



Lucas Sanches Gomes

README

Desafio técnico – Renova BR

São Paulo

2023

INTRODUÇÃO E OBJETIVOS

A segunda etapa do processo seletivo é composta por um desafio técnico, o qual foi disponibilizado duas bases de dados, uma referente aos resultados do primeiro turno das eleições de 2020 para prefeito e vereadores, e outra sobre o perfil do eleitorado.

O intuito geral é analisar os dados do estado de São Paulo e seus municípios.

Foi proposto também a utilização das ferramentas Python, SQL e/ou Excel. Para resolução dos desafios utilizei o Excel para realizar o tratamento, filtragem geral e limpeza dos bancos de dados, tratando o maior número de inconsistências possíveis, e posteriormente utilizei o software R para as demais análises, uma vez que o mesmo é muito similar ao Python, com a pequena diferença de ser voltado para estatísticos. De qualquer modo, ao final deste documento, disponibilizei o código em linguagem Python, que nos fornece a mesma análise e resultados.

O objetivo deste desafio é resolver os seguintes tópicos:

1. Tratamento dos dados e inconsistências.
2. Criação de uma tabela para cada base de dados, com os dados já tratados e estruturados.
3. Consulta JOIN entre tabelas:
 - 3.1. Qual candidato foi mais votado em cada município.
 - 3.2. Em qual município o candidato X foi mais votado.
4. Qual o perfil do eleitorado mais votou em cada candidato.
5. Novas ideias e insights:
6. Códigos de programação

Segue os resultados e interpretações para cada um dos tópicos.

1. Tratamento dos dados e inconsistências

A ideia ao realizar este tópico foi melhorar a visualização dos dados, limpá-los, corrigir dados inseridos de forma incorreta e eliminar informações que não são relevantes e possam atrapalhar análises futuras.

Para isso, utilizei o software Excel, o primeiro passo foi eliminar variáveis (colunas) que não são interessantes para a análise em cada uma das bases.

A respeito da base do perfil do eleitorado:

- DT_GERACAO
- HH_GERACAO
- ANO_DE_ELEICAO
- CD_MUNICIPIO
- CD_MUN_SIT_BIOMETRIA
- DS_MUN_SIT_BIOMÉTRICA
- CD_GENERO
- CD_ESTADO CIVIL
- CD_FAIXA ETÁRIA
- CD_GRAU_DE_ESCOLARIADADE
- QT_ELEITORES_BIOMETRIA

As três primeiras variáveis dizem respeito apenas a criação da base, a terceira apresenta o ano de 2020 e não é relevante, os códigos (CD) podem ser considerados como variáveis “duplicadas” em relação a DS, por isso são excluídas.

A respeito da DS_MUN_SIT_BIOMETRICA e QT_ELEITORES_BIOMETRIA não apresentam uma significância para o tipo de análise pedida, uma vez que independente delas o eleitor pode votar da mesma maneira.

Já na base do resultado do primeiro turno:

- DT_GERACAO
- HH_GERACAO
- ANO_ELEICAO
- NM_TIPO_ELEICAO

- CD_TIPO_ELEICAO
- CD_PLEITO
- DT_PLEITO
- NR_TURNO
- CD_ELEICAO
- DS_ELEICAO
- CD_MUNICIPIO
- NR_SECAO
- NR_LOCAL_VOTACAO
- CD_CARGO_PERGUNTA
- NR_PARTIDO
- NM_PARTIDO
- DT_BU_RECEBIDO
- CD_TIPO_URNA
- DS_TIPO_URNA
- CD_TIPO_VOTAVEL
- NR_VOTAVEL
- NR_URNA_EFETIVADA
- CD_CARGA_1_URNA_EFETIVADA
- CD_CARGA_2_URNA_EFETIVADA
- CD_FLASHCARD_URNA_EFETIVADA
- DT_CARGA_URNA_EFETIVADA
- DS_CARGO_PERGUNTA_SECAO
- DS_AGREGADAS
- DT_ABERTURA
- DT_ENCERRAMENTO
- QT_ELEITORES_BIOMETRIA_NH
- DT_EMISSAO_BU
- NR_JUNTA_APURADORA
- NR_TURMA_APURADORA
- DS_TIPO_VOTÁVEL

Para a exclusão destas variáveis utilizei a mesma lógica anterior, eliminando as redundantes, nulas, “duplicadas” e as irrelevantes para a análise.

Formatei as células das variáveis restantes de ambas as bases para que garanta que números estejam em formato numérico, texto quando houver uma descrição (exemplo grau de escolaridade), e assim sucessivamente.

Na base do perfil do eleitorado, utilizei a filtragem do Excel para manter na base de dados apenas os que se referam a SG_UF de São Paulo, uma vez que queremos analisar o estado de São Paulo e seus municípios, pois a base continha outros estados. Após filtrar os dados por sigla, apenas os copiei e criei uma nova planilha para obter dados apenas do estado de interesse.

Na base de turno, coluna SG_PARTIDO, todas as células que possuem #NULO# devem ser mudadas, retirando o caractere especial “#”, para não prejudicar a análise futura.

Para isso selecionei a coluna SG_PARTIDO, na página inicial usei a ferramenta localizar e selecionar para substituir todas as células #NULO# por NULO, removendo o caractere especial e limpando visualmente a base, com um total de 38859 substituições, isto é, 38859 votos nulos.

Fiz o mesmo para a coluna NM_VOTAVEL, primeiro com as células contidas "Branco" para "BRANCO", com 20118 substituições e votos em branco, e depois para "Nulo", deixando em maiúsculo, possibilitando uma melhor padronização dos dados e resultando num total de 19439 substituições, ou seja, votos nulos totais no estado de sp.

2. Criação de uma tabela para cada base de dados

Para resolução deste tópico, selecionei todas as células com o comando ctrl+shift+space, na aba inserir selecionei "tabela" para melhorar ainda mais a visualização e ser possível selecionar informações de cada coluna conforme o interesse, tornando as tabelas mais dinâmicas com os dados já tratados e estruturados.

As bases disponíveis no repositório são essas.

3. Consulta JOIN entre tabelas

3.1. Qual candidato foi mais votado em cada município

Através do software R, filtrei os cargos de Prefeito e Vereador separadamente, criei funções que agrupavam por município (para que eles não se repitam) os candidatos. Por fim realizei um JOIN através da função `left_join` criando uma tabela que me informa o município, o prefeito e vereador mais votado nos mesmos.

NM_MUNICIPIO	Prefeito_mais_votado	Vereador_mais_votado
<chr>	<chr>	<chr>
1 AGUDOS	FERNANDO OCTAVIANI	JUNINHO ARTIOLI
2 ALAMBARI	JOÃO PAULO CARIOCA	VITÓRIO DA AMBULAN
CIA		
3 ÁLVARO DE CARVALHO	ADILSON	EDER
4 ALVINLÂNDIA	ABIGAIL	JADDER BERGAMIN
5 AMPARO	CARLOS ALBERTO	CARLOS CAZOTTI
6 ANDRADINA	MARIO CELSO	WELLINGTON
7 ANHEMBI	ROGERIÃO	RODRIGO MOURA
8 APIAÍ	DONIZETTI BARBOSA	SILMERIA DA SAUDE
9 ARAÇATUBA	DILADOR BORGES	DR JAIME
10 ARAMINA	MARIA	TUNIQUEINHO
11 ARAPEÍ	RENÊ	MAICON UILTON
12 ARARAQUARA	EDINHO SILVA	SILVANI SILVA
13 ARARAS	PEDRINHO ELISEU	NULO
14 AREIAS	PAULO HENRIQUE PH	EDINHO DA VAN
15 ASSIS	JOSE FERNANDES	BRANCO
16 ATIBAIA	EMIL ONO	JULIO MENDES
17 AVARÉ	JÔ SILVESTRE	BRANCO
18 BANANAL	DR WILLIAM	CRISTINA FONTES
19 BARIRI	ABELARDINHO	AIRTON PEGORARO
20 BARRA DO CHAPÉU	IVAN DA ANUNCIATA	RUBINHO DO ÔNIBUS
21 BARRETOS	PAULA LEMOS	DR VAGNER CHIAPETT
I		
22 BATATAIS	JUNINHO GASPAR	BRANCO
23 BAURU	DR RAUL	BRANCO
24 BEBEDOURO	FERNANDO PIFFER	MANÉ DA PARABOLICA
25 BIRIGUI	LEANDRO MAFFEIS	NULO
26 BOFETE	TORÃO	CARLINHOS DO CARMO
27 BOM JESUS DOS PERDÕES	DR BENEDITO	HÉLIO GONÇALVES
28 BORÁ	LUIZ	CECILIA
29 BOTUCATU	PARDINI	RICARDO DO MONTE M
OR		
30 BRAGANÇA PAULISTA	JESUS CHEDID	DRA. ANA MALENGO
31 BREJO ALEGRE	MAYSA	MARQUINHOS
32 BROTAS	LEANDRO CORRÊA	TATU
33 BURITIZAL	DANIEL SARRETA	DONIZETI SAPECA
34 CAÇAPAVA	PÉTALA DO CONVIVER	TUKA
35 CAÇONDE	JOÃO FILIPE	EDVALDO DO TRANSPO
RTE ESCOL...		
36 CAFELÂNDIA	CAPITÃO CAPUTTI	LUIZINHO
37 CAJURU	ALEX MORETINI	GUERREIRO
38 CAMPINAS	ARTUR ORSI	CARLINHOS CAMELO
39 CAMPOS DO JORDÃO	MARCELO PADOVAN	JHONNY SALVINO
40 CANANÉIA	PAULO RANGEL	FERNANDA RANGEL DO
ARIRI		
41 CAPÃO BONITO	JULIO FERNANDO	VALDECI DO TURVO
42 CAPIVARI	VITÃO	ODÉSIO FERRAZ

43	CASA BRANCA	MARCO CÉSAR AGA	RUBINHO DA LAGOA
44	CÁSSIA DOS COQUEIROS	SARGENTO JORGE	LUCÉLIA GONÇALVES
TÉIA			
45	CASTILHO	PAULO BOAVENTURA	DANIEL
46	CATANDUVA	PADRE OSVALDO	MANOEL GOL DE OURO
47	CLEMENTINA	NELSON	JULIANO DO BIDU
48	COLÔMBIA	TUTA	SANDRÃO
49	CONCHAS	BALU	DARCISO
50	COROADOS	TEREZINHA	TÂNIA MIRANDA
51	CRUZEIRO	THALES GABRIEL	ANDERSON FERRER LE
PRECHAL			
52	CUNHA	ZÉ ÉDER	JOÃO CIRINEU
53	DESCALVADO	LUISINHO PANONE	REINALDO NINJA
54	ECHAPORÃ	GUSTAVO	EVERTON PSICOLOGO
55	ELISIÁRIO	CASSIO BERTELLI	MARQUINHO DO ESPOR
TE			
56	FERNÃO	JOSÉ FODRA	BILL
57	FRANCA	ALEXANDRE FERREIRA	BRANCO
58	GÁLIA	RENATO	ZÉ ADAO
59	GARÇA	JOÃO CARLOS DOS SANTOS	LUCAS CATETO
60	GAVIÃO PEIXOTO	ADRIANO	NEY SUB ZERO
61	GUAPIARA	MATHEUS FREITAS	JAIR DO ALEGRE
62	GUARANTÃ	MARCOS FRUGERI	CRISTINA VILARRUBI
A			
63	GUARATINGUETÁ	MARCUS SOLIVA	FERRI
64	IACANGA	ELI CARDOSO	CLAUDIO DARIVA
65	IBITINGA	CRISTINA ARANTES	ZÉ NILSON
66	IGARAPAVA	ZÉ RICARDO	NANAU MOTORISTA
67	IGUAPE	WILSON	FAUSTO JAIRÊ
68	ILHA COMPRIDA	MARA VENTURA	CARLA DE PEDRINHAS
69	ITAJU	LEDINÉL	JUNINHO LOPES
70	ITAOCA	FREDERICO	DARZI SILVA
71	ITAPETINGA	SIMONE MARQUETTO	GÊ DE ARAÚJO
72	ITAPIRAPUÃ PAULISTA	CESAR CONSTRUVALE	XARRET
73	ITATINGA	JOÃO BOSCO	ROSALVA RIBEIRO
74	ITIRAPUÃ	GERSON ALVES	PASTORA RAQUEL
75	ITOBI	MARANATA	JU PEREIRA
76	JAMBEIRO	CASQUINHA	ALAN DA QUITANDA
77	JARINU	DEBORA PRADO	CEARÁ DO CAMPO LAR
GO			
78	JÚLIO MESQUITA	TIRSINHO	RODRIGO DO ZÉ DA C
ARNE			
79	LAVRINHAS	ZEZINHO DO AÇOUGUE	SÉRGIO
80	LUTÉCIA	LEATI	PAULINHO ENFERMEIR
O			
81	MOMBUCA	FORMIGUINHA	FELIPE GIGIO
82	MONTE ALEGRE DO SUL	EDSON RODRIGO	VALTER BUENO
83	MURUTINGA DO SUL	GILSON PIMENTEL	MANINHO
84	NAZARÉ PAULISTA	MURILO	JEFFERSON PEREIRA
85	NOVA EUROPA	LUIZÃO	PROFESSOR JORGE
86	NOVA INDEPENDÊNCIA	FERNANDINHO	PIPA CARMONA
87	OSCAR BRESSANE	PAPINHA	ANSELMO
88	PARAGUAÇU PAULISTA	IAN SALOMÃO	PROFESSORA DELMIRA
89	PARDINHO	DR JOSE LUIZ	RENAN EBURNEO
90	PATROCÍNIO PAULISTA	DR MAURO	VAGUINHO VETERINÁR
IO			
91	PAULISTÂNIA	PAULO	ALEXSANDRO LEITE
92	PEREIRAS	MIGUEL TOMAZELA	NETTO MARQUES
93	PIRATININGA	MAJOR JORGE LUÍS	CLAUDINEI
94	QUELUZ	LAURINDO	ZÉ DA FARMÁCIA
95	RAFARD	FABINHO	PEDRINHO
96	RIBEIRA	ARI DA CESP	NALVA
97	RIBEIRÃO GRANDE	MARCELO NUNES	JOÃO LUIZ DO BOA V
ISTA			
98	SANTA CRUZ DA ESPERANÇA	MARQUIM DO DIDA	ELAINE DO MINEIRIN
HO			
99	SANTA CRUZ DAS PALMEIRAS	ZÉ DA FARMÁCIA	RAFAEL CUSTODIO LA
LA			
100	SANTÓPOLIS DO AGUAPEÍ	LÔI	NATO DO NICO

101	SÃO JOSÉ DO BARREIRO	LÊ BRAGA	ANDERSON DE FORMOS
102	SÃO PAULO	BRUNO COVAS	MARCOS MS
103	SARAPUÍ	PROFESSOR GUSTAVO	LEANDRO TEIXEIRA
104	TABATINGA	DR. EDUARDO	ANGÉLICA DE CURUPÁ
105	TAPIRATIBA	RAMON	ELAINE FERREIRA
106	TORRINHA	RENE BLUMER	MARQUINHO CELIN
107	VALINHOS	CAPITÃ LUCIMARA	DINHO ALMEIDA

3.2. Qual município o candidato foi mais votado

Já aproveitando o JOIN anterior, criei uma função que agrupava os dados, só que dessa vez pelos nomes dos candidatos, criando uma tabela que me informa o nome do candidato, seu cargo o município e sua quantidade de votos.

Não apresentei a tabela inteira aqui para não aumentar a quantidade de páginas, porém ao rodar o código é possível ver a lista completa de todos os candidatos, ela informa todos os candidatos, prefeitos e vereadores.

NM_VOTAVEL	Cargo	Cidade_Mais_Votado	Qt_De_Votos
<chr>	<chr>	<chr>	<dbl>
1 A BELEZINHA DA AMBULÂNCIA	Vereador	TORRINHA	9
2 A MARICATO	Vereador	ITAPETININGA	2
3 AARON TRESNY	Vereador	VALINHOS	2
4 ABDENOR DENTISTA	Vereador	BATATAIS	8
5 ABDIAS	Vereador	AVARÉ	7
6 ABEL DOMINGUES	Vereador	GUAPIARA	45
7 ABELARDINHO	Prefeito	BARIRI	154
8 ABELARDO	Vereador	BOTUCATU	65
9 ABELIM BETTONI	Vereador	GUARATINGUETÁ	10
10 ABELINO	Vereador	BARRETOS	1
# i 15,397 more rows			
# i Use `print(n = ...)` to see more rows			

4. Qual perfil do eleitorado mais votou em cada candidato

Neste tópico encontramos um grande problema.

O JOIN pode sim ser feito entre as bases de dados, principalmente pela coluna NM_MUNICIPIOS, porém, existem grandes chances de obtermos dados errôneos e prejudicarmos as demais análises. Isso se deve a alguns fatores como:

- A base dos resultados do primeiro turno não possui um documento README, o que impossibilita sabermos se cada linha representa as respectivas da base do perfil eleitorado, realizar o JOIN poderia

associar os votos as características dos eleitores de forma equivocada, causando uma análise tendenciosa.

- As duas bases não possuem as mesmas quantidades de linhas, o que reforça o pensamento anterior.

Deste modo, obteríamos resultados e dados não verdadeiros, por isso optei por não realizar este tópico. O erro pode ter sido cometido em algumas etapas, como por exemplo no planejamento da pesquisa, coleta e montagem da base de dados do perfil, que não consta a intenção de voto e/ou o candidato que o eleitor ou grupo votou (tendo essa informação poderíamos confirmar até mesmo se mentiram ou não a responder a pesquisa, baseado nos resultados do primeiro turno) e nem mesmo garante representatividade da amostra ou mesmo resultados não tendenciosos.

Um lado positivo, é que conseguimos realizar nesta base algumas análises interessantes e muito úteis que serão abordadas no próximo tópico.

5. Novas ideias e insights

O desafio nos possibilita irmos além, e trazermos novos horizontes.

Para isso irei realizar algumas análises a respeito das duas bases de dados, utilizando de ferramentas/técnicas estatísticas ensinadas no curso de graduação.

5.1. Melhor visualização do perfil do eleitorado

Neste subtópico irei melhorar ainda mais a visualização da base, até o momento (base disponibilizada no repositório) a mesma apresenta algumas variáveis que se repetem diversas vezes conforme linhas e características da coluna, como por exemplo faixa etária, se repete diversas vezes por linhas mas se relaciona a outras colunas, a ideia então foi agrupar os municípios, para que essas características apareçam de forma resumida, e colunas como a de estado civil por idade apresentem contagem de quantos dos casados de 45 a 49 anos

são deficientes, quantos possuem inclusão de nome social, e etc (através de contagem). Mesclando as características e enumerando suas quantidades divididas por faixas de interesse.

A tabela a seguir representa esta ideia, mas novamente de forma resumida a fim de não ocupar muitas páginas, ao rodar o código é possível visualizar a base completa para todas as cidades e faixas de interesse.

	NM_MUNICIPIO	DS_FAIXA_ETARIA	Soma_QT_ELEITORES_PERFIL	Soma_QT_ELEITORES_DEFICIENCIA	Estados_Civis_Unicos
1	AGUAÍ	18 anos	1	0	CASADO
2	AGUAÍ	19 anos	4	0	CASADO
3	AGUAÍ	20 anos	8	0	SOLTEIRO, CASADO
4	AGUAÍ	21 a 24 anos	960	3	SOLTEIRO, CASADO, SI
5	AGUAÍ	25 a 29 anos	1192	19	SOLTEIRO, CASADO, VI
6	AGUAÍ	30 a 34 anos	1215	10	SOLTEIRO, CASADO, VI
7	AGUAÍ	35 a 39 anos	1227	3	SOLTEIRO, CASADO, VI
8	AGUAÍ	40 a 44 anos	1023	3	SOLTEIRO, CASADO, VI
9	AGUAÍ	45 a 49 anos	933	3	SOLTEIRO, CASADO, VI
10	AGUAÍ	50 a 54 anos	882	7	SOLTEIRO, CASADO, VI
11	AGUAÍ	55 a 59 anos	842	9	SOLTEIRO, CASADO, VI
12	AGUAÍ	60 a 64 anos	748	8	SOLTEIRO, CASADO, VI
13	AGUAÍ	65 a 69 anos	610	17	SOLTEIRO, CASADO, VI
14	AGUAÍ	70 a 74 anos	452	11	SOLTEIRO, CASADO, VI
15	AGUAÍ	75 a 79 anos	202	10	SOLTEIRO, CASADO, VI
16	AGUAÍ	80 a 84 anos	120	3	SOLTEIRO, CASADO, VI
17	AGUAÍ	85 a 89 anos	51	1	SOLTEIRO, CASADO, VI
18	AGUAÍ	90 a 94 anos	21	1	SOLTEIRO, CASADO, VI

Ao rolarmos o scroll inferior ainda das mesmas linhas.

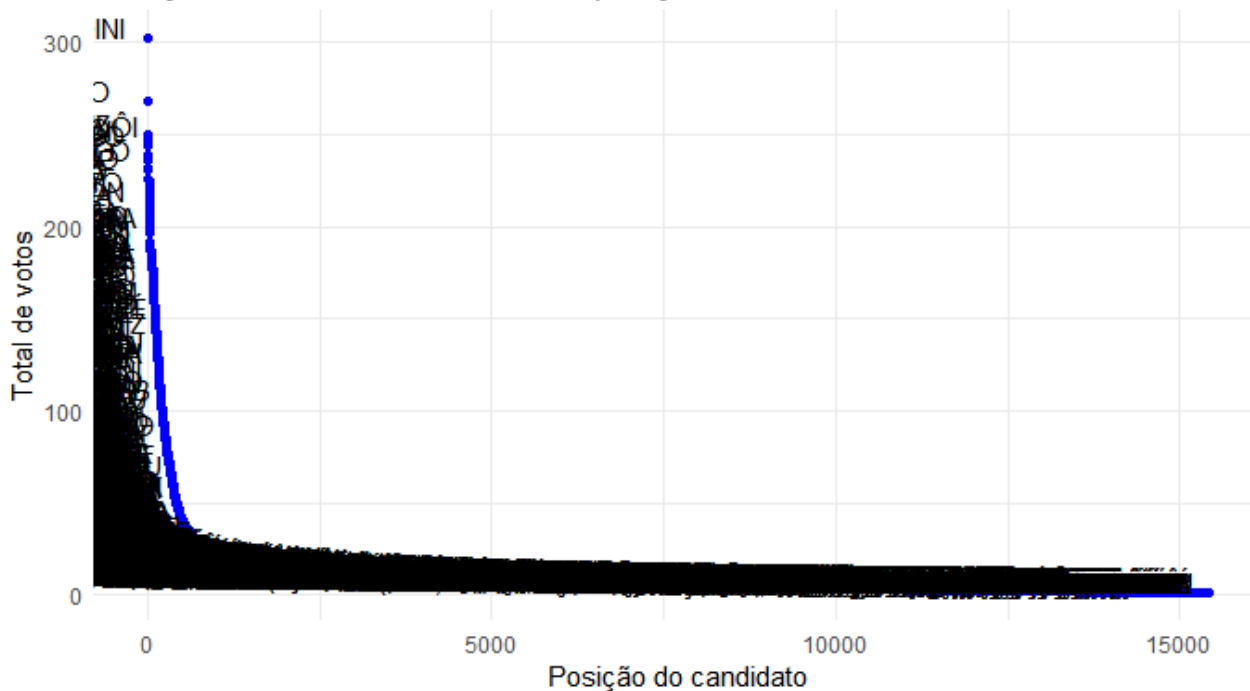
Estados_Civis_Unicos	Contagem_Estados_Civis	Generos_Unicos	Contagem_Generos
CASADO	1	FEMININO	1
CASADO	2	FEMININO	2
SOLTEIRO, CASADO	1, 1	FEMININO	2
SOLTEIRO, CASADO, SEPARADO JUDICIALMENTE, DIVORCIA...	6, 2, 1, 8	FEMININO	17
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE, DI...	6, 5, 3, 7, 1	FEMININO	22
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	7, 3, 8, 2	FEMININO	20
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 5, 8, 3	FEMININO	24
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 6, 8, 4	FEMININO	26
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 6, 8, 6	FEMININO	28
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 6, 8, 7	FEMININO	29
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 8, 8, 8	FEMININO	32
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 6, 8, 8	FEMININO	30
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 7, 8, 8	FEMININO	31
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	8, 5, 8, 6	FEMININO	27
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	6, 5, 5, 7	FEMININO	23
SOLTEIRO, CASADO, VIÚVO, SEPARADO JUDICIALMENTE	7, 1, 6, 6	FEMININO	20
SOLTEIRO, CASADO, VIÚVO	4, 4, 5	FEMININO	13
SOLTEIRO, CASADO, VIÚVO	1, 1, 3	FEMININO	5

Ao rolarmos o scroll inferior ainda das mesmas linhas encontramos ainda a coluna de contagem de eleitores que possuem nome social para cada faixa, ao rolarmos o scroll vertical encontramos outras faixas para as características e outros municípios.

5.2. Análise geral

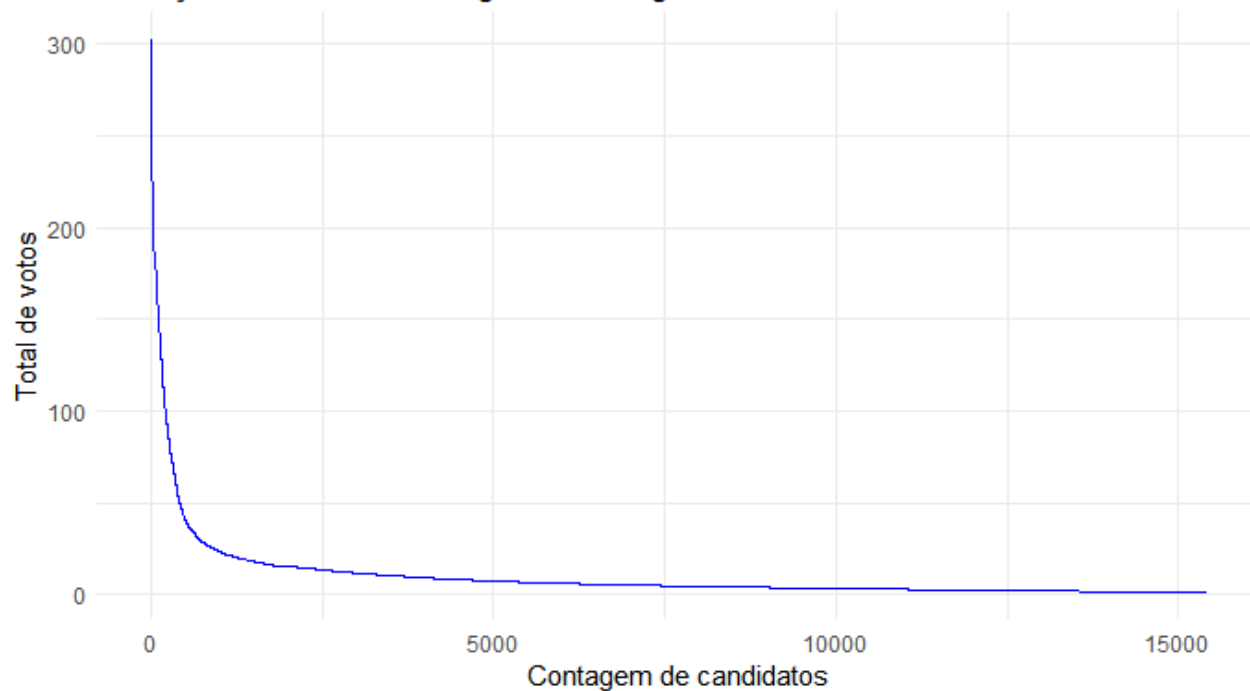
A análise geral se refere a todos os candidatos e municípios. Criei uma nova coluna na base de dados dos candidatos que representa sua colocação, dessa forma fiz um gráfico de dispersão para visualizar as posições em relação ao total de votos de cada um.

Relação entre número de votos e posição do candidato



É possível notar que a visualização do gráfico de dispersão (pontos) não ficou boa devido a essa parte preta que representa os nomes dos candidatos, então irei fazer um gráfico de linha para suavizar a imagem.

Variação dos votos ao longo da contagem de candidatos



Ao retirar os nomes dos candidatos podemos notar uma melhor visualização e suavização do gráfico, é possível notar alguns outliers (valores discrepantes) que alguns candidatos representam em relação a outros, porém neste caso não os devemos descartar por motivos óbvios de interferência nas análises futuras.

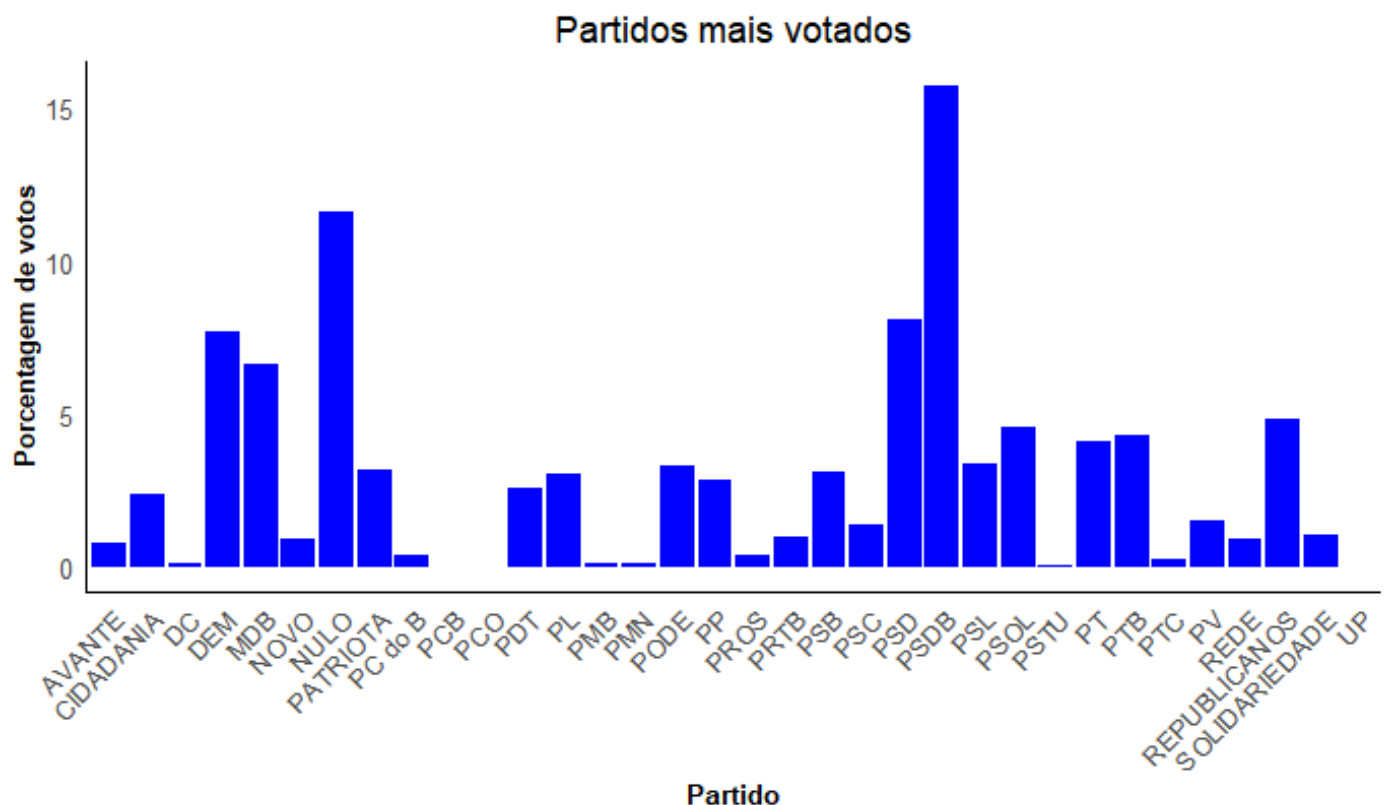
Esta análise é importante para sabermos a discrepância de votos de alguns candidatos quando comparados a outros, e enxergarmos as distribuições dos votos.

5.3. Análise dos partidos mais votados

Agora iremos verificar uma informação importante, quais partidos tiveram mais votos no primeiro turno.

Para isso criei uma função que agrupa os dados por SG_PARTIDO, e criei uma coluna referente a porcentagem de votos de cada partido.

A forma mais eficiente desta visualização é o histograma.



Antes de tudo, é viável observar que o voto nulo foi o que obteve a segunda maior porcentagem, votos esses que podem ser convertidos até o segundo turno através de campanhas de marketing.

Podemos observar então que o top 3 no ranking de partidos com mais votos foram respectivamente: PSDB, PSD e DEM.

5.4. Análise do principal município de interesse

Supondo então que agora o município de maior interesse é São Paulo, vamos comparar os três candidatos mais votados para prefeito e vereador separadamente, e depois realizar uma análise em geral.

Podemos de acordo ao interesse alterar o código para outro município ou candidatos que nos seja interessante.

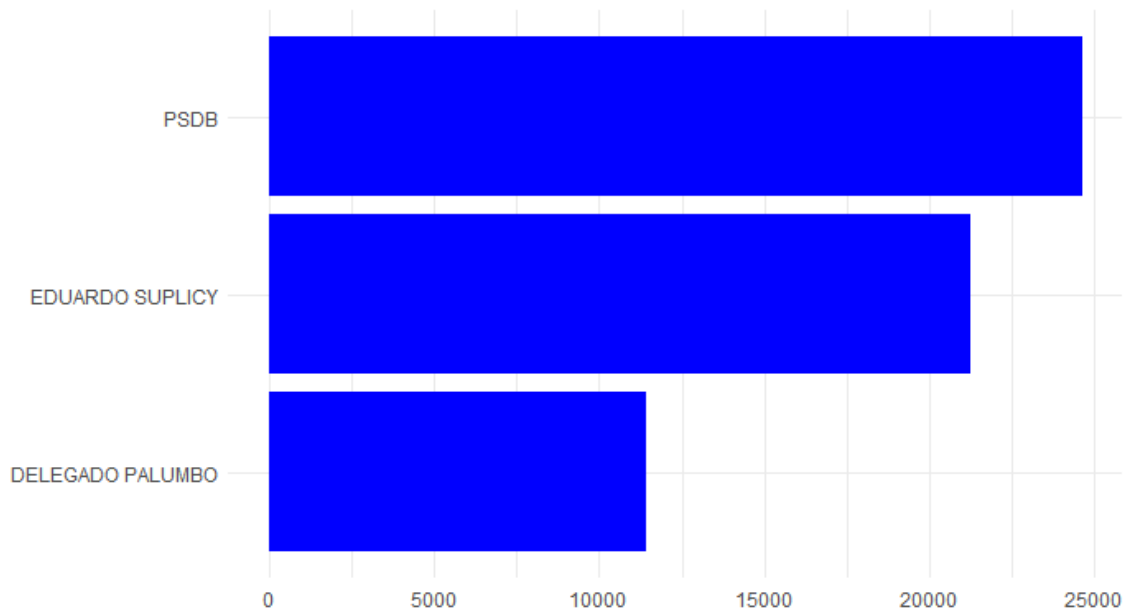
Por enquanto ficaremos com o que eu comentei. O primeiro passo é criar uma função que separe os top 3 candidatos mais votados em cada cargo na cidade de São Paulo.

NM_VOTAVEL	Total_Votos
<chr>	<dbl>
1 BRUNO COVAS	209115
2 GUILHERME BOULOS	144351
3 MÁRCIO FRANÇA	71252

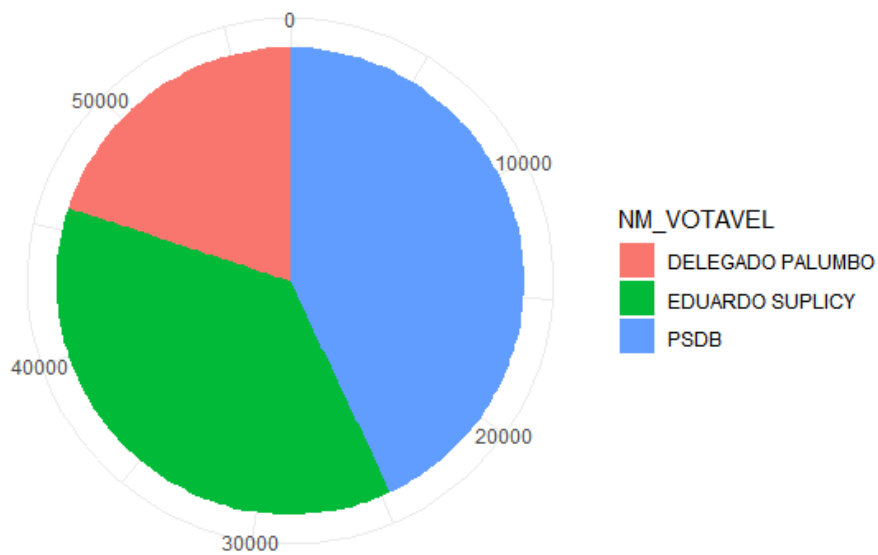
NM_VOTAVEL	Total_Votos
<chr>	<dbl>
1 PSDB	24628
2 EDUARDO SUPPLY	21276
3 DELEGADO PALUMBO	11423

Agora realizaremos alguns gráficos para melhorar a visualização.

Vereadores mais votados em São Paulo

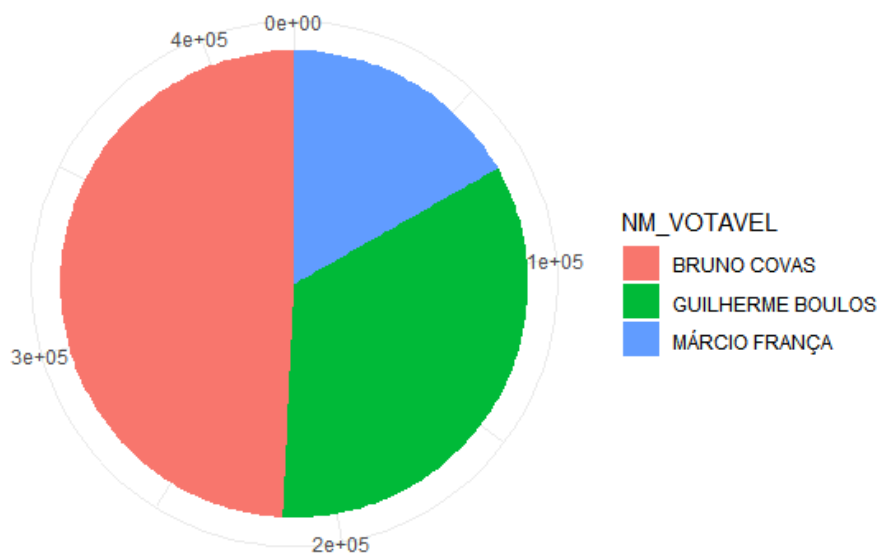


TOP 3 vereadores mais votados





TOP 3 prefeitos mais votados



Agora realizarei uma análise geral, porém no município de interesse.

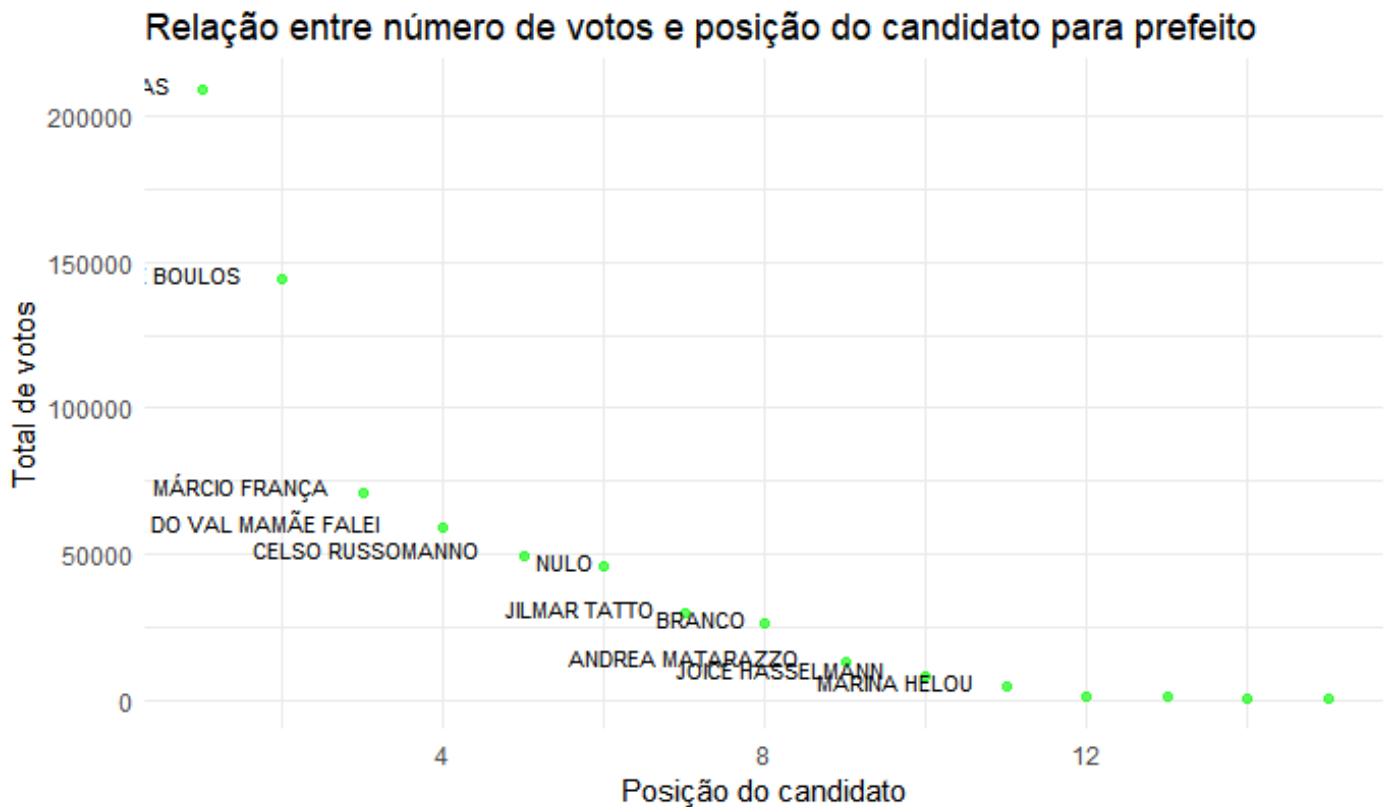
Utilizei da mesma lógica, criei uma função que separa os dados de interesse em São Paulo e plotei alguns gráficos para visualização.

NM_VOTAVEL	Total_Votos
<chr>	<dbl>
1 BRUNO COVAS	209115
2 GUILHERME BOULOS	144351
3 MÁRCIO FRANÇA	71252

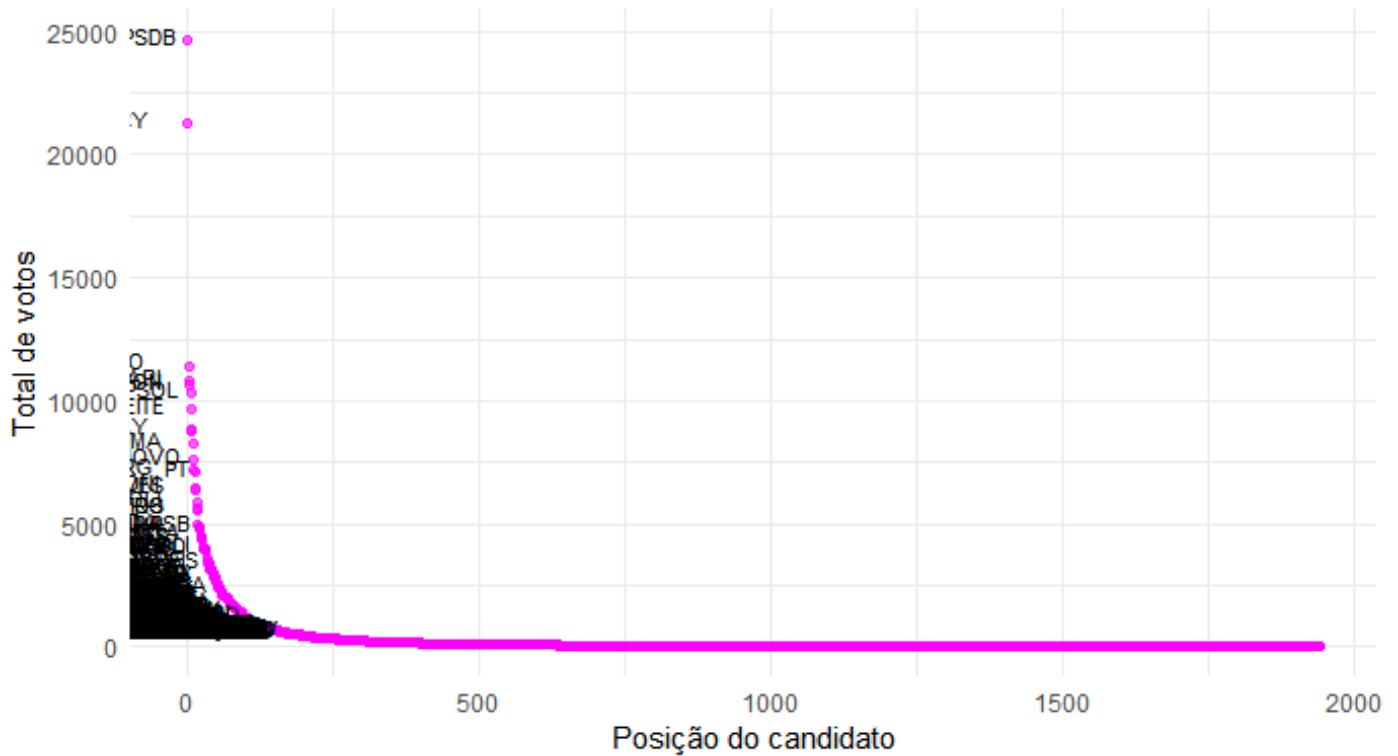
4	ARTHUR DO VAL MAMÃE FALEI	58929
5	CELSO RUSSOMANNO	49532
6	NULO	45502
7	JILMAR TATTO	29837
8	BRANCO	26182
9	ANDREA MATARAZZO	13026
10	JOICE HASSELMANN	8365
11	MARINA HELOU	4655
12	LEVY FIDELIX	1157
13	ORLANDO SILVA	996
14	VERA	335
15	ANTÔNIO CARLOS	89

NM_VOTAVEL	Total_Votos
<chr>	<dbl>
1 PSDB	24628
2 EDUARDO SUPPLY	21276
3 DELEGADO PALUMBO	11423
4 FELIPE BECARI	10789
5 ERIKA HILTON	10673
6 PSOL	10346
7 MILTON LEITE	9637
8 FERNANDO HOLIDAY	8823
9 SILVIA DA BANCADA FEMINISTA	8765
10 JANAÍNA LIMA	8227

i 1,931 more rows
i Use `print(n = ...)` to see more rows



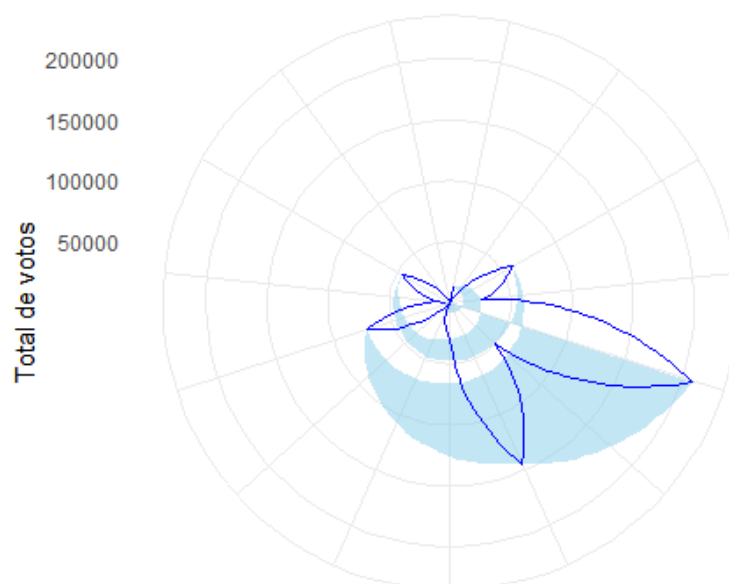
Relação entre número de votos e posição do candidato para vereador



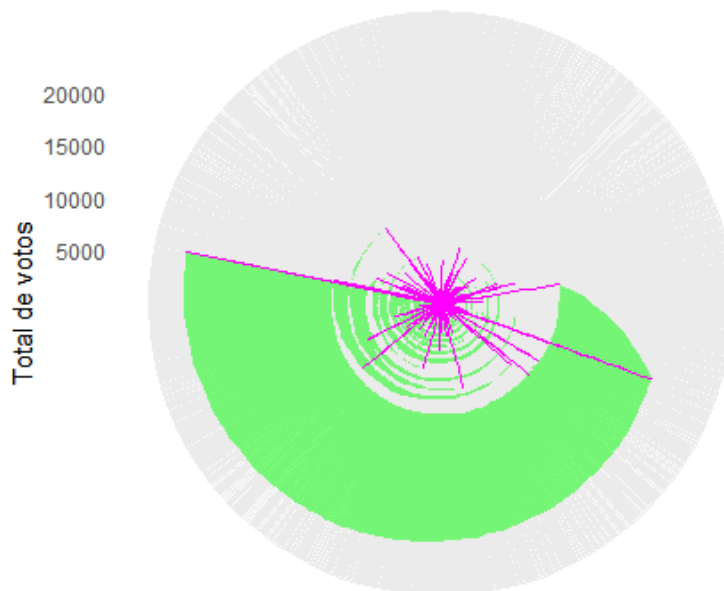
Os gráficos podem ser suavizados, transformando-os em gráficos de linhas como foi feito nos tópicos anteriores se for interessante, e a legenda deve ser melhorada.

Agora utilizei um gráfico de radar, importantíssimo para visualização das características em relações candidato-candidato.

Características dos candidatos para prefeito



Características dos candidatos para vereador



Os eixos radiais representam o total de votos recebidos, e as linhas cada candidato específico que se liga a outro. A distância da linha em relação ao centro nos informa o valor da variável quantidade de votos (quanto mais próximo ao centro, menos votos), mas pode facilmente ser alterada no código para verificarmos alguma outra característica de interesse que desejamos comparar entre os candidatos.

Esse tipo de gráfico é importante pois nos permite identificar rapidamente quais candidatos se destacam mais na área de interesse.

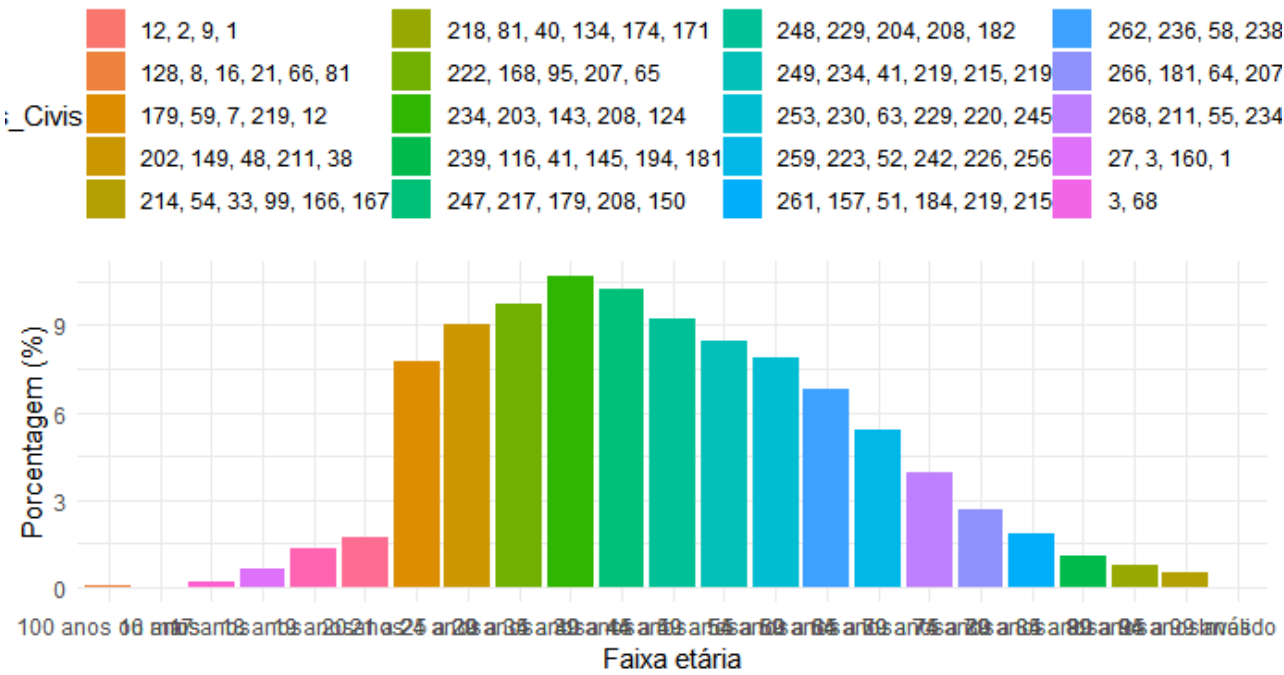
5.5. Sugestões para análise do perfil de eleitorado

Para finalizar minha análise, criei uma função para filtrar o município de São Paulo, e criei uma coluna de porcentagens.

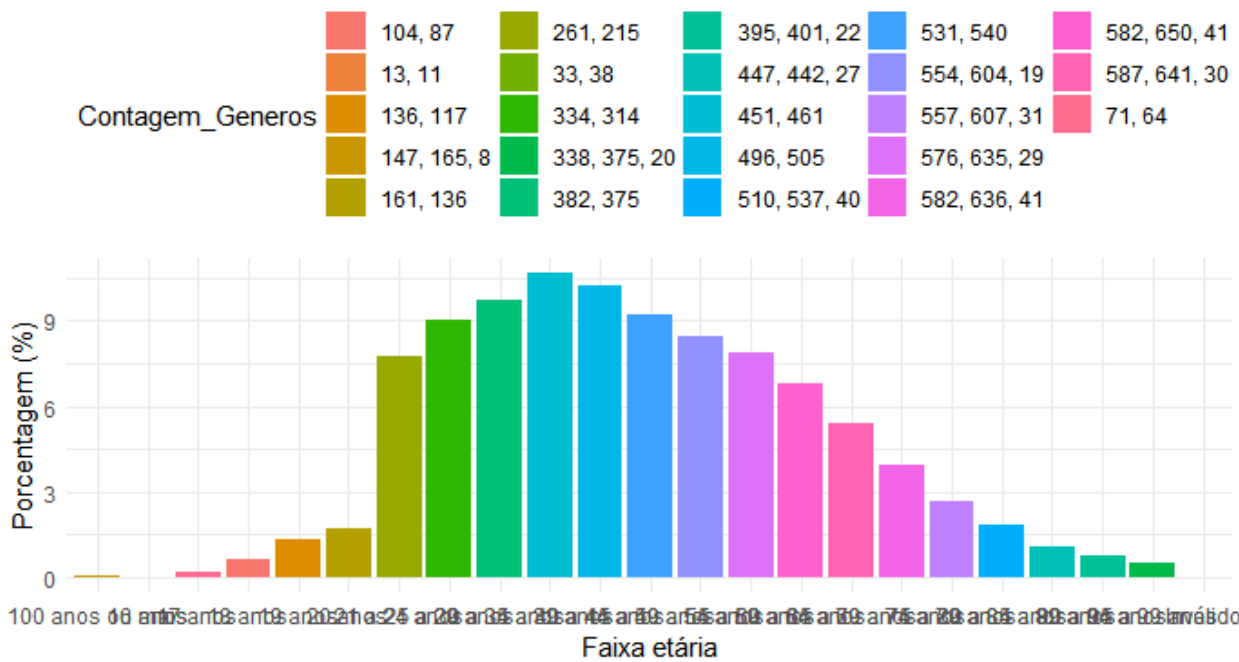
```
[1] 0.05655799 0.02235172 0.18632464 0.66023534 1.32001823 1.737
77816
[7] 7.74609173 9.05638198 9.72467118 10.67570509 10.24052529 9.20
447342
[13] 8.47039596 7.87214818 6.82161746 5.39241969 3.95566578 2.65
379135
[19] 1.82659682 1.07509951 0.76434729 0.53567203 0.00113116
```

Essa coluna é juntada a tabela do perfil de eleitorado, e enfim realizei dois gráficos baseados nessas porcentagens por variável de interesse, no caso estado civil e gênero.

Porcentagem de eleitores por estado civil em São Paulo



Porcentagem de eleitores por gênero em São Paulo



As legendas podem ser alteradas para melhor visualização.

Visualmente, os dados aparentam ter uma distribuição normal, o que é excelente, pois se isso se comprovar verdadeiro, futuramente é possível realizar análises mais profundas e complexas para previsão e testes de hipóteses, ou mesmo inferências.

Para se verificar a normalidade dos dados, poderíamos usar alguns testes, tais como Shapiro-Wilk, Kolmogorov-Smirnov, Jarque-Bera. Que podem ser feitos no software com apenas uma linha de programação.

6. Códigos de programação

Por fim deixo os códigos de programação, primeiro em linguagem R e posteriormente em Python.

Nos códigos deixo comentários seguidos do caractere “#” que explicam passo a passo o que fiz e o que é necessário para reproduzir o código e resultados em qualquer computador.

Vale lembrar que as tabelas apresentadas podem ser convertidas em tabelas do Excel através da função que deixei no código.

Código em R:

```
#Bibliotecas que iremos utilizar
library(dplyr)
library(readxl)
library(ggplot2)

#Insereindo os dados tratados no excel
perfil_eleitorado_reduzido_sp_final <-
read_excel("C:/Users/romba/OneDrive/Área de Trabalho/perfil eleitoral
reduzido sp final.xlsx")
View(perfil_eleitorado_reduzido_sp_final)
```

```

SP_turno_1_reduzido_final <-
read_excel("C:/Users/romba/OneDrive/Área de Trabalho/SP_turno_1-reduzido
final.xlsx")
View(SP_turno_1_reduzido_final)

#-----
---

#Qual candidato foi mais votado em cada município

#Filtrando os cargos
dados_prefeito <- SP_turno_1_reduzido_final %>%
  filter(DS_CARGO_PERGUNTA == "Prefeito")

dados_vereador <- SP_turno_1_reduzido_final %>%
  filter(DS_CARGO_PERGUNTA == "Vereador")

#Obtendo os candidatos mais votados em cada município
pref_mais_vot <- dados_prefeito %>%
  group_by(NM_MUNICIPIO) %>%
  summarize(Prefeito_mais_votado =
NM_VOTAVEL[which.max(QT_VOTOS)])

vere_mais_vot <- dados_vereador %>%
  group_by(NM_MUNICIPIO) %>%
  summarize(Vereador_mais_votado =
NM_VOTAVEL[which.max(QT_VOTOS)])

#Join dos dados
dados_juntados <- left_join(dados_prefeito, pref_mais_vot, by =
"NM_MUNICIPIO")
dados_juntados <- left_join(dados_juntados, vere_mais_vot, by =
"NM_MUNICIPIO")
View(dados_juntados)

```

```

resultado_final <- dados_juntados %>%
group_by(NM_MUNICIPIO) %>%
summarize(Prefeito_mais_votado = first(Prefeito_mais_votado),
          Vereador_mais_votado = first(Vereador_mais_votado))

#Mostrando a tabela
print(resultado_final,n=nrow(resultado_final))
View(resultado_final)

#-----
#Qual município o candidato foi mais votado

#Separando infos dos candidatos
cand_mun_mais_voto <- bind_rows(dados_prefeito, dados_vereador)
%>%
group_by(NM_VOTAVEL) %>%
summarize(Cargo = first(DS_CARGO_PERGUNTA),
          Cidade_Mais_Votado =
NM_MUNICIPIO[which.max(QT_VOTOS)],
          Qt_De_Votos = max(QT_VOTOS))

#Mostrando o resultado
print(cand_mun_mais_voto)
View(cand_mun_mais_voto)

#-----
#Melhor visualização dos perfis
base1_consolidada <- SP_turno_1_reduzido_final %>%
  distinct(NM_MUNICIPIO, .keep_all = TRUE)
View(base1_consolidada)

base2_consolidada <- perfil_eleitorado_reduzido_sp_final %>%

```

```

group_by(NM_MUNICIPIO, DS_FAIXA_ETARIA) %>%
  summarize(Soma_QT_ELEITORES_PERFIL =
sum(QT_ELEITORES_PERFIL),
            Soma_QT_ELEITORES_DEFICIENCIA =
sum(QT_ELEITORES_DEFICIENCIA),
            Estados_Civis_Unicos = paste(unique(DS_ESTADO_CIVIL),
collapse = ", "),
            Contagem_Estados_Civis = paste(table(DS_ESTADO_CIVIL),
collapse = ", "),
            Generos_Unicos = paste(unique(DS_GENERO), collapse = ", "),
            Contagem_Generos = paste(table(DS_GENERO), collapse = ",
"),
            Soma_QT_ELEITORES_INC_NM_SOCIAL =
sum(QT_ELEITORES_INC_NM_SOCIAL))
View(base2_consolidada)
print(base2_consolidada )

```

```

#-----

```

```

#Análises gerais

```

```

descricao_geral <- cand_mun_mais_voto %>%
  mutate(Qt_De_Votos = as.numeric(Qt_De_Votos)) %>%
  arrange(desc(Qt_De_Votos))

```

```

#Criando uma coluna para a posição dos candidatos

```

```

descricao_geral <- descricao_geral %>%
  mutate(Posicao = seq_along(NM_VOTAVEL))

```

```

# Gráfico de Dispersão

```

```

grafico_dispersao <- ggplot(descricao_geral, aes(x = Posicao, y =
Qt_De_Votos, label = NM_VOTAVEL)) +
  geom_point(color = "blue") +
  geom_text(hjust = 1.2, vjust = 0) +
  labs(title = "Relação entre número de votos e posição do candidato",

```



```

    x = "Posição do candidato",
    y = "Total de votos") +
theme_minimal()

# Mostrando o gráfico de dispersão
print(grafico_dispersao)

#A visualização não ficou legal, iremos transformar em gráfico de linhas
#Gráfico de Linhas
grafico_linhas <- ggplot(descricao_geral, aes(x = Posicao, y =
Qt_De_Votos)) +
  geom_line(color = "blue") +
  labs(title = "Variação dos votos ao longo da contagem de candidatos",
    x = "Contagem de candidatos",
    y = "Total de votos") +
  theme_minimal()

#Mostrando o gráfico de linhas
print(grafico_linhas)

#-----
#Análise dos partidos mais votados

partidos_mais_votados <- SP_turno_1_reduzido_final %>%
  group_by(SG_PARTIDO) %>%
  summarize(Total_Votos = sum(QT_VOTOS)) %>%
  arrange(desc(Total_Votos)) %>%
  mutate(Porcentagem_Votos = (Total_Votos / sum(Total_Votos)) * 100)

#Mostrando os partidos mais votados em ordem decrescente
print(partidos_mais_votados, n = nrow(partidos_mais_votados) )

#Criando um histograma para os partidos

```

```

hist_partidos <- ggplot(partidos_mais_votados, aes(x = SG_PARTIDO, y
= Porcentagem_Votos)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Partidos mais votados", x = "Partido", y = "Porcentagem de
votos") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black"),
        axis.title = element_text(size = 10, face = "bold"),
        axis.text = element_text(size = 10),
        legend.title = element_blank(),
        legend.text = element_text(size = 10))

```

#Mostrando o histograma

```
print(hist_partidos)
```

#-----

#Análises de principais cidades de interesse

#Comparando os resultados dos 3 mais votados de sp

```
dados_vereador_sp <- dados_vereador %>%
```

```
  filter(NM_MUNICIPIO == "SÃO PAULO") %>%
```

```
  filter(NM_VOTAVEL != "NULO" & NM_VOTAVEL != "BRANCO")
```

#Retirando branco e nulo pois são os que mais votaram

#Obtendo os candidatos mais votados em São Paulo para prefeito

```
pref_mais_vot_sp <- dados_prefeito_sp %>%
```

```
  group_by(NM_VOTAVEL) %>%
```

```
  summarize(Total_Votos = sum(QT_VOTOS)) %>%
```

```
  top_n(3, Total_Votos) %>%
```

```

    arrange(desc(Total_Votos))

#Obtendo os candidatos mais votados em São Paulo para vereador
vere_mais_vot_sp <- dados_vereador_sp %>%
  group_by(NM_VOTAVEL) %>%
  summarize(Total_Votos = sum(QT_VOTOS)) %>%
  top_n(3, Total_Votos) %>%
  arrange(desc(Total_Votos))

#Visualizando a tabela de São Paulo para candidatos
print(pref_mais_vot_sp)
print(vere_mais_vot_sp)

#Tornando a variável numérica
descricao_vereador_sp <- vere_mais_vot_sp %>%
  mutate(Total_Votos = as.numeric(Total_Votos)) %>%
  arrange(desc(Total_Votos))

descricao_prefeito_sp <- pref_mais_vot_sp %>%
  mutate(Total_Votos = as.numeric(Total_Votos)) %>%
  arrange(desc(Total_Votos))

#Tema dos gráficos, padronizando os gráficos a seguir
tema_personalizado <- theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold"),
        legend.position = "right")

#Gráfico de barras vereadores
grafico_barras_vere <- ggplot(descricao_vereador_sp, aes(x =
reorder(NM_VOTAVEL, Total_Votos), y = Total_Votos)) +
  geom_bar(stat = "identity", fill = "blue") +
  coord_flip() +

```

```
labs(title = "Vereadores mais votados em São Paulo",  
      x = "Vereador",  
      y = "Total de votos") +  
tema_personalizado
```

```
#Gráfico de pizza vereadores  
grafico_pizza_vere <- ggplot(descricao_vereador_sp, aes(x = "", y =  
Total_Votos, fill = NM_VOTAVEL)) +  
  geom_bar(stat = "identity", width = 1) +  
  coord_polar("y", start = 0) +  
  labs(title = "TOP 3 vereadores mais votados") +  
  tema_personalizado
```

```
#Gráfico de barras prefeitos  
grafico_barras_pref <- ggplot(descricao_prefeito_sp, aes(x =  
reorder(NM_VOTAVEL, Total_Votos), y = Total_Votos)) +  
  geom_bar(stat = "identity", fill = "blue") +  
  coord_flip() +  
  labs(title = "Prefeitos mais votados em São Paulo",  
        x = "Prefeito",  
        y = "Total de votos") +  
  tema_personalizado
```

```
#Gráfico de pizza para prefeitos  
grafico_pizza_pref <- ggplot(descricao_prefeito_sp, aes(x = "", y =  
Total_Votos, fill = NM_VOTAVEL)) +  
  geom_bar(stat = "identity", width = 1) +  
  coord_polar("y", start = 0) +  
  labs(title = "TOP 3 prefeitos mais votados") +  
  tema_personalizado
```

```
#Mostrando os gráficos  
plot(grafico_barras_vere)  
plot(grafico_pizza_vere)
```

```
plot(grafico_barras_pref)
plot(grafico_pizza_pref)
```

```
#Análise geral do estado de sp
pref_geral<-dados_prefeito_sp %>%
  group_by(NM_VOTAVEL) %>%
  summarize(Total_Votos = sum(QT_VOTOS))
```

```
vere_geral<-dados_vereador_sp %>%
  group_by(NM_VOTAVEL) %>%
  summarize(Total_Votos = sum(QT_VOTOS))
```

```
#Dados dos prefeitos, geral
pref_geral <- dados_prefeito_sp %>%
  group_by(NM_VOTAVEL) %>%
  summarize(Total_Votos = sum(QT_VOTOS)) %>%
  arrange(desc(Total_Votos))
```

```
#Dados dos vereadores, geral
vere_geral <- dados_vereador_sp %>%
  filter(NM_VOTAVEL != "NULO" & NM_VOTAVEL != "BRANCO") %>%
  group_by(NM_VOTAVEL) %>%
  summarize(Total_Votos = sum(QT_VOTOS)) %>%
  arrange(desc(Total_Votos))
```

```
#Gráfico de Dispersão para Prefeito
grafico_dispersao_pref <- ggplot(pref_geral, aes(x =
seq_along(NM_VOTAVEL), y = Total_Votos)) +
  geom_point(color = "green", alpha = 0.6) +
  geom_text(aes(label = ifelse(Total_Votos > max(Total_Votos) * 0.02,
NM_VOTAVEL, "")),
  hjust = 1.2, vjust = 0.2, size = 3) +
```

```
labs(title = "Relação entre número de votos e posição do candidato  
para prefeito",
```

```
  x = "Posição do candidato",  
  y = "Total de votos") +  
  theme_minimal() +  
  theme(legend.position = "bottom") +  
  guides(color = guide_legend(title = "Candidatos"))
```

```
#Gráfico de Dispersão para Vereador
```

```
grafico_dispersao_vere <- ggplot(vere_geral, aes(x =  
seq_along(NM_VOTAVEL), y = Total_Votos)) +  
  geom_point(color = "magenta", alpha = 0.6) +  
  geom_text(aes(label = ifelse(Total_Votos > max(Total_Votos) * 0.02,  
NM_VOTAVEL, "")),  
            hjust = 1.2, vjust = 0.2, size = 3) +
```

```
labs(title = "Relação entre número de votos e posição do candidato  
para vereador",
```

```
  x = "Posição do candidato",  
  y = "Total de votos") +  
  theme_minimal() +  
  theme(legend.position = "bottom") +  
  guides(color = guide_legend(title = "Candidatos"))
```

```
#Mostrando os gráficos
```

```
print(grafico_dispersao_pref)  
print(grafico_dispersao_vere)
```

```
#Gráfico de Linhas para Prefeito
```

```
grafico_linhas_pref <- ggplot(pref_geral, aes(x =  
seq_along(NM_VOTAVEL), y = Total_Votos)) +  
  geom_line(color = "green") +  
  labs(title = "Variação dos votos ao longo da contagem de candidatos  
para prefeito",  
        x = "Contagem de candidatos",
```

```
y = "Total de votos") +  
theme_minimal()
```

```
#Gráfico de Linhas para Vereador  
grafico_linhas_vere <- ggplot(vere_geral, aes(x =  
seq_along(NM_VOTAVEL), y = Total_Votos)) +  
  geom_line(color = "magenta") +  
  labs(title = "Variação dos votos ao longo da contagem de candidatos  
para vereador",  
        x = "Contagem de candidatos",  
        y = "Total de votos") +  
  theme_minimal()
```

```
#Mostrando os gráficos  
print(grafico_linhas_pref)  
print(grafico_linhas_vere)
```

```
#Gráfico de Radar para Prefeito  
grafico_radar_pref <- ggplot(pref_geral, aes(x = NM_VOTAVEL, y =  
Total_Votos)) +  
  geom_polygon(aes(group = 1), fill = "skyblue", alpha = 0.5) +  
  geom_line(aes(group = 1), color = "blue") +  
  coord_polar(start = 0) +  
  labs(title = "Características dos candidatos para prefeito",  
        y = "Total de votos") +  
  theme_minimal() +  
  theme(axis.title.x = element_blank(),  
        axis.text.x = element_blank(),  
        axis.ticks.x = element_blank())
```

```
#Gráfico de Radar para Vereador  
grafico_radar_vere <- ggplot(vere_geral, aes(x = NM_VOTAVEL, y =  
Total_Votos)) +  
  geom_polygon(aes(group = 1), fill = "green", alpha = 0.5) +
```

```

geom_line(aes(group = 1), color = "magenta") +
coord_polar(start = 0) +
labs(title = "Características dos candidatos para vereador",
      y = "Total de votos") +
theme_minimal() +
theme(axis.title.x = element_blank(),
      axis.text.x = element_blank(),
      axis.ticks.x = element_blank())

#Mostrando os gráficos
print(grafico_radar_pref)
print(grafico_radar_vere)

#-----
#Sugestões de análises do banco de perfil do eleitorado

#Filtrando o estado de São Paulo
base_sp <- base2_consolidada %>%
  filter(NM_MUNICIPIO == "SÃO PAULO")

#Porcentagens
base_sp_percentual$Porcentagem <- base_sp %>%
  mutate(Porcentagem = Soma_QT_ELEITORES_PERFIL /
sum(Soma_QT_ELEITORES_PERFIL) * 100)

#Tabela com porcentagens
View(base_sp_percentual)
print(base_sp_percentual, n = nrow(base_sp_percentual) )

#Gráfico de barras para Estado Civil
grafico_estado_civil <- ggplot(base_sp_percentual, aes(x =
DS_FAIXA_ETARIA, y = Porcentagem, fill = Contagem_Estados_Civis)) +
  geom_bar(stat = "identity") +
  labs(title = "Porcentagem de eleitores por estado civil em São Paulo",

```



```
x = "Faixa etária",  
y = "Porcentagem (%)" +  
theme_minimal() +  
theme(legend.position = "top")
```

```
#Gráfico de barras para Gênero  
grafico_genero <- ggplot(base_sp_percentual, aes(x =  
DS_FAIXA_ETARIA, y = Porcentagem, fill = Contagem_Generos)) +  
geom_bar(stat = "identity") +  
labs(title = "Porcentagem de eleitores por gênero em São Paulo",  
x = "Faixa etária",  
y = "Porcentagem (%)" +  
theme_minimal() +  
theme(legend.position = "top")
```

#Se for do interesse, podemos fazer o mesmo para verificar os eleitores com inclusão de nome social

```
#Mostrando os gráficos  
print(grafico_estado_civil)  
print(grafico_genero)
```

#Os dados aparentemente seguem aparentemente uma distribuição normal

```
#Seria possível utilizar testes estatísticos para essa comprovação  
#Tais como Shapiro-Wilk, Kolmogorov-Smirnov, Jarque-Bera  
#Se comprovarmos a normalidade dos dados, outros testes estatísticos  
podem ser realizados para uma análise mais profunda  
#Se existir alguma tabela feita que interesse estar em formato excel, é  
possível usar comandos para gera-las  
#Basta usar a biblioteca library(openxlsx)
```

Código em Python:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Lendo os dados tratados do Excel
perfil_eleitorado_reduzido_sp_final =
pd.read_excel("C:/Users/romba/OneDrive/Área de Trabalho/perfil eleitoral
reduzido sp final.xlsx")

SP_turno_1_reduzido_final =
pd.read_excel("C:/Users/romba/OneDrive/Área de Trabalho/SP_turno_1-
reduzido final.xlsx")

# Qual candidato foi mais votado em cada município
dados_prefeito =
SP_turno_1_reduzido_final[SP_turno_1_reduzido_final['DS_CARGO_PERGUN
TA'] == "Prefeito"]

dados_vereador =
SP_turno_1_reduzido_final[SP_turno_1_reduzido_final['DS_CARGO_PERGUN
TA'] == "Vereador"]

pref_mais_vot =
dados_prefeito.groupby('NM_MUNICIPIO')['NM_VOTAVEL'].apply(lambda x:
x[x.index[x['QT_VOTOS'] == x['QT_VOTOS'].max()][0]])

vere_mais_vot =
dados_vereador.groupby('NM_MUNICIPIO')['NM_VOTAVEL'].apply(lambda x:
x[x.index[x['QT_VOTOS'] == x['QT_VOTOS'].max()][0]])

dados_juntados = pd.merge(dados_prefeito, pref_mais_vot,
on='NM_MUNICIPIO')

dados_juntados = pd.merge(dados_juntados, vere_mais_vot,
on='NM_MUNICIPIO')
```

```
    resultado_final =  
dados_juntados.groupby('NM_MUNICIPIO').agg(Prefeito_mais_votado=('NM_V  
OTAVEL_x', 'first'),
```

```
Vereador_mais_votado=('NM_VOTAVEL_y', 'first'))
```

```
print(resultado_final)  
print("Número de municípios:", resultado_final.shape[0])
```

```
# Qual município o candidato foi mais votado  
cand_mun_mais_voto = pd.concat([dados_prefeito, dados_vereador],  
ignore_index=True)  
cand_mun_mais_voto =  
cand_mun_mais_voto.groupby('NM_VOTAVEL').agg(Cargo=('DS_CARGO_PE  
RGUNTA', 'first'),
```

```
Cidade_Mais_Votado=('NM_MUNICIPIO', lambda x: x[x.index[x['QT_VOTOS']  
== x['QT_VOTOS'].max()][0]]),
```

```
Qt_De_Votos=('QT_VOTOS',  
'max'))
```

```
print(cand_mun_mais_voto)  
print("Número de candidatos:", cand_mun_mais_voto.shape[0])
```

```
# Melhor visualização dos perfis  
base1_consolidada =  
SP_turno_1_reduzido_final.drop_duplicates(subset='NM_MUNICIPIO',  
keep='first')  
base2_consolidada =  
perfil_eleitorado_reduzido_sp_final.groupby(['NM_MUNICIPIO',  
'DS_FAIXA_ETARIA']).agg(  
    Soma_QT_ELEITORES_PERFIL=('QT_ELEITORES_PERFIL', 'sum'),
```

```

Soma_QT_ELEITORES_DEFICIENCIA=('QT_ELEITORES_DEFICIENCIA',
'sum'),

    Estados_Civis_Unicos=('DS_ESTADO_CIVIL', lambda x: ',
'.join(np.unique(x))),

    Contagem_Estados_Civis=('DS_ESTADO_CIVIL', lambda x: ',
'.join(map(str, x.value_counts().tolist()))),

    Generos_Unicos=('DS_GENERO', lambda x: ', '.join(np.unique(x))),
    Contagem_Generos=('DS_GENERO', lambda x: ', '.join(map(str,
x.value_counts().tolist()))),

Soma_QT_ELEITORES_INC_NM_SOCIAL=('QT_ELEITORES_INC_NM_SOCIAL', 'sum')).reset_index()

print(base2_consolidada)
print("Número de municípios:", base2_consolidada.shape[0])

# Análises gerais
descricao_geral = cand_mun_mais_voto.copy()
descricao_geral['Qt_De_Votos'] =
pd.to_numeric(descricao_geral['Qt_De_Votos'])
descricao_geral = descricao_geral.sort_values(by='Qt_De_Votos',
ascending=False)
descricao_geral['Posicao'] = np.arange(1, descricao_geral.shape[0] + 1)

# Gráfico de Dispersão
plt.figure(figsize=(10, 6))
sns.scatterplot(data=descricao_geral, x='Posicao', y='Qt_De_Votos',
label='NM_VOTAVEL', color='blue')
plt.title("Relação entre número de votos e posição do candidato")
plt.xlabel("Posição do candidato")
plt.ylabel("Total de votos")
plt.legend()
plt.show()

```

```

# Gráfico de Linhas
plt.figure(figsize=(10, 6))
sns.lineplot(data=descricao_geral, x='Posicao', y='Qt_De_Votos',
color='blue')
plt.title("Variação dos votos ao longo da contagem de candidatos")
plt.xlabel("Contagem de candidatos")
plt.ylabel("Total de votos")
plt.show()

# Análise dos partidos mais votados
partidos_mais_votados =
SP_turno_1_reduzido_final.groupby('SG_PARTIDO')['QT_VOTOS'].sum().reset
_index()
partidos_mais_votados =
partidos_mais_votados.sort_values(by='QT_VOTOS', ascending=False)
partidos_mais_votados['Porcentagem_Votos'] =
(partidos_mais_votados['QT_VOTOS'] /
partidos_mais_votados['QT_VOTOS'].sum()) * 100

print(partidos_mais_votados)
print("Número de partidos:", partidos_mais_votados.shape[0])

plt.figure(figsize=(10, 6))
sns.barplot(data=partidos_mais_votados, x='SG_PARTIDO',
y='Porcentagem_Votos', color='blue')
plt.title("Partidos mais votados")
plt.xlabel("Partido")
plt.ylabel("Porcentagem de votos (%)")
plt.xticks(rotation=45, ha='right')
plt.show()

# Sugestões de análises do banco de perfil do eleitorado

```

```
base_sp = base2_consolidada[base2_consolidada['NM_MUNICIPIO'] ==  
"SÃO PAULO"]  
base_sp_percentual = base_sp.copy()  
base_sp_percentual['Porcentagem'] =  
(base_sp_percentual['Soma_QT_ELEITORES_PERFIL'] /  
base_sp_percentual['Soma_QT_ELEITORES_PERFIL'].sum()) * 100
```

```
plt.figure(figsize=(10, 6))  
sns.barplot(data=base_sp_percentual, x='DS_FAIXA_ETARIA',  
y='Porcentagem', hue='Contagem_Estados_Civis')  
plt.title("Porcentagem de eleitores por estado civil em São Paulo")  
plt.xlabel("Faixa etária")  
plt.ylabel("Porcentagem (%)")  
plt.legend(title="Estado Civil")  
plt.xticks(rotation=45, ha='right')  
plt.show()
```

```
plt.figure(figsize=(10, 6))  
sns.barplot(data=base_sp_percentual, x='DS_FAIXA_ETARIA',  
y='Porcentagem', hue='Contagem_Generos')  
plt.title("Porcentagem de eleitores por gênero em São Paulo")  
plt.xlabel("Faixa etária")  
plt.ylabel("Porcentagem (%)")  
plt.legend(title="Gênero")  
plt.xticks(rotation=45, ha='right')  
plt.show()
```

