Autor : Daniel Ivan Medina Barreras

Datos (https://github.com/noelDz/COVID19_Mx_opendata)

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import scipy as sc


        import plotly
        import plotly.graph_objs as go
        from plotly.offline import init_notebook_mode, iplot
        import plotly.offline as py
        import plotly.figure_factory as ff

        import seaborn as sb

        import pandas_profiling
```

```
In [2]: init_notebook_mode(connected=True)
```

```
In [3]: %matplotlib inline
```

Lectura de la fuente de información

```
In [4]: df = pd.read_csv('./data_COVID19Mx/Confirmados/COVID19Mx_confirmados28-03-20.cs
        v',index_col=0)
```

Los metadatos de la fuente de información

```
In [5]: df.columns
```

```
Out[5]: Index(['index', 'NumCaso', 'Estado', 'Sexo', 'Edad',
               'Fecha_de_Inicio_de_Síntomas',
               'IdentificacionRT-PCR_COVID19(tiemporeal)', 'Procedencia',
               'Llegada_México'],
              dtype='object')
```

In [6]:
```python
df.head()
```

Out[6]:

	index	NumCaso	Estado	Sexo	Edad	Fecha_de_Inicio_de_Síntomas	IdentificacionRT-PCR_COVID19(tiemporeal)	Proceder
0	313	314	MÉXICO	M	44	2020-02-19	confirmado	
1	5	6	MÉXICO	M	71	2020-02-21	confirmado	
2	0	1	CDMX	M	35	2020-02-22	confirmado	
3	1	2	SINALOA	M	41	2020-02-22	confirmado	
4	2	3	CDMX	M	59	2020-02-23	confirmado	

In [7]:
```python
df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 848 entries, 0 to 847
Data columns (total 9 columns):
index                                    848 non-null int64
NumCaso                                  848 non-null int64
Estado                                   848 non-null object
Sexo                                     848 non-null object
Edad                                     848 non-null int64
Fecha_de_Inicio_de_Síntomas              848 non-null object
IdentificacionRT-PCR_COVID19(tiemporeal) 848 non-null object
Procedencia                              848 non-null object
Llegada_México                           567 non-null object
dtypes: int64(3), object(6)
memory usage: 66.2+ KB
```

In [8]:
```python
DataFrameAnalisisEstadistico=df.describe()
```

In [9]:
```python
DataFrameAnalisisEstadistico
```

Out[9]:

	index	NumCaso	Edad
count	848.000000	848.000000	848.000000
mean	423.500000	424.500000	42.443396
std	244.940809	244.940809	15.583014
min	0.000000	1.000000	0.000000
25%	211.750000	212.750000	30.000000
50%	423.500000	424.500000	41.000000
75%	635.250000	636.250000	53.000000
max	847.000000	848.000000	88.000000

In [10]:
```python
d=ff.create_table(DataFrameAnalisisEstadistico)
```

In [11]: `iplot(d)`

index	NumCaso	E
848.0	848.0	8
423.5	424.5	4
244.94080917642123	244.94080917642123	1
0.0	1.0	0
211.75	212.75	3

In [12]:
```python
pandas_profiling.ProfileReport(df)
```

```
/home/saxsa/anaconda3/lib/python3.7/site-packages/pandas_profiling/describe.py:
392: FutureWarning:

The join_axes-keyword is deprecated. Use .reindex or .reindex_like on the resul
t to achieve the same functionality.
```

`Out[12]:`

Overview

Dataset info

Number of variables	9
Number of observations	848
Total Missing (%)	3.7%
Total size in memory	66.2 KiB
Average record size in memory	80.0 B

Variables types

Numeric	2
Categorical	5
Boolean	0
Date	0
Text (Unique)	0
Rejected	2
Unsupported	0

Warnings

- `NumCaso` is highly correlated with `index` (ρ = 1) `Rejected`
- `IdentificacionRT-PCR_COVID19(tiemporeal)` has constant value confirmado `Rejected`
- `Llegada_México` has 281 / 33.1% missing values `Missing`

Variables

index
Numeric

Distinct count	848		**Mean**	423.5
Unique (%)	100.0%		**Minimum**	0
Missing (%)	0.0%		**Maximum**	847
Missing (n)	0		**Zeros (%)**	0.1%
Infinite (%)	0.0%			
Infinite (n)	0			

To

~~NumCaso~~
Highly correlated

This variable is highly correlated with `index` *and should be ignored for analysis*

Correlation	1

In [13]:
```python
AnalisisEstadisticoPromedioXGenero=df.groupby('Sexo').mean()
```

In [14]:
```python
AnalisisEstadisticoPromedioXGenero
```

Out[14]:

Sexo	index	NumCaso	Edad
F	446.986150	447.986150	41.958449
M	406.090349	407.090349	42.802875

In [15]:
```python
ColoresComponentesTabla=[[0,'#4d004c'],[.5,'#f2e5ff'],[1,'#ffffff']]
```

In [16]:
```python
ColoresLetra=['#000000']
```

In [17]:
```python
dz=ff.create_table(AnalisisEstadisticoPromedioXGenero,index=True,colorscale=ColoresComponentesTabla,font_colors=ColoresLetra)
iplot(dz)
```

	index	NumCaso
F	446.9861495844875	447.9861495844875

Despliegue gráfico de resultados

In [18]:
```python
col="Sexo"
AnalisisFrecuenciaXGenero=df[col].value_counts().reset_index()
```

In [19]:
```python
AnalisisFrecuenciaXGenero
```

Out[19]:

	index	Sexo
0	M	487
1	F	361

In [20]:
```python
AnalisisFrecuenciaXGenero=AnalisisFrecuenciaXGenero.rename(columns={col:"count","index":col})
```
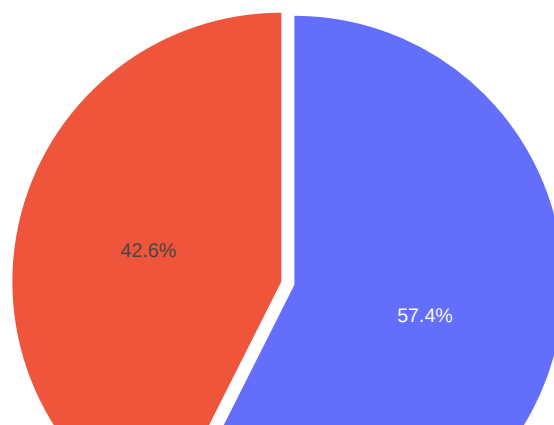
In [21]:
```python
AnalisisFrecuenciaXGenero
```

Out[21]:

	Sexo	count
0	M	487
1	F	361

Gráfica distribución registros X género

In [22]:
```
trace=go.Pie(labels=AnalisisFrecuenciaXGenero[col],values=AnalisisFrecuenciaXGe
nero['count'],pull=[0.05,0])
layout={'title':'Género(Hombre, Mujer)'}
fig=go.Figure(data=[trace],layout=layout)
iplot(fig)
```
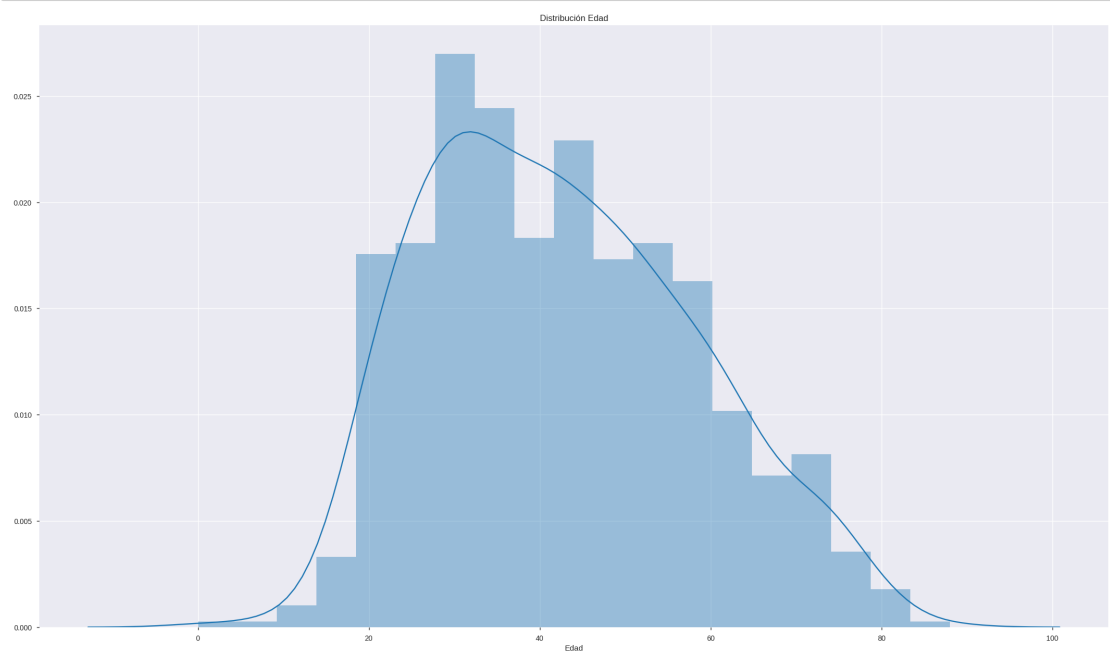
Género(Hombre, Mujer)

42.6%

57.4%

Distribución Edad

In [23]:
```python
plt.figure(figsize=(90,15))
plt.subplot(1,3,1)
sns.distplot(df["Edad"])
plt.title('Distribución Edad')

plt.show()
```


In [24]:
```python
x=df
```

In [25]:
```python
d1=x[x['Sexo']=='M']
```

In [26]:
```python
d1
```

Out[26]:

	index	NumCaso	Estado	Sexo	Edad	Fecha_de_Inicio_de_Síntomas	IdentificacionRT-PCR_COVID19(tiemporeal)
0	313	314	MÉXICO	M	44	2020-02-19	confirmado
1	5	6	MÉXICO	M	71	2020-02-21	confirmado
2	0	1	CDMX	M	35	2020-02-22	confirmado
3	1	2	SINALOA	M	41	2020-02-22	confirmado
4	2	3	CDMX	M	59	2020-02-23	confirmado
...	...	...	...	...	...	...	...
841	741	742	SINALOA	M	43	2020-03-25	confirmado
842	561	562	QUINTANA_ROO	M	75	2020-03-25	confirmado
843	805	806	GUANAJUATO	M	30	2020-03-25	confirmado
845	837	838	TABASCO	M	58	2020-03-26	confirmado
846	845	846	MÉXICO	M	35	2020-03-26	confirmado

487 rows × 9 columns

```
In [27]: d2=x[x['Sexo']=='F']
```

```
In [28]: d2
```

Out[28]:

	index	NumCaso	Estado	Sexo	Edad	Fecha_de_Inicio_de_Síntomas	IdentificacionRT PCR_COVID19(tiemporea
5	4	5	CHIAPAS	F	18	2020-02-25	confirmad
6	3	4	COAHUILA	F	20	2020-02-27	confirmad
8	686	687	JALISCO	F	50	2020-03-01	confirmad
9	32	33	JALISCO	F	64	2020-03-01	confirmad
10	25	26	QUINTANA_ROO	F	71	2020-03-01	confirmad
...	...	...	...	...	...	...	.
838	716	717	AGUASCALIENTES	F	5	2020-03-25	confirmad
839	718	719	AGUASCALIENTES	F	37	2020-03-25	confirmad
840	824	825	CDMX	F	35	2020-03-25	confirmad
844	835	836	MÉXICO	F	73	2020-03-26	confirmad
847	847	848	MÉXICO	F	61	2020-03-26	confirmad

361 rows × 9 columns

In [29]:
```python
x=df
col='Edad'
v1=x[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x: 100 *x/sum(v1['count']))
v1=v1.sort_values(col)
trace1=go.Bar(x=v1[col],y=v1["count"],name="0",marker=dict(color="rgb(63, 72, 2
04)"))
y=[trace1]
layout={'title':"Distribución X Edad ",'xaxis':{'title':"Edad"},'yaxis':{'title
':"Frecuencia"}}
fig=go.Figure(data=y,layout=layout)
fig.layout.template='presentation'
iplot(fig)
```
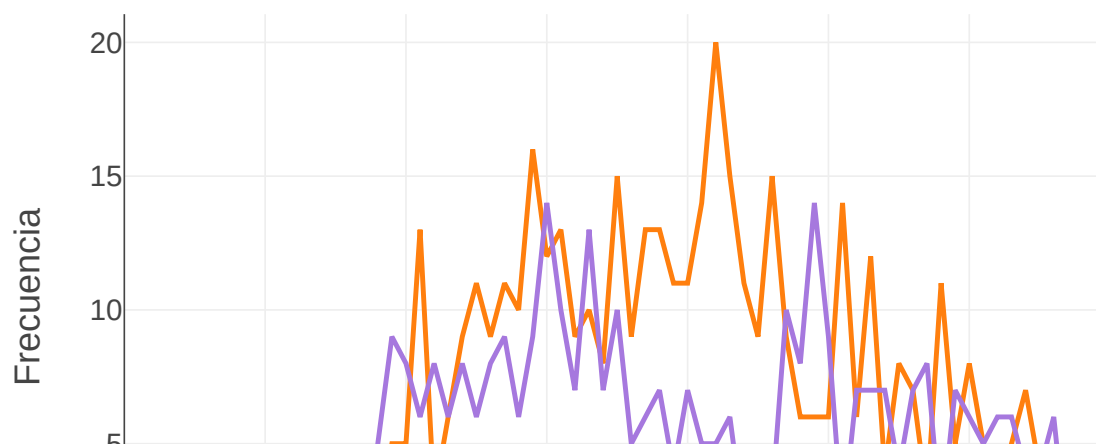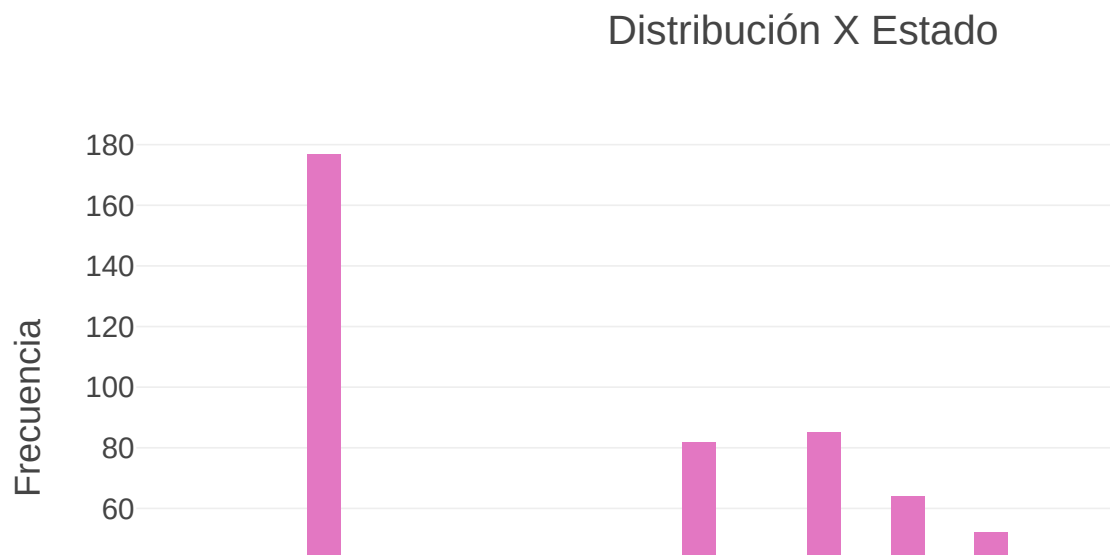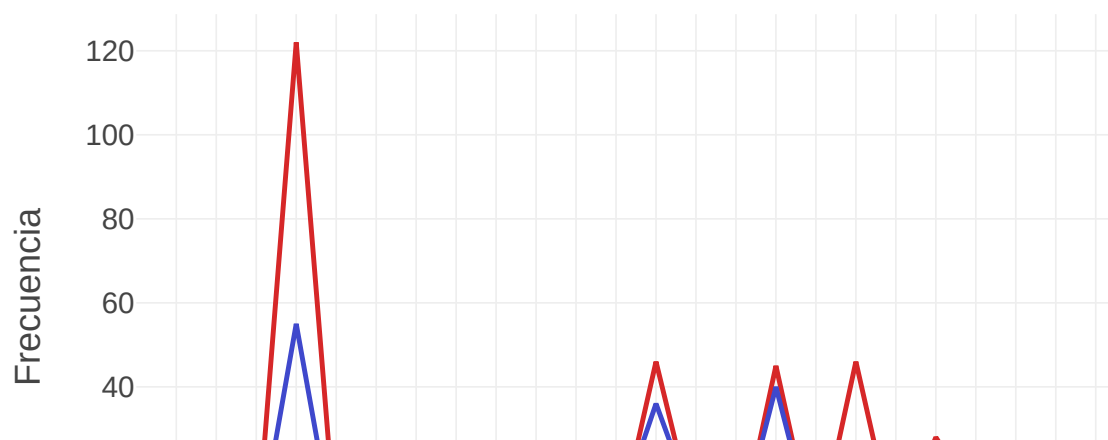

Distribución X Edad
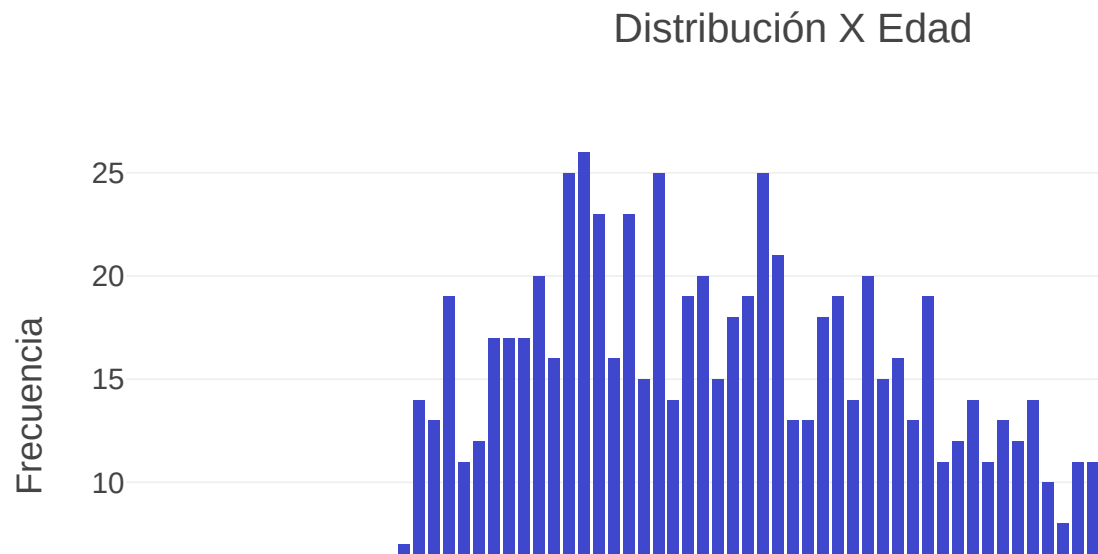
In [30]:
```python
col='Edad'

v1=d1[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x:100*x/sum(v1['count']))
v1=v1.sort_values(col)
v2=d2[col].value_counts().reset_index()
v2=v2.rename(columns={col:'count','index':col})
v2['percent']=v2['count'].apply(lambda x:100*x/sum(v2['count']))
v2=v2.sort_values(col)
trace1=go.Scatter(x = v1[col], y = v1["count"],name="Hombre",marker=dict(colo
r="#ff7f0e"))
trace2= go.Scatter(x = v2[col],y = v2["count"], name = "Mujer", marker = dict(c
olor='#a678de'))
y = [trace1, trace2]
layout={'title':"Comparación X Edad [[ Hombre vs Mujer ]] ",'xaxis':{'title':"E
dad"},'yaxis':{'title':"Frecuencia"}}
fig=go.Figure(data = y, layout = layout)
fig.layout.template='presentation'
iplot(fig)
```

In [31]:
```python
x=df
col='Estado'
v1=x[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x: 100 *x/sum(v1['count']))
v1=v1.sort_values(col)
trace1=go.Bar(x=v1[col],y=v1["count"],name="0",marker=dict(color="#e377c2"))
y=[trace1]
layout={'title':"Distribución X Estado ",'xaxis':{'title':""},'yaxis':{'title
':"Frecuencia"}}
fig=go.Figure(data=y,layout=layout)
fig.layout.template='presentation'
iplot(fig)
```
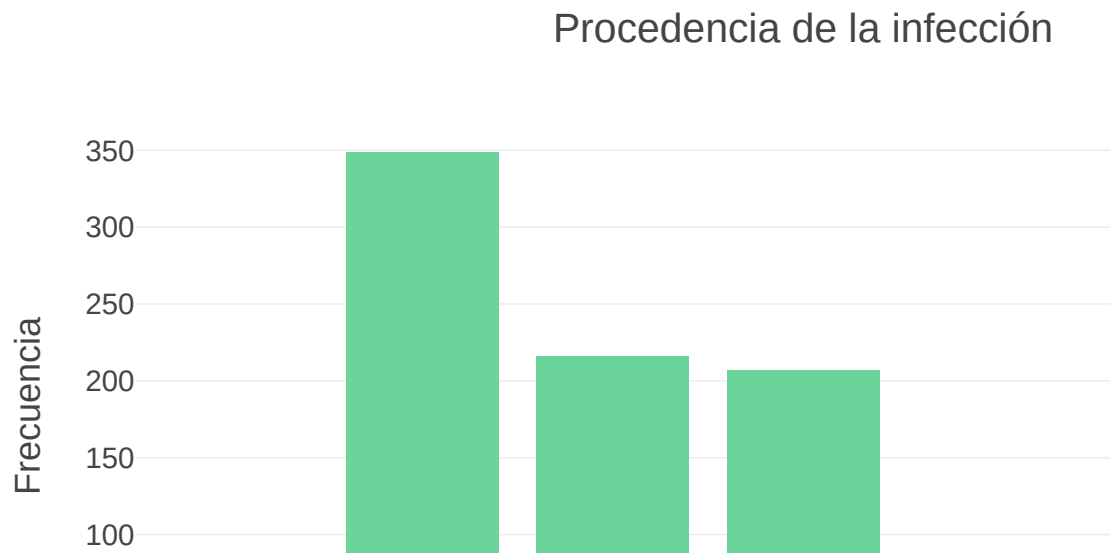

Distribución X Estado

In [32]:
```python
col='Estado'

v1=d1[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x:100*x/sum(v1['count']))
v1=v1.sort_values(col)
v2=d2[col].value_counts().reset_index()
v2=v2.rename(columns={col:'count','index':col})
v2['percent']=v2['count'].apply(lambda x:100*x/sum(v2['count']))
v2=v2.sort_values(col)
trace1=go.Scatter(x = v1[col], y = v1["count"],name="Hombre",marker=dict(colo
r="#d62728"))
trace2= go.Scatter(x = v2[col],y = v2["count"], name = "Mujer", marker = dict(c
olor='rgb(63, 72, 204)'))
y = [trace1, trace2]
layout={'title':"Comparación X Estado [[ Hombre vs Mujer ]] ",'xaxis':{'title
':"Estado"},'yaxis':{'title':"Frecuencia"}}
fig=go.Figure(data = y, layout = layout)
fig.layout.template='presentation'
iplot(fig)
```

In [33]:
```python
x = df
col='Procedencia'
v1=x[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x: 100 *x/sum(v1['count']))
v1=v1.sort_values(col)
trace1=go.Bar(x=v1[col],y=v1["count"],name="0",marker=dict(color="#6ad49b"))
y=[trace1]
layout={'title':"Procedencia de la infección ",'xaxis':{'title':"Procedenci
a"},'yaxis':{'title':"Frecuencia"}}
fig=go.Figure(data=y,layout=layout)
fig.layout.template='presentation'
iplot(fig)
```

In [34]:
```python
col='Procedencia'

v1=d1[col].value_counts().reset_index()
v1=v1.rename(columns={col:'count','index':col})
v1['percent']=v1['count'].apply(lambda x:100*x/sum(v1['count']))
v1=v1.sort_values(col)
v2=d2[col].value_counts().reset_index()
v2=v2.rename(columns={col:'count','index':col})
v2['percent']=v2['count'].apply(lambda x:100*x/sum(v2['count']))
v2=v2.sort_values(col)
trace1=go.Scatter(x = v1[col], y = v1["count"],name="Hombre",marker=dict(colo
r="#17becf"))
trace2= go.Scatter(x = v2[col],y = v2["count"], name = "Mujer", marker = dict(c
olor='#a678de'))
y = [trace1, trace2]
layout={'title':"Comparación X Procedencia [[ Hombre vs Mujer ]] ",'xaxis':{'ti
tle':"Procedencia"},'yaxis':{'title':"Frecuencia"}}
fig=go.Figure(data = y, layout = layout)
fig.layout.template='presentation'
iplot(fig)
```


Comparación X Procedencia [[Hombre vs
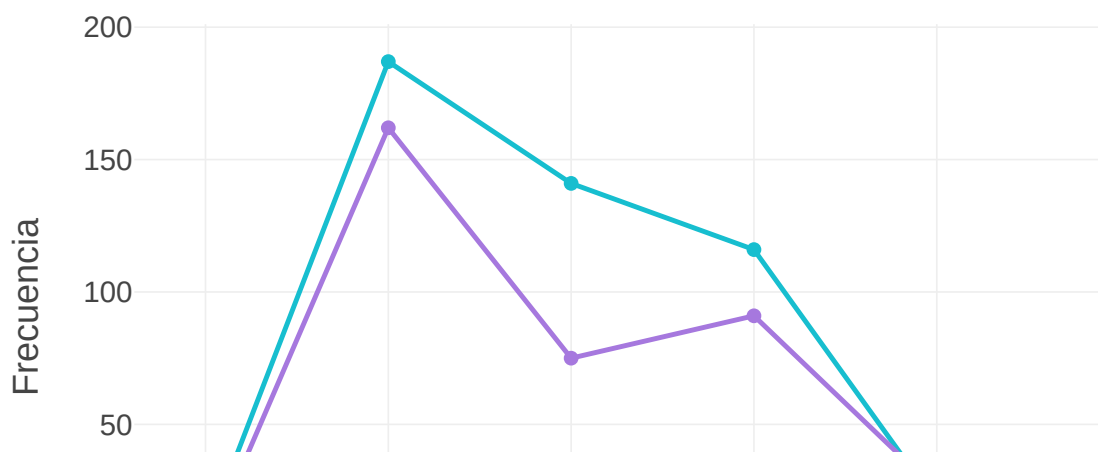
Serie de tiempo: Casos por fecha

Casos de COVID19 de hombres en México por día

In [37]:
```python
casos_fecha_M = pd.DataFrame(d1['IdentificacionRT-PCR_COVID19(tiemporeal)'].groupby(d1['Fecha_de_Inicio_de_Síntomas']).count())
fig = go.Figure()
fig.add_trace(go.Scatter(
                x=casos_fecha_M.index,
                y=casos_fecha_M["IdentificacionRT-PCR_COVID19(tiemporeal)"],
                mode='lines+markers',
                name="Casos Nuevos Hombres",
                line_color='deepskyblue',
                opacity=0.8))

fig.add_trace(go.Scatter(
                x=casos_fecha_M.index,
                y=sc.cumsum(casos_fecha_M["IdentificacionRT-PCR_COVID19(tiemporeal)"]),
                mode='lines+markers',
                name="Casos Totales Hombres",
                line_color='#ff7f0e',
                opacity=0.8))
# Edit the layout
fig.update_layout(title='Casos de COVID19 de hombres en México por día',
                xaxis_title='Fecha',
                yaxis_title='Número de casos')
fig.show()
```

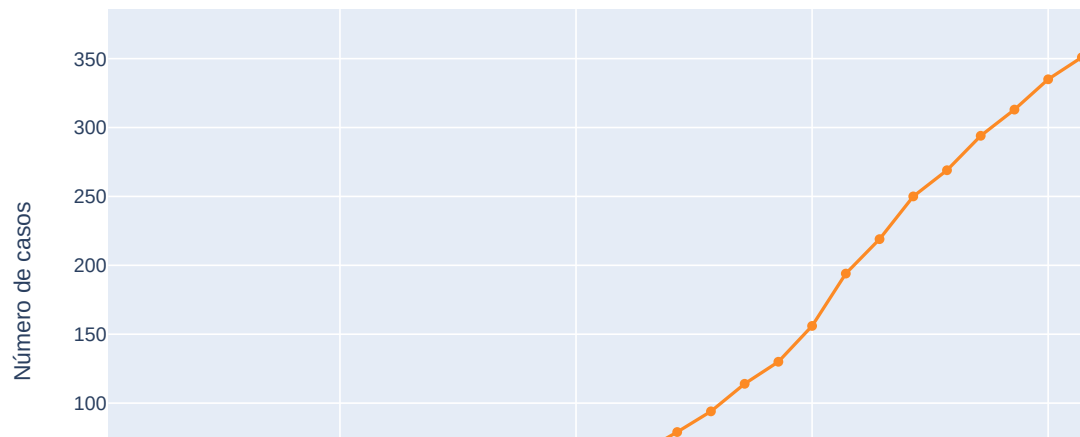Casos de COVID19 de hombres en México por día
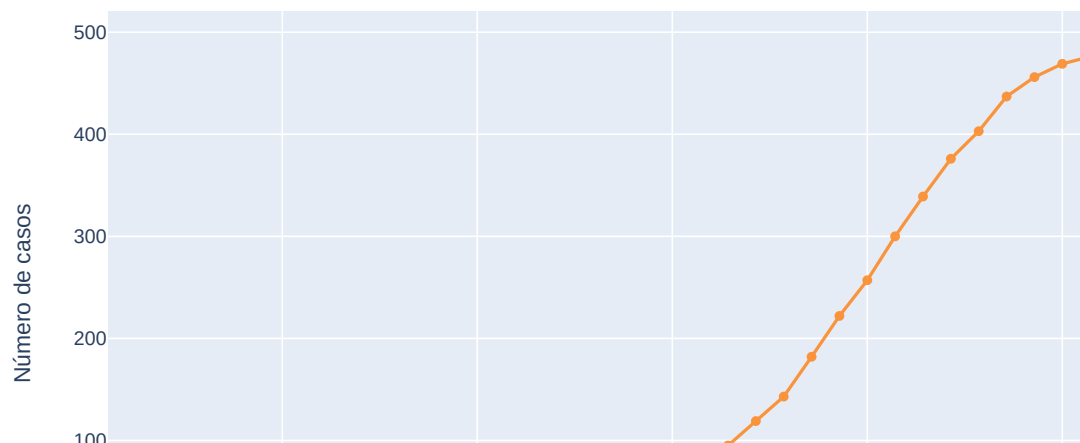
Casos de COVID19 de mujeres en México por día

In [38]:
```python
casos_fecha_F = pd.DataFrame(d2['IdentificacionRT-PCR_COVID19(tiemporeal)'].gro
upby(d2['Fecha_de_Inicio_de_Síntomas']).count())
fig = go.Figure()
fig.add_trace(go.Scatter(
                x=casos_fecha_F.index,
                y=casos_fecha_F["IdentificacionRT-PCR_COVID19(tiemporeal)"],
                mode='lines+markers',
                name="Casos Nuevos Mujeres",
                line_color='#a678de',
                opacity=0.8))

fig.add_trace(go.Scatter(
                x=casos_fecha_F.index,
                y=sc.cumsum(casos_fecha_F["IdentificacionRT-PCR_COVID19(tiempor
eal)"]),
                mode='lines+markers',
                name="Casos Totales Mujeres",
                line_color='#ff7f0e',
                opacity=0.9))
# Edit the layout
fig.update_layout(title='Casos de COVID19 de Mujeres en México por día',
                xaxis_title='Fecha',
                yaxis_title='Número de casos')
fig.show()
```

Casos de COVID19 de Mujeres en México por día

Casos de COVID19 [Hombres vs Mujeres] en México por día

In [39]:
```python
casos_fecha_F = pd.DataFrame(d2['IdentificacionRT-PCR_COVID19(tiemporeal)'].gro
upby(d2['Fecha_de_Inicio_de_Síntomas']).count())
fig = go.Figure()

fig.add_trace(go.Scatter(
                x=casos_fecha_M.index,
                y=casos_fecha_M["IdentificacionRT-PCR_COVID19(tiemporeal)"],
                mode='lines+markers',
                name="Casos Nuevos Hombres'",
                line_color='deepskyblue',
                opacity=0.8))

fig.add_trace(go.Scatter(
                x=casos_fecha_F.index,
                y=casos_fecha_F["IdentificacionRT-PCR_COVID19(tiemporeal)"],
                mode='lines+markers',
                name="Casos Nuevos Mujeres",
                line_color='#e377c2',
                opacity=0.8))
# Edit the layout
fig.update_layout(title='Casos de COVID19 [Hombres vs Mujeres] en México por dí
a',
                xaxis_title='Fecha',
                yaxis_title='Número de casos')
fig.show()
```
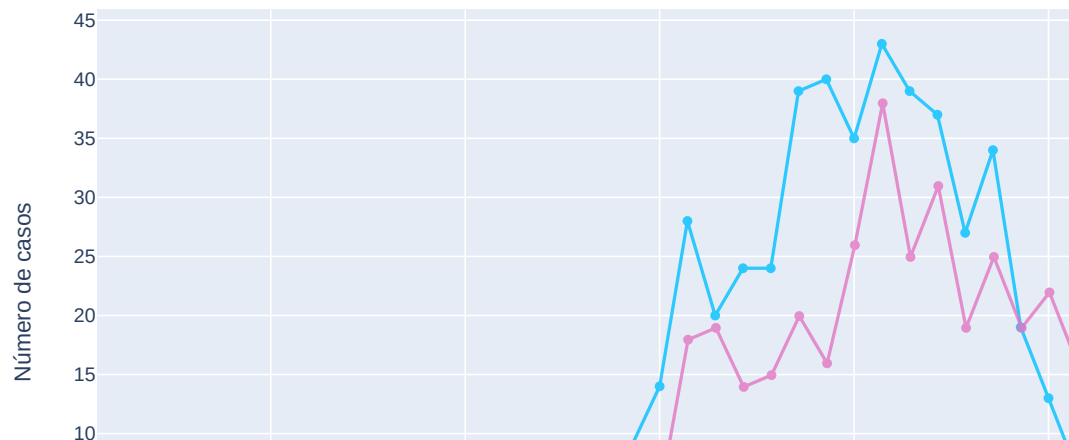
Casos de COVID19 [Hombres vs Mujeres] en México por día

Casos de COVID19 en México por día

In [40]:
```python
casos_fecha = pd.DataFrame(df['IdentificacionRT-PCR_COVID19(tiemporeal)'].group
by(df['Fecha_de_Inicio_de_Síntomas']).count())
fig = go.Figure()
fig.add_trace(go.Scatter(
                x=casos_fecha.index,
                y=casos_fecha["IdentificacionRT-PCR_COVID19(tiemporeal)"],
                mode='lines+markers',
                name="Casos Nuevos",
                line_color='deepskyblue',
                opacity=0.8))

fig.add_trace(go.Scatter(
                x=casos_fecha.index,
                y=sc.cumsum(casos_fecha["IdentificacionRT-PCR_COVID19(tiemporea
l)"]),
                mode='lines+markers',
                name="Casos Totales",
                line_color='#ff7f0e',
                opacity=0.8))
# Edit the layout
fig.update_layout(title='Casos de COVID19 en México por día',
                xaxis_title='Fecha',
                yaxis_title='Número de casos')
fig.show()
```

Casos de COVID19 en México por día