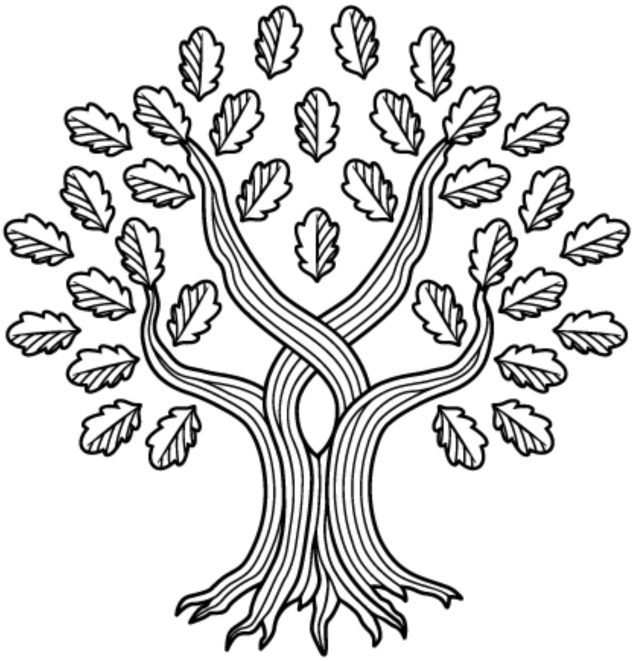
–trigger-event-filters=“bucket=YOUR_STORAGE_BUCKET
“

Divorces

Big Data

Universidad Panamericana
Facultad de Ingeniería



Noviembre 27 , 2024

Hernández Toledo Daniel 0243179@up.edu.mx
Lorenzo Reinoso Fuentes 0212511@up.edu.mx

Contents

1 Introduction

This project aims to address the challenge of loading and processing data from INEGI in cloud function to update a BigQuery table that will store a comprehensive historical record of divorces in Mexico.

The primary goal is to analyze this historical data to identify patterns and trends related to divorces in the country. The insights gained from this analysis could provide valuable information about the reasons behind marital dissolution in Mexico, which may help develop strategies and policies to promote stronger and healthier relationships.

2 Architecture

This project is center of using the Google Cloud tools, to process, analyze and store data from the divorces in Mexico.

2.1 Bucket / Cloud Storage / Cloud Functions / BigQuery

For this project we use a script that call a request from the INEGI web page to download the data from the years (2020-2023) then we store this files in a bucket.

The process is a lit it bit manually because the function is going to be trigger google.cloud.storage.object.v1.finalized by this object in cloud storage that is when a new file overwrite or update. The reason for that is that this data is only updated one time in the year.

2.2 / Cloud Functions

The cloud function is a script in python that allow us to process 30 tables in the buckets and process the information automatic and then it is allow us to wrote the result in bigquery.

2.2.1 Schema

tipo_div,ent_regis,mun_regis,loc_regis,tloc_regis,ent_mat,mun_mat,local_mat,tloc_mat,dia_mat,mes_mat,a
vor,causa,hijos,hij_men,custodia,cus_hij,pat_pot,pat_hij,pension,pen_hij,naci_div1,edad_div1,nacim_div1,e

2.2.2 Script

```
import polars as pl
import os
import os
from google.cloud import storage
from pandas_gbq import to_gbq

########### Funtion To Processs all Joins #############

def do_joins_to_main(lista_archivos, conjunto_de_datos):

    for archivo in lista_archivos:
            list_joins = diccionario_de_datos.filter(pl.col("catalogo") ==
archivo)[["nemonico", "catalogo"]]

                name_csv  =  list_joins.select(pl.col("catalogo").unique())
["catalogo"].to_list()
          name_column_real_dataset = list_joins.select(pl.col("nemonico"))
["nemonico"].to_list()

        for column in name_column_real_dataset:
            if len(name_csv) < 1 :
                print("not working")
                continue
            else:
                blob = bucket.blob(f"catalogos/{name_csv[0]}.csv")

                with blob.open("r") as f:
                    file_csv = pl.read_csv(f,truncate_ragged_lines=True)
```

```
                    conjunto_de_datos = conjunto_de_datos.join(file_csv,
left_on=[column], right_on=["clave"], how="inner")

                conjunto_de_datos = conjunto_de_datos.with_columns(
                    pl.col("descripción").alias(column)
                )
                conjunto_de_datos = conjunto_de_datos.drop("descripción")

    return  conjunto_de_datos

########## Main ##########

os.environ["GOOGLE_APPLICATION_CREDENTIALS"]                                =
"lrf-bigdata-08d173e4f9bf.json"

storage_client = storage.Client()

bucket = storage_client.bucket("proyecto_final_divorcios")

########### Main Data ############
blob = bucket.blob("conjunto_de_datos/conjunto_de_datos_ed2023.csv")
with blob.open("r") as f:
    conjunto_de_datos = pl.read_csv(f)

############ Read Dictionary ###########
blob = bucket.blob("diccionario_de_datos/diccionario_datos_ed2023.csv")
with blob.open("r") as f:
    diccionario_de_datos = pl.read_csv(f)

############# Read State ###############
blob = bucket.blob("entidad_municipio_localidad_2022.csv")
with blob.open("r") as f:
        entidades  =  pl.read_csv(f).filter((pl.col("cve_mun")  ==  0)  &
(pl.col("cve_loc") == 0 ))["cve_ent", "nom_loc"]

############# Read all  #############
ruta_carpeta = 'catalogos/'
blobs = bucket.list_blobs(prefix=ruta_carpeta)
lista = []
for blob in blobs:
    nombre_archivo = os.path.basename(blob.name)
    lista.append(nombre_archivo)

######## Replace csv an clean columns  ##########

lista_archivos = [archivo.replace('.csv', '') for archivo in lista]
```

```
lista_columns = ["año_nacimiento", "año_ejecutoria", "dia", "año_sentencia",
"año_registro", "edad", "numero_de_hijos", "duracion_matrimonio"]

for rem in lista_columns:
    lista_archivos.remove(rem)

########## Execute Function And Join to obtain the State  #########

conjunto_de_datos            =            do_joins_to_main(lista_archivos,
conjunto_de_datos).join(entidades,  left_on="ent_mat",  right_on="cve_ent",
how="inner").with_columns(
    pl.col("nom_loc").alias("ent_mat")

).drop("nom_loc")

conjunto_de_datos = conjunto_de_datos.to_pandas()

to_gbq(conjunto_de_datos,
destination_table='lrf-bigdata.divorcios.conjunto_datos',  project_id='lrf-
bigdata', if_exists='replace')
```

2.3 BigQuery

Our table has 64 columns with different information about the divorce, the most relevant columns are the year of the marriage, the date of the divorce, the years that lives the marriage, the genre, the age, the cause of the divorce, who start the divorce. It contain half millions rows in three years.

Then we use the extension of Jupyter with BigQuery to analize this dataset with spark.

2.3.1 Schema

tipo_div,ent_regis,mun_regis,loc_regis,tloc_regis,ent_mat,mun_mat, .

local_mat,tloc_mat,dia_mat,mes_mat,anio_mat,dia_reg,mes_reg,

anio_reg,dia_sen,mes_sen,anio_sen,dia_eje,mes_eje,anio_eje,

ini_juic,favor,causa,hijos,hij_men,custodia,cus_hij,pat_pot,pat_hij,

pension,pen_hij,naci_div1,edad_div1,nacim_div1,eciv_adiv1,ent_rhdiv1,

mun_rhdiv1,loc_rhdiv1,tloc_div1,escol_div1,con_acdiv1,dedic_div1,postr_div1,

sexo_div1,naci_div2,edad_div2,nacim_div2,eciv_adiv2,ent_rhdiv2,mun_rhdiv2,

loc_rhdiv2,tloc_div2,escol_div2,con_acdiv2,dedic_div2,postr_div2,

sexo_div2,dura_soc,dura_leg,edad_mdiv1,edad_mdiv2,t_dvante,dis_reoax

3 Results

3.1 Distribution of Divorces in Mexico States

```
SELECT ent_mat, COUNT(*) FROM table GROUP BY ent_mat SORT BY COUNT(*) DESC")
```

	ent_mat	count(1)
0	México	50752
1	Nuevo León	47685
2	Ciudad de México	35591
3	Guanajuato	31307
4	Sinaloa	26025
5	Coahuila de Zaragoza	25980
6	Chihuahua	25271
7	Jalisco	24480
8	Tamaulipas	24099
9	Michoacán de Ocampo	20635
10	Veracruz de Ignacio de la Llave	18541

Figure 1: Top 10 States

3.2 Divorces By Year

```
    SELECT anio_eje, COUNT(*) as cantidad FROM table GROUP BY anio_eje ORDER BY
cantidad DESC
```

	anio_eje	cantidad
0	2022	164314
1	2023	162490
2	2021	142370
3	2020	62862

Figure 2: Divorces Per Year

3.3 Causes of Divorce

```
SELECT causa,
       COUNT(*) as cantidad,
       ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER (), 2) as porcentaje
FROM table
GROUP BY causa
ORDER BY cantidad DESC
LIMIT 10;
```

	causa	cantidad	porcentaje
0	Sin causa (incausado)	350774	65.93
1	Mutuo consentimiento	173916	32.69
2	Separación del hogar conyugal por más de 1 año	2943	0.55
3	La separación por 2 años o más independienteme...	1237	0.23
4	Incompatibilidad de caracteres	1086	0.20
5	Abandono de hogar por más de 3 o 6 meses	824	0.15
6	No especificada	403	0.08
7	Si un cónyuge solicitó el divorcio por causa i...	244	0.05
8	Sevicia	184	0.03
9	Adulterio o infidelidad sexual	151	0.03

Figure 3: Causes of the Divorce

3.4 Schollarity of the people of the divorce

```
SELECT escol_div1,
       ROUND(AVG(dura_soc), 2) as duracion_promedio_years,
       COUNT(*) as cantidad_divorcios
FROM table
GROUP BY escol_div1
ORDER BY duracion_promedio_years DESC;
```

	escol_div1	duracion_promedio_years	cantidad_divorcios
0	4 a 5 años de primaria	31.81	2723
1	1 a 3 años de primaria	30.74	3427
2	Primaria completa	28.81	31428
3	No especificada	26.39	143103
4	Otra	26.29	4772
5	Sin escolaridad	25.99	5507
6	Secundaria o equivalente	22.18	105304
7	Preparatoria o equivalente	20.46	111007
8	Superior	19.73	97337
9	Carrera técnica	17.83	27428

Figure 4: Schollarity

3.5 Age

```
SELECT
    'Primer divorciante' as divorciante,
    ROUND(AVG(CASE WHEN sexo_div1 = 'Hombre' AND edad_div1 < 150 THEN edad_div1
END), 2) as edad_promedio_hombres,
    ROUND(AVG(CASE WHEN sexo_div1 = 'Mujer' AND edad_div1 < 150 THEN edad_div1
END), 2) as edad_promedio_mujeres
FROM table
```

```
UNION ALL
SELECT
    'Segundo divorciante',
    ROUND(AVG(CASE WHEN sexo_div2 = 'Hombre' AND edad_div2 < 150 THEN edad_div2
END), 2),
    ROUND(AVG(CASE WHEN sexo_div2 = 'Mujer' AND edad_div2 < 150 THEN edad_div2
END), 2)
FROM table;


sql_result.toPandas()
```

	divorciante	edad_promedio_hombres	edad_promedio_mujeres
0	Primer divorciante	43.17	39.64
1	Segundo divorciante	41.85	40.55

Figure 5: Age and Sex

Also this metric in general is almost the same it doesn't change significantly

3.6 Children's by Divorce

```
SELECT
    CASE
        WHEN hijos = 0 THEN 'Sin hijos'
        WHEN hijos = 1 THEN '1 hijo'
        WHEN hijos = 2 THEN '2 hijos'
        WHEN hijos > 2 THEN 'Más de 2 hijos'
    END as categoria_hijos,
    COUNT(*) as cantidad_divorcios,
    ROUND(AVG(dura_soc), 2) as duracion_promedio_matrimonio
FROM table
GROUP BY categoria_hijos
ORDER BY cantidad_divorcios DESC;
```

	categoria_hijos	cantidad_divorcios	duracion_promedio_matrimonio
0	Más de 2 hijos	158478	38.82
1	2 hijos	136639	17.64
2	1 hijo	118769	13.43
3	Sin hijos	118150	16.94

Figure 6: Childrens by Divorce

It's interesting that when you have more children's the marriage has a longest duration.

3.7 Economic

```
SELECT
    d.dedic_div1,
    d.dedic_div2,
    COUNT(*) as cantidad_divorcios,
    ROUND(AVG(dura_soc), 2) as duracion_promedio_matrimonio
FROM table d
GROUP BY d.dedic_div1, d.dedic_div2
ORDER BY cantidad_divorcios DESC
LIMIT 7;
```

	dedic_div1	dedic_div2	cantidad_divorcios	duracion_promedio_matrimonio
0	Trabaja	Trabaja	238695	20.71
1	No especificada	No especificada	101012	29.26
2	Trabaja	Al hogar	56624	26.31
3	Trabaja	No especificada	43931	19.00
4	Al hogar	Trabaja	38554	18.20
5	No especificada	Trabaja	9862	17.20
6	Al hogar	No especificada	8536	20.44

Figure 7: Profesion of the Marriage

It's interesting that when the two work's it's when is more commun to get divorce. Also is interesting that there are many people that doesn't like to tell his economical situation

3.8 Sex and Economic

```
WITH empleo_sexo AS (
    SELECT
        'Primer divorciante' as divorciante,
        sexo_div1 as sexo,
        dedic_div1 as empleo,
        COUNT(*) as cantidad
    FROM table
    GROUP BY sexo_div1, dedic_div1

    UNION ALL

    SELECT
        'Segundo divorciante' as divorciante,
        sexo_div2 as sexo,
        dedic_div2 as empleo,
        COUNT(*) as cantidad
    FROM table
    GROUP BY sexo_div2, dedic_div2
),
totales AS (
    SELECT
        sexo,
        SUM(cantidad) as total_sexo
```

```
    FROM empleo_sexo
    WHERE sexo IN ('Hombre', 'Mujer')
        AND empleo IS NOT NULL
    GROUP BY sexo
)
SELECT
    e.sexo,
    e.empleo,
    SUM(e.cantidad) as total,
    ROUND(100.0 * SUM(e.cantidad) / MAX(t.total_sexo), 2) as porcentaje
FROM empleo_sexo e
JOIN totales t ON e.sexo = t.sexo
WHERE e.sexo IN ('Hombre', 'Mujer')
    AND e.empleo IS NOT NULL
GROUP BY e.sexo, e.empleo
ORDER BY e.sexo, porcentaje DESC;
```

It's important to notice that there were more man working in the divorce than women but it's not so hight in other combinations the percentage of change is to low so it's irrelevant.

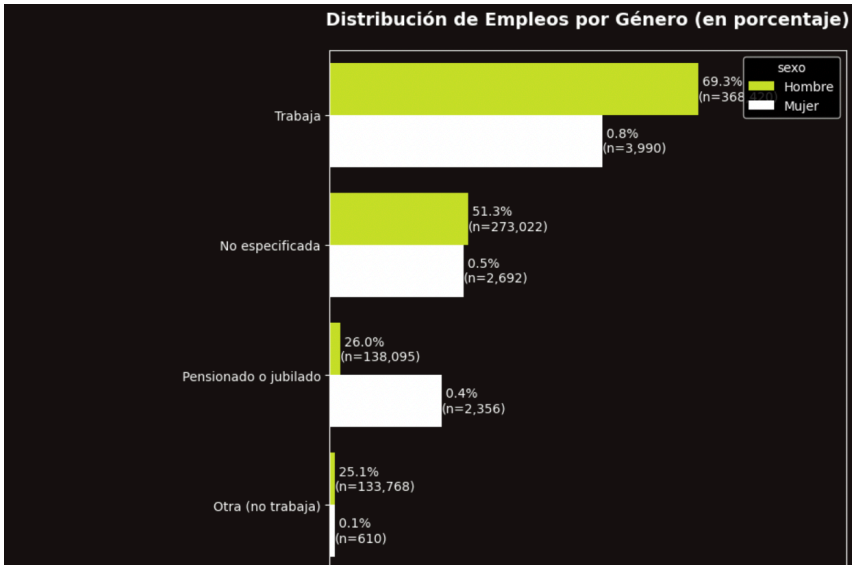


Figure 8: Distribution Profesion/Sex

3.9 Age of DIfference

```
SELECT
    CASE
        WHEN ABS(edad_div1 - edad_div2) = 0 THEN 'Misma edad'
        WHEN ABS(edad_div1 - edad_div2) BETWEEN 1 AND 5 THEN '1-5 años'
        WHEN ABS(edad_div1 - edad_div2) BETWEEN 6 AND 10 THEN '6-10 años'
        WHEN ABS(edad_div1 - edad_div2) > 10 THEN 'Más de 10 años'
    END as diferencia_edad,
    COUNT(*) as cantidad,
    ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER (), 2) as porcentaje,
    ROUND(AVG(dura_soc), 2) as duracion_promedio_matrimonio
FROM table
```

```
WHERE edad_div1 < 999 AND edad_div2 < 999
GROUP BY
    CASE
        WHEN ABS(edad_div1 - edad_div2) = 0 THEN 'Misma edad'
        WHEN ABS(edad_div1 - edad_div2) BETWEEN 1 AND 5 THEN '1-5 años'
        WHEN ABS(edad_div1 - edad_div2) BETWEEN 6 AND 10 THEN '6-10 años'
        WHEN ABS(edad_div1 - edad_div2) > 10 THEN 'Más de 10 años'
    END
ORDER BY
    CASE diferencia_edad
        WHEN 'Misma edad' THEN 1
        WHEN '1-5 años' THEN 2
        WHEN '6-10 años' THEN 3
        WHEN 'Más de 10 años' THEN 4
    END;
```

	diferencia_edad	cantidad	porcentaje	duracion_promedio_matrimonio
0	Misma edad	64355	13.80	21.72
1	1-5 años	287983	61.75	23.24
2	6-10 años	79299	17.00	23.92
3	Más de 10 años	34741	7.45	24.93

3.10

```
SELECT
    CASE
        WHEN edad_mdiv1 < 20 THEN 'Menos de 20'
        WHEN edad_mdiv1 BETWEEN 20 AND 24 THEN '20-24'
        WHEN edad_mdiv1 BETWEEN 25 AND 29 THEN '25-29'
        WHEN edad_mdiv1 BETWEEN 30 AND 34 THEN '30-34'
        WHEN edad_mdiv1 BETWEEN 35 AND 39 THEN '35-39'
        WHEN edad_mdiv1 BETWEEN 40 AND 44 THEN '40-44'
        WHEN edad_mdiv1 BETWEEN 45 AND 49 THEN '45-49'
        WHEN edad_mdiv1 BETWEEN 50 AND 54 THEN '50-54'
        WHEN edad_mdiv1 BETWEEN 55 AND 59 THEN '55-59'
        WHEN edad_mdiv1 >= 60 THEN '60 o más'
    END as rango_edad,
    COUNT(*) as cantidad,
    ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER (), 2) as porcentaje,
    ROUND(AVG(dura_soc), 2) as duracion_promedio_matrimonio
FROM table
WHERE edad_mdiv1 < 999  -- Excluir valores no válidos
GROUP BY
    CASE
```

```
        WHEN edad_mdiv1 < 20 THEN 'Menos de 20'
        WHEN edad_mdiv1 BETWEEN 20 AND 24 THEN '20-24'
        WHEN edad_mdiv1 BETWEEN 25 AND 29 THEN '25-29'
        WHEN edad_mdiv1 BETWEEN 30 AND 34 THEN '30-34'
        WHEN edad_mdiv1 BETWEEN 35 AND 39 THEN '35-39'
        WHEN edad_mdiv1 BETWEEN 40 AND 44 THEN '40-44'
        WHEN edad_mdiv1 BETWEEN 45 AND 49 THEN '45-49'
        WHEN edad_mdiv1 BETWEEN 50 AND 54 THEN '50-54'
        WHEN edad_mdiv1 BETWEEN 55 AND 59 THEN '55-59'
        WHEN edad_mdiv1 >= 60 THEN '60 o más'
    END
ORDER BY
    CASE rango_edad
        WHEN 'Menos de 20' THEN 1
        WHEN '20-24' THEN 2
        WHEN '25-29' THEN 3
        WHEN '30-34' THEN 4
        WHEN '35-39' THEN 5
        WHEN '40-44' THEN 6
        WHEN '45-49' THEN 7
        WHEN '50-54' THEN 8
        WHEN '55-59' THEN 9
        WHEN '60 o más' THEN 10
    END;
```

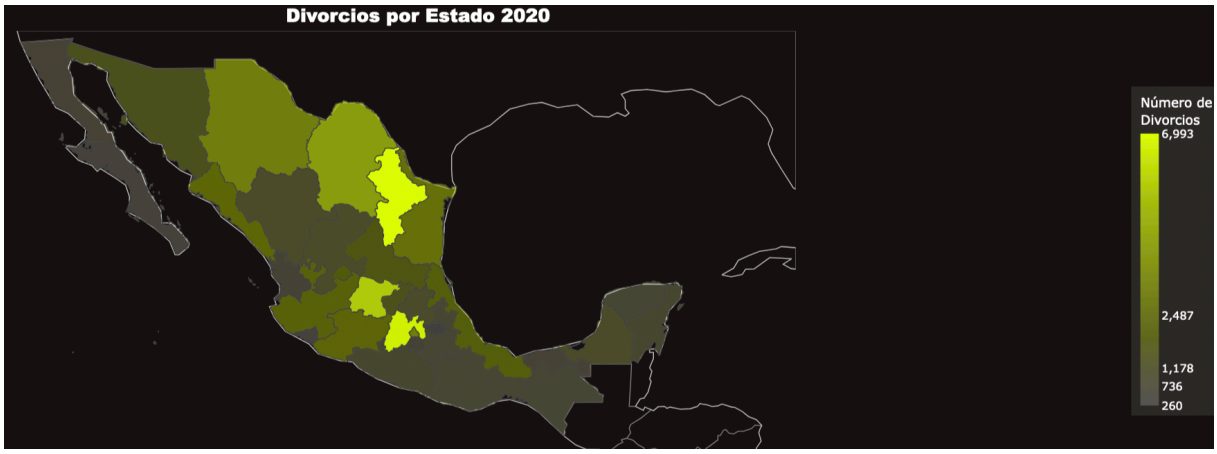
	rango_edad	cantidad	porcentaje	duracion_promedio_matrimonio
0	Menos de 20	92030	17.30	22.13
1	20-24	163039	30.64	20.43
2	25-29	103785	19.51	18.55
3	30-34	46498	8.74	18.24
4	35-39	21092	3.96	18.90
5	40-44	11263	2.12	20.47
6	45-49	6570	1.23	20.69
7	50-54	3878	0.73	22.17
8	55-59	2341	0.44	23.98
9	60 o más	81540	15.33	38.15

Figure 9: Age/Duration *Range*

3.11 Map

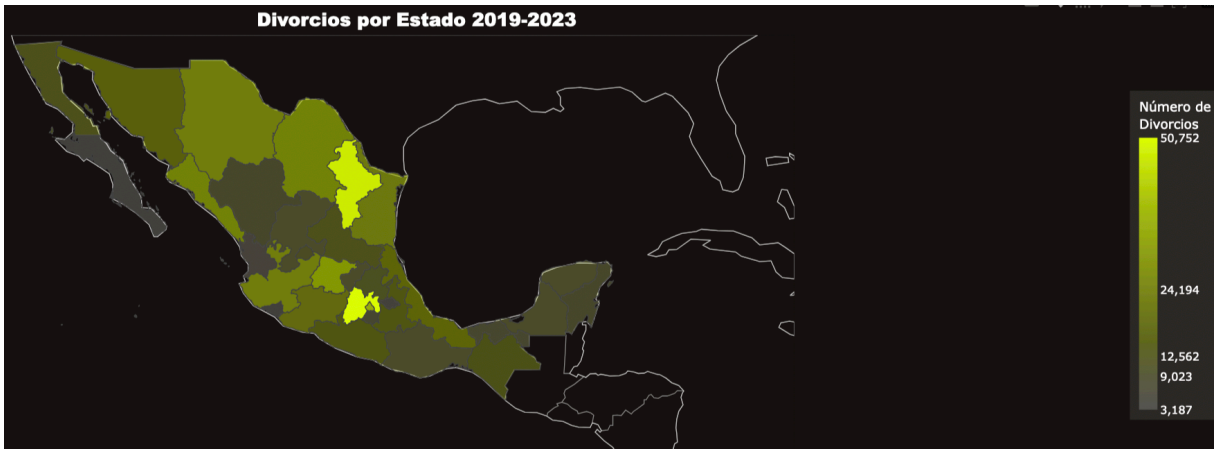
It's interesting depend of the year the state changes

3.11.1 2020



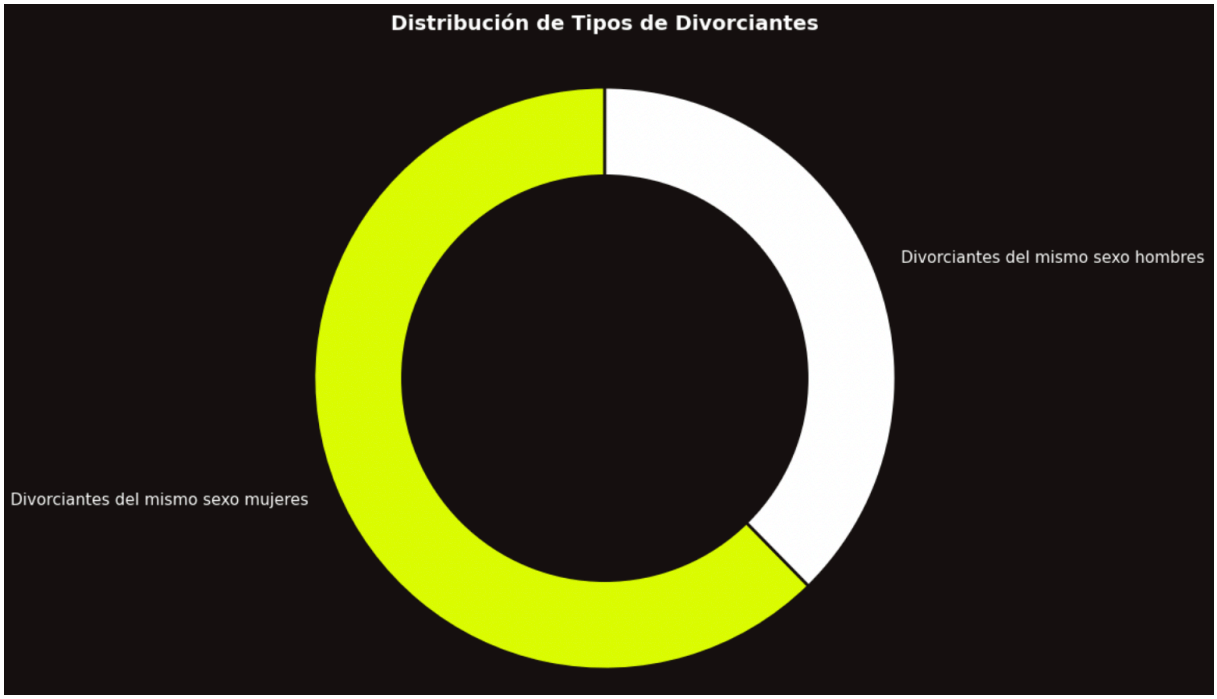
This is helpfull becasuse this tell us in which

3.11.2 2020- 2023



3.11.3 2023

3.12 LGBT


Distribución de Tipos de Divorciantes
Divorciantes del mismo sexo hombres
Divorciantes del mismo sexo mujeres