

Ideando_historia

October 31, 2024

1 Entrega Final - Proyecto de Visualización de la Información

Universidad del Desarrollo Profesor: Fernando Becerra Saavedra

2 Fluctuaciones en procesos electorales chilenos por región

31 de Octubre de 2024

Nombre Estudiantes: - Gabriel Álvarez - Diego León - Constanza Pérez - César Soto

2.1 Contexto

Debido a la desigualdad de cantidad de habitantes por región en Chile, existe la creencia popular de que en procesos electorales siempre va a triunfar aquello que triunfa en la Región Metropolitana. Sin embargo, en cada proceso electoral, cada una de las 16 regiones presentes en Chile presenta variaciones de población y también de tendencias de sector político. Se plantea entonces las siguientes interrogantes: históricamente en los resultados de procesos electorales, ¿sólo valen las variaciones de la región metropolitana para marcar la tendencia nacional? ¿Cada región es invariante en el tiempo en tendencia política? ¿Existen zonas o grupos de regiones que presentan tendencias políticas similares?

El Servicio de Votaciones Electorales (SERVEL) disponibiliza datos abiertos de cada proceso electoral, incluyendo elecciones presidenciales, municipales y plebiscitos. Dados estos datos, se busca poder afirmar o desmentir esta información con la información presente de las 16 regiones chilenas.

Para el presente caso, se abordará el análisis de las últimas 5 elecciones presidenciales de Chile, desde el año 2006 hasta el año 2021. Se busca analizar la tendencia de votación en general del país y por región y cómo esta ha variado en el tiempo.

Se importan los paquetes necesarios para la generación del análisis visual de los datos.

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import geopandas as gpd
from matplotlib.patches import Circle, FancyArrowPatch
from matplotlib.colors import LinearSegmentedColormap
from matplotlib.font_manager import FontProperties
import plotly.graph_objects as go
from shapely.ops import unary_union
```

```
from flexitext import flexitext
from IPython.display import Image
```

2.2 Elecciones presidenciales

Se descargan los datos abiertos del SERVEL: <https://www.servel.cl/servel/modulo-de-archivos/> para los procesos de elección presidencial entre 2006 y 2021 y se estudia en detalle las variaciones de preferencias por sector político

Dataset de elecciones desde 1989 hasta 2017. Sólo se ocupará desde 2006 hasta 2017

```
[2]: data = pd.read_excel('resultados_elecciones_presidenciales_ce_1989_2017_Chile.
    ↪xlsx',
                           sheet_name = 'Presidenciales Chile')
data.head(5)
```

```
[2]: Tipo de Elección      Cargo Fecha de Elección  Año de Elección  \
0      PRESIDENCIAL  PRESIDENTE      1989-12-14      1989
1      PRESIDENCIAL  PRESIDENTE      1989-12-14      1989
2      PRESIDENCIAL  PRESIDENTE      1989-12-14      1989
3      PRESIDENCIAL  PRESIDENTE      1989-12-14      1989
4      PRESIDENCIAL  PRESIDENTE      1989-12-14      1989

      Inicio de Período  Fin de Período      Período Votación Presidencial  \
0              1990              1994  1990 a 1994      UNICA VOTACIÓN
1              1990              1994  1990 a 1994      UNICA VOTACIÓN
2              1990              1994  1990 a 1994      UNICA VOTACIÓN
3              1990              1994  1990 a 1994      UNICA VOTACIÓN
4              1990              1994  1990 a 1994      UNICA VOTACIÓN

      Id Región      Región  ...  Sexo Mesa      Candidato (a)  \
0          1  DE TARAPACA  ...    HOMBRE  FRANCISCO JAVIER ERRAZURIZ
1          1  DE TARAPACA  ...    HOMBRE      HERNAN BUCHI
2          1  DE TARAPACA  ...    HOMBRE    PATRICIO AYLWIN
3          1  DE TARAPACA  ...    HOMBRE      VOTOS NULOS
4          1  DE TARAPACA  ...    HOMBRE    VOTOS EN BLANCO

      Nombres Apellido Paterno Apellido Materno      Sexo Electo(a)  \
0  FRANCISCO JAVIER      ERRAZURIZ      TALAVERA  HOMBRE      NaN
1      HERNAN      BUCHI      BUC  HOMBRE      NaN
2    PATRICIO      AYLWIN      AZOCAR  HOMBRE      SI
3          NaN      NaN      NaN  NaN      NaN
4          NaN      NaN      NaN  NaN      NaN

      Partido Sigla Partido Votos Totales
0    INDEPENDIENTE    INDEP      7065
1    INDEPENDIENTE    INDEP     11549
2  DEMOCRATA CRISTIANO      DC     23230
```

3	NaN	NaN	561
4	NaN	NaN	392

[5 rows x 24 columns]

Dataset de último proceso electoral presencial en 2021

```
[3]: data_2021 = pd.read_excel('Resultados_mesa_presidencial_TRICEL_2v_2021.xlsx',
    sheet_name = 'CHILE')
data_2021.head(5)
```

```
[3]:
```

	Nro.	Región	Región	Provincia	Circ. Senatorial	Distrito	\
0	1.0	DE TARAPACA	IQUIQUE		2.0	2.0	
1	1.0	DE TARAPACA	IQUIQUE		2.0	2.0	
2	1.0	DE TARAPACA	IQUIQUE		2.0	2.0	
3	1.0	DE TARAPACA	IQUIQUE		2.0	2.0	
4	1.0	DE TARAPACA	IQUIQUE		2.0	2.0	

	Comuna	Circ. Electoral	Local	\
0	ALTO HOSPICIO	ALTO HOSPICIO	ANEXO DE COLEGIO SAN ANTONIO DE MATILLA	
1	ALTO HOSPICIO	ALTO HOSPICIO	ANEXO DE COLEGIO SAN ANTONIO DE MATILLA	
2	ALTO HOSPICIO	ALTO HOSPICIO	ANEXO DE COLEGIO SAN ANTONIO DE MATILLA	
3	ALTO HOSPICIO	ALTO HOSPICIO	ANEXO DE COLEGIO SAN ANTONIO DE MATILLA	
4	ALTO HOSPICIO	ALTO HOSPICIO	ANEXO DE COLEGIO SAN ANTONIO DE MATILLA	

	Nro. Mesa	Tipo de mesa	Mesas Fusionadas	Electores	Nro. en Voto	\
0	20.0	M	20M	333.0	1.0	
1	20.0	M	20M	333.0	2.0	
2	20.0	M	20M	333.0	900.0	
3	20.0	M	20M	333.0	901.0	
4	21.0	M	21M	340.0	1.0	

	Candidato	Votos TRICEL
0	GABRIEL BORIC FONT	71.0
1	JOSE ANTONIO KAST RIST	92.0
2	VOTOS NULOS	1.0
3	VOTOS EN BLANCO	0.0
4	GABRIEL BORIC FONT	82.0

```
[4]: data = data[(data["Votación Presidencial"] == "SEGUNDA VOTACIÓN")
    & (data["Año de Elección"] >= 2006)]
data.reset_index(inplace=True, drop=True)
data.head(5)
```

```
[4]:
```

	Tipo de Elección	Cargo	Fecha de Elección	Año de Elección	\
0	PRESIDENCIAL	PRESIDENTE	2006-01-15	2006	
1	PRESIDENCIAL	PRESIDENTE	2006-01-15	2006	
2	PRESIDENCIAL	PRESIDENTE	2006-01-15	2006	

3	PRESIDENCIAL	PRESIDENTE	2006-01-15	2006
4	PRESIDENCIAL	PRESIDENTE	2006-01-15	2006

	Inicio de Período	Fin de Período	Período	Votación	Presidencial \
0	2006	2010	2006 a 2010	SEGUNDA VOTACIÓN	
1	2006	2010	2006 a 2010	SEGUNDA VOTACIÓN	
2	2006	2010	2006 a 2010	SEGUNDA VOTACIÓN	
3	2006	2010	2006 a 2010	SEGUNDA VOTACIÓN	
4	2006	2010	2006 a 2010	SEGUNDA VOTACIÓN	

	Id Región	Región	...	Sexo	Mesa	Candidato (a)	Nombres \
0	1	DE TARAPACA	...	HOMBRE	MICHELLE	BACHELET	MICHELLE
1	1	DE TARAPACA	...	HOMBRE	SEBASTIAN	PIÑERA	SEBASTIAN
2	1	DE TARAPACA	...	HOMBRE	VOTOS	NULOS	NaN
3	1	DE TARAPACA	...	HOMBRE	VOTOS	EN BLANCO	NaN
4	1	DE TARAPACA	...	MUJER	MICHELLE	BACHELET	MICHELLE

	Apellido Paterno	Apellido Materno	Sexo	Electo(a)	Partido \
0	BACHELET	JERIA	MUJER	SI	SOCIALISTA DE CHILE
1	PIÑERA	ECHENIQUE	HOMBRE	NaN	RENOVACION NACIONAL
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	BACHELET	JERIA	MUJER	SI	SOCIALISTA DE CHILE

	Sigla Partido	Votos Totales
0	PS	3910
1	RN	3430
2	NaN	195
3	NaN	47
4	PS	3170

[5 rows x 24 columns]

Se agrupa por sector político: izquierda o derecha

```
[7]: data_agg_2021 = data_2021.groupby(["Nro. Región", "Región"]).apply((lambda s:
    pd.Series({
        "Votos izquierda": s[s["Candidato"].isin(['GABRIEL BORIC FONT'])]["Votos_
    TRICEL"].astype(int).sum(),
        "Votos derecha": s[s["Candidato"].isin(['JOSE ANTONIO KAST RIST'])]["Votos_
    TRICEL"].astype(int).sum()
    }))
)
data_agg_2021.reset_index(inplace = True)
data_agg_2021["% izquierda"] = round(
    data_agg_2021["Votos izquierda"] / (data_agg_2021["Votos izquierda"] +
    data_agg_2021["Votos derecha"]), 2)
```

```

data_agg_2021["% derecha"] = round(
    data_agg_2021["Votos derecha"] / (data_agg_2021["Votos izquierda"] +
    ↪data_agg_2021["Votos derecha"]), 2)
data_agg_2021["Participación efectiva"] = data_agg_2021["Votos izquierda"] +
    ↪data_agg_2021["Votos derecha"]
data_agg_2021.rename(columns = {"Nro. Región": "Id Región"}, inplace = True)
data_agg_2021["Id Región"] = data_agg_2021["Id Región"].astype(int)
data_agg_2021.insert(0, 'Año de Elección', 2021)
data_agg_2021.head(5)

```

C:\Users\Diego\AppData\Local\Temp\ipykernel_22300\2741349133.py:1:

DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.

```

data_agg_2021 = data_2021.groupby(["Nro. Región", "Región"]).apply((lambda s:
pd.Series({

```

```

[7]:

```

	Año de Elección	Id Región	Región	Votos izquierda	Votos derecha \
0	2021	1	DE TARAPACA	55477	58485
1	2021	2	DE ANTOFAGASTA	128009	86097
2	2021	3	DE ATACAMA	76194	40235
3	2021	4	DE COQUIMBO	199259	115610
4	2021	5	DE VALPARAISO	544874	374058

	% izquierda	% derecha	Participación efectiva
0	0.49	0.51	113962
1	0.60	0.40	214106
2	0.65	0.35	116429
3	0.63	0.37	314869
4	0.59	0.41	918932

```

[8]:
izq_lista = ["EDUARDO FREI", 'MICHELLE BACHELET JERIA', 'MICHELLE BACHELET',
'ALEJANDRO GUILLIER ALVAREZ']
der_lista = ['SEBASTIAN PIÑERA', 'SEBASTIAN PIÑERA ECHENIQUE', 'EVELYN MATTHEI_
↪FORNET']

data_agg = data.groupby(["Año de Elección", "Id Región", "Región"]).
    ↪apply((lambda s: pd.Series({
        "Votos izquierda": s[s["Candidato (a)"].isin(izq_lista)]["Votos Totales"].
    ↪astype(int).sum(),
        "Votos derecha": s[s["Candidato (a)"].isin(der_lista)]["Votos Totales"].
    ↪astype(int).sum()
    })))
)
data_agg.reset_index(inplace = True)

```

```

data_agg["% izquierda"] = round(
    data_agg["Votos izquierda"] / (data_agg["Votos izquierda"] +
    ↪data_agg["Votos derecha"]), 2)
data_agg["% derecha"] = round(
    data_agg["Votos derecha"] / (data_agg["Votos izquierda"] + data_agg["Votos_
    ↪derecha"]), 2)
data_agg["Participación efectiva"] = data_agg["Votos izquierda"] +
    ↪data_agg["Votos derecha"]
data_agg = pd.concat((data_agg, data_agg_2021))
data_agg.head(5)

```

C:\Users\Diego\AppData\Local\Temp\ipykernel_22300\1558685333.py:5:

DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.

```

data_agg = data.groupby(["Año de Elección", "Id Región",
"Región"]).apply((lambda s: pd.Series({

```

```

[8]:
  Año de Elección  Id Región  Región  Votos izquierda  Votos derecha \
0          2006         1  DE TARAPACA          86120          84901
1          2006         2  DE ANTOFAGASTA        115979          73399
2          2006         3    DE ATACAMA          66055          43878
3          2006         4    DE COQUIMBO        158908         108237
4          2006         5    DE VALPARAISO        389696         384079

    % izquierda  % derecha  Participación efectiva
0          0.50      0.50          171021
1          0.61      0.39          189378
2          0.60      0.40          109933
3          0.59      0.41          267145
4          0.50      0.50          773775

```

2.2.1 Participación electoral

Intuitivamente, se puede creer que debido al aumento constante de población en el país (particularmente en la región Metropolitana), existe un aumento de participación en las votaciones. Sin embargo, el gráfico muestra que para la Región Metropolitana esto no es cierto, donde en 2013 se registra una participación electoral más baja que en los procesos anteriores. En 2021 se alcanza un peak de participación en los últimos procesos desde 2006

```

[7]: fig, ax = plt.subplots(1, 1, figsize = (6, 5))

data_met = data_agg[data_agg["Id Región"] == 13].copy()

for temp_iter in data_met.groupby(["Año de Elección"]):
    temp_df = temp_iter[1]

```

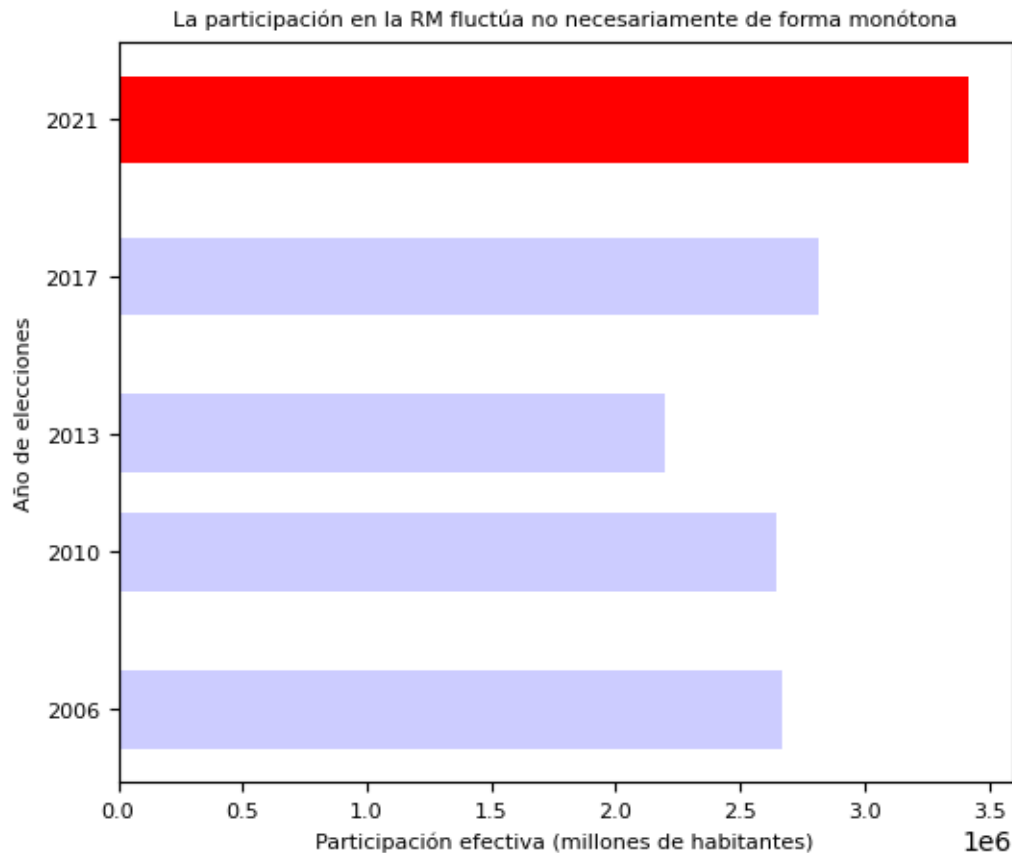
```

anio = temp_iter[0][0]
if anio == 2021:
    ax.barh(temp_df["Año de Elección"], temp_df["Participación efectiva"],
    height = 2.2, color = 'r')
else:
    ax.barh(temp_df["Año de Elección"], temp_df["Participación efectiva"],
    height = 2, color = 'b', alpha = 0.2)

ax.set_yticks(data_met["Año de Elección"].values)
ax.set_yticklabels(data_met["Año de Elección"].astype(str))
ax.tick_params(axis='both', labelsize=8)
ax.set_xlabel("Participación efectiva (millones de habitantes)", fontsize = 8)
ax.set_ylabel("Año de elecciones", fontsize = 8)
ax.set_title("La participación en la RM fluctúa no necesariamente de forma
monótona", fontsize = 8)

```

[7]: Text(0.5, 1.0, 'La participación en la RM fluctúa no necesariamente de forma monótona')



El gráfico mostrado arriba **desmiente** totalmente cualquier tendencia monótona creciente

o decreciente en la participación electoral en la región metropolitana en los últimos años

2.2.2 Resultados por tendencia política en las regiones de Chile

Sabiendo los porcentajes de preferencias por región. Se puede saber en cada proceso por separado si los resultados de **todas las regiones** tuvieron una tendencia hacia derecha, centro o izquierda política. Existe la posibilidad que solo la Región Metropolitana (y otras regiones de la mayor cantidad de participación) exhiba una tendencia hacia el sector ganador

```
[8]: fig, ax = plt.subplots(3, 2, figsize=(10, 10))
    custom_linestyle = (0, (5, 5)) # 5 unidades de línea, 5 unidades de espacio

    for (idx, temp_iter) in enumerate(data_agg.groupby(["Año de Elección"])):
        temp_df = temp_iter[1]
        anio = temp_iter[0][0]
        region = temp_df["Id Región"].unique()[0]

        ax[idx // 2, idx % 2].plot(np.arange(0.3, 0.8, 0.1),
                                   np.arange(0.3, 0.8, 0.1),
                                   color = 'k',
                                   linestyle=custom_linestyle)
        ax[idx // 2, idx % 2].scatter(0.5, 0.5, marker = 'x', color = "k", label = "Igualdad entre sectores (50%)")

        ax[idx // 2, idx % 2].scatter(temp_df["% derecha"], temp_df["% izquierda"],
                                       s=temp_df['Participación efectiva']/temp_df['Participación efectiva'].max()*300,
                                       c="lightblue", alpha = 0.8, label = "Región \n(tamaño por participantes)")

        ax[idx // 2, idx % 2].set_xlim(0.3, 0.7)
        ax[idx // 2, idx % 2].set_ylim(0.3, 0.7)
        ax[idx // 2, idx % 2].set_xticks(np.arange(0.3, 0.8, 0.1))
        ax[idx // 2, idx % 2].set_yticks(np.arange(0.3, 0.8, 0.1))
        ax[idx // 2, idx % 2].set_xlabel('Tasa de votos a la der. política',
                                       fontsize=6)
        ax[idx // 2, idx % 2].set_ylabel('Tasa de votos a la izq. política',
                                       fontsize=6)

        if anio == 2010:
            ax[idx // 2, idx % 2].text(0.33, 0.5, r'$\uparrow$ Más % votos'+
                                       '\n al centro\npolítico', fontsize=8)
            ax[idx // 2, idx % 2].set_ylabel('Tasa de votos al centro político',
                                       fontsize=6)
        else:
            ax[idx // 2, idx % 2].text(0.33, 0.5, r'$\uparrow$ Más % votos'+
                                       '\n a la izquierda\npolítico', fontsize=8)
```

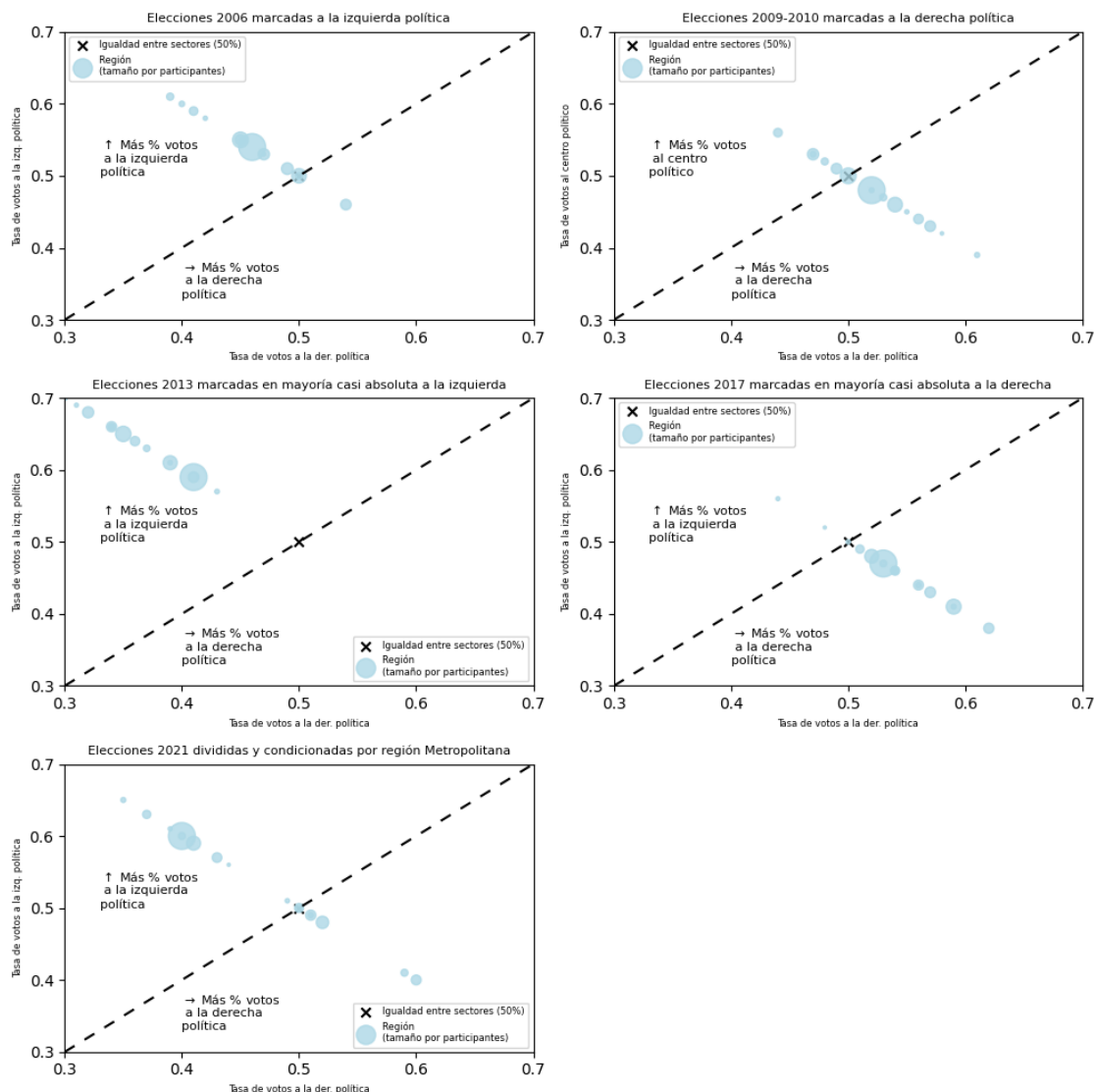


```

ax[idx // 2, idx % 2].text(0.4, 0.33, r'$\rightarrow$ Más % votos'+
                           '\n a la derecha\npolítica', fontsize=8)
ax[idx // 2, idx % 2].add_patch(Circle((-0.5, 7), 0.03, color='none',
ec='grey'))
ax[idx // 2, idx % 2].legend(fontsize = 6)

ax[0, 0].set_title("Elecciones 2006 marcadas a la izquierda política", fontsize=
    8)
ax[0, 1].set_title("Elecciones 2009-2010 marcadas a la derecha política",
    fontsize = 8)
ax[1, 0].set_title("Elecciones 2013 marcadas en mayoría casi absoluta a la
    izquierda", fontsize = 8)
ax[1, 1].set_title("Elecciones 2017 marcadas en mayoría casi absoluta a la
    derecha", fontsize = 8)
ax[2, 0].set_title("Elecciones 2021 divididas y condicionadas por región
    Metropolitana", fontsize = 8)
ax[2, 1].set_axis_off() #ocultar ejes originales
fig.tight_layout()

```



En los gráficos mostrados, en todos los procesos electorales cuando un sector (derecha o izquierda) triunfa, también las regiones con mayor tamaño poblacional muestran su apoyo a ese sector. Se puede observar que:

- En el proceso de 2013 todas las regiones sin excepción tuvieron inclinación por el sector de izquierda política (todos los puntos que representan regiones en el triángulo superior), donde la RM fue incluso una de las que mostró más similitud entre ambos sectores. Esto desmiente totalmente que en 2013 la decisión de la RM fue la más determinante, ya que se observa que tiende a ser la más dividida.
- En el proceso de 2017, se puede observar una inversión en la tendencia hacia el sector de derecha política (triángulo inferior), donde nuevamente la RM fue una de las que mostró más similitud entre ambos sectores.
- El único proceso que puede dar cuenta una gran influencia de la RM fue el de 2021, donde esta región tiende a ser la con mayor inclinación hacia el sector de izquierda.

Como ya se vio la tendencia general de las regiones, ahora se exploran tendencias por ubicación demográfica. Para lo anterior, se definen los siguientes mapas de colores

```
[9]: gradient_iz_der = LinearSegmentedColormap.from_list('my_gradient', (
    # Edit this gradient at https://eltos.github.io/gradient/
    ↪#FF2400-DCDCDC-009BFF
    (0.000, (1.000, 0.141, 0.000)),
    (0.500, (0.863, 0.863, 0.863)),
    (1.000, (0.000, 0.608, 1.000))))
gradient_iz_der
```

[9]:



Bordes de regiones

Usando los mapas de color anteriores. Se grafican las tendencias de sector político en el mapa de Chile:

```
[ ]: def graficar_mapa_votaciones(list_regiones=[]):
    gdf = gpd.read_file("regiones/regiones.shp")
    gdf['Id Región'] = gdf["COD_REGI"].astype(int)
    data_agg_copy = data_agg.copy()
    perc_izq_2006_1 = data_agg_copy.loc[(data_agg_copy['Año de Elección'] == 2006) & (data_agg_copy['Id Región'] == 1), '% izquierda'].values[0]
    ↪perc_der_2006_1 = data_agg_copy.loc[(data_agg_copy['Año de Elección'] == 2006) & (data_agg_copy['Id Región'] == 1), '% derecha'].values[0]
    ↪perc_izq_2006_10 = data_agg_copy.loc[(data_agg_copy['Año de Elección'] == 2006) & (data_agg_copy['Id Región'] == 10), '% izquierda'].values[0]
    ↪perc_der_2006_10 = data_agg_copy.loc[(data_agg_copy['Año de Elección'] == 2006) & (data_agg_copy['Id Región'] == 10), '% derecha'].values[0]
    list_temp = [{'Año de Elección': 2006, 'Id Región': 15, '% izquierda': ↪
    ↪perc_izq_2006_1, '% derecha': perc_der_2006_1},
    ↪{'Año de Elección': 2006, 'Id Región': 14, '% izquierda': ↪
    ↪perc_izq_2006_10, '% derecha': perc_der_2006_10}
    ]
    data_agg_copy = pd.concat([data_agg_copy, pd.DataFrame(list_temp)])
    if list_regiones:
        data_agg_copy = data_agg_copy[data_agg_copy['Id Región'].
    ↪isin(list_regiones)]
    gdf = gdf.merge(data_agg_copy, on = 'Id Región')

    fig, ax = plt.subplots(1, 5, figsize=(30, 15))
    for (idx, temp_iter) in enumerate(gdf.groupby(["Año de Elección"])):
```

```

temp_df = temp_iter[1]
temp_df['% diferencia'] = temp_df['% derecha'] - temp_df['% izquierda']
if idx == 1:
    temp_df.loc[temp_df['% izquierda'] > temp_df['% derecha'], '%
↳diferencia'] = 0
    cmap = gradient_iz_der
    temp_df.plot(column='% diferencia', ax=ax[idx], legend=False,
                  legend_kwds={'label': "% diferencia",
                                'orientation': "horizontal"},
                  # cmap=cmap, vmin=-0.5, vmax=0.5)
                  cmap=cmap, vmin=-0.5, vmax=0.5, edgecolor='#231f20',
↳linewidth=0.1)

    ax[idx].set_axis_off() #ocultar ejes originales
    # Agregar el colorbar de forma manual
    sm = plt.cm.ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin=-0.5, vmax=0.
↳5))
    sm.set_array([]) # Esto es para evitar un warning
    cbar = fig.colorbar(sm, ax=ax, orientation='vertical', fraction=0.02, pad=0.
↳15)
    font = FontProperties(family='Arial', size=14) # Cambiar tamaño y fuente
    cbar.set_label('% votos de diferencia en elecciones presidenciales',
↳fontproperties=font)

    text = "<name:Arial, size:18, color:#231f20, weight:bold>Elecciones
↳presidenciales 2006\n</><name:Arial, size:18, color:#231f20>Victoria
↳izquierda con 53.5%\nMichelle Bachelet</>"
    flexitext(0.5, 1.24, text, va="top", ax=ax[0], ha='center', ma='center')
    text = "<name:Arial, size:18, color:#231f20, weight:bold>Elecciones
↳presidenciales 2010\n</><name:Arial, size:18, color:#231f20>Victoria derecha
↳con 51,6%\nSebastián Piñera</>"
    flexitext(0.5, 1.24, text, va="top", ax=ax[1], ha='center', ma='center')
    text = "<name:Arial, size:18, color:#231f20, weight:bold>Elecciones
↳presidenciales 2013\n</><name:Arial, size:18, color:#231f20>Victoria
↳izquierda con 62,1%\nMichelle Bachelet</>"
    flexitext(0.5, 1.24, text, va="top", ax=ax[2], ha='center', ma='center')
    text = "<name:Arial, size:18, color:#231f20, weight:bold>Elecciones
↳presidenciales 2017\n</><name:Arial, size:18, color:#231f20>Victoria derecha
↳con 54.5%\nSebastián Piñera</>"
    flexitext(0.5, 1.24, text, va="top", ax=ax[3], ha='center', ma='center')
    text = "<name:Arial, size:18, color:#231f20, weight:bold>Elecciones
↳presidenciales 2021\n</><name:Arial, size:18, color:#231f20>Victoria
↳izquierda con 55.8%\nGabriel Boric</>"
    flexitext(0.5, 1.24, text, va="top", ax=ax[4], ha='center', ma='center')

```

```

text = "<name:Arial, size:14, color:#231f20>Para las elecciones del_
↳2006\n aún se mantenía el apoyo a la\n izquierda luego de la vuelta a\n la_
↳democracia</>"
flexitext(0.5, 1.17, text, va="top", ax=ax[0], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20>Para las elecciones del_
↳2012\n fue la única elección de las\n últimas 5 en que la izquierda_
↳no\n participó de la segunda vuelta</>"
flexitext(0.5, 1.17, text, va="top", ax=ax[1], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20>Para las elecciones del_
↳2013\n en todo el país se observa\n una mayoría por el voto a\n la izquierda</>"
flexitext(0.5, 1.17, text, va="top", ax=ax[2], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20>Para las elecciones del_
↳2017\n sólo las regiones de Aysén\n y de Magallanes mantuvieron\n una mayoría_
↳de votos a la izquierda</>"
flexitext(0.5, 1.17, text, va="top", ax=ax[3], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20>Para las elecciones del_
↳2021\n se observa una mayor polarización\n en la diferencia de votos\n entre_
↳izquierda y derecha</>"
flexitext(0.5, 1.17, text, va="top", ax=ax[4], ha='center', ma='center')

text = "<name:Arial, size:14, color:#231f20, weight:bold>2006</>"
flexitext(0.5, 1.08, text, va="top", ax=ax[0], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20, weight:bold>2010</>"
flexitext(0.5, 1.08, text, va="top", ax=ax[1], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20, weight:bold>2013</>"
flexitext(0.5, 1.08, text, va="top", ax=ax[2], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20, weight:bold>2017</>"
flexitext(0.5, 1.08, text, va="top", ax=ax[3], ha='center', ma='center')
text = "<name:Arial, size:14, color:#231f20, weight:bold>2021</>"
flexitext(0.5, 1.08, text, va="top", ax=ax[4], ha='center', ma='center')

fig.text(0.21, 0.92, "Revolución pingüina", va="top", ha='right',_
↳ma='right', rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.34, 0.92, "Terremoto 27F", va="top", ha='right', ma='right',_
↳rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.39, 0.92, "Movilizaciones\n estudiantiles", va="top", ha='right',_
↳ma='right', rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.51, 0.92, "Reforma tributaria\n y educativa", va="top",_
↳ha='right', ma='right', rotation=45, fontsize=14, color='#231f20',_
↳fontfamily='Arial')
fig.text(0.61, 0.92, "Incendios forestales", va="top", ha='right',_
↳ma='right', rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.68, 0.92, "Estallido social", va="top", ha='right', ma='right',_
↳rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.71, 0.92, "Pandemia COVID-19", va="top", ha='right', ma='right',_
↳rotation=45, fontsize=14, color='#231f20', fontfamily='Arial')

```

```

fig.text(0.89, 0.835, "Diferencia en favor\nde la derecha", va="top",
↪ha='center', ma='center', fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.89, 0.18, "Diferencia en favor\nde la izquierda", va="top",
↪ha='center', ma='center', fontsize=14, color='#231f20', fontfamily='Arial')

plt.annotate(
    '', xy=(0.7, 0.85), xycoords='figure fraction', # Punto final de la
↪flecha
    xytext=(0.08, 0.85), textcoords='figure fraction', # Punto inicial de
↪la flecha
    arrowprops=dict(arrowstyle="→", color='black', lw=2)
)
list_lines_coor = [(0.16, 0.55), (0.78, 0.55)], [(0.16, 0.28), (0.78, 0.
↪28)], [(0.16, 0.46), (0.78, 0.46)], [(0.16, 0.7), (0.78, 0.7)], [(0.16, 0.
↪8), (0.78, 0.8)],
                    [(0.2, 0.93), (0.2, 0.96)], [(0.33, 0.93), (0.33, 0.
↪96)], [(0.38, 0.93), (0.38, 0.96)], [(0.5, 0.93), (0.5, 0.96)], [(0.6, 0.
↪93), (0.6, 0.96)], [(0.67, 0.93), (0.67, 0.96)], [(0.7, 0.93), (0.7, 0.96)]
    for line_coor in list_lines_coor:
        line_hor = FancyArrowPatch(
            line_coor[0], line_coor[1], # Coordinadas de inicio y fin
            transform=fig.transFigure, # Usar coordenadas de la figura
            color="black", lw=0.5, alpha=0.5, arrowstyle="→"
        )
        line_hor.set_zorder(0)
        fig.add_artist(line_hor)

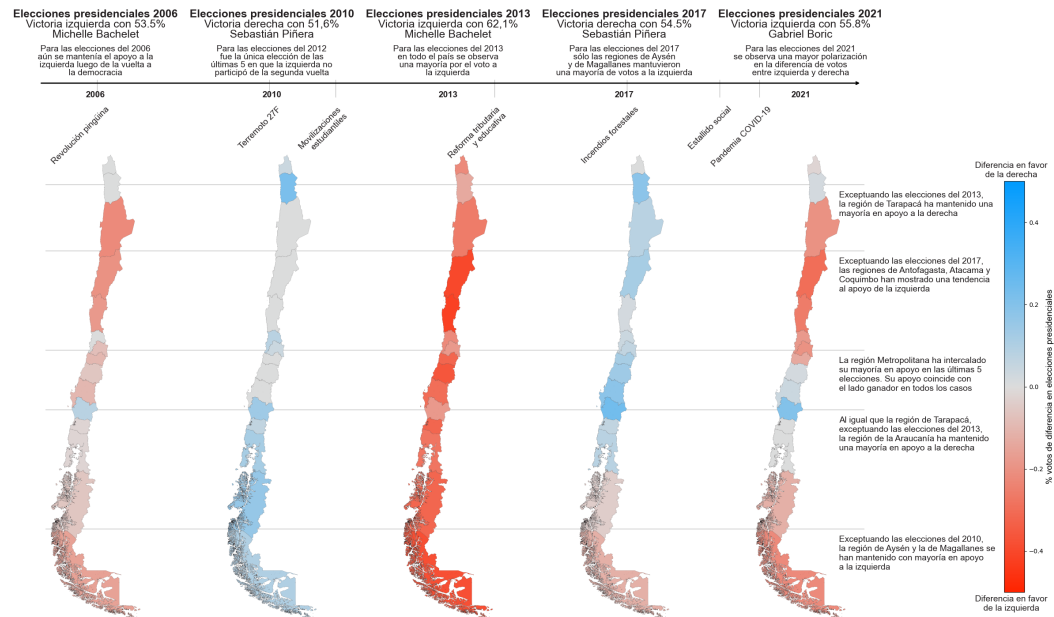
fig.text(0.76, 0.79, "Exceptuando las elecciones del 2013,\nla región de
↪Tarapacá ha mantenido una\nmayoría en apoyo a la derecha", va="top",
↪ha='left', ma='left', fontsize=14, color='#231f20', fontfamily='Arial')
fig.text(0.76, 0.69, "Exceptuando las elecciones del 2017,\nlas regiones de
↪Antofagasta, Atacama y\nCoquimbo han mostrado una tendencia\nal apoyo de la
↪izquierda", va="top", ha='left', ma='left', fontsize=14, color='#231f20',
↪fontfamily='Arial')
fig.text(0.76, 0.54, "La región Metropolitana ha intercalado\nsu mayoría en
↪apoyo en las últimas 5\nelecciones. Su apoyo coincide con\nel lado ganador
↪en todos los casos", va="top", ha='left', ma='left', fontsize=14,
↪color='#231f20', fontfamily='Arial')
fig.text(0.76, 0.45, "Al igual que la región de Tarapacá,\nexceptuando las
↪elecciones del 2013,\nla región de la Araucanía ha mantenido\nuna mayoría en
↪apoyo a la derecha", va="top", ha='left', ma='left', fontsize=14,
↪color='#231f20', fontfamily='Arial')

```

```

fig.text(0.76, 0.27, "Exceptuando las elecciones del 2010,\nla región de
↪Aysén y la de Magallanes se\nhan mantenido con mayoría en apoyo\nna la
↪izquierda", va="top", ha='left', ma='left', fontsize=14, color='#231f20',
↪fontfamily='Arial')
graficar_mapa_votaciones()

```



Se puede ver que en varios procesos electorales presidenciales el país completo puede llegar a apoyar a un sector político en particular. En el 2013 todo el país de norte a sur tuvo una inclinación hacia la izquierda, mientras que en 2017 la relación se invirtió (exceptuando las 2 últimas regiones en el sur de Chile), observando una **transición de sector político**. En el resto de procesos se observa una división, la cual tiende a estar marcada demográficamente. Ejemplo de esto último es que prácticamente todo el norte de Chile apoyó a la izquierda en 2021, mientras que el sur tiende a estar polarizado o dividido. Similarmente, en el 2010 una zona muy marcada en la 2da, 3a y 4a región tuvo una preferencia por el centro político.

Cabe destacar la existencia de regiones con una clara tendencia hacia un sector político en particular, como las regiones de Aysén y Magallanes, las cuales tienden a tener una preferencia por la izquierda política los procesos electorales; mientras que las regiones de Tarapacá y de la Araucanía tienden a tener una preferencia por la derecha política en los procesos electorales.

La región metropolitana fluctúa y ha intercalado su preferencia por sector político en los últimos años, donde en todos los procesos ha apoyado al sector político que resultó vencedor. Esto puede mostrar la influencia que tiene esta región en los resultados electorales del país, en especial al ser la región con mayor cantidad de habitantes (cerca del 40% del total del país). En este punto cabe consultar si la región metropolitana es la región que más influye en los resultados electorales del país o si son todas las regiones las que siguen una tendencia similar de fluctuación en apoyo de sector político.

2.2.3 Transiciones de sector político entre procesos electorales presidenciales

Transiciones en el tiempo Anteriormente, se ha visto las tendencias por proceso electoral. Sin embargo, se ha omitido cómo varían entre procesos. Para lo anterior, primero se grafican los porcentajes de apoyo y su evolución en el tiempo:

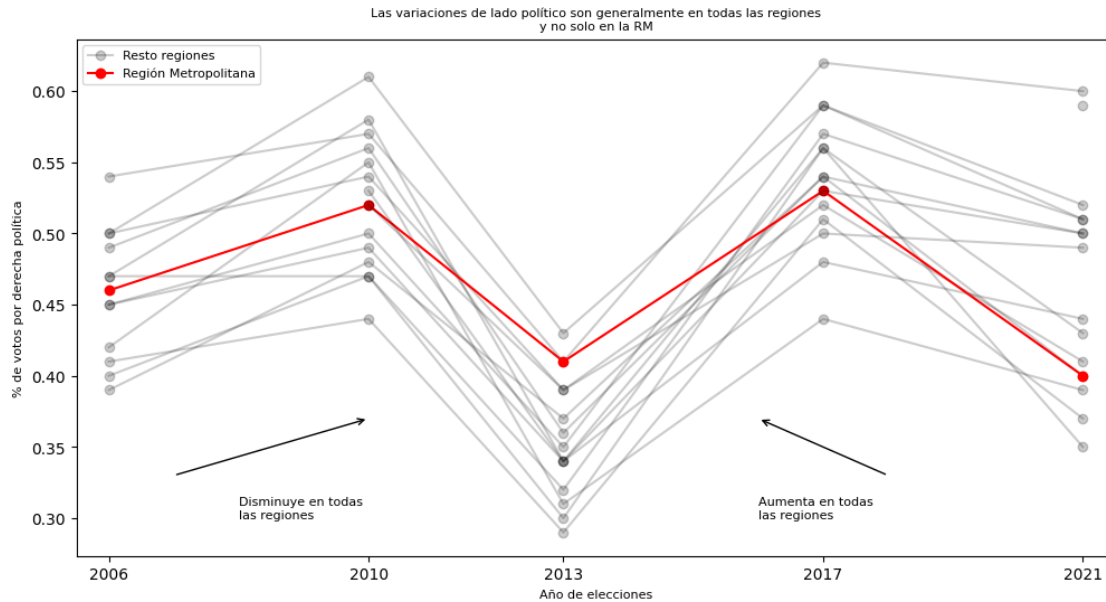
```
[13]: fig, ax = plt.subplots(1, 1, figsize = (12, 6))
flag_regiones = True
for temp_iter in data_agg.groupby(["Id Región"]):
    temp_df = temp_iter[1]
    region = temp_iter[0][0]

    if region == 13:
        ax.plot(temp_df["Año de Elección"], temp_df["% derecha"], alpha = 1, color='r',
        ↪marker = 'o', label = 'Región Metropolitana')

    else:
        if flag_regiones:
            label = "Resto regiones"
            flag_regiones = False
        else:
            label = '_nolegend_'
        ax.plot(temp_df["Año de Elección"], temp_df["% derecha"], alpha = 0.2,
        ↪color = 'k', marker = 'o', label = label)

    ax.set_xticks([2006, 2010, 2013, 2017, 2021])
    ax.set_xticklabels(["2006", "2010", "2013", "2017", "2021"])
    ax.set_xlim(2005.5, 2021.5)
    ax.set_xlabel("Año de elecciones", fontsize = 8)
    ax.set_ylabel("% de votos por derecha política", fontsize = 8)
    ax.set_title("Las variaciones de lado político son generalmente en todas las
    ↪regiones\ny no solo en la RM",
        fontsize = 8)
    ax.annotate('', xy=(2010, 0.37), xytext=(2007, 0.33),
        arrowprops=dict(arrowstyle='->', lw=1, color='black'))
    ax.annotate('', xy=(2016, 0.37), xytext=(2018, 0.33),
        arrowprops=dict(arrowstyle='->', lw=1, color='black'))
    ax.text(2008, 0.3, 'Disminuye en todas\nlas regiones', fontsize=8)
    ax.text(2016, 0.3, 'Aumenta en todas\nlas regiones', fontsize=8)
    ax.legend(fontsize = 8)
```

```
[13]: <matplotlib.legend.Legend at 0x1e3a7131220>
```

Si se observan los procesos juntos, se puede observar que en general a través de los años existe una transición intercalada de sector político en las preferencias. Al ocurrir una transición de izquierda a derecha o viceversa, en general **el cambio de preferencia es en todas las regiones**. Esto muestra una evidencia que las transiciones en el ganador resultante de cada proceso es un **comportamiento país** y no determinado únicamente por la Región Metropolitana

Flujos de votos entre sectores Agregando nulos y blancos

```
[5]: data_San_2021 = data_2021.groupby(["Nro. Región", "Región"]).apply(lambda s: pd.
    ↳Series({
        "Votos izquierda": s[s["Candidato"] == 'GABRIEL BORIC FONT']["Votos_
    ↳TRICEL"].fillna(0).astype(int).sum(),
        "Votos derecha": s[s["Candidato"] == 'JOSE ANTONIO KAST RIST']["Votos_
    ↳TRICEL"].fillna(0).astype(int).sum(),
        "Votos Nulos": s[s["Candidato"] == 'VOTOS NULOS']["Votos TRICEL"].fillna(0).
    ↳astype(int).sum(),
        "Votos en Blanco": s[s["Candidato"] == 'VOTOS EN BLANCO']["Votos TRICEL"].
    ↳fillna(0).astype(int).sum()
    ↳}))

# Resetting the index
data_San_2021.reset_index(inplace=True)

# Calculating effective participation
#data_Sand_2021["Participación efectiva"] = data_Sand_2021["Votos izquierda"] +
    ↳data_Sand_2021["Votos derecha"]+ data_Sand_2021["Votos en Blanco"]+
    ↳data_Sand_2021["Votos Nulos"]
```

```

data_San_2021["Participación efectiva"] = data_San_2021["Votos izquierda"] +
    ↪data_San_2021["Votos derecha"]
data_San_2021.rename(columns={"Nro. Región": "Id Región"}, inplace=True)
data_San_2021["Id Región"] = data_San_2021["Id Región"].astype(int)

# Inserting election year at the beginning
data_San_2021.insert(0, 'Año de Elección', 2021)

izq_lista = ["EDUARDO FREI", 'MICHELLE BACHELET JERIA', 'MICHELLE BACHELET',
             'ALEJANDRO GUILLIER ALVAREZ']
der_lista = ['SEBASTIAN PIÑERA', 'SEBASTIAN PIÑERA ECHENIQUE', 'EVELYN MATTHEI',
             ↪FORNET']

data['Candidato (a)'] = data['Candidato (a)'].replace('VOTOS EN BLANCO ', 'VOTOS_
    ↪EN BLANCO')
data['Candidato (a)'] = data['Candidato (a)'].replace('VOTOS NULOS ', 'VOTOS_
    ↪NULOS')

data_San = data.groupby(["Año de Elección", "Id Región", "Región"]).
    ↪apply((lambda s: pd.Series({
        "Votos izquierda": s[s["Candidato (a)"].isin(izq_lista)]["Votos Totales"].
        ↪astype(int).sum(),
        "Votos derecha": s[s["Candidato (a)"].isin(der_lista)]["Votos Totales"].
        ↪astype(int).sum(),
        "Votos Nulos": s[s["Candidato (a)"] == 'VOTOS NULOS']["Votos Totales"].
        ↪fillna(0).astype(int).sum(),
        "Votos en Blanco": s[s["Candidato (a)"] == 'VOTOS NULOS']["Votos Totales"].
        ↪fillna(0).astype(int).sum()
    })))
)
data_San.reset_index(inplace = True)

#data_Sand["Participación efectiva"] = data_Sand["Votos izquierda"] +
    ↪data_Sand["Votos derecha"]+ data_Sand["Votos en Blanco"]+ data_Sand["Votos_
    ↪Nulos"]
data_San["Participación efectiva"] = data_San["Votos izquierda"] +
    ↪data_San["Votos derecha"]
data_San_agg = pd.concat((data_San, data_San_2021))

df_san=data_San_agg[['Año de Elección','Votos izquierda','Votos derecha','Votos_
    ↪Nulos','Votos en Blanco','Participación efectiva']].groupby(['Año de_
    ↪Elección']).sum().reset_index()
df_san['Nulos y Blancos']=(df_san['Votos Nulos']+df_san['Votos en Blanco'])

```

```

df_san['%Nulos y Blancos']=(df_san['Nulos y Blancos']/df_san['Participación_
↪efectiva'])*100
df_san['% Izquierda'] = (df_san['Votos izquierda'] / df_san['Participación_
↪efectiva']) * 100
df_san['% Derecha'] = (df_san['Votos derecha'] / df_san['Participación_
↪efectiva']) * 100
df_san.drop(columns=['Votos Nulos','Votos en Blanco'],inplace=True)
df_san.set_index('Año de Elección', inplace=True)
df_san

```

C:\Users\Diego\AppData\Local\Temp\ipykernel_17664\3304849650.py:1:

DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.

```

data_San_2021 = data_2021.groupby(["Nro. Región", "Región"]).apply(lambda s:
pd.Series({

```

C:\Users\Diego\AppData\Local\Temp\ipykernel_17664\3304849650.py:27:

DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.

```

data_San = data.groupby(["Año de Elección", "Id Región",
"Región"]).apply((lambda s: pd.Series({

```

```

[5]:          Votos izquierda  Votos derecha  Participación efectiva \
Año de Elección
2006                3723019        3236394                6959413
2010                3367790        3591182                6958972
2013                3470379        2111891                5582270
2017                3147868        3788641                6936509
2021                4596579        3640606                8237185

```

```

          Nulos y Blancos  %Nulos y Blancos  % Izquierda  % Derecha
Año de Elección
2006                309944        4.453594    53.496164    46.503836
2010                378980        5.445919    48.394935    51.605065
2013                165832        2.970691    62.167882    37.832118
2017                112508        1.621969    45.381156    54.618844
2021                92147         1.118671    55.802789    44.197211

```

```

[6]: df_padron=pd.read_csv('people_enabled_to_vote_chile_2006_2021.csv')
df_padron.head()
resultado=pd.merge(df_padron,df_san,left_on='Year',right_on='Año de_
↪Elección',how='left')

```

```

resultado['Participación']=resultado['Participación efectiva']+resultado['Nulos_
↳y Blancos']
resultado=resultado[['Year', 'Padron', 'Participación']]
resultado['No_participación']=resultado['Padron']-resultado['Participación']

resultado['%Participación']=resultado['Participación']/resultado['Padron']*100
resultado['%No_participación']=resultado['No_participación']/
↳resultado['Padron']*100

resultado

```

```

[6]:   Year    Padron  Participación  No_participación  %Participación  \
0  2006   8220897      7269357           951540      88.425351
1  2010   8285186      7337952           947234      88.567137
2  2013  13573143      5748102          7825041      42.349086
3  2017  14308151      7049017          7259134      49.265744
4  2021  15030974      8329332          6701642      55.414453

      %No_participación
0          11.574649
1          11.432863
2          57.650914
3          50.734256
4          44.585547

```

Con el fin de visualizar los flujos de votos entre sectores, se parte primero por graficar los flujos de participación:

```

[7]: # Definir tendencias de análisis: solo Participación y No_participación
tendencias = ['Participación', 'No_participación']

# Función para calcular la transición entre dos años consecutivos
def calcular_transicion(df, año_anterior, año_actual):
    valores_anterior = df.loc[df['Year'] == año_anterior, tendencias].iloc[0]
    valores_actual = df.loc[df['Year'] == año_actual, tendencias].iloc[0]
    cambios = valores_actual - valores_anterior

    ganadores = cambios[cambios > 0]
    perdedores = cambios[cambios < 0] * -1

    total_ganancias = ganadores.sum()
    total_perdidas = perdedores.sum()

    # Crear matriz de transición vacía
    transicion = pd.DataFrame(0.0, index=tendencias, columns=tendencias)

    # Calcular transiciones de Participación y No_participación entre años

```

```

    if total_ganancias > 0 and total_perdidas > 0:
        for perdedor in perdedores.index:
            for ganador in ganadores.index:
                transicion.loc[perdedor, ganador] = perdedores[perdedor] *  $\frac{ganadores[ganador]}{total\_ganancias}$ 
            elif total_perdidas > 0:
                for perdedor in perdedores.index:
                    transicion.loc[perdedor, 'No_participación'] += perdedores[perdedor]

    # Mantener o transferir valores de Participación y No_participación
    for tendencia in tendencias:
        if cambios[tendencia] >= 0:
            transicion.loc[tendencia, tendencia] +=  $\frac{cambios[tendencia]}{total\_ganancias}$ 
        min(valores_anterior[tendencia], valores_actual[tendencia])
    else:
        transicion.loc[tendencia, tendencia] += valores_actual[tendencia]

    return transicion

# Generar diccionario de transiciones para cada par de años consecutivos
transiciones = {}
for i in range(1, len(resultado)):
    año_anterior = resultado['Year'].iloc[i - 1]
    año_actual = resultado['Year'].iloc[i]
    transiciones[f"{año_anterior}_to_{año_actual}"] =  $\frac{transicion.loc[año\_anterior, año\_actual]}{total\_ganancias}$ 
    calcular_transicion(resultado, año_anterior, año_actual)

# Preparar listas para los datos del gráfico Sankey
source_labels_list = []
target_labels_list = []
values_list = []

# Iterar sobre cada matriz de transición en el diccionario `transiciones`
for key, transicion in transiciones.items():
    año_anterior, año_actual = key.split("_to_")

    for i, tendencia_origen in enumerate(tendencias):
        for j, tendencia_destino in enumerate(tendencias):
            source_label = f"{año_anterior}-{tendencia_origen}"
            target_label = f"{año_actual}-{tendencia_destino}"

            value = transicion.iloc[i, j]

            if value > 0:
                source_labels_list.append(source_label)
                target_labels_list.append(target_label)
                values_list.append(value)

```

```

# Crear lista de etiquetas únicas y mapear a índices, organizando
↳ "Participación" primero
labels = sorted(list(set(source_labels_list + target_labels_list)), key=lambda
↳ x: (int(x.split('-')[0]), x.split('-')[1] == 'No_participación'))
label_to_index = {label: idx for idx, label in enumerate(labels)}

source_indices = [label_to_index[label] for label in source_labels_list]
target_indices = [label_to_index[label] for label in target_labels_list]

labels_with_percentages = []
for label in labels:
    año, categoria = label.split('-')
    año = int(año)

    # Buscar el valor adecuado según la categoría del nodo
    if categoria == 'Participación':
        percentage = resultado.loc[resultado['Year'] == año, '%Participación'].
↳ values[0]
        labels_with_percentages.append(f"{percentage:.2f}%")
    elif categoria == 'No_participación':
        percentage = resultado.loc[resultado['Year'] == año,
↳ '%No_participación'].values[0]
        labels_with_percentages.append(f"{percentage:.2f}%")

# Asignar colores a los nodos según la categoría
node_colors = []
for label in labels:
    if 'Participación' in label:
        node_colors.append('rgba(0, 162, 208, 1)') # Azul para Participación
    elif 'No_participación' in label:
        node_colors.append('rgba(101, 141, 158, 1)') # Verde para
↳ No_participación

link_colors = []
for i in range(len(source_labels_list)):
    source_label = source_labels_list[i]
    if 'Participación' in source_label:
        link_colors.append('rgba(0, 162, 208, 0.5)')
    elif 'No_participación' in source_label:
        link_colors.append('rgba(101, 141, 158, 0.5)')

# Crear el diagrama Sankey
fig = go.Figure(data=[go.Sankey(
    node=dict(
        pad=30,
        thickness=20,

```

```

        line=dict(color="black", width=0.5),
        label=labels_with_percentages,
        color=node_colors
    ),
    link=dict(
        source=source_indices,
        target=target_indices,
        value=values_list,
        color=link_colors
    )
)]

# Añadir anotaciones para crear una leyenda personalizada debajo del gráfico
fig.add_annotation(
    x=0.1, y=-0.05, xref="paper", yref="paper",
    text="<span style='color:rgba(0, 162, 208, 0.5)''><b>% Participación</b></span>",
    showarrow=False, font=dict(size=12)
)

fig.add_annotation(
    x=0.7, y=-0.05, xref="paper", yref="paper",
    text="<span style='color:rgba(101, 141, 158, 0.5)''><b>% No participación</b></span>",
    showarrow=False, font=dict(size=12)
)

fig.update_layout(
    title=dict(
        text="<b>Transiciones de Participación y No Participación (Por Año)</b>",
        font=dict(size=16, color="black", family="Arial", weight="bold")
    ),
    shapes=[
        dict(
            type="line",
            x0=-0.01, x1=1, y0=1.1, y1=1.1,
            xref="paper", yref="paper",
            line=dict(color="black", width=2)
        )
    ],
    font_size=10,
    width=1400,
    height=900,
    margin=dict(t=100, b=100)
)

```

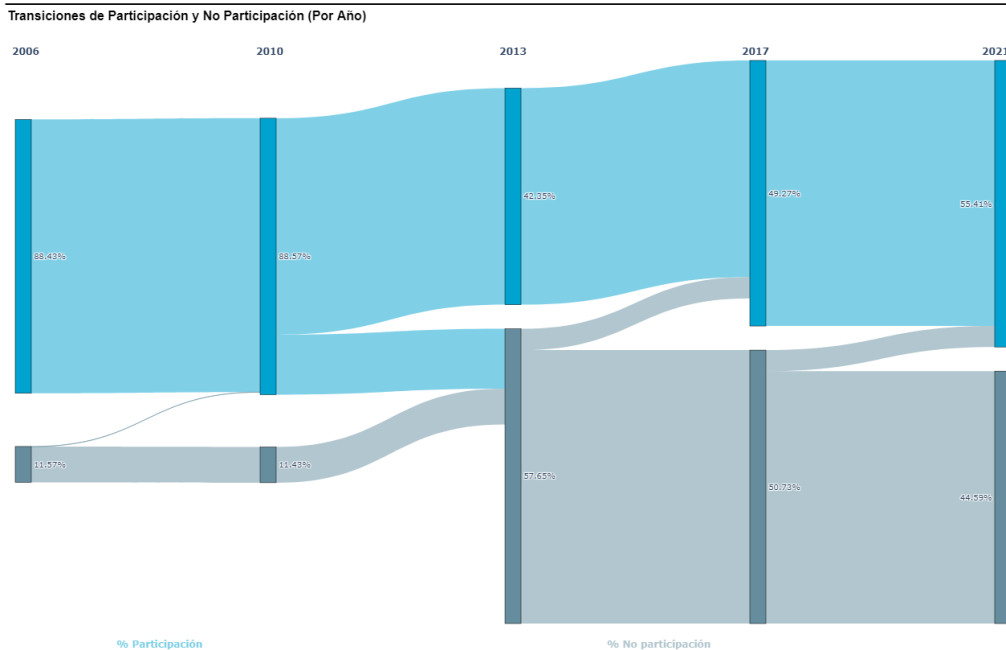
```

# Añadir años en la parte superior
fig.add_annotation(
    x=-0.005, y=1.03, xref="paper", yref="paper",
    text="<b>2006</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.24, y=1.03, xref="paper", yref="paper",
    text="<b>2010</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.5, y=1.03, xref="paper", yref="paper",
    text="<b>2013</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.76, y=1.03, xref="paper", yref="paper",
    text="<b>2017</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=1, y=1.03, xref="paper", yref="paper",
    text="<b>2021</b>",
    showarrow=False, font=dict(size=12)
)

fig.write_image("sankey_participaciones.png")
Image("sankey_participaciones.png")

```

[7]:



Se puede ver que existe un aumento explosivo de no participación desde 2013 en adelante. Este porcentaje de no participación se ha mantenido en el tiempo desde 2013 con pequeños flujos hacia el sector de la población que si participó. Adicionalmente, se observa un flujo importante de participación hacia el sector no participante entre 2010 y 2013. Esto puede explicarse porque el 2012 dejó de regir el voto obligatorio y pasó a ser voluntario, lo que puede haber generado un aumento de no participación en 2013.

Se genera el mismo tipo de visualización, pero esta vez sobre los sectores políticos y los flujos de votos:

```
[8]: import plotly.graph_objects as go
import pandas as pd

# Definir tendencias de análisis
tendencias = ['% Izquierda', '% Derecha', '%Nulos y Blancos']
df = df_san

def calcular_transicion(df, año_anterior, año_actual):
    porcentajes_anterior = df.loc[año_anterior, tendencias]
    porcentajes_actual = df.loc[año_actual, tendencias]
    cambios = porcentajes_actual - porcentajes_anterior

    # Identificar ganadores y perdedores
    ganadores = cambios[cambios > 0]
    perdedores = cambios[cambios < 0] * -1 # Convertir a valores positivos
```

```

# Sumar total de ganancias y pérdidas
total_ganancias = ganadores.sum()
total_perdidas = perdedores.sum()

# Crear matriz de transición vacía
transicion = pd.DataFrame(0.0, index=tendencias, columns=tendencias)

# Si hay perdedores y ganadores
if total_ganancias > 0 and total_perdidas > 0:
    for perdedor in perdedores.index:
        for ganador in ganadores.index:
            # Proporción de la pérdida del perdedor que va al ganador
            transicion.loc[perdedor, ganador] = perdedores[perdedor] *
↳ (ganadores[ganador] / total_ganancias)
# Si no hay ganadores, los votos perdidos van a VB_VN
elif total_perdidas > 0:
    for perdedor in perdedores.index:
        transicion.loc[perdedor, 'VB-VN'] += perdedores[perdedor]

# Los que mantienen o aumentan mantienen sus votos anteriores
for tendencia in tendencias:
    if cambios[tendencia] >= 0:
        transicion.loc[tendencia, tendencia] +=
↳ min(porcentajes_anterior[tendencia], porcentajes_actual[tendencia])
    else:
        transicion.loc[tendencia, tendencia] +=
↳ porcentajes_actual[tendencia]

return transicion

# Listas para almacenar los datos necesarios para el gráfico Sankey
source_labels_transicion = []
target_labels_transicion = []
values_transicion = []

# Procesar cada par de años para construir las listas de source, target y values
for idx in range(len(df.index) - 1):
    año_anterior = df.index[idx]
    año_actual = df.index[idx + 1]
    transicion = calcular_transicion(df, año_anterior, año_actual)

# Excluir columnas y filas adicionales
matriz_sin_totales = transicion.loc[tendencias, tendencias]

# Construir las etiquetas de origen y destino, y los valores
for i, tendencia_origen in enumerate(tendencias):

```

```

for j, tendencia_destino in enumerate(tendencias):
    source_label = f"{año_anterior}-{tendencia_origen}"
    target_label = f"{año_actual}-{tendencia_destino}"
    value = matriz_sin_totales.iloc[i, j]

    # Solo agregamos conexiones con valor positivo
    if value > 0:
        source_labels_transicion.append(source_label)
        target_labels_transicion.append(target_label)
        values_transicion.append(value)

# Crear una lista de etiquetas únicas
labels_transicion = list(set(source_labels_transicion +
    ↪target_labels_transicion))

# Crear un diccionario para mapear etiquetas a índices
label_to_index_transicion = {label: idx for idx, label in
    ↪enumerate(labels_transicion)}

# Mapear las etiquetas de origen y destino a índices
source_indices_transicion = [label_to_index_transicion[label] for label in
    ↪source_labels_transicion]
target_indices_transicion = [label_to_index_transicion[label] for label in
    ↪target_labels_transicion]

# Función para convertir hex a RGBA
def hex_to_rgba(hex_color, alpha=1.0):
    hex_color = hex_color.lstrip('#')
    r, g, b = tuple(int(hex_color[i:i+2], 16) for i in (0, 2, 4))
    return f'rgba({r}, {g}, {b}, {alpha})'

# Mapa de colores para cada tendencia
color_map = {
    '% Derecha': '#7C87B0',      # Azul
    '% Nulos y Blancos': '#E8B141',    # Naranja
    '% Izquierda': '#ED7A73'        # Rojo
}

# Generar lista de colores RGBA para los nodos con opacidad completa
node_colors_transicion = []
for label in labels_transicion:
    tendencia = label.split('-')[1]
    hex_color = color_map.get(tendencia, '#7f7f7f') # Gris por defecto
    rgba_color = hex_to_rgba(hex_color, alpha=1.0) # Opacidad completa para
    ↪nodos
    node_colors_transicion.append(rgba_color)

```

```

# Generar lista de colores RGBA para los enlaces con menor opacidad
link_colors_transicion = []
for src_idx in source_indices_transicion:
    # Obtener la tendencia del nodo de origen
    tendencia = labels_transicion[src_idx].split('-')[1]
    hex_color = color_map.get(tendencia, '#7f7f7f') # Gris por defecto
    # Convertir a RGBA con opacidad reducida (por ejemplo, 0.3)
    rgba_color = hex_to_rgba(hex_color, alpha=0.3)
    link_colors_transicion.append(rgba_color)

# Preparar etiquetas de nodos con porcentajes
labels_with_percentages_transicion = []
for label in labels_transicion:
    tendencia = label.split('-')[1]
    año = label.split('-')[0]

    # Obtener el porcentaje correspondiente a la tendencia y el año
    porcentaje = df.loc[int(año), tendencia]
    label_with_percentage = f"{porcentaje:.2f}%" # Formatear con solo el
    porcentaje
    labels_with_percentages_transicion.append(label_with_percentage)

# Crear el gráfico Sankey con las etiquetas actualizadas
fig = go.Figure(data=[go.Sankey(
    node=dict(
        pad=60, # Controlar el espacio entre los nodos
        thickness=30, # Controlar el grosor de las líneas
        line=dict(color="black", width=0.5),
        label=labels_with_percentages_transicion, # Usar solo los porcentajes
    en los nodos
        color=node_colors_transicion
    ),
    link=dict(
        source=source_indices_transicion,
        target=target_indices_transicion,
        value=values_transicion,
        color=link_colors_transicion
    )
)])

# Añadir línea sobre el título
fig.update_layout(
    title=dict(
        text="<b>Análisis de la Evolución Política por Año</b>",
        font=dict(size=16, color="black", family="Arial", weight="bold")
    ),
    shapes=[

```

```

        dict(
            type="line",
            x0=-0.01, x1=1, y0=1.1, y1=1.1,
            xref="paper", yref="paper",
            line=dict(color="black", width=2)
        )
    ],
    font_size=10,
    width=1400,
    height=900,
    margin=dict(t=100, b=100)
)

# Añadir anotaciones para crear una leyenda personalizada debajo del gráfico
fig.add_annotation(
    x=0.1, y=-0.05, xref="paper", yref="paper",
    text="<span style='color:#7C87B0'><b>% Derecha</b></span> - Azul",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.4, y=-0.05, xref="paper", yref="paper",
    text="<span style='color:#ED7A73'><b>% Izquierda</b></span> - Rojo",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.7, y=-0.05, xref="paper", yref="paper",
    text="<span style='color:#E8B141'><b>% Nulos<br>y Blancos</b></span>",
    showarrow=False, font=dict(size=12)
)

# Añadir años en la parte superior
fig.add_annotation(
    x=-0.005, y=1.03, xref="paper", yref="paper",
    text="<b>2006</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.24, y=1.03, xref="paper", yref="paper",
    text="<b>2010</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(
    x=0.5, y=1.03, xref="paper", yref="paper",
    text="<b>2013</b>",
    showarrow=False, font=dict(size=12)
)
fig.add_annotation(

```

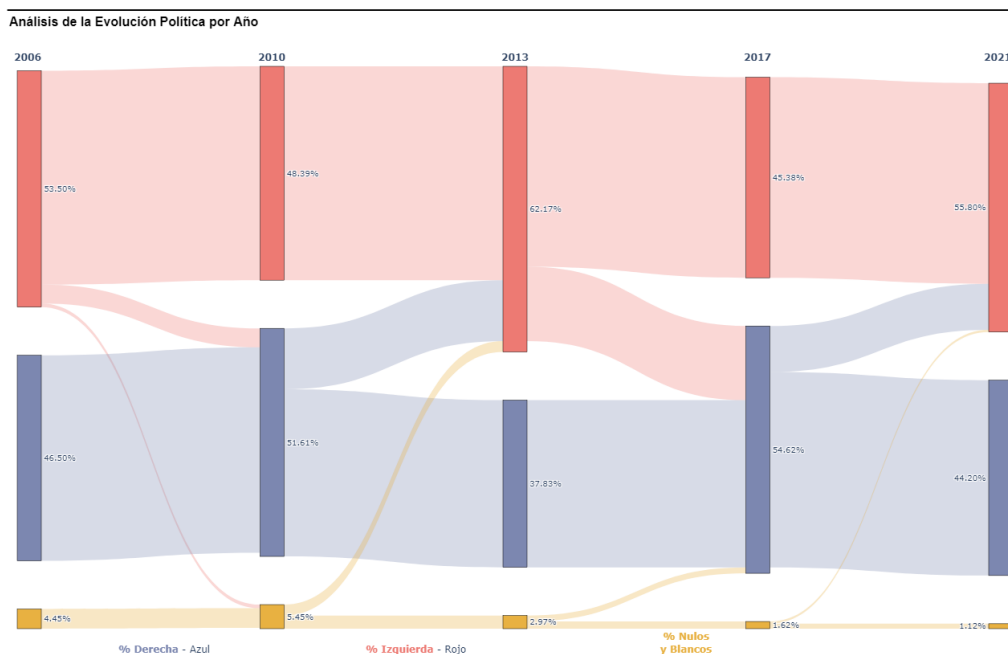
```

x=0.76, y=1.03, xref="paper", yref="paper",
text="<b>2017</b>",
showarrow=False, font=dict(size=12)
)
fig.add_annotation(
x=1, y=1.03, xref="paper", yref="paper",
text="<b>2021</b>",
showarrow=False, font=dict(size=12)
)

fig.write_image("sankey_sectores.png")
Image("sankey_sectores.png")

```

[8]:



Se puede ver que de 2010 a 2013 y de 2013 a 2017 hay un **flujo importante de votos que cambian de sector político**. Esta tendencia también está presente de 2006 a 2010 y desde 2017 a 2021, pero menos notorio. Estos flujos intercalados de votos de un sector a otro son los que se pueden atribuir a un resultado también intercalado del sector político ganador. Finalmente, se junta la visualización anterior con el de flujos de participación electoral:

```

[9]: import plotly.graph_objects as go

# Crear la figura para incluir ambos gráficos uno encima del otro
fig = go.Figure()

```

```

# Primer gráfico Sankey (Transiciones de Participación y No Participación)
fig.add_trace(go.Sankey(
    domain=dict(y=[0.73, 1.0]), # Posiciona el gráfico superior en la mitad
    ↪superior
    node=dict(
        pad=30,
        thickness=30,
        line=dict(color="black", width=0.5),
        label=labels_with_percentages,
        color=node_colors
    ),
    link=dict(
        source=source_indices,
        target=target_indices,
        value=values_list,
        color=link_colors
    )
))

# Segundo gráfico Sankey (Análisis de la Evolución Política con solo
    ↪porcentajes)
fig.add_trace(go.Sankey(
    domain=dict(y=[0, 0.65]), # Posiciona el gráfico inferior en la mitad
    ↪inferior
    node=dict(
        pad=30,
        thickness=30,
        line=dict(color="black", width=0.5),
        label=labels_with_percentages_transicion,
        color=node_colors_transicion
    ),
    link=dict(
        source=source_indices_transicion,
        target=target_indices_transicion,
        value=values_transicion,
        color=link_colors_transicion
    )
))

# Configuración del layout general
# Añadir línea sobre el título
fig.update_layout(
    title=dict(
        text="<b>Análisis de la Evolución Política por Año en Elecciones
    ↪Presidenciales</b>",
        font=dict(size=16, color="black", family="Arial", weight="bold")
    ),

```

```

shapes=[
    dict(
        type="line",
        x0=-0.01, x1=1, y0=1.1, y1=1.1,
        xref="paper", yref="paper",
        line=dict(color="black", width=2)
    )
],
font_size=10,
width=1400,
height=900,
margin=dict(t=100, b=100)
)

# Añadir la nota de la fuente en el extremo inferior izquierdo
fig.add_annotation(
    x=0, y=-0.12,
    xref="paper", yref="paper",
    text="Fuente: Servicio Electoral de Chile",
    showarrow=False,
    font=dict(size=10, color="gray"),
    align="left"
)

fig.add_annotation(
    x=0.975, y=0.95, # Ajusta la posición según la ubicación de "Participación
↪2021"
    xref="paper", yref="paper",
    text="Mayor Participación<br>en Elecciones Presidenciales<br>7,61 Millones
↪de Votantes",
    showarrow=True,
    arrowhead=2,
    ax=-110, # Ajuste de posición del texto
    ay=-15, # Ajuste de posición del texto
    font=dict(size=10, color="black"),
    ##bgcolor="lightyellow", # Fondo de la nota para destacarla
    #bordercolor="black",
    #borderwidth=1
)

fig.add_annotation(
    x=0.48, y=0.76, # Ajusta la posición según la ubicación de "Participación
↪2021"
    xref="paper", yref="paper",

```



```

text="Inicio Inscripción Automática<br>y Voto Voluntarios",
showarrow=True,
arrowhead=2,
ax=-110, # Ajuste de posición del texto
ay=5, # Ajuste de posición del texto
font=dict(size=10, color="black"),
##bgcolor="lightyellow", # Fondo de la nota para destacarla
#bordercolor="black",
#borderwidth=1
)

fig.add_annotation(
    x=0.985, y=-0.01, # Ajusta la posición según la ubicación de
    ↪ "Participación 2021"
    xref="paper", yref="paper",
    text="Menor cantidad de <br>Votos Blancos o Nulos",
    showarrow=True,
    arrowhead=2,
    ax=-60, # Ajuste de posición del texto
    ay=40, # Ajuste de posición del texto
    font=dict(size=10, color="black"),
    ##bgcolor="lightyellow", # Fondo de la nota para destacarla
    #bordercolor="black",
    #borderwidth=1
)
#ED7A73
fig.add_annotation(x=1.02, y=0.14, xref="paper", yref="paper",
    text="<span style='color:#7C87B0'><b>% Izquierda</b></span> ",
    showarrow=False, font=dict(size=12), textangle=-90)
fig.add_annotation(x=1.02, y=0.5, xref="paper", yref="paper",
    text="<span style='color:#ED7A73'><b>% Derecha</b></span>",
    showarrow=False, font=dict(size=12), textangle=-90)
fig.add_annotation(x=1.03, y=-0.07, xref="paper", yref="paper",
    text="<span style='color:#E8B141'><b>% Nulos<br>y Blancos</b></span>",
    showarrow=False, font=dict(size=12), textangle=-90)
fig.add_annotation(x=1.02, y=1.02, xref="paper", yref="paper",
    text="<span style='color:#00A2D0'><b>% Participación</b></span> ",
    showarrow=False, font=dict(size=11), textangle=-90)
fig.add_annotation(x=1.02, y=0.85, xref="paper", yref="paper",
    text="<span style='color:#658D9E'><b>% No Participación</b></span>",
    showarrow=False, font=dict(size=11), textangle=-90)

fig.add_annotation(x=-0.005, y=1.03, xref="paper", yref="paper",
    text="<b>2006</b>", showarrow=False, font=dict(size=12))
fig.add_annotation(x=0.24, y=1.03, xref="paper", yref="paper",
    text="<b>2010</b>", showarrow=False, font=dict(size=12))
fig.add_annotation(x=0.5, y=1.03, xref="paper", yref="paper",

```

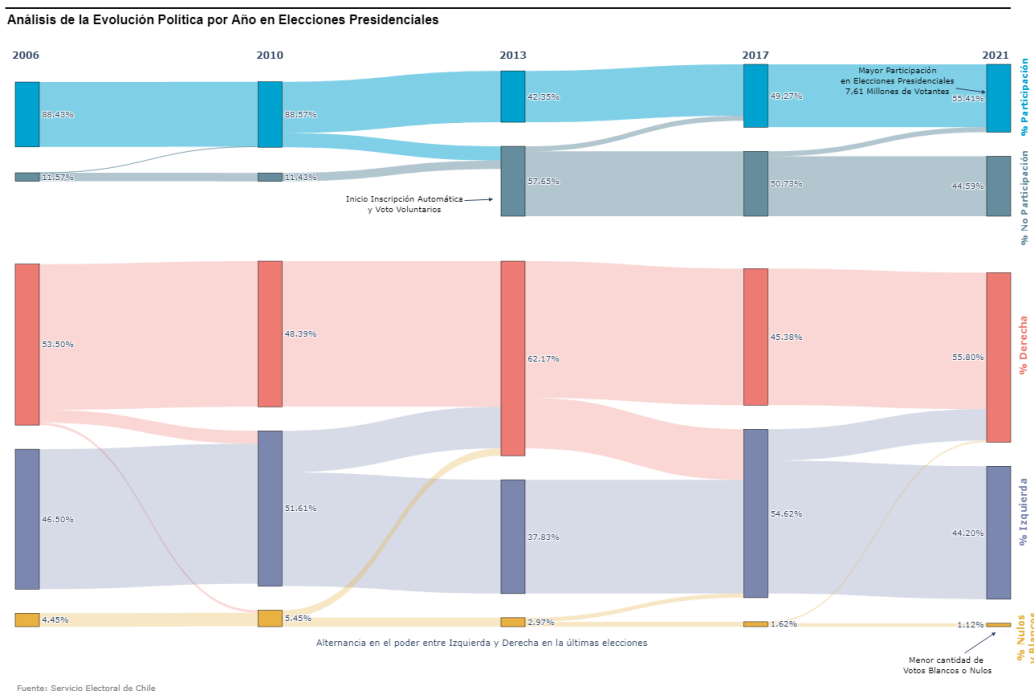
```

        text="<b>2013</b>", showarrow=False, font=dict(size=12))
fig.add_annotation(x=0.76, y=1.03, xref="paper", yref="paper",
        text="<b>2017</b>", showarrow=False, font=dict(size=12))
fig.add_annotation(x=1, y=1.03, xref="paper", yref="paper",
        text="<b>2021</b>", showarrow=False, font=dict(size=12))
fig.add_annotation(x=0.3, y=-0.04,
        xref="paper", yref="paper",
        text="Alternancia en el poder entre Izquierda y Derecha en la últimas",
        showarrow=False,
        font=dict(size=11),
        align="left")

fig.write_image("sankey_unificado.png")
Image("sankey_unificado.png")

```

[9]:



Aquí se puede ver unificado los resultados expuesto anteriormente por separado. Al ver la participación unificada con los votos, permite visualizar los periodos donde hubieron más cambios tanto de participación como de cambios de tendencia política. En este caso, se puede ver que desde 2013 en adelante, pese a ver fluctuaciones y resultados intercalados (izquierda-derecha como ganador), la participación en las elecciones no ha mostrado grandes cambios a nivel de país (no separado por regiones),

2.3 Conclusiones

Se puede concluir, que en general existe un comportamiento altamente dinámico en los procesos electorales presidenciales chilenos. Han existido fuertes cambios de participación electoral, tanto a nivel país como a nivel de regiones y además el país completo puede cambiar de preferencia de sector político a través del tiempo. Mediante las visualizaciones expuestas, se corrobora que en gran parte de los procesos electorales la tendencia país se torna hacia el sector ganador y no necesariamente la RM es la determinante, pese a ser la zona del país con mayor cantidad de habitantes.

El comportamiento por región es altamente dinámico, donde se pudo ver, que a través de los distintos procesos electorales en general el apoyo a la derecha política sube y baja de manera intercalada **en prácticamente todas las regiones en cada elección presidencial**. No obstante, sí existen tendencias regionales que se pueden observar a través de los años. Estos son los casos destacables de las regiones de Aysén y Magallanes, las cuales tienden a tener una preferencia por la izquierda política los procesos electorales; mientras que las regiones de Tarapacá y de la Araucanía tienden a tener una preferencia por la derecha política en los procesos electorales. Estas tendencias son sostenidas en el tiempo y, aunque existen cambios de preferencia en un año, este cambio es menor en comparación a la tendencia general del país.

Finalmente, a través de los gráficos de flujos de votos y participación, se pudo constatar como ha sido la dinámica de las transiciones de sector político. En la participación hubo una baja importante desde 2013 en adelante y se pudo ver claramente que hay un sector importante de la población que intercala sus preferencia por izquierda o derecha (votos de izquierda que pasan a derecha y viceversa), por lo que no es de un sector definido necesariamente.