Project Deliverables

[#A] Data Imports

In [2]:
```
!pip install wbdata
!pip install cufflinks
!pip install iso3166

import iso3166 #iso3166.countries.get('country details')
import wbdata
import cufflinks as cf
import pandas as pd
import numpy as np
import plotly
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.offline as py
import plotly.graph_objs as go
cf.go_offline()
```

```
Collecting wbdata
  Using cached wbdata-0.3.0-py3-none-any.whl (14 kB)
Requirement already satisfied: decorator>=4.0 in /opt/conda/lib/python3.9/s
ite-packages (from wbdata) (5.0.9)
Requirement already satisfied: requests>=2.0 in /opt/conda/lib/python3.9/si
te-packages (from wbdata) (2.26.0)
Requirement already satisfied: appdirs<2.0,>=1.4 in /opt/conda/lib/python3.
9/site-packages (from wbdata) (1.4.4)
Requirement already satisfied: tabulate>=0.8.5 in /opt/conda/lib/python3.9/
site-packages (from wbdata) (0.9.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/sit
e-packages (from requests>=2.0->wbdata) (3.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python
3.9/site-packages (from requests>=2.0->wbdata) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/pyth
on3.9/site-packages (from requests>=2.0->wbdata) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/lib/
python3.9/site-packages (from requests>=2.0->wbdata) (2.0.0)
Installing collected packages: wbdata
Successfully installed wbdata-0.3.0
Requirement already satisfied: cufflinks in /opt/conda/lib/python3.9/site-p
ackages (0.17.3)
Requirement already satisfied: ipywidgets>=7.0.0 in /opt/conda/lib/python3.
9/site-packages (from cufflinks) (7.7.2)
Requirement already satisfied: colorlover>=0.2.1 in /opt/conda/lib/python3.
9/site-packages (from cufflinks) (0.3.0)
Requirement already satisfied: setuptools>=34.4.1 in /opt/conda/lib/python
3.9/site-packages (from cufflinks) (58.2.0)
Requirement already satisfied: six>=1.9.0 in /opt/conda/lib/python3.9/site-
packages (from cufflinks) (1.16.0)
Requirement already satisfied: ipython>=5.3.0 in /opt/conda/lib/python3.9/s
ite-packages (from cufflinks) (8.9.0)
Requirement already satisfied: pandas>=0.19.2 in /opt/conda/lib/python3.9/s
ite-packages (from cufflinks) (1.3.5)
Requirement already satisfied: plotly>=4.1.1 in /opt/conda/lib/python3.9/si
te-packages (from cufflinks) (5.2.1)
Requirement already satisfied: numpy>=1.9.2 in /opt/conda/lib/python3.9/sit
e-packages (from cufflinks) (1.21.6)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.9/site
-packages (from ipython>=5.3.0->cufflinks) (4.8.0)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.9/site
-packages (from ipython>=5.3.0->cufflinks) (0.7.5)
Requirement already satisfied: stack-data in /opt/conda/lib/python3.9/site-
packages (from ipython>=5.3.0->cufflinks) (0.6.2)
Requirement already satisfied: matplotlib-inline in /opt/conda/lib/python3.
9/site-packages (from ipython>=5.3.0->cufflinks) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.30 in /opt/conda/
lib/python3.9/site-packages (from ipython>=5.3.0->cufflinks) (3.0.36)
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.9/
site-packages (from ipython>=5.3.0->cufflinks) (2.14.0)
Requirement already satisfied: traitlets>=5 in /opt/conda/lib/python3.9/sit
e-packages (from ipython>=5.3.0->cufflinks) (5.9.0)
Requirement already satisfied: decorator in /opt/conda/lib/python3.9/site-p
ackages (from ipython>=5.3.0->cufflinks) (5.0.9)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.9/site-
packages (from ipython>=5.3.0->cufflinks) (0.18.2)
```

```
Requirement already satisfied: backcall in /opt/conda/lib/python3.9/site-pa
ckages (from ipython>=5.3.0->cufflinks) (0.2.0)
Requirement already satisfied: ipykernel>=4.5.1 in /opt/conda/lib/python3.
9/site-packages (from ipywidgets>=7.0.0->cufflinks) (6.19.4)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /opt/conda/lib/
python3.9/site-packages (from ipywidgets>=7.0.0->cufflinks) (3.6.1)
Requirement already satisfied: jupyterlab-widgets<3,>=1.0.0 in /opt/conda/l
ib/python3.9/site-packages (from ipywidgets>=7.0.0->cufflinks) (1.1.1)
Requirement already satisfied: ipython-genutils~=0.2.0 in /opt/conda/lib/py
thon3.9/site-packages (from ipywidgets>=7.0.0->cufflinks) (0.2.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/lib/pyt
hon3.9/site-packages (from pandas>=0.19.2->cufflinks) (2.8.0)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.9/sit
e-packages (from pandas>=0.19.2->cufflinks) (2021.1)
Requirement already satisfied: tenacity>=6.2.0 in /opt/conda/lib/python3.9/
site-packages (from plotly>=4.1.1->cufflinks) (8.1.0)
Requirement already satisfied: comm>=0.1.1 in /opt/conda/lib/python3.9/site
-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (0.1.2)
Requirement already satisfied: jupyter-client>=6.1.12 in /opt/conda/lib/pyt
hon3.9/site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks)
(7.2.0)
Requirement already satisfied: nest-asyncio in /opt/conda/lib/python3.9/sit
e-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (1.5.6)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-p
ackages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (21.3)
Requirement already satisfied: debugpy>=1.0 in /opt/conda/lib/python3.9/sit
e-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (1.6.6)
Requirement already satisfied: pyzmq>=17 in /opt/conda/lib/python3.9/site-p
ackages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (25.0.0)
Requirement already satisfied: tornado>=6.1 in /opt/conda/lib/python3.9/sit
e-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (6.2)
Requirement already satisfied: psutil in /opt/conda/lib/python3.9/site-pack
ages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->cufflinks) (5.9.4)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/conda/lib/python
3.9/site-packages (from jedi>=0.16->ipython>=5.3.0->cufflinks) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.9/
site-packages (from pexpect>4.3->ipython>=5.3.0->cufflinks) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.9/site-pac
kages (from prompt-toolkit<3.1.0,>=3.0.30->ipython>=5.3.0->cufflinks) (0.2.
6)
Requirement already satisfied: notebook>=4.4.1 in /opt/conda/lib/python3.9/
site-packages (from widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflink
s) (6.4.12)
Requirement already satisfied: executing>=1.2.0 in /opt/conda/lib/python3.
9/site-packages (from stack-data->ipython>=5.3.0->cufflinks) (1.2.0)
Requirement already satisfied: asttokens>=2.1.0 in /opt/conda/lib/python3.
9/site-packages (from stack-data->ipython>=5.3.0->cufflinks) (2.2.1)
Requirement already satisfied: pure-eval in /opt/conda/lib/python3.9/site-p
ackages (from stack-data->ipython>=5.3.0->cufflinks) (0.2.2)
Requirement already satisfied: jupyter-core>=4.9.2 in /opt/conda/lib/python
3.9/site-packages (from jupyter-client>=6.1.12->ipykernel>=4.5.1->ipywidget
s>=7.0.0->cufflinks) (5.2.0)
Requirement already satisfied: entrypoints in /opt/conda/lib/python3.9/site
-packages (from jupyter-client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7.0.0
->cufflinks) (0.4)
Requirement already satisfied: prometheus-client in /opt/conda/lib/python3.
```

```
9/site-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidget
s>=7.0.0->cufflinks) (0.16.0)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.9/site-pack
ages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->c
ufflinks) (3.1.2)
Requirement already satisfied: nbformat in /opt/conda/lib/python3.9/site-pa
ckages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0-
>cufflinks) (5.7.3)
Requirement already satisfied: Send2Trash>=1.8.0 in /opt/conda/lib/python3.
9/site-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidget
s>=7.0.0->cufflinks) (1.8.0)
Requirement already satisfied: terminado>=0.8.3 in /opt/conda/lib/python3.
9/site-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidget
s>=7.0.0->cufflinks) (0.17.1)
Requirement already satisfied: nbconvert>=5 in /opt/conda/lib/python3.9/sit
e-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.
0.0->cufflinks) (7.2.9)
Requirement already satisfied: argon2-cffi in /opt/conda/lib/python3.9/site
-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.
0.0->cufflinks) (21.3.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/lib/p
ython3.9/site-packages (from packaging->ipykernel>=4.5.1->ipywidgets>=7.0.0
->cufflinks) (3.0.9)
Requirement already satisfied: platformdirs>=2.5 in /opt/conda/lib/python3.
9/site-packages (from jupyter-core>=4.9.2->jupyter-client>=6.1.12->ipykerne
l>=4.5.1->ipywidgets>=7.0.0->cufflinks) (2.6.2)
Requirement already satisfied: markupsafe>=2.0 in /opt/conda/lib/python3.9/
site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.
0->ipywidgets>=7.0.0->cufflinks) (2.1.2)
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ip
ywidgets>=7.0.0->cufflinks) (0.7.1)
Requirement already satisfied: bleach in /opt/conda/lib/python3.9/site-pack
ages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywid
gets>=7.0.0->cufflinks) (6.0.0)
Requirement already satisfied: tinycss2 in /opt/conda/lib/python3.9/site-pa
ckages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipyw
idgets>=7.0.0->cufflinks) (1.2.1)
Requirement already satisfied: jupyterlab-pygments in /opt/conda/lib/python
3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=
3.6.0->ipywidgets>=7.0.0->cufflinks) (0.2.2)
Requirement already satisfied: nbclient>=0.5.0 in /opt/conda/lib/python3.9/
site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.
0->ipywidgets>=7.0.0->cufflinks) (0.7.2)
Requirement already satisfied: beautifulsoup4 in /opt/conda/lib/python3.9/s
ite-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0
->ipywidgets>=7.0.0->cufflinks) (4.9.3)
Requirement already satisfied: importlib-metadata>=3.6 in /opt/conda/lib/py
thon3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextensi
on~=3.6.0->ipywidgets>=7.0.0->cufflinks) (6.0.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /opt/conda/lib/pytho
n3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~
=3.6.0->ipywidgets>=7.0.0->cufflinks) (1.5.0)
Requirement already satisfied: mistune<3,>=2.0.3 in /opt/conda/lib/python3.
9/site-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.
6.0->ipywidgets>=7.0.0->cufflinks) (2.0.4)
```

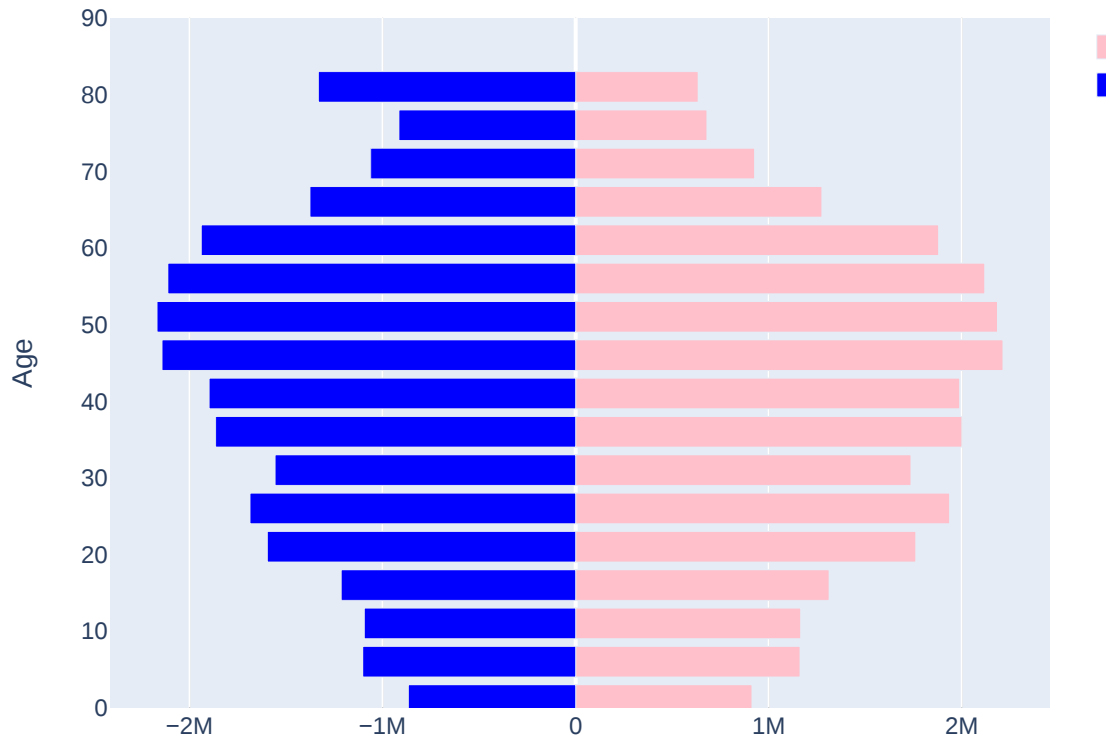
```
Requirement already satisfied: fastjsonschema in /opt/conda/lib/python3.9/site-packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in /opt/conda/lib/python3.9/site-packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (4.17.3)
Requirement already satisfied: argon2-cffi-bindings in /opt/conda/lib/python3.9/site-packages (from argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (21.2.0)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.9/site-packages (from importlib-metadata>=3.6->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (3.12.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /opt/conda/lib/python3.9/site-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (0.19.3)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.9/site-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (19.3.0)
Requirement already satisfied: cffi>=1.0.1 in /opt/conda/lib/python3.9/site-packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (1.14.6)
Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.9/site-packages (from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (2.3.2.post1)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.9/site-packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (0.5.1)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.9/site-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.0.0->cufflinks) (2.20)
Collecting iso3166
  Using cached iso3166-2.1.1-py3-none-any.whl (9.8 kB)
Installing collected packages: iso3166
Successfully installed iso3166-2.1.1
/opt/conda/lib/python3.9/site-packages/geopandas/_compat.py:111: UserWarning:

The Shapely GEOS version (3.10.3-CAPI-1.16.1) is incompatible with the GEOS version PyGEOS was compiled with (3.10.4-CAPI-1.16.2). Conversions between both will be slow.
```

In [3]:
```
#wbdata.get_source() to get all sources
#wbdata.get_topic()
#wbdata.get_indicator()

source_id = 40 #Population Estimates and Projections
indicators = wbdata.get_indicator(source=source_id)
```

[#A] Population Pyramids

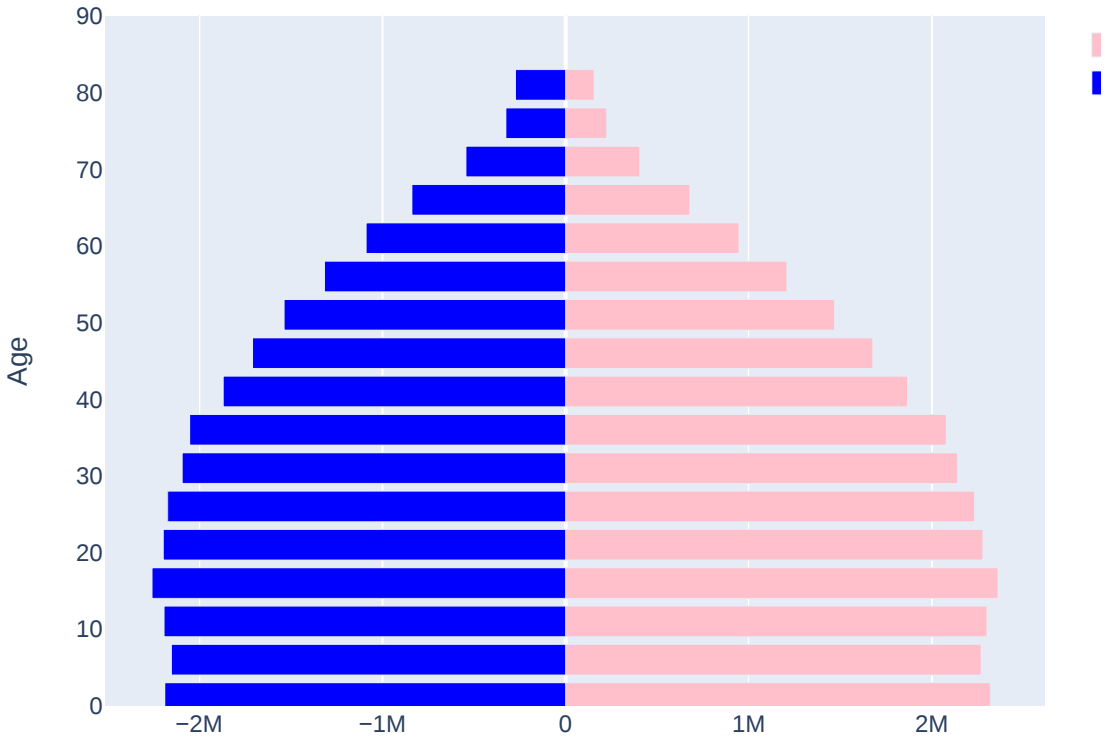
In [4]:
```
# Data from WDI on age-sex comes in the forms of variables
# which take the form "SP.POP.LLHH.MA" for males
```

```
# and "SP.POP.LLHH.FE" for females, where LL is the *low* end of
# age range, like "05" for 5-yo, and HH is the *high* end.

# We construct a list of age-ranges.

# Start with an empty list of age-rages
age_ranges = []

# Ranges top out at 80, and go in five year increments
for i in range(0,80,5):
    age_ranges.append(f"{i:02d}"+f"{i+4:02d}")

age_ranges.append("80UP")

male_variables = {"SP.POP."+age_range+".MA":"Males "+age_range for age_range
female_variables = {"SP.POP."+age_range+".FE":"Females "+age_range for age_r

variables = male_variables
variables.update(female_variables)

# WLD is the World; substitute your own code or list of codes.
# Remember you can search for the appropriate codes using
# wbdata.search_countries("")

df = wbdata.get_dataframe(variables,country="KOR")

py.init_notebook_mode(connected=True)

layout = go.Layout(barmode='overlay',
                   yaxis=go.layout.YAxis(range=[0, 90], title='Age'),
                   xaxis=go.layout.XAxis(title='Number'))

year = 2020

bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
               y = [int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Men',
               marker=dict(color='pink'),
               hoverinfo='skip'
               ),

        go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
               y=[int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Women',
               marker=dict(color='blue'),
               hoverinfo='skip',
               )
        ]
py.iplot(dict(data=bins, layout=layout))
```



In [5]:
```
df = wbdata.get_dataframe(variables,country="MMR")

py.init_notebook_mode(connected=True)

layout = go.Layout(barmode='overlay',
                   yaxis=go.layout.YAxis(range=[0, 90], title='Age'),
                   xaxis=go.layout.XAxis(title='Number'))

year = 2020

bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
               y = [int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Men',
               marker=dict(color='pink'),
               hoverinfo='skip'
               ),

        go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
               y=[int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Women',
               marker=dict(color='blue'),
               hoverinfo='skip',
```

```
                )
        ]
py.iplot(dict(data=bins, layout=layout))
```

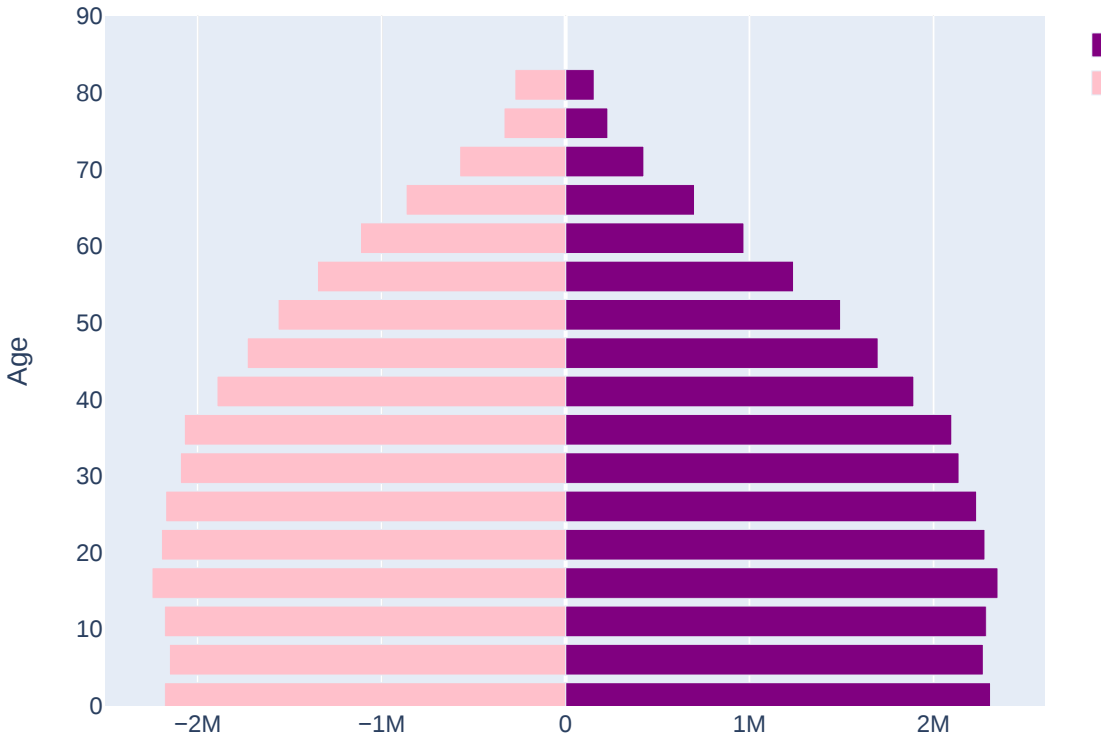


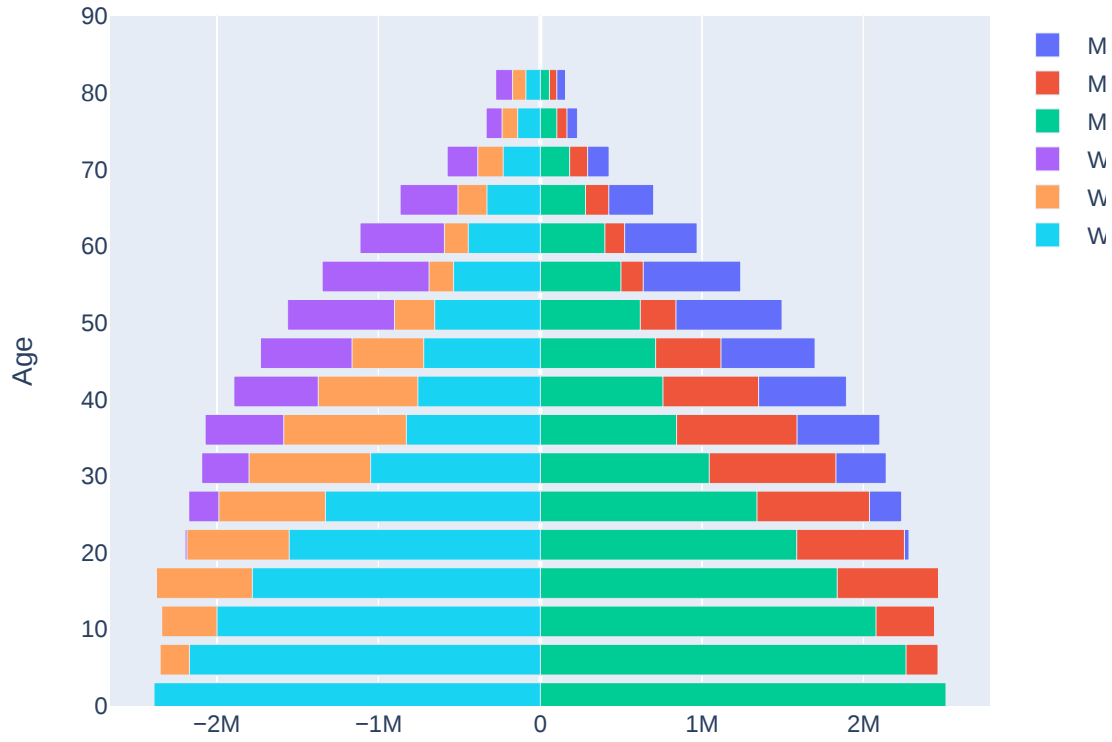
[C#] Animated Population Pyramids

In [26]:
```
age_ranges = []

# Ranges top out at 80, and go in five year increments
for i in range(0,80,5):
    age_ranges.append(f"{i:02d}"+f"{i+4:02d}")

age_ranges.append("80UP")

print(age_ranges)

male_variables = {"SP.POP."+age_range+".MA":"Males "+age_range for age_range
female_variables = {"SP.POP."+age_range+".FE":"Females "+age_range for age_r

variables = male_variables
variables.update(female_variables)

print(variables)
```

```
# MMR
df = wbdata.get_dataframe(variables,country="MMR")

py.init_notebook_mode(connected=True)

layout = go.Layout(barmode='overlay',
                   yaxis=go.layout.YAxis(range=[0, 90], title='Age'),
                   xaxis=go.layout.XAxis(title='Number'))

year = 2021

bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
               y = [int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Men',
               marker=dict(color='purple'),
               hoverinfo='skip'
               ),

        go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
               y=[int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Women',
               marker=dict(color='pink'),
               hoverinfo='skip',
               )
        ]
py.iplot(dict(data=bins, layout=layout))

# Count down by increments of 20 years
years = range(2021,1961,-20)

# This makes a list of graphs, year by year
bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
               y = [int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Men {:d}'.format(year),
               hoverinfo='skip'
              )
        for year in years]

bins += [go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
                y=[int(s[:2])+1 for s in age_ranges],
                orientation='h',
                name='Women {:d}'.format(year),
                hoverinfo='skip',
               )
         for year in years]

py.iplot(dict(data=bins, layout=layout))
```

['0004', '0509', '1014', '1519', '2024', '2529', '3034', '3539', '4044', '4549', '5054', '5559', '6064', '6569', '7074', '7579', '80UP']
{'SP.POP.0004.MA': 'Males 0004', 'SP.POP.0509.MA': 'Males 0509', 'SP.POP.1014.MA': 'Males 1014', 'SP.POP.1519.MA': 'Males 1519', 'SP.POP.2024.MA': 'Males 2024', 'SP.POP.2529.MA': 'Males 2529', 'SP.POP.3034.MA': 'Males 3034', 'SP.POP.3539.MA': 'Males 3539', 'SP.POP.4044.MA': 'Males 4044', 'SP.POP.4549.MA': 'Males 4549', 'SP.POP.5054.MA': 'Males 5054', 'SP.POP.5559.MA': 'Males 5559', 'SP.POP.6064.MA': 'Males 6064', 'SP.POP.6569.MA': 'Males 6569', 'SP.POP.7074.MA': 'Males 7074', 'SP.POP.7579.MA': 'Males 7579', 'SP.POP.80UP.MA': 'Males 80UP', 'SP.POP.0004.FE': 'Females 0004', 'SP.POP.0509.FE': 'Females 0509', 'SP.POP.1014.FE': 'Females 1014', 'SP.POP.1519.FE': 'Females 1519', 'SP.POP.2024.FE': 'Females 2024', 'SP.POP.2529.FE': 'Females 2529', 'SP.POP.3034.FE': 'Females 3034', 'SP.POP.3539.FE': 'Females 3539', 'SP.POP.4044.FE': 'Females 4044', 'SP.POP.4549.FE': 'Females 4549', 'SP.POP.5054.FE': 'Females 5054', 'SP.POP.5559.FE': 'Females 5559', 'SP.POP.6064.FE': 'Females 6064', 'SP.POP.6569.FE': 'Females 6569', 'SP.POP.7074.FE': 'Females 7074', 'SP.POP.7579.FE': 'Females 7579', 'SP.POP.80UP.FE': 'Females 80UP'}

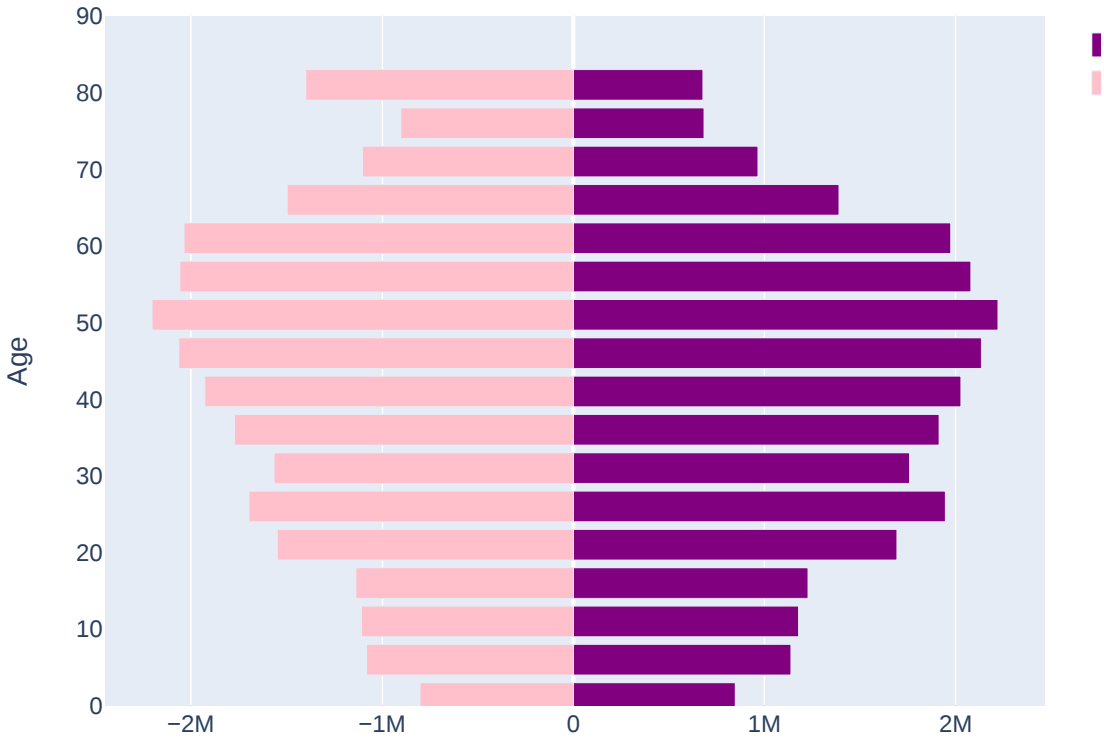


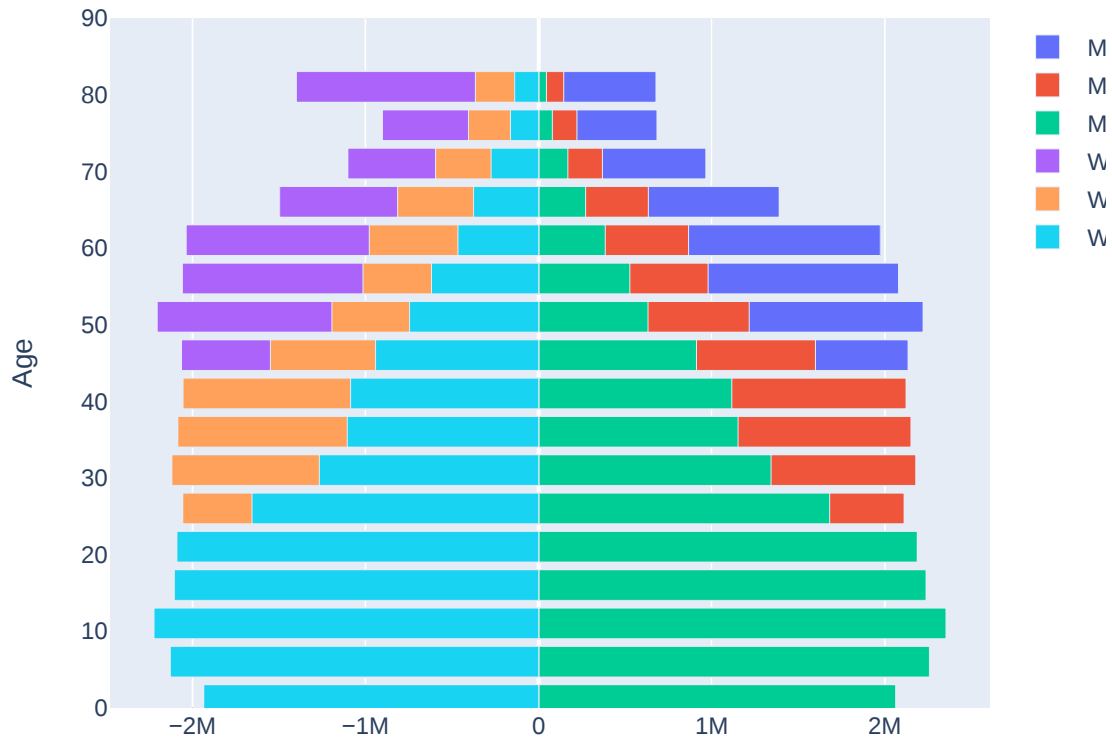


In [27]:

```
# KOR

df = wbdata.get_dataframe(variables,country="KOR")

py.init_notebook_mode(connected=True)

layout = go.Layout(barmode='overlay',
                   yaxis=go.layout.YAxis(range=[0, 90], title='Age'),
                   xaxis=go.layout.XAxis(title='Number'))

year = 2021

bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
               y = [int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Men',
               marker=dict(color='purple'),
               hoverinfo='skip'
               ),

        go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
               y=[int(s[:2])+1 for s in age_ranges],
               orientation='h',
               name='Women',
```

```
                marker=dict(color='pink'),
                hoverinfo='skip',
                )
        ]
py.iplot(dict(data=bins, layout=layout))

# Count down by increments of 20 years
years = range(2021,1961,-20)

# This makes a list of graphs, year by year
bins = [go.Bar(x = df.loc[str(year),:].filter(regex="Male").values,
                y = [int(s[:2])+1 for s in age_ranges],
                orientation='h',
                name='Men {:d}'.format(year),
                hoverinfo='skip'
               )
        for year in years]

bins += [go.Bar(x = -df.loc[str(year),:].filter(regex="Female").values,
                 y=[int(s[:2])+1 for s in age_ranges],
                 orientation='h',
                 name='Women {:d}'.format(year),
                 hoverinfo='skip',
                )
         for year in years]

py.iplot(dict(data=bins, layout=layout))
```


90
80
70
60
50
40
30
20
10
0
Age
−2M
−1M
0
1M
2M



[#A] Population Dataframe

In [6]:

```
def pop_df(year='2021', group='all', age_lower=0, age_upper=100, location='w
    df = pop_df_helper(year, age_lower, age_upper, location)

    if group == 'Male':
        return df.drop(columns = ['Female'])
    elif group == 'Female':
        return df.drop(columns = ['Male'])
    else:
        total_pop = df["Female"] + df["Male"]
        df["Total"] = total_pop
    return df

# Returns a list of input strings for population age ranges
def pop_df_helper(year, age_lower, age_upper, location):
    if location != 'WLD':
        country_alpha3 = iso3166.countries.get(location).alpha3
        country_name = iso3166.countries.get(location).apolitical_name
    else:
        country_alpha3 = 'WLD'
        country_name = 'World'
    inputs = list_of_age_inputs(age_lower, age_upper)
```

```
    #create indicator-dictionaries required for WBData API
    indicator_dict_m = {}
    indicator_dict_f = {}
    for i in inputs:
        key_m = "SP.POP.{}.{}".format(i, 'MA')
        key_f = "SP.POP.{}.{}".format(i, 'FE')
        value = "{}-{}".format(i[:2], i[2:])
        indicator_dict_m[key_m] = value
        indicator_dict_f[key_f] = value

#source_id = 40, refer to wbdata.get_source()
    wbdf_m = wbdata.get_dataframe(indicator_dict_m, country=country_alpha3,
    wbdf_f = wbdata.get_dataframe(indicator_dict_f, country=country_alpha3,
    datas_m = wbdf_m.query("date=='{}'".format(year)).sum(axis=0).tolist()
    datas_f = wbdf_f.query("date=='{}'".format(year)).sum(axis=0).tolist()
    df = pd.DataFrame({
        'Country': country_name,
        'Year': year,
        'Age': list(indicator_dict_m.values()),
        'Female': datas_f,
        'Male': datas_m
    })
    return df

# Returns a list of input strings for population age ranges
def list_of_age_inputs(age_lower, age_upper):
    def round_down(n):
        return max(0, n - n % 5)

    def round_up(n):
        return n - n % 5 + 5

    results = []
    r_lower_bound, r_upper_bound = round_down(age_lower), round_up(age_upper
    while r_lower_bound < min(79, r_upper_bound):
        results.append("{:02d}{:02d}".format(r_lower_bound, r_lower_bound +
        r_lower_bound += 5
    if age_upper >= 80:
        results.append('80UP')
    return results

df = pop_df(year=2018,group='Total',age_lower = 0, age_upper = 100,location=
df
```

Out[6]:

	Country	Year	Age	Female	Male	Total
0	World	2018	00-04	333098585.0	354088174.0	687186759.0
1	World	2018	05-09	325470037.0	347389724.0	672859761.0
2	World	2018	10-14	308035159.0	329029796.0	637064955.0
3	World	2018	15-19	295004074.0	315050425.0	610054499.0
4	World	2018	20-24	288393154.0	306524173.0	594917327.0
5	World	2018	25-29	295497148.0	310713277.0	606210425.0
6	World	2018	30-34	285117787.0	296708276.0	581826063.0
7	World	2018	35-39	260070368.0	268841828.0	528912196.0
8	World	2018	40-44	238289828.0	244013351.0	482303179.0
9	World	2018	45-49	235273671.0	237876277.0	473149948.0
10	World	2018	50-54	215725423.0	214089804.0	429815227.0
11	World	2018	55-59	181123022.0	174951704.0	356074726.0
12	World	2018	60-64	162791878.0	151598168.0	314390046.0
13	World	2018	65-69	132182084.0	117713443.0	249895527.0
14	World	2018	70-74	92955177.0	79116689.0	172071866.0
15	World	2018	75-79	68827619.0	53385227.0	122212846.0
16	World	2018	80-UP	88876981.0	53954172.0	142831153.0

[A#] Population Statistics

In [7]:

```
def population(year='', sex='', age_range=(0), place=''):
    age_lower, age_upper = age_range
    df = pop_df(year, sex, age_lower, age_upper, place)
    inputs = list_of_age_inputs(age_lower, age_upper);
    age_l = inputs[0][0:2]

    if sex.lower() == 'people':
        g = 'people'
    elif sex.lower() == 'male':
        g = "males"
    elif sex.lower() == 'female':
        g = "females"
    else:
        g = sex

    if age_upper >= 80:
        age_h = '80 or over'
    else:
        age_h = inputs[-1][2:4]

    if place == 'WLD' and sex.lower() != "people":
        loc = 'the world'
        print("In {}, there are {} {} aged {} to aged {} living in {}.".form
```

```
    elif place == "WLD" and sex.lower() == "people":
        loc = "the world"
        print("In {}, there are {} {} aged {} to aged {} living in {}.".form

    elif place != "WLD" and sex.lower() == "people":
        loc = iso3166.countries.get(place).alpha3
        print("In {}, there are {} {} aged {} to aged {} living in {}.".form
    else:
        loc = iso3166.countries.get(place).alpha3
        print("In {}, there are {} {} aged {} to aged {} living in {}.".form

population(year=2010,sex='People',age_range=(0,50),place='WLD')
```

```
In 2010, there are 5889158289 people aged 00 to aged 54 living in the worl
d.
```

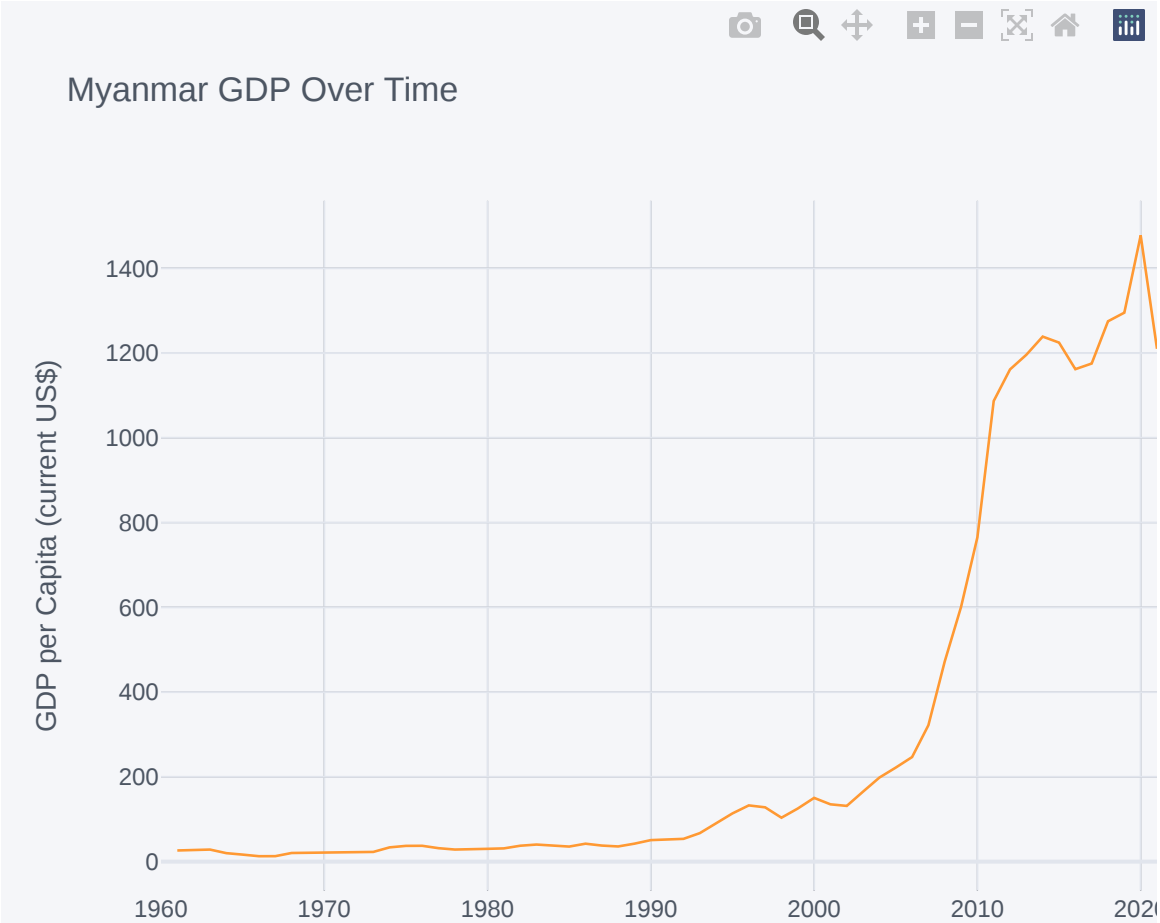
In [ ]:

[#C] Other Visualization Tools

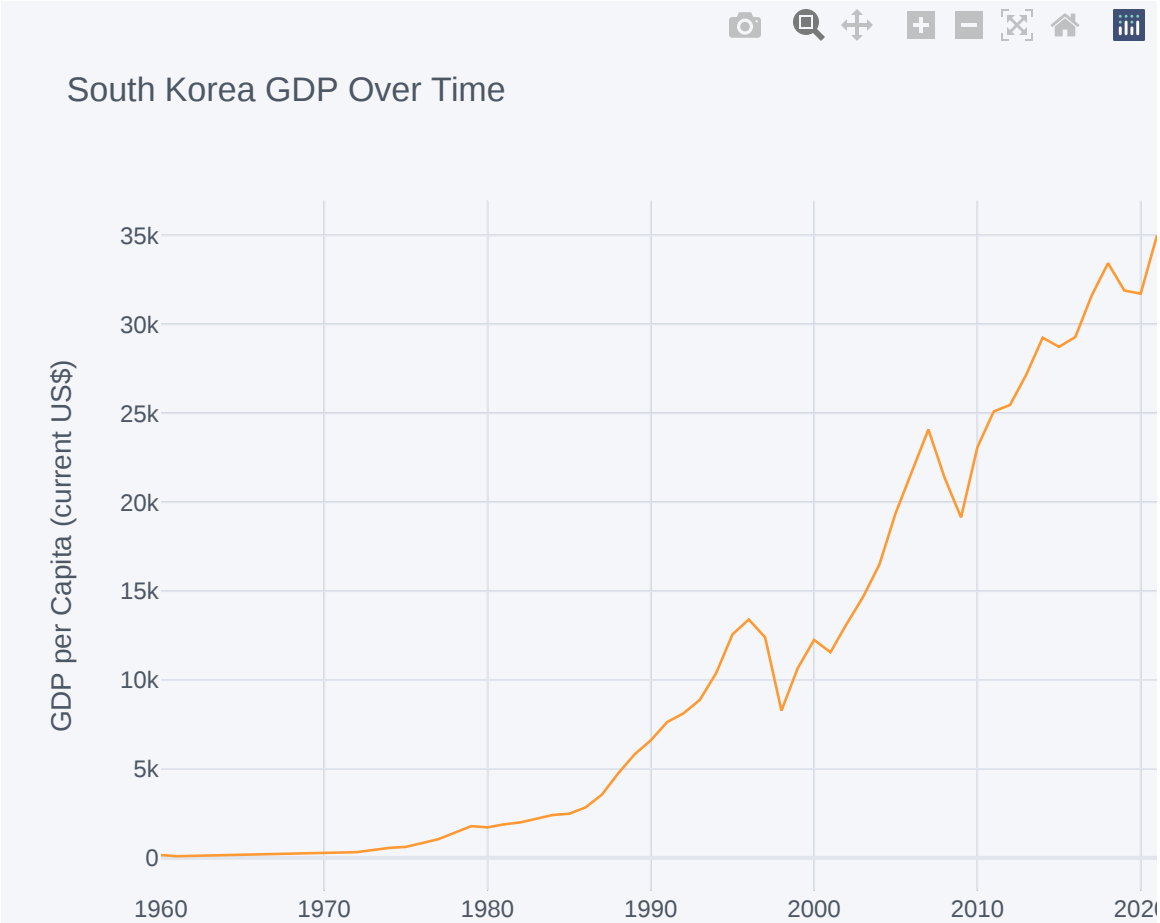
1: GDP Visualizations showing the growth rate of GDP over time in South Korea vs Myanmar and how GDP has changed over time in each country

In [13]:

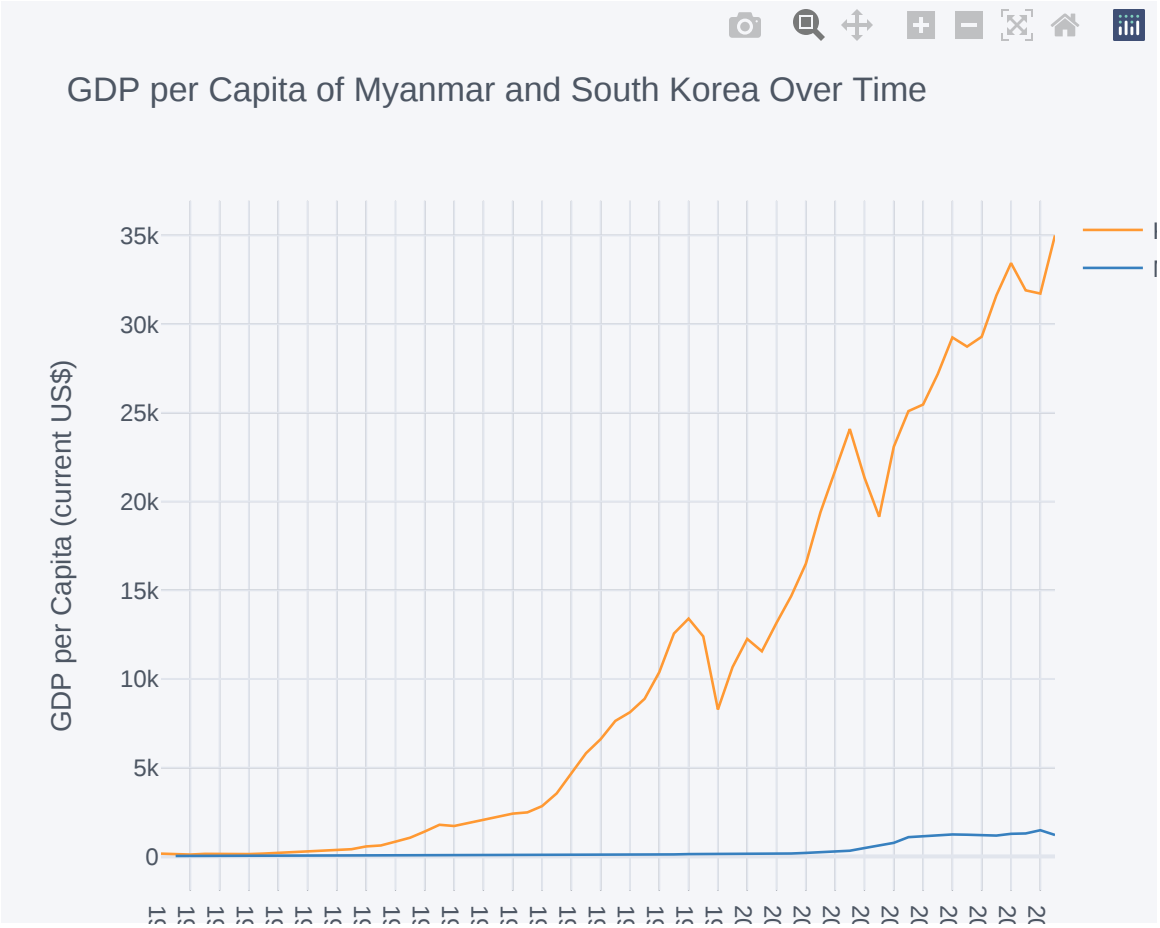
```
# Give variable for clarity
variable_labels = {"NY.GDP.PCAP.CD":"GDP per capita"}

myanmar = wbdata.get_dataframe(variable_labels, country="MMR")

# Date index is of type string; change to integers
myanmar.index = myanmar.index.astype(int)

# Print a few years' data
myanmar.head()

myanmar.iplot(title="Myanmar GDP Over Time",xTitle='Year',yTitle='GDP per Ca
```



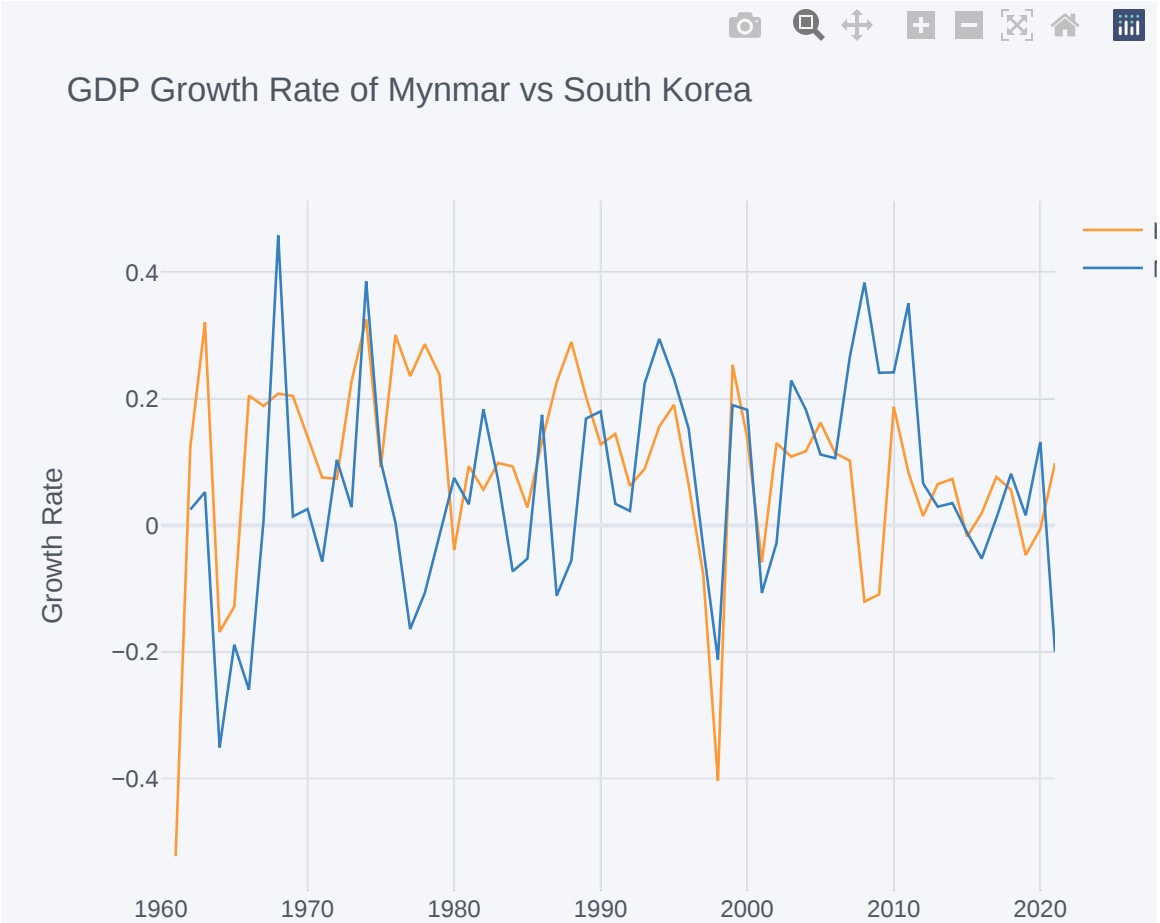
In [12]:
```
variable_labels = {"NY.GDP.PCAP.CD":"GDP per capita"}
south_korea = wbdata.get_dataframe(variable_labels, country="KOR")
south_korea.index = south_korea.index.astype(int)
south_korea.head()
south_korea.iplot(title="South Korea GDP Over Time",xTitle='Year',yTitle='GD
```



In [14]:
```
gdp_indicator = {"NY.GDP.PCAP.CD":"GDP per capita"}
countries = {"MMR":"Mynmar",
             "KOR":"South Korea"}

gdp = wbdata.get_dataframe(gdp_indicator, country = countries).squeeze().uns
gdp.iplot(title="GDP per Capita of Myanmar and South Korea Over Time",xTitle
```

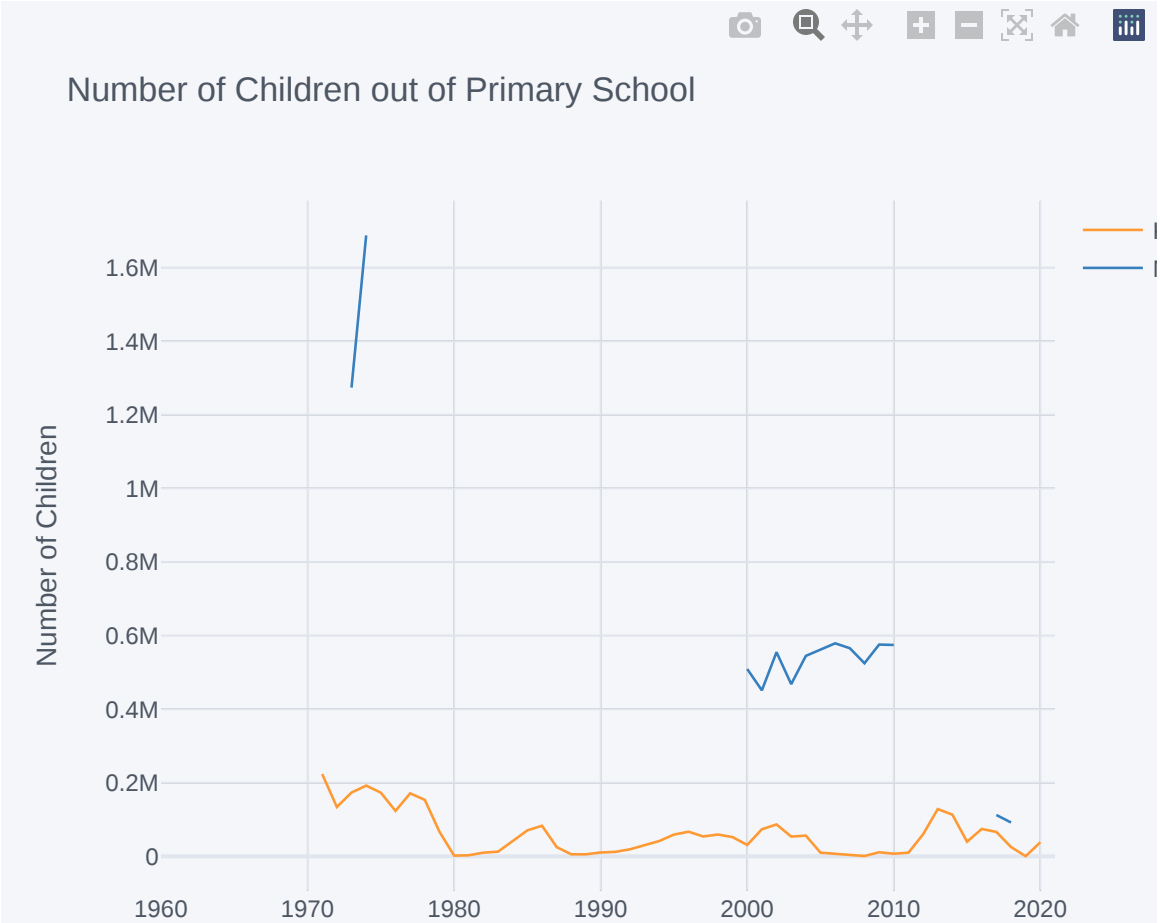


In [15]:
```
variable_labels = {"NY.GDP.PCAP.CD":"GDP per capita"}

# Three letter codes come from wbdata.get_country()
countries = {"MMR":"Mynmar",
             "KOR":"South Korea"}

df = wbdata.get_dataframe(variable_labels, country = countries).squeeze()

df = df.unstack('country')
# Date index is of type string; change to integers
df.index = df.index.astype(int)

# Differences (over time) in logs give us growth rates
np.log(df).diff().iplot(title="GDP Growth Rate of Mynmar vs South Korea",
                        yTitle="Growth Rate",xTitle='Year')
```

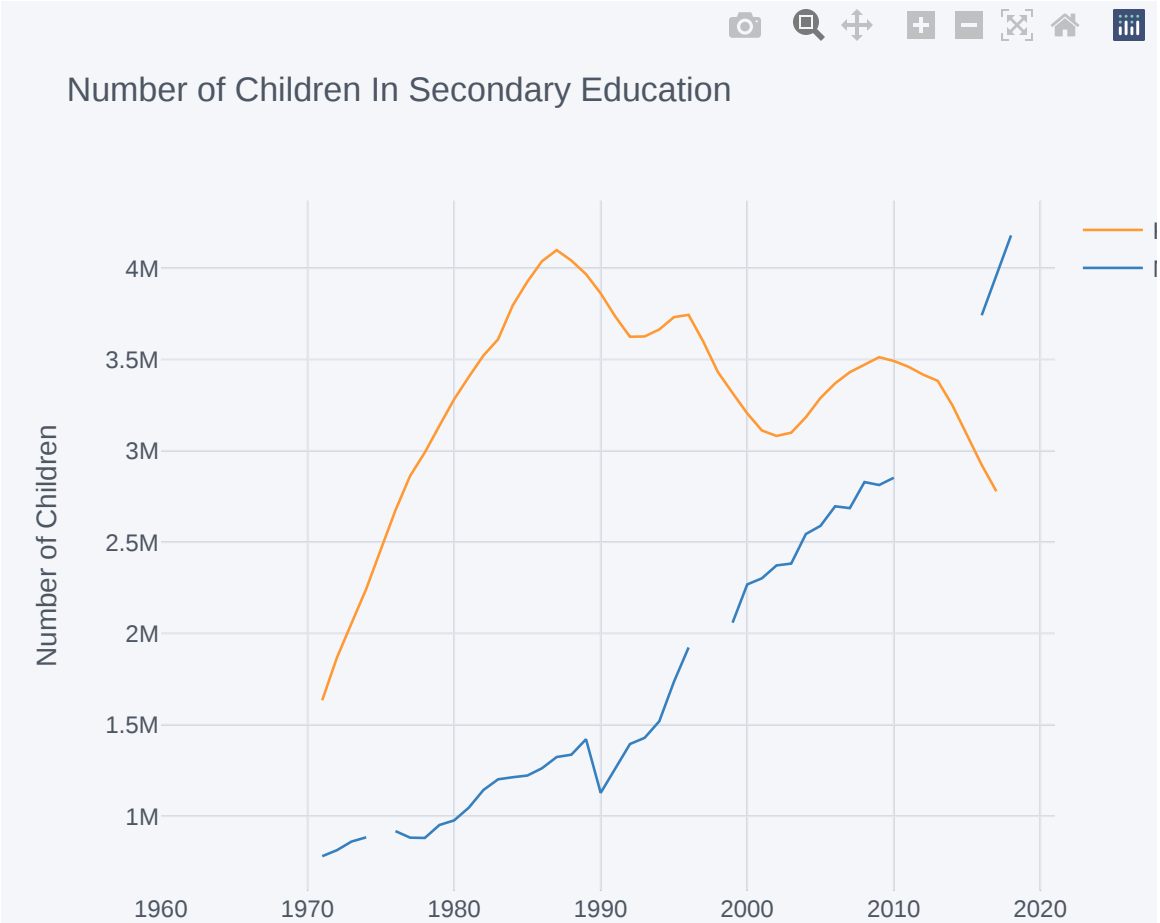


2. Education Visualizations of South Korea and Myanmar showing the number of children out of primary school, the number of children in secondary education, and the percent of children enrolled in primary school

In [16]:
```
#variable_labels = {"SE.PRM.UNER":  "Children out of primary school"}

# Three letter codes come from wbdata.get_country()
countries = {"MMR":"Myanmar",
             "KOR":"South Korea"}
variable_labels1 = {"SE.PRM.UNER":  "Children out of primary school"}

df = wbdata.get_dataframe(variable_labels1, country = countries).squeeze()

df = df.unstack('country')
# Date index is of type string; change to integers
df.index = df.index.astype(int)

# Differences (over time) in logs give us growth rates
df.iplot(title="Number of Children out of Primary School",
                       yTitle="Number of Children",xTitle='Year')
```

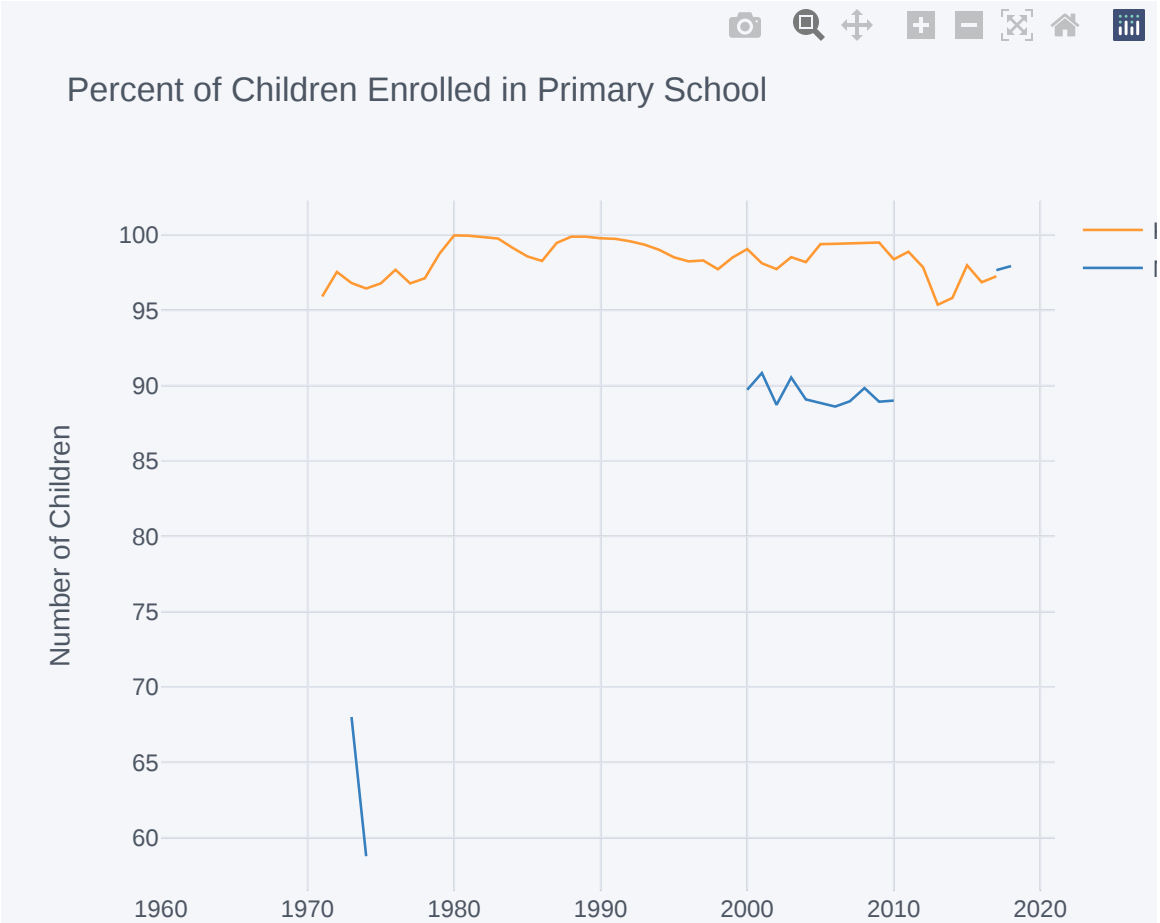


In [17]:
```
variable_labels2 = {"SE.SEC.ENRL.GC": "Secondary Education"}
countries = {"MMR":"Myanmar",
             "KOR":"South Korea"}

df = wbdata.get_dataframe(variable_labels2, country = countries).squeeze()

df = df.unstack('country')
# Date index is of type string; change to integers
df.index = df.index.astype(int)

# Differences (over time) in logs give us growth rates
df.iplot(title="Number of Children In Secondary Education",
                        yTitle="Number of Children",xTitle='Year')
```



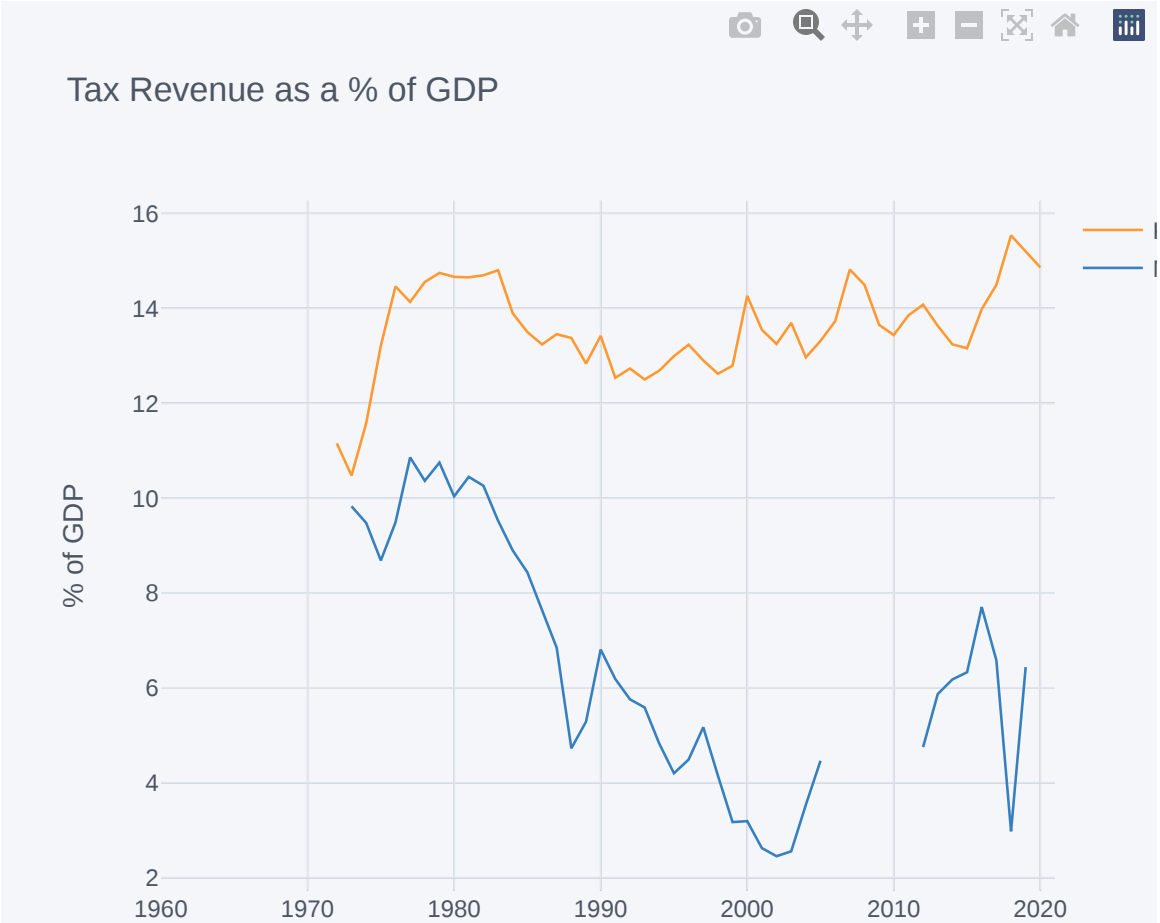
In [18]:

```
variable_labels3 = {"SE.PRM.NENR": "Percent Net Primary School Enrollment"}
countries = {"MMR":"Myanmar",
             "KOR":"South Korea"}

df = wbdata.get_dataframe(variable_labels3, country = countries).squeeze()

df = df.unstack('country')
# Date index is of type string; change to integers
df.index = df.index.astype(int)

# Differences (over time) in logs give us growth rates
df.iplot(title="Percent of Children Enrolled in Primary School",
                        yTitle="Number of Children",xTitle='Year')
```



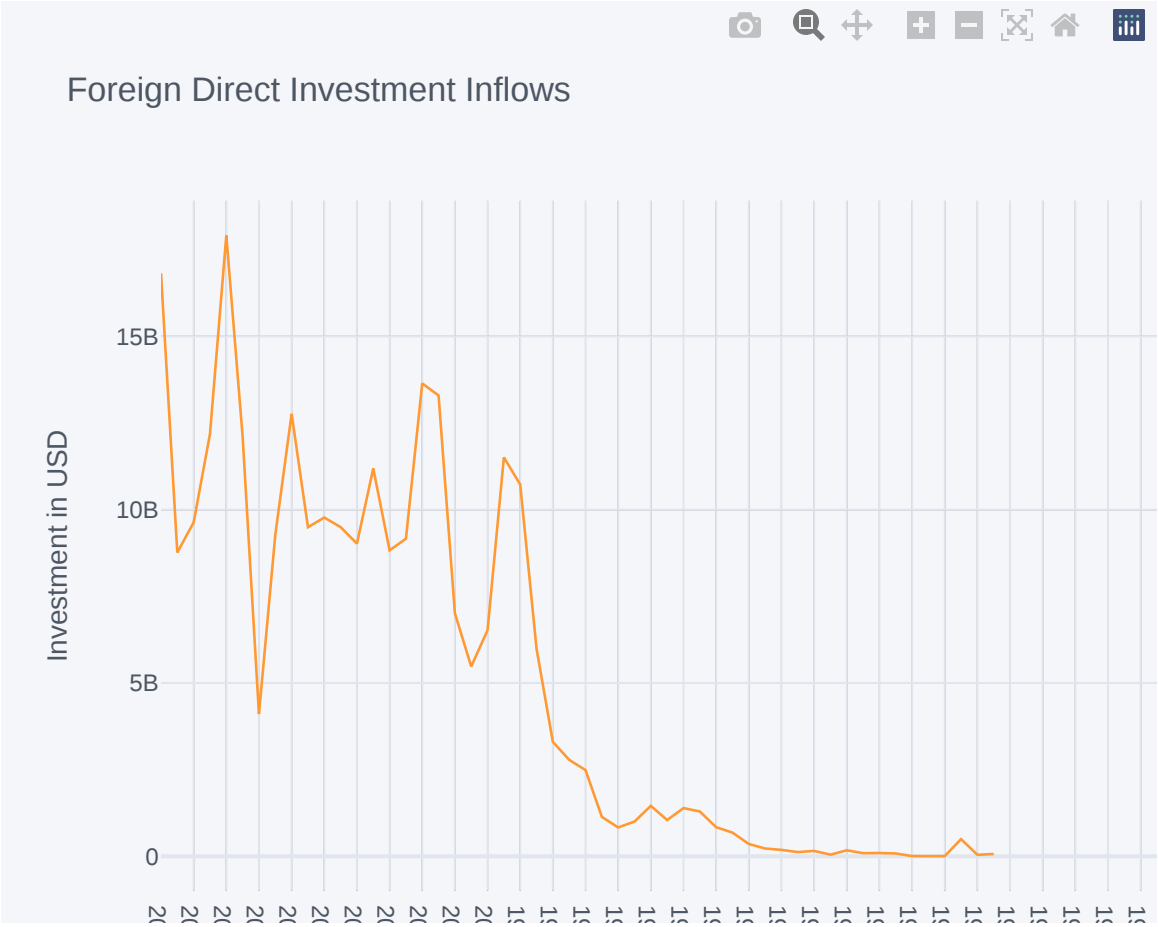
3. Tax Visualizations showing the tax revenue over time for Myanmar and South Korea

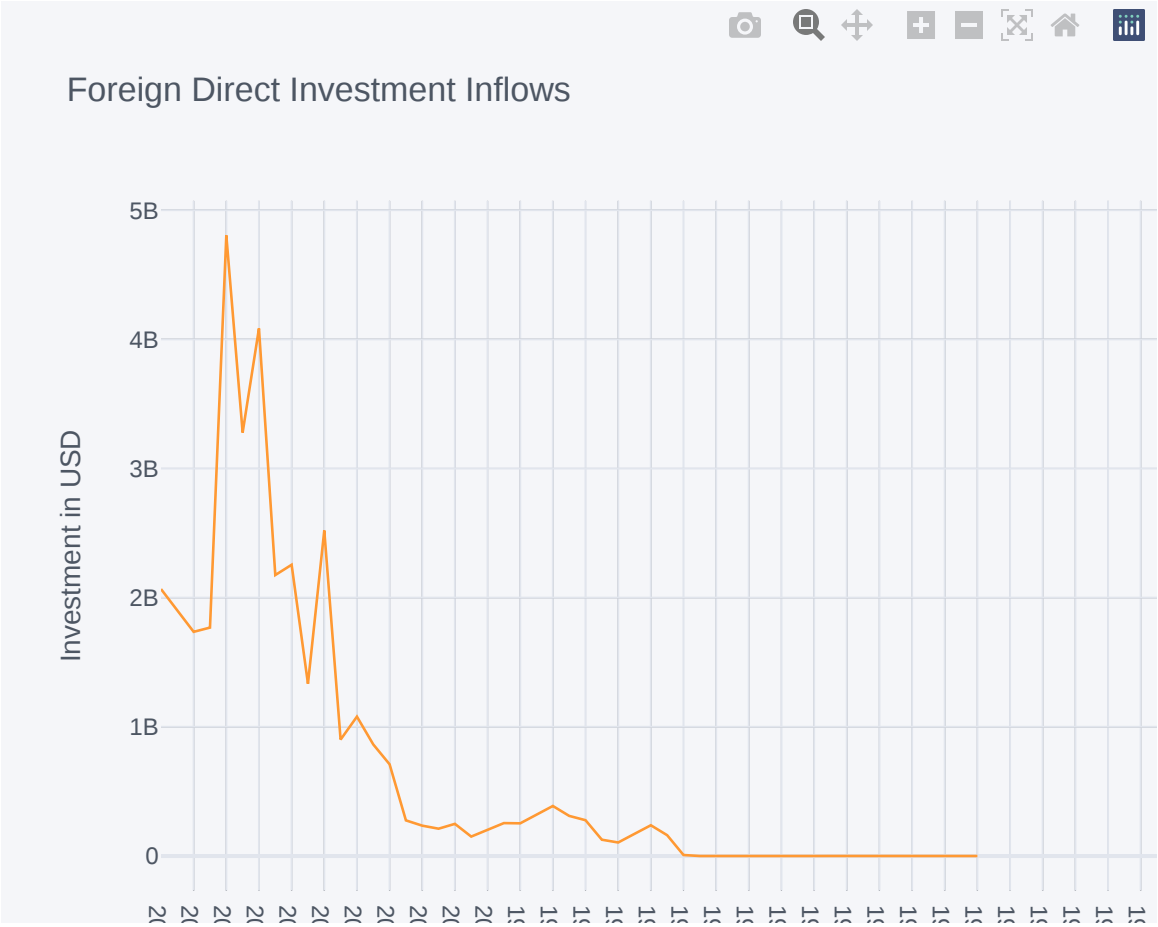
In [19]:
```
variable_labels4 = {"GC.TAX.TOTL.GD.ZS": "Tax Revenue"}

Myanmar4 = wbdata.get_dataframe(indicators=variable_labels2,country="MMR")
Korea4 = wbdata.get_dataframe(variable_labels4, country="KOR")

countries = {"MMR":"Myanmar",
             "KOR":"South Korea"}

df = wbdata.get_dataframe(variable_labels4, country = countries).squeeze()

df = df.unstack('country')
# Date index is of type string; change to integers
df.index = df.index.astype(int)

# Differences (over time) in logs give us growth rates
df.iplot(title="Tax Revenue as a % of GDP",
                      yTitle="% of GDP",xTitle='Year')
```

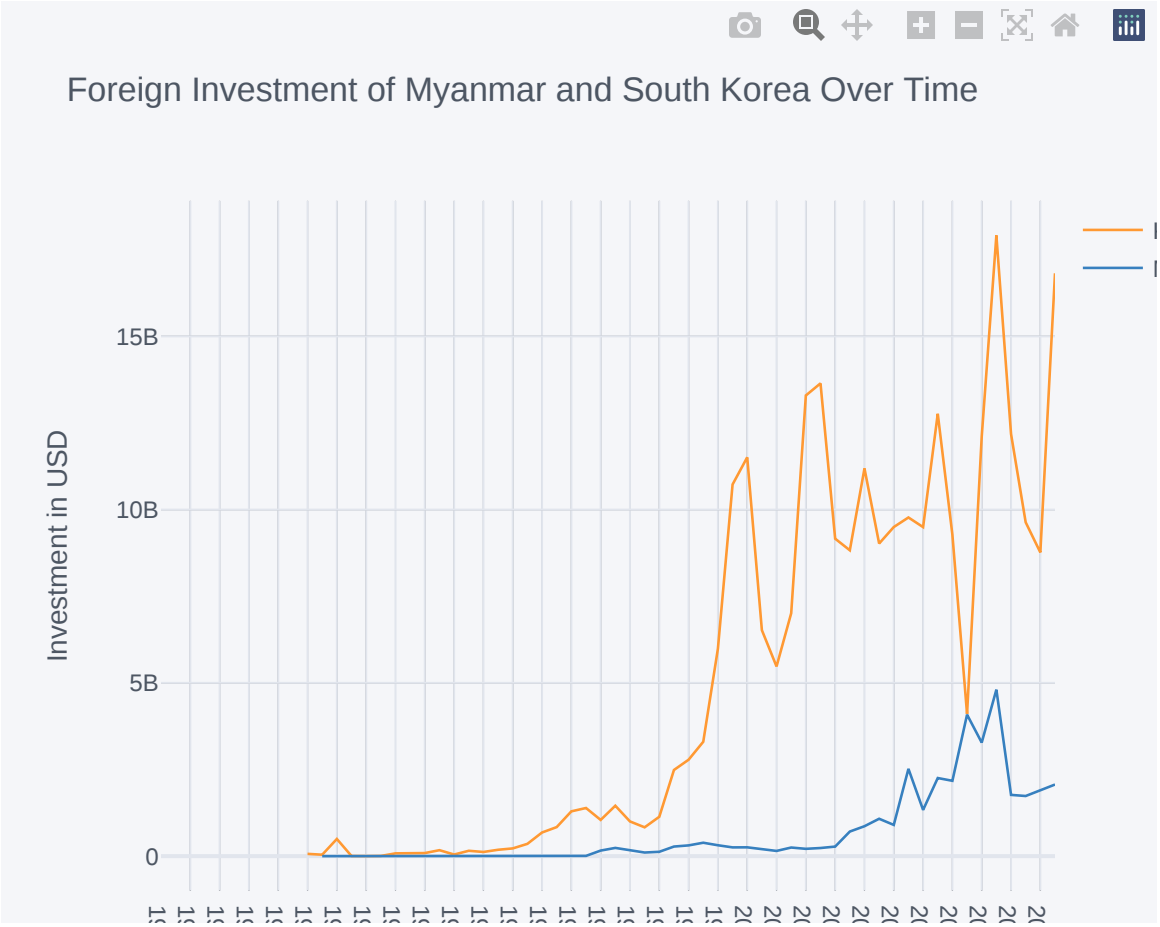


4. Foreign Direct Investment Visualizations showing how FDI changes between Myanmar and South Korea

In [20]:
```
variable_labels4 = {"BX.KLT.DINV.CD.WD":"Foreign Direct Investment, net infl

korea_fdi = wbdata.get_dataframe(variable_labels4, country="KOR")
myanmar_fdi = wbdata.get_dataframe(variable_labels4, country = "MMR")

korea_fdi.iplot(title="Foreign Direct Investment Inflows",xTitle='Year',yTit
myanmar_fdi.iplot(title="Foreign Direct Investment Inflows",xTitle='Year',yT

foreign_investment= wbdata.get_dataframe(variable_labels4, country= countrie

foreign_investment.iplot(title= "Foreign Investment of Myanmar and South Kor
```

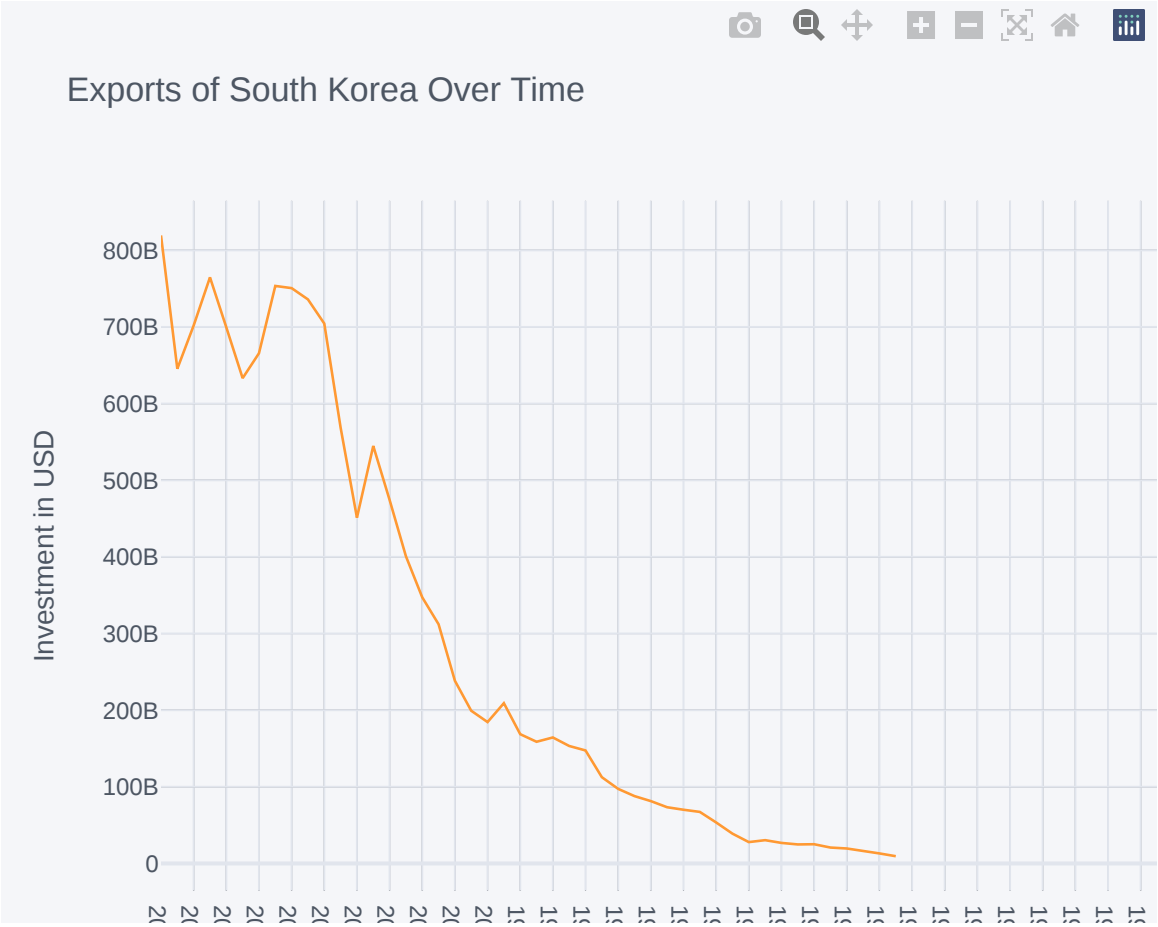

Foreign Direct Investment Inflows
Investment in USD
15B
10B
5B
0

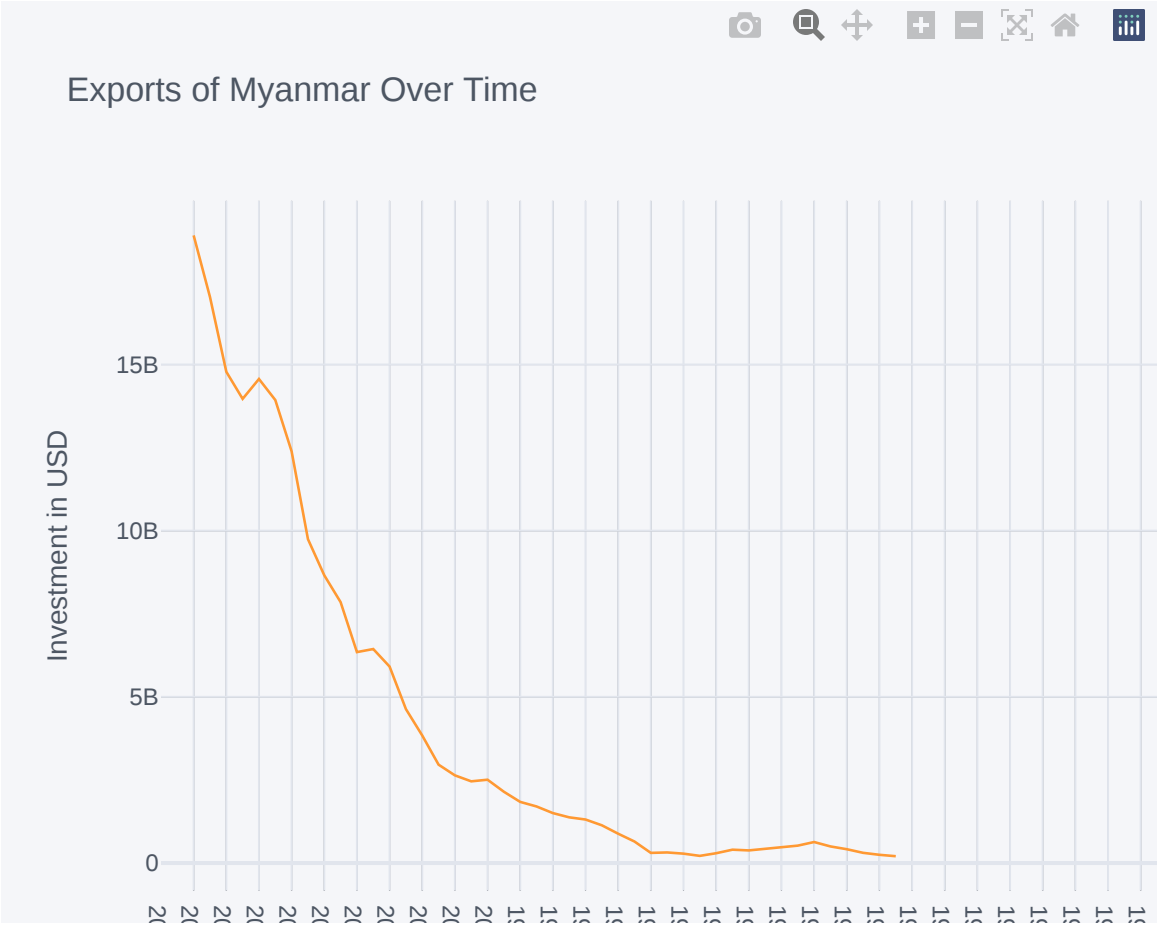

Foreign Direct Investment Inflows
Investment in USD
5B
4B
3B
2B
1B
0

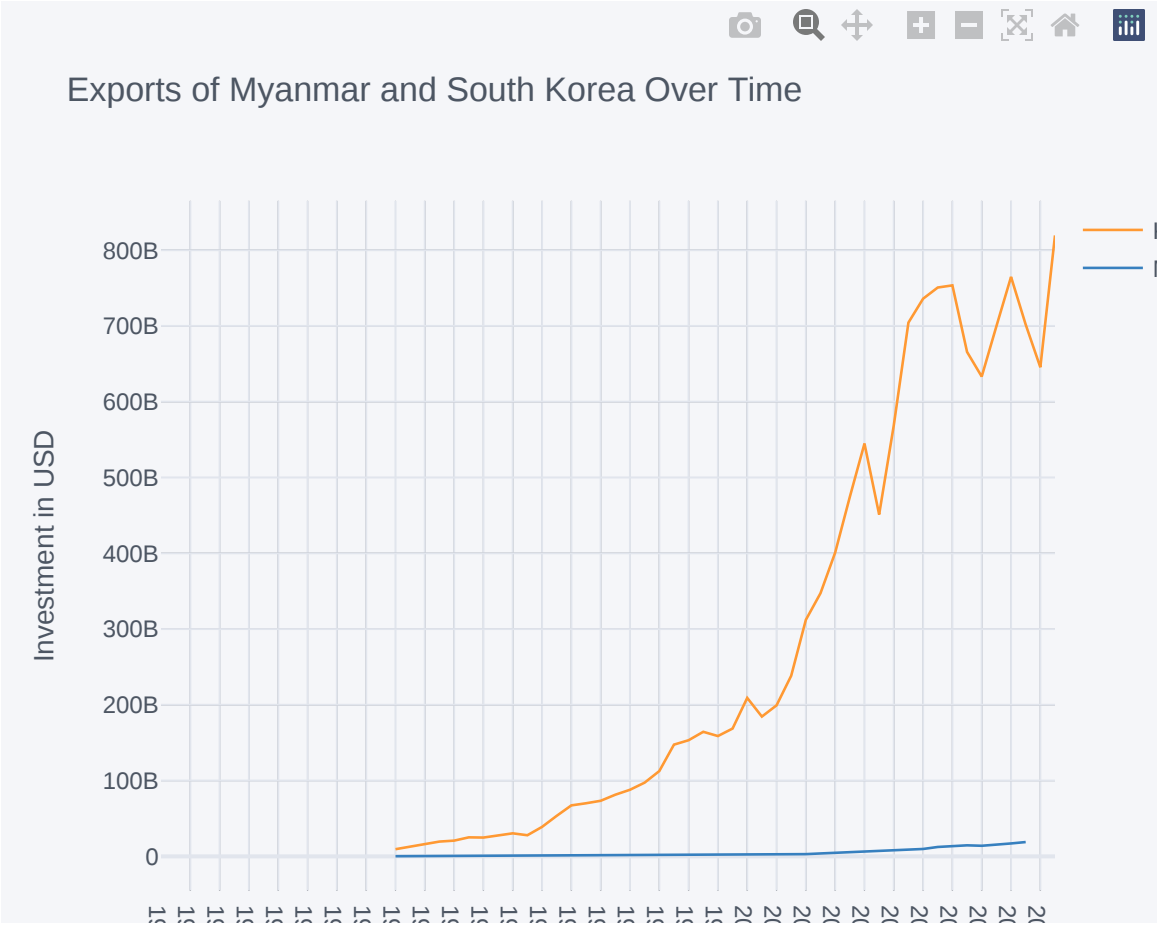


5. Export of goods and services visualizations showing how FDI changes between Myanmar and South Korea

In [25]:
```
variable_labels5 = {"BX.GSR.TOTL.CD":"Exports of Goods and Services in USD"}

korea_exports = wbdata.get_dataframe(variable_labels5, country="KOR")
myanmar_exports = wbdata.get_dataframe(variable_labels5, country = "MMR")

exports= wbdata.get_dataframe(variable_labels5, country= countries).squeeze(

korea_exports.iplot(title= "Exports of South Korea Over Time", xTitle= 'Year
myanmar_exports.iplot(title= "Exports of Myanmar Over Time", xTitle= 'Year',
exports.iplot(title= "Exports of Myanmar and South Korea Over Time", xTitle=
```


Exports of South Korea Over Time
Investment in USD
800B
700B
600B
500B
400B
300B
200B
100B
0


Exports of Myanmar Over Time
Investment in USD
15B
10B
5B
0


Exports of Myanmar and South Korea Over Time
Investment in USD
800B
700B
600B
500B
400B
300B
200B
100B
0

In [ ]: