # Total de solicitações por país
    taxa_deferido = DEFERIDO / solicitacoes, # Calcula taxa de deferimento
    taxa_indeferido = INDEFERIDO / solicitacoes # Calcula taxa de indeferimento
  ) %>%
  arrange(desc(taxa_deferido)) # Ordena por taxa de deferimento (maior primeiro)

# ANALISAR DISTRIBUIÇÃO DO NÚMERO DE SOLICITAÇÕES POR PAÍS
summary(df_taxa_pais$solicitacoes) # Mostra quartis, mediana, média, etc.

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0      4.0     19.5   303.7   112.5 21468.0

```

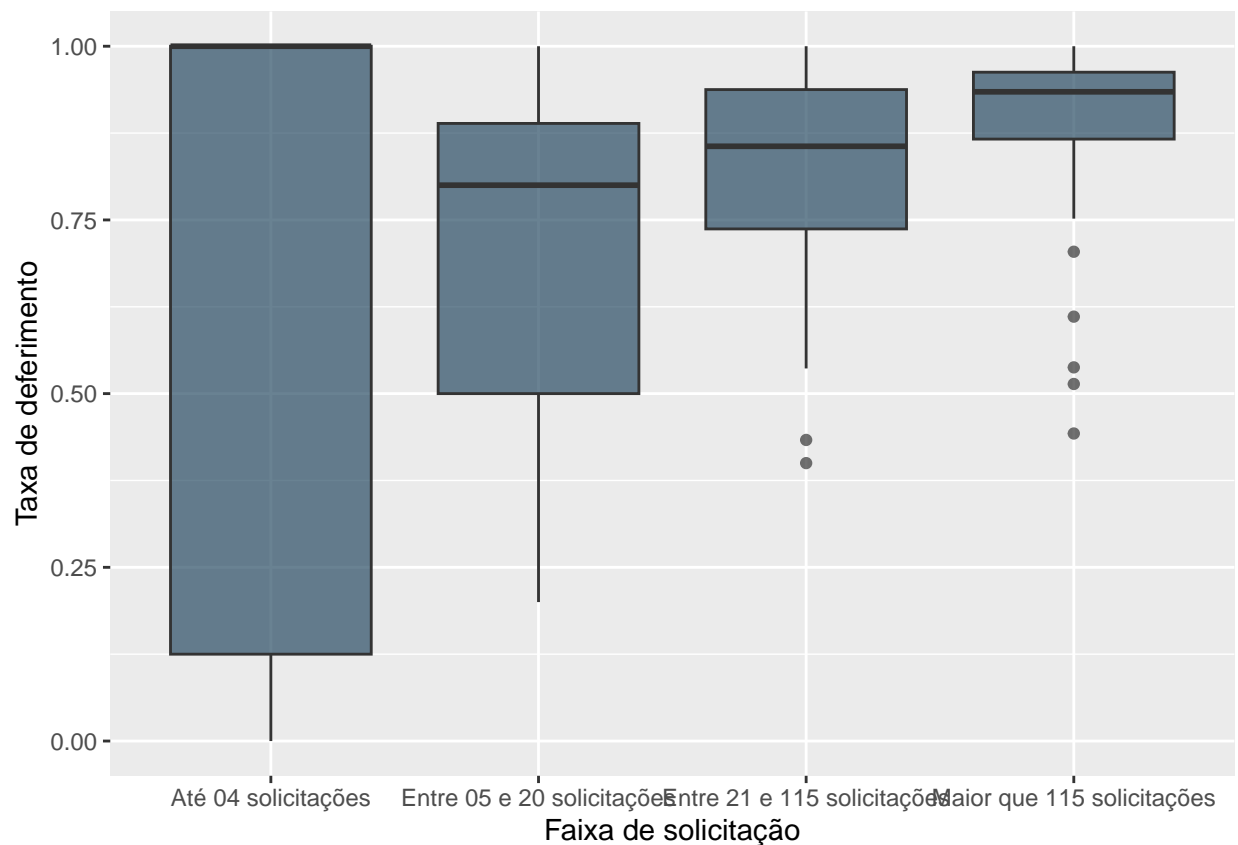
```

# CRIANDO FAIXAS DE SOLICITAÇÕES BASEADAS NOS QUARTIS:
# 1 a 4      (primeiro quartil)
# 5 a 20     (segundo quartil)
# 21 a 115   (terceiro quartil)
# maior que 115 (quarto quartil)

df_taxa_pais$faixa_solicitacao <- case_when(
  df_taxa_pais$solicitacoes <= 4 ~ "Até 04 solicitações",
  df_taxa_pais$solicitacoes > 4 &
    df_taxa_pais$solicitacoes <= 20 ~ "Entre 05 e 20 solicitações",
  df_taxa_pais$solicitacoes > 20 &
    df_taxa_pais$solicitacoes <= 115 ~ "Entre 21 e 115 solicitações",
  df_taxa_pais$solicitacoes > 115 ~ "Maior que 115 solicitações"
)

# GRÁFICO DE BOXPLOT DA TAXA DE DEFERIMENTO POR FAIXA DE SOLICITAÇÃO
ggplot(df_taxa_pais,
  aes(x = faixa_solicitacao, y = taxa_deferido)) +
  geom_boxplot(fill = "#284b63", alpha = 0.7) +      # Boxplot com cor e transparência
  labs(x = "Faixa de solicitação",
       y = "Taxa de deferimento")

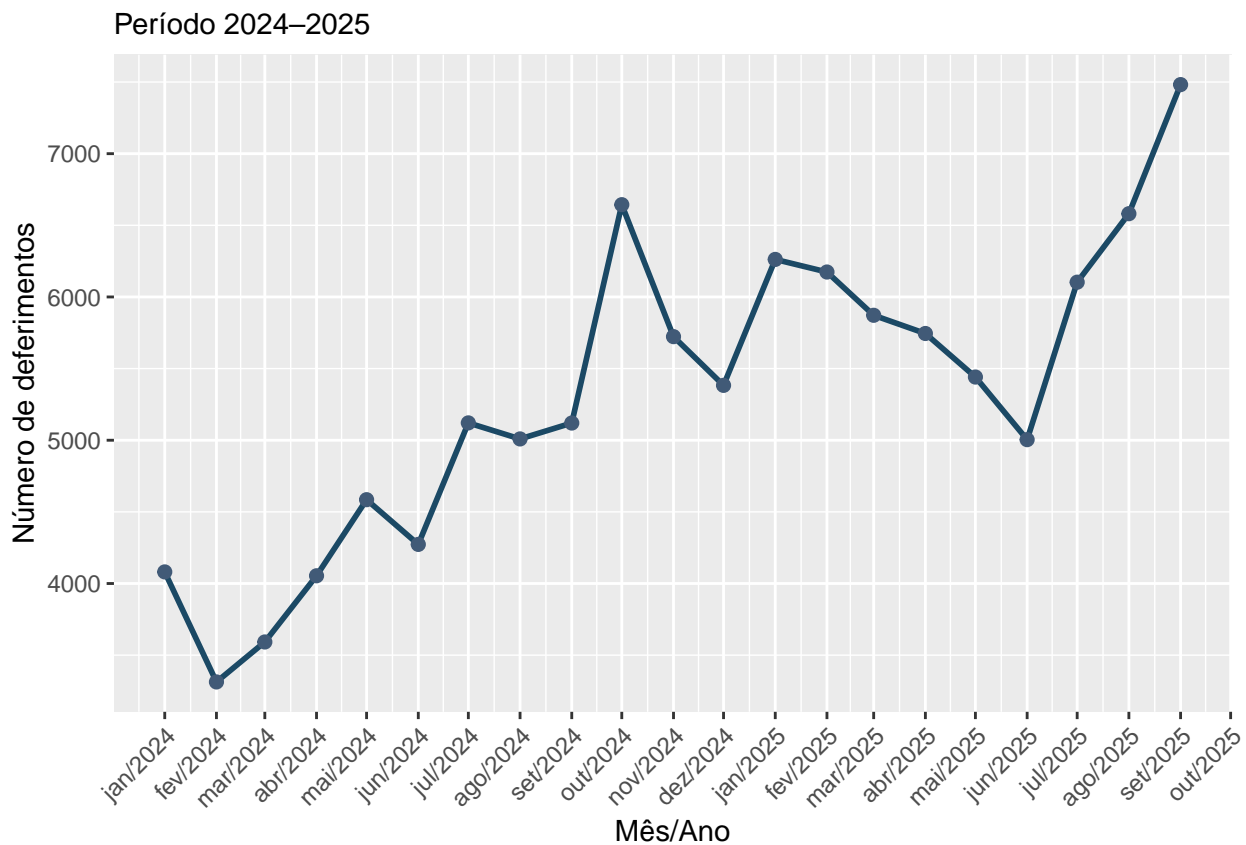
```



## Gráficos de Linhas

```
# Agrupa dados por ano e mês, criando variável de data formatada
df_serie <- df %>%
  count(ano, mes) %>% # Conta processos por ano e mês
  mutate(
    data = as.Date(paste(ano, mes, "01", sep = "-")) # Cria data no formato AAA-MM-DD
  ) # Usa "01" como dia padrão para primeiro do mês

ggplot(df_serie, aes(x = data, y = n, group = 1)) + # group=1 conecta pontos em série
  geom_line(color = "#1b4965", linewidth = 1) + # Linha principal da série
  geom_point(color = "#415a77", size = 2) + # Pontos sobre a linha
  scale_x_date(
    date_labels = "%b/%Y", # Formata datas como "Jan/2024"
    date_breaks = "1 month" # Quebra do eixo a cada mês
  ) +
  # scale_y_continuous(
  #   breaks = seq(0, max(df_serie$n, na.rm = TRUE), by = 500) # Escala Y de 500 em 500
  # ) +
  labs(
    x = "Mês/Ano", # Rótulo eixo X
    y = "Número de deferimentos", # Rótulo eixo Y
    subtitle = "Período 2024-2025" # Subtítulo
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Inclina labels do eixo X 45°
  )
```



## Mapas de calor

### Visualização geográfica - Distribuição por UF

```
# PREPARAR DADOS: Contar imigrantes deferidos por UF em 2025
df_summary_uf <- df %>%
  filter(andamento == "DEFERIDO", ano == 2025) %>% # Filtra apenas deferimentos de 2025
  count(uf_estrangeiro, name = "n_imigrantes")      # Conta por UF

# CARREGAR MAPA DO BRASIL: Obter shapes dos estados brasileiros
# Esse objeto tem o formato sf (Simple Features), com polígonos de cada UF
mapa_brasil_uf <- read_state(showProgress = T, year = 2020) # Carrega mapa dos estados

## Using year/date 2020

# UNIFORMIZANDO OS CARACTERES DOS ESTADOS - Padronizar nomes para fazer o merge

# Padroniza nomes no mapa: remove acentos e converte para maiúsculas
mapa_brasil_uf$name_state <- mapa_brasil_uf$name_state %>%
  iconv(from = "UTF-8", to = "ASCII//TRANSLIT") %>% # Remove acentos
  toupper()                                          # Transforma para caixa alta

# Padroniza nomes nos dados: remove acentos e converte para maiúsculas
df_summary_uf$uf_estrangeiro <- df_summary_uf$uf_estrangeiro %>%
  iconv(from = "UTF-8", to = "ASCII//TRANSLIT") %>% # Remove acentos
  toupper()                                          # Transforma para caixa alta

# COMBINAR DADOS COM MAPA: Juntar informações de imigrantes com shapes dos estados
mapa_uf <- merge(
  x = df_summary_uf,          # Dados com contagem de imigrantes
  y = mapa_brasil_uf,        # Shapes dos estados
  by.x = "uf_estrangeiro",    # Coluna de união nos dados
  by.y = "name_state",        # Coluna de união no mapa
  all.x = TRUE                # Mantém todos os estados dos dados
)
```

*# 1. ESBOÇO DO MAPA BASE - Visualização inicial sem dados*

```
ggplot(mapa_brasil_uf) +  
  geom_sf(fill = "#457b9d", color = "white") + # Desenha estados com cor uniforme  
  theme_void()                                # Remove todos os elementos do tema
```



```

# 2. MAPA TEMÁTICO - Adicionar as informações de número de imigrantes
mapa_uf <- st_as_sf(mapa_uf) # Garante que é objeto sf (Simple Features)

# Cria mapa coroplético (cores proporcionais aos valores)
ggplot(mapa_uf) +
  geom_sf(aes(fill = n_imigrantes), # Preenche por número de imigrantes
          color = "white", size = 0.2) + # Bordas brancas finas
  scale_fill_viridis_c(
    option = "plasma", # Escala de cores "plasma"
    trans = "log10", # Transformação logarítmica (melhor visualização)
    name = "Nº de Imigrantes (escala log)", # Título da legenda
    labels = scales::number_format(accuracy = 1) # Formata rótulos da legenda
  ) +
  theme_minimal() +
  labs(
    title = "",
    subtitle = "",
    caption = ""
  ) +
  theme(
    axis.text = element_blank(), # Remove textos dos eixos
    axis.ticks = element_blank(), # Remove marcadores dos eixos
    panel.grid = element_blank() # Remove grade do fundo
  )

```

