TASK #1: PLOT INTERACTIVE SCATTERPLOT USING PLOTLY EXPRESS

In [1]:

```
# Make sure to install plotly
!pip install plotly==4.14.3
```

```
Requirement already satisfied: plotly==4.14.3 in c:\users\tashf\anaconda3\lib\site-packa
ges (4.14.3)
Requirement already satisfied: retrying>=1.3.3 in c:\users\tashf\anaconda3\lib\site-pack
ages (from plotly==4.14.3) (1.3.4)
Requirement already satisfied: six in c:\users\tashf\anaconda3\lib\site-packages (from p
lotly==4.14.3) (1.16.0)
```

In [2]:

```
# The plotly Python package empowers anyone to create, manipulate and render graphical f
# The figures are represented by data structures referred to as figures.
# The rendering process uses the Plotly.js JavaScript library under the hood but you nev
# Figures can be represented in Python either as dictionaries or as instances of the plo

# Note:
# Plotly Express is the recommended entry-point into the plotly package
# PLotly Express is the high-level plotly.express module that consists of Python functio
# plotly.express module contains functions that can create interactive figures using a v
# Plotly Express is refered to as px.
# Plotly Express is a built-in part of the plotly library
# Plotly Express function uses graph objects internally and returns a plotly.graph_objec
# check out the documentation here: https://plotly.com/python/plotly-express/

import plotly.express as px
import pandas as pd
```

In [3]:

```
salary_df = pd.read_csv(r"C:\Users\tashf\Desktop\Python Data Visualiazation with plotly\
```

In [4]:

```
salary_df
```

Out[4]:

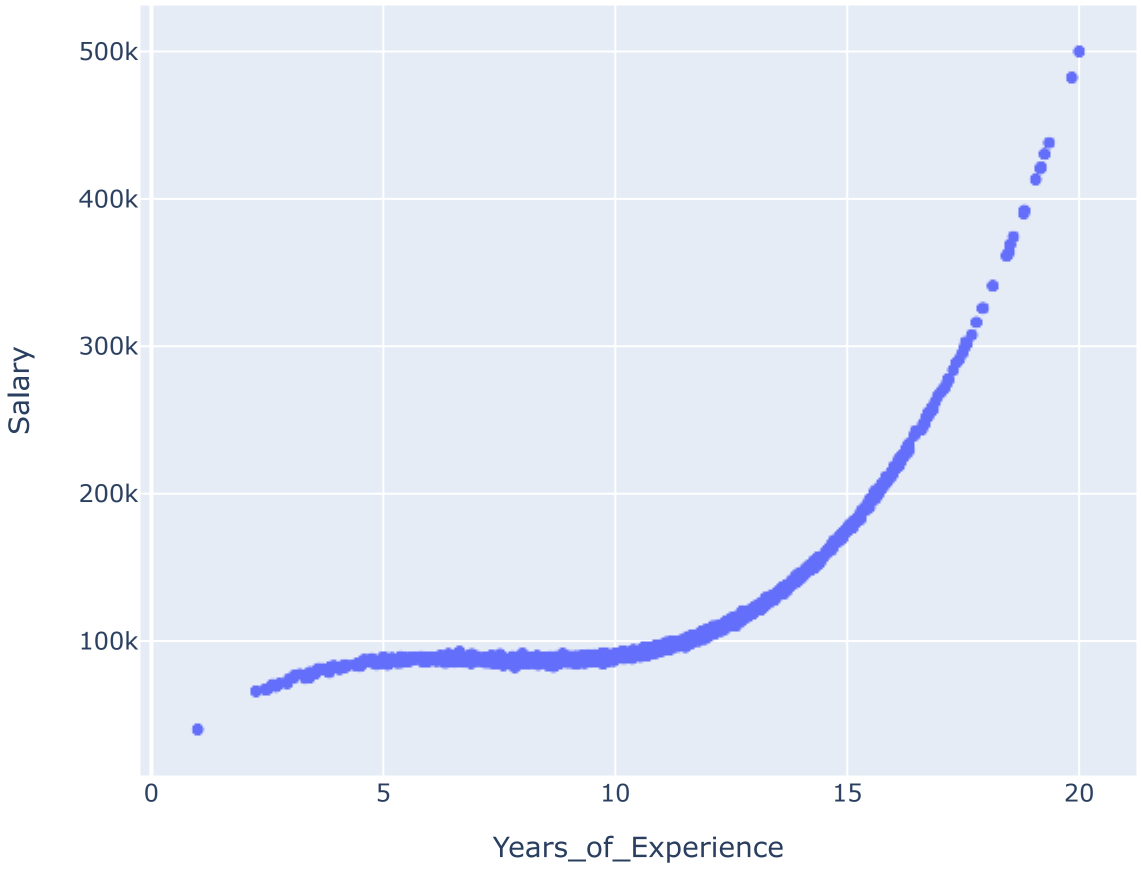
	Years_of_Experience	Salary
0	1.000000	40000.00000
1	2.257942	65979.42119
2	2.450875	67253.57549
3	2.498713	67342.43510
4	2.613729	70532.20448
...	...	...
1995	19.178575	421534.69100
1996	19.254499	430478.02650
1997	19.353369	438090.84540
1998	19.842520	482242.16080
1999	20.000000	500000.00000

2000 rows × 2 columns

In [5]:

```
# Plot Years of Experience Vs. Salary Using Ploltly Express
```

```
fig = px.scatter(salary_df, x = 'Years_of_Experience', y = 'Salary')
fig.show()
```



In [6]:

```
# Let's import another more advanced dataset entitled University admission (university_a

# GRE Scores (out of 340)
# TOEFL Scores (out of 120)
# University Rating (out of 5)
# Statement of Purpose (SOP)
# Letter of Recommendation (LOR) Strength (out of 5)
# Undergraduate GPA (out of 10)
# Research Experience (either 0 or 1)
# Chance of admission (ranging from 0 to 1)

admission_df = pd.read_csv(r"C:\Users\tashf\Desktop\Python Data Visualiazation with plot
admission_df
```

Out[6]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...

495	496	332	108	5	4.5	4.0	9.02	1	0.87
496	497	337	117	5	5.0	5.0	9.87	1	0.96
497	498	330	120	5	4.5	5.0	9.56	1	0.93
498	499	312	103	4	4.0	5.0	8.43	0	0.73
499	500	327	113	4	4.5	4.5	9.04	0	0.84

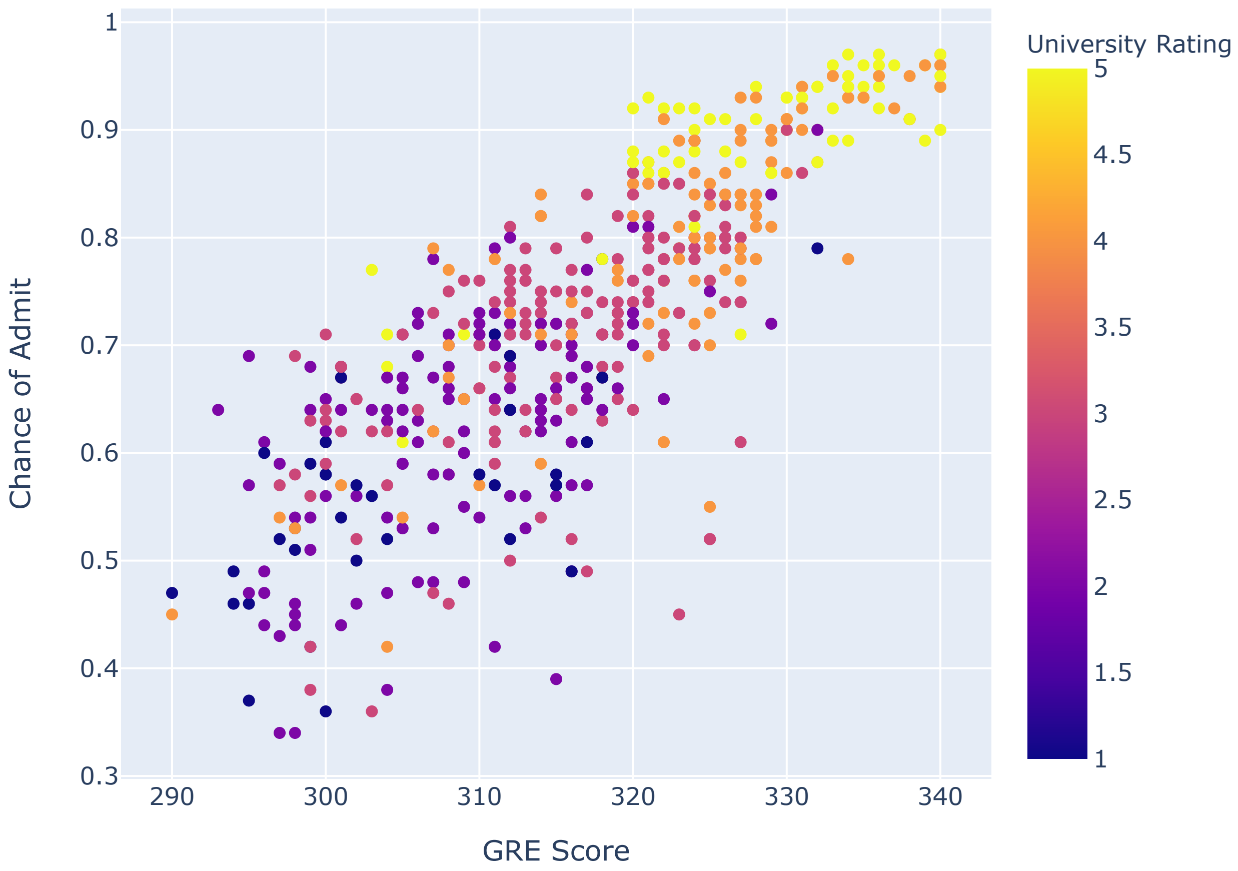
500 rows × 9 columns

MINI CHALLENGE #1:

- **Plot the scatter plot for GRE Score vs. chance of admission**
- **What do you infer from that plot?**
- **Use the color attribute to show the university rating as a third dimension**

In [7]:

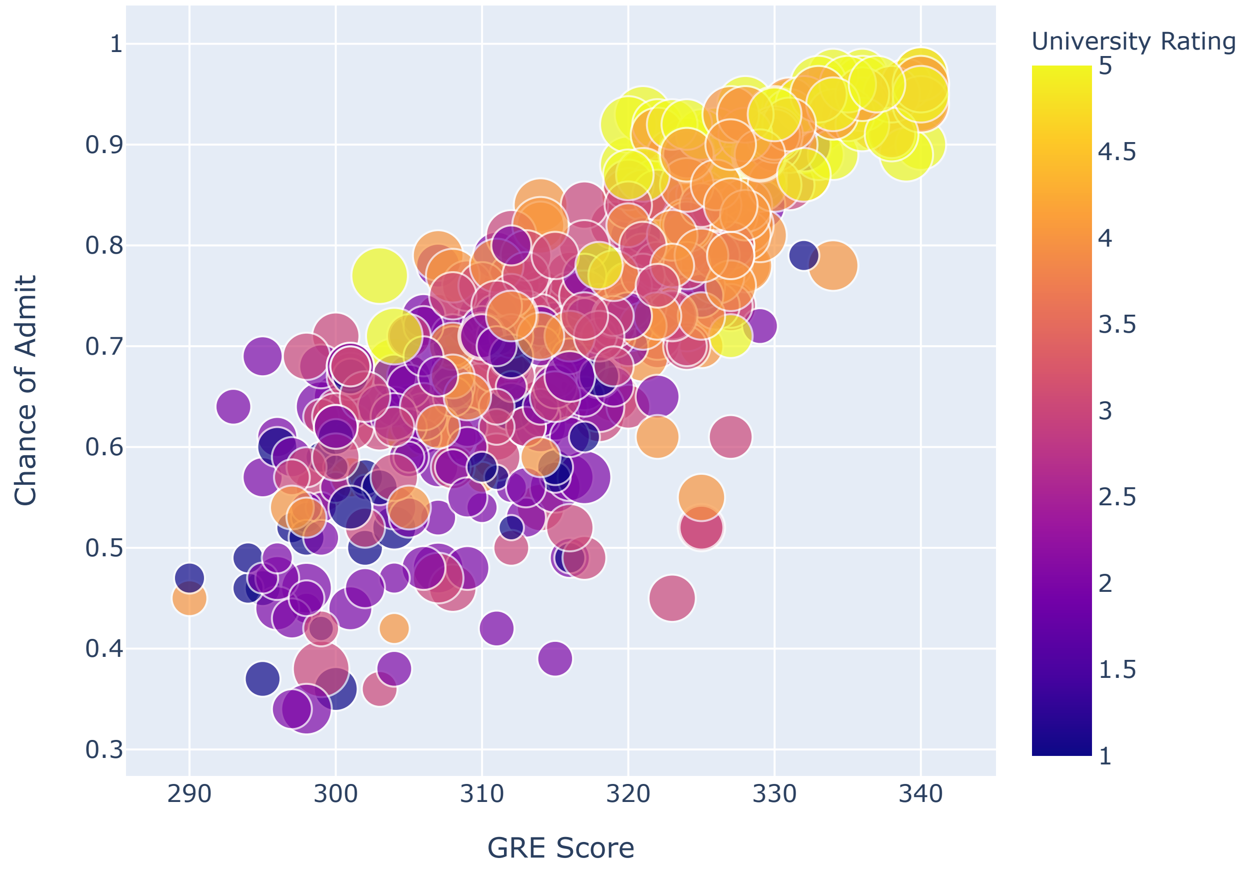
```
fig = px.scatter(admission_df, x = 'GRE Score', y ='Chance of Admit', color = 'Universit
fig.show()
```



TASK #2: PLOT INTERACTIVE BUBBLE CHART (SCATTERPLOT WITH SIZE)

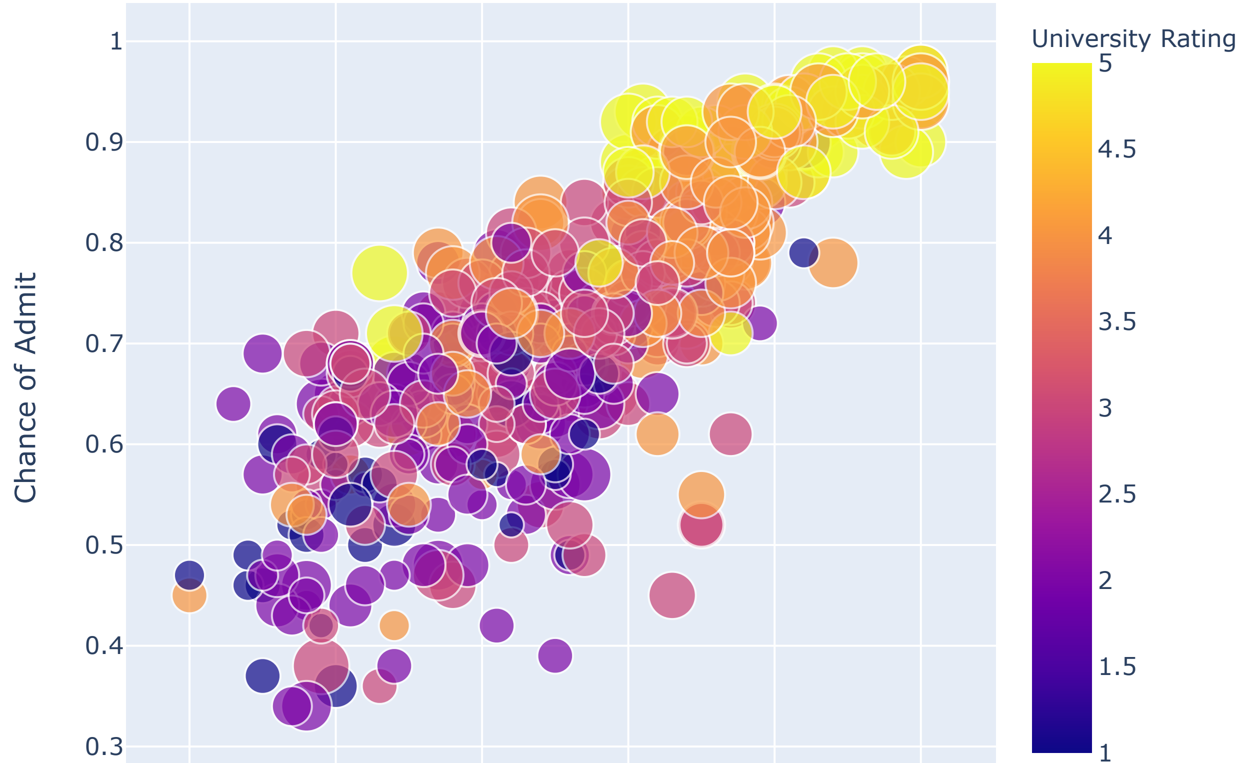
In [8]:

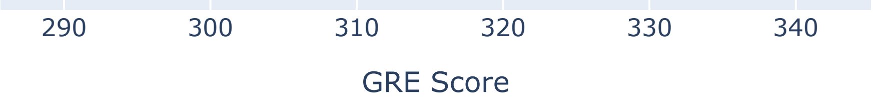
```
# Let's add a fourth variable "SOP" as the size
fig = px.scatter(admission_df, x = 'GRE Score', y ='Chance of Admit', color = 'Universit
fig.show()
```



In [9]:

```
# You can also add more data on hover using hover_data
fig = px.scatter(admission_df, x = 'GRE Score', y ='Chance of Admit', color = 'Universit
fig.show()
```

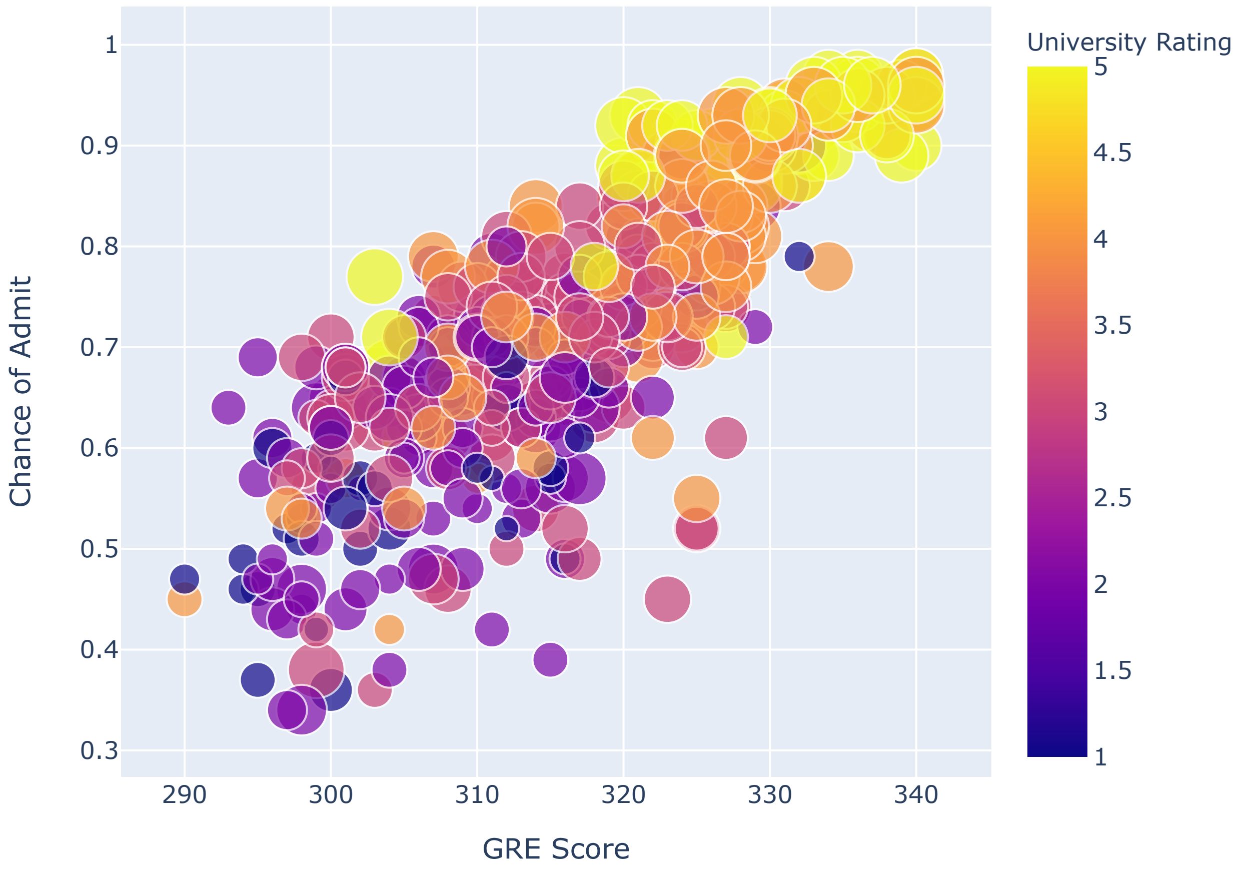




MINI CHALLENGE #2:

- **Modify the SOP column to make the bubble size variations more prominent**

In [10]:
```
fig = px.scatter(admission_df, x = 'GRE Score', y ='Chance of Admit', color = 'Universit
fig.show()
```



TASK #3: PLOT INTERACTIVE SINGLE LINEPLOT USING PLOTLY EXPRESS

In [11]:
```
# Import the crypto currency dataset
crypto_prices = pd.read_csv(r"C:\Users\tashf\Desktop\Python Data Visualiazation with plo
crypto_prices
```

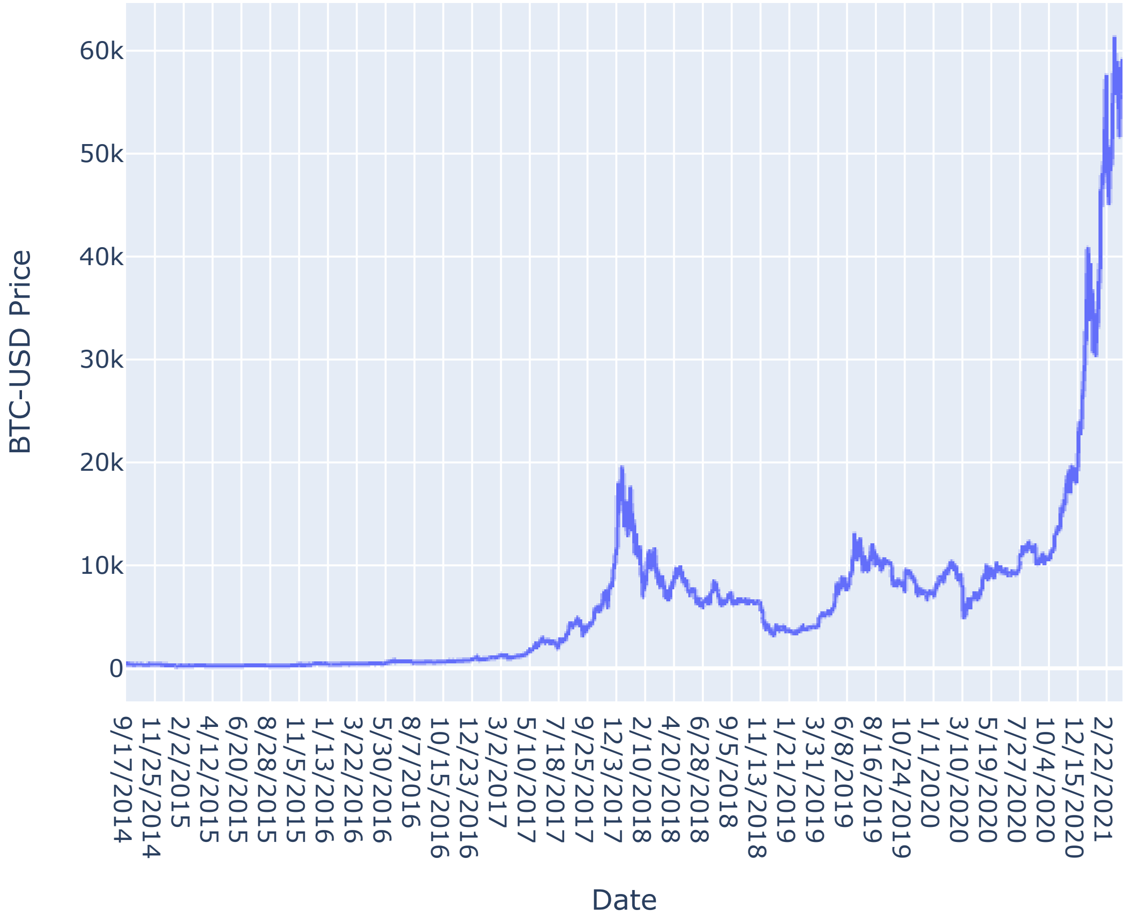
Out[11]:

	Date	BTC-USD Price	ETH-USD Price	LTC-USD Price
0	9/17/2014	457.334015	NaN	5.058550
1	9/18/2014	424.440002	NaN	4.685230
2	9/19/2014	394.795990	NaN	4.327770

3	9/20/2014	408.903992	NaN	4.286440
4	9/21/2014	398.821014	NaN	4.245920
...	...	...	...	...
2380	3/28/2021	55950.746090	1691.355957	185.028488
2381	3/29/2021	57750.199220	1819.684937	194.474777
2382	3/30/2021	58917.691410	1846.033691	196.682098
2383	3/31/2021	58918.832030	1918.362061	197.499100
2384	4/1/2021	59095.808590	1977.276855	204.112518

2385 rows × 4 columns

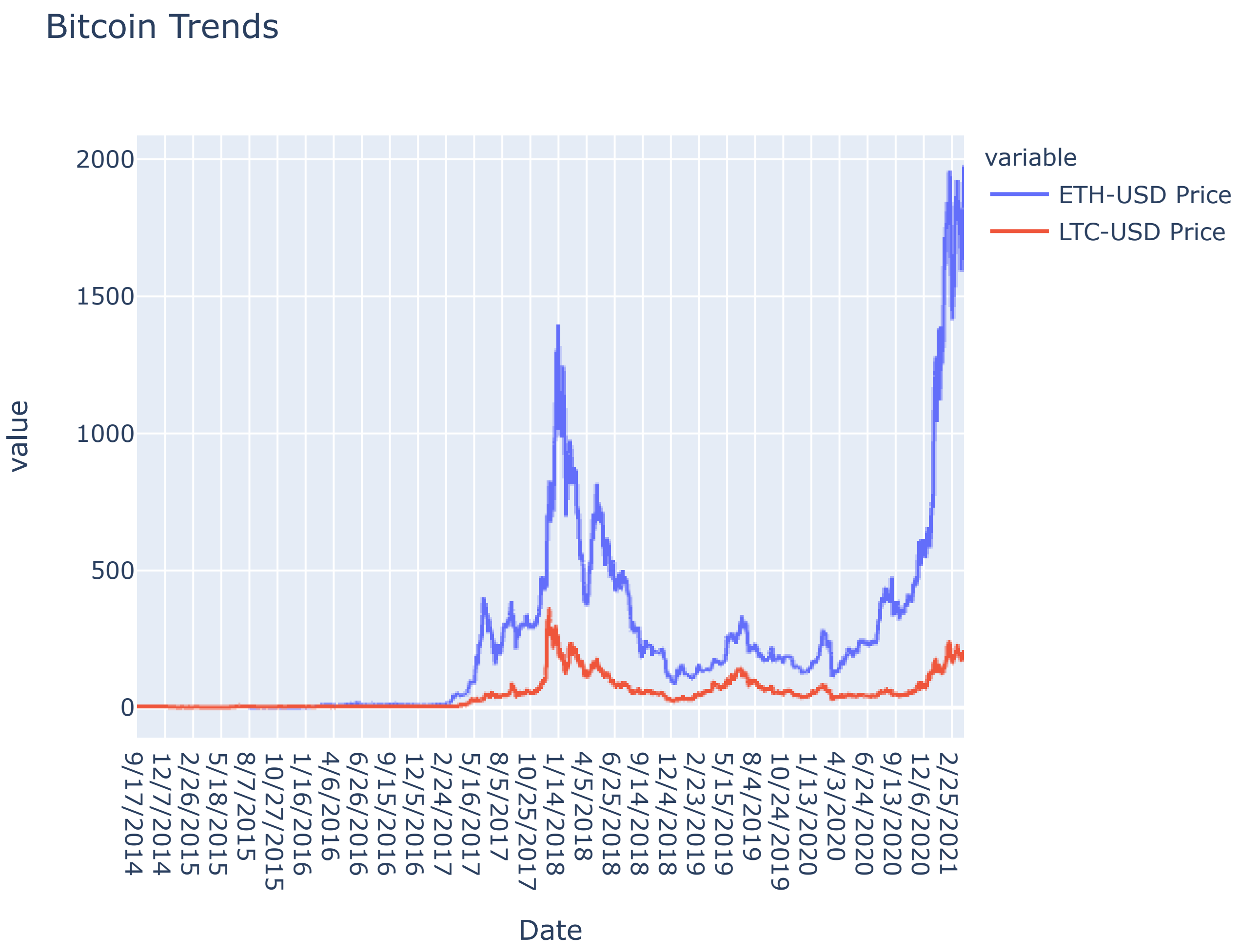
In [12]:
```python
fig = px.line(crypto_prices, x = 'Date', y = 'BTC-USD Price')
fig.show()
```



MINI CHALLENGE #3:

- **Plot interactive line plot for Ethereum and Litecoin.**
- **What is the maximum price of Bitcoin, Ethereum and Litecoin over the specified time period?**
- **Indicate the date when these peak prices took place**

In [13]:
```python
fig = px.line(crypto_prices, x = 'Date', y = ['ETH-USD Price','LTC-USD Price'], title =
fig.show()
```



TASK #4: PLOT INTERACTIVE MULTIPLE LINE PLOTS USING PLOTLY EXPRESS

In [14]:

```
crypto_prices
```

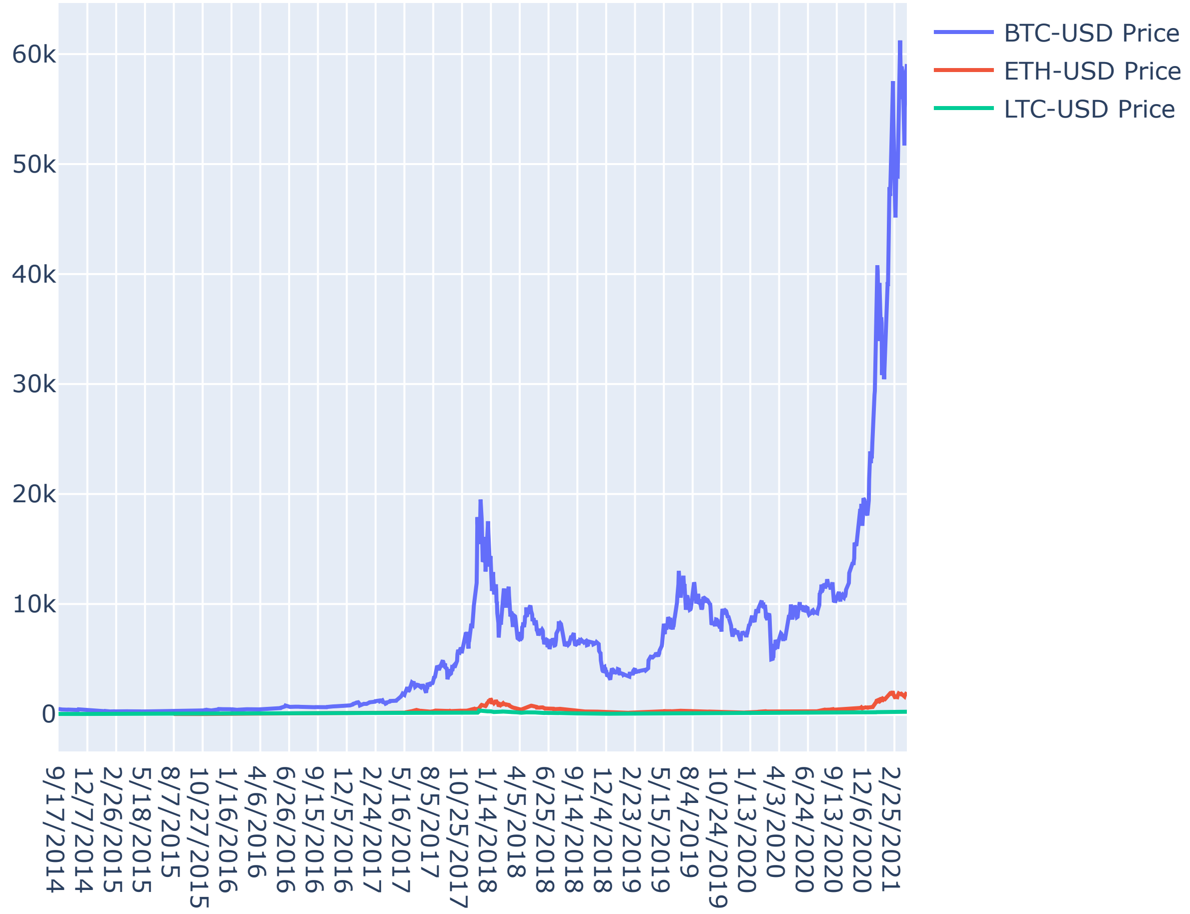
Out[14]:

	Date	BTC-USD Price	ETH-USD Price	LTC-USD Price
0	9/17/2014	457.334015	NaN	5.058550
1	9/18/2014	424.440002	NaN	4.685230
2	9/19/2014	394.795990	NaN	4.327770
3	9/20/2014	408.903992	NaN	4.286440
4	9/21/2014	398.821014	NaN	4.245920
...	...	...	...	...
2380	3/28/2021	55950.746090	1691.355957	185.028488
2381	3/29/2021	57750.199220	1819.684937	194.474777
2382	3/30/2021	58917.691410	1846.033691	196.682098
2383	3/31/2021	58918.832030	1918.362061	197.499100
2384	4/1/2021	59095.808590	1977.276855	204.112518

2385 rows × 4 columns

In [15]:

```
fig = px.line()
for i in crypto_prices.columns[1:]:
        fig.add_scatter(x =crypto_prices['Date'], y = crypto_prices[i], name = i)
fig.show()
```



MINI CHALLENGE #4:

- **Use "compare data on hover" feature to indicate the prices of LTC and ETH when BTC price peaked.**
- **Use Pandas operations to filter out to filter out the DataFrame and confirm your answers**

In [16]:

```
crypto_prices['BTC-USD Price'].max()
```

Out[16]:

```
61243.08594
```

In [17]:

```
crypto_prices[crypto_prices['BTC-USD Price'] == crypto_prices['BTC-USD Price'].max()]
```

Out[17]:

	Date	BTC-USD Price	ETH-USD Price	LTC-USD Price
2365	3/13/2021	61243.08594	1924.685425	226.578293

TASK #5. PLOT INTERACTIVE PIE CHARTS

In [18]:
```
# Define a dictionary with all crypto allocation in a portfolio
# Note that total summation = 100%
my_dict = {'allocation %' : [20, 20, 20, 20,20]}
my_dict
```

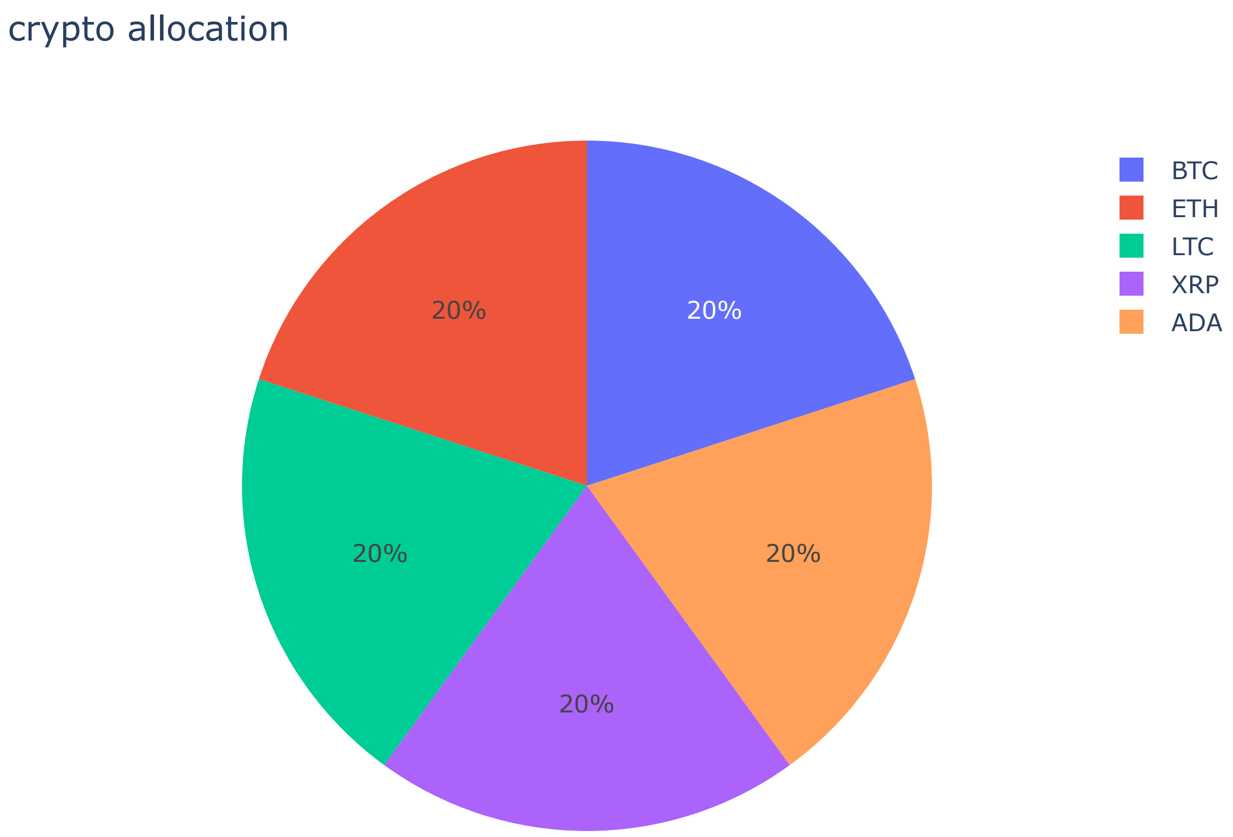
Out[18]:
```
{'allocation %': [20, 20, 20, 20, 20]}
```

In [19]:
```
crypto_df = pd.DataFrame(data = my_dict, index = ['BTC', 'ETH', 'LTC', 'XRP', 'ADA'])
crypto_df
```

Out[19]:

	allocation %
BTC	20
ETH	20
LTC	20
XRP	20
ADA	20

In [20]:
```
# Use Plotly Express to plot a pie chart
fig = px.pie(crypto_df, values= 'allocation %', names = ['BTC', 'ETH', 'LTC', 'XRP', 'AD
fig.show()
```



MINI CHALLENGE #5:

- **Assume that you became bullish on XRP and decided to allocate 60% of your assets in it. You also decided to equally divide the rest of your assets in other coins (BTC, LTC, ADA, and ETH). Change**

the allocations and plot the pie chart.

- **Use 'hole' attribute and see its impact on the pie chart (External Research is Required)**

In [21]:

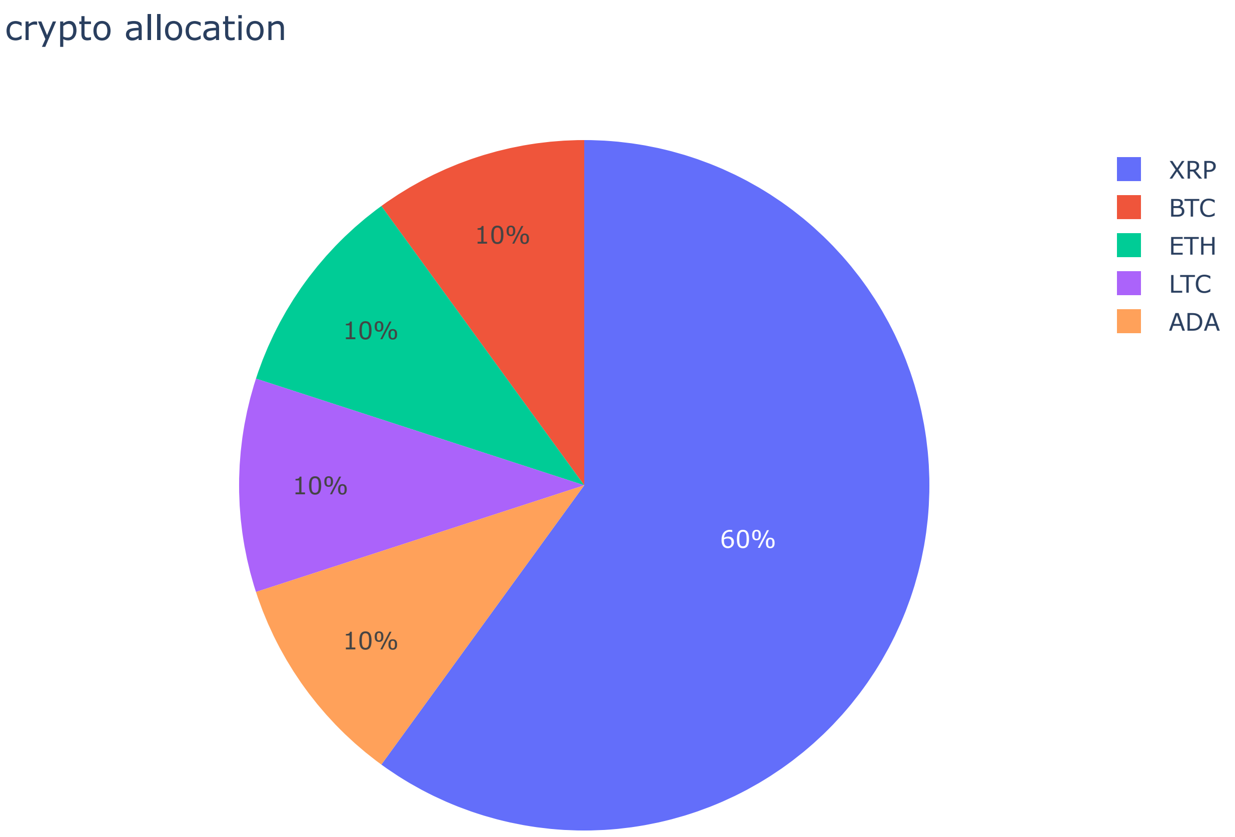
```
my_dict = {'allocation %' : [10, 10, 10, 60, 10]}
my_dict
```

Out[21]:

```
{'allocation %': [10, 10, 10, 60, 10]}
```

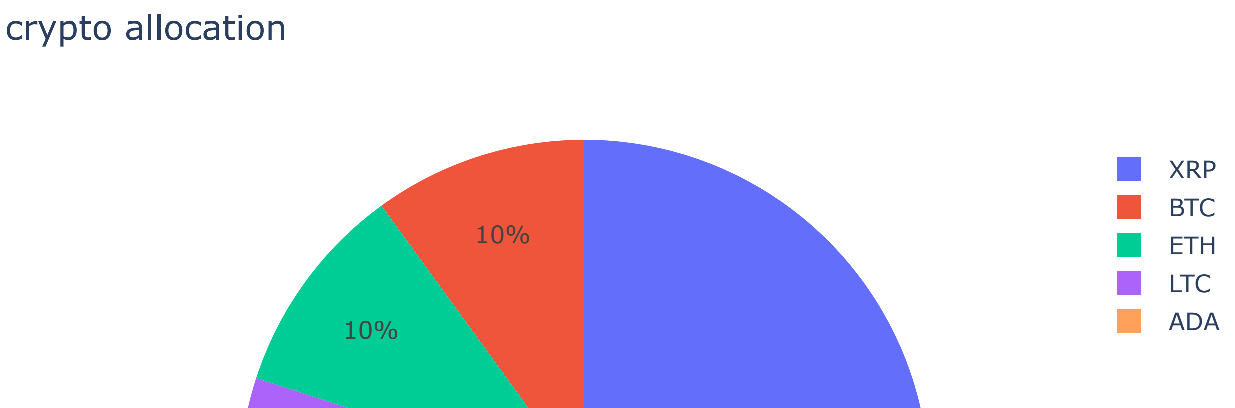
In [22]:

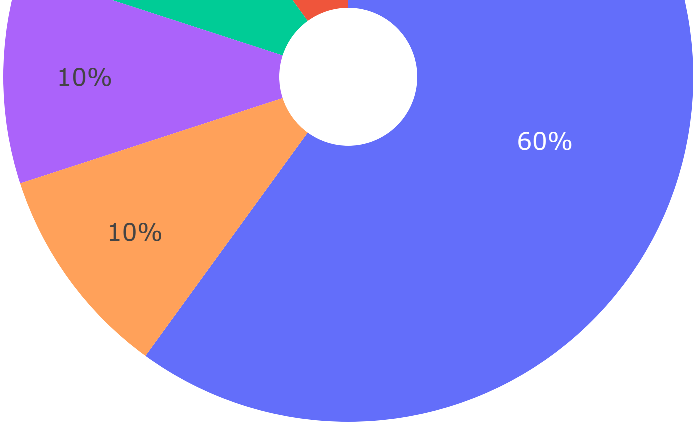
```
crypto_df = pd.DataFrame(data = my_dict, index = ['BTC', 'ETH', 'LTC', 'XRP', 'ADA'])
crypto_df
fig = px.pie(crypto_df, values= 'allocation %', names = ['BTC', 'ETH', 'LTC', 'XRP', 'AD
fig.show()
```



In [23]:

```
fig = px.pie(crypto_df, values= 'allocation %', names = ['BTC', 'ETH', 'LTC', 'XRP', 'AD
fig.show()
```





TASK #6: PLOT INTERACTIVE BAR CHART

In [24]:
```
# Gapminder combines data from multiple sources in a time-series format
# Check this out: https://www.gapminder.org/data/
data = px.data.gapminder()
data
```

Out[24]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4
...	...	...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306	ZWE	716
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786	ZWE	716
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960	ZWE	716
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623	ZWE	716
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

1704 rows × 8 columns

In [25]:
```
# Gapminder combines data from multiple sources in a time-series format
# Check this out: https://www.gapminder.org/data/
# You can read the data directly as follows: data = px.data.gapminder()

# Alternatively, you can import the data as follows:
canada_df = data[data.country == 'Canada']
canada_df
```

Out[25]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
240	Canada	Americas	1952	68.750	14785584	11367.16112	CAN	124

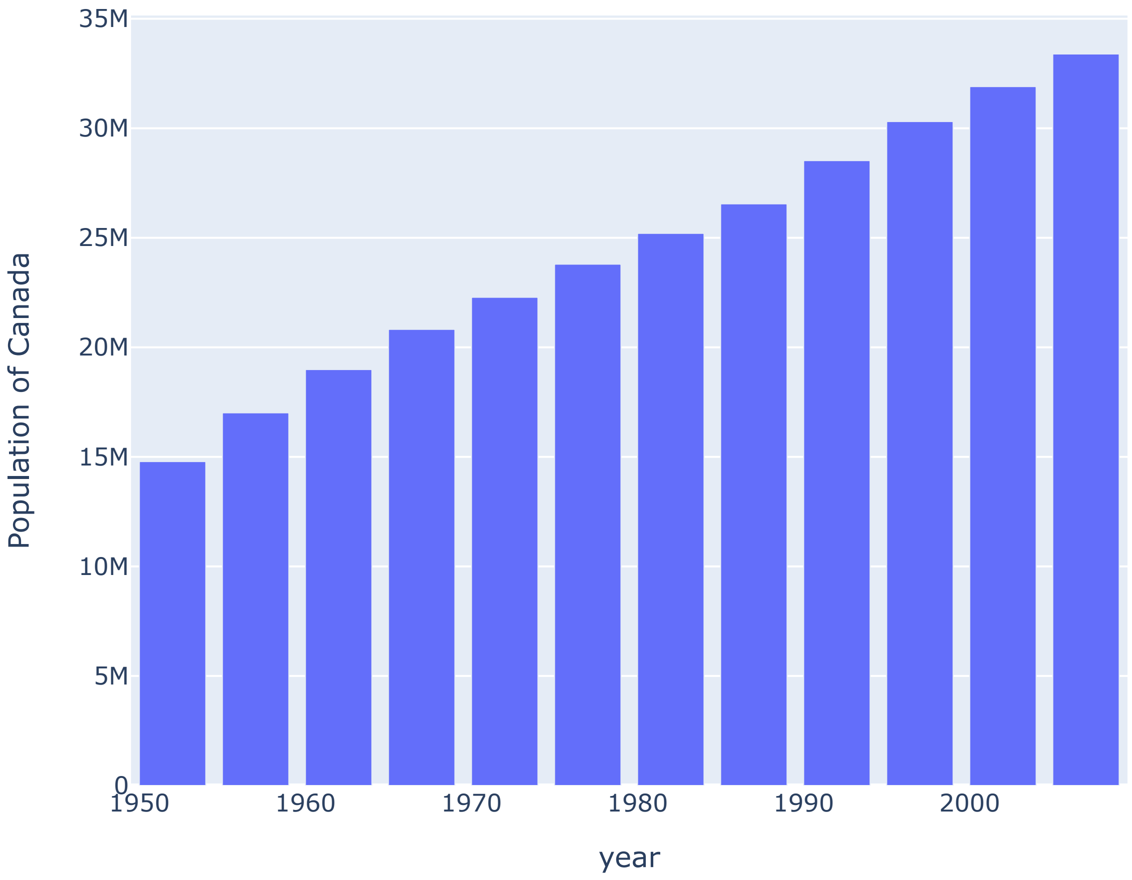
241	Canada	Americas	1957	69.960	17010154	12489.95006	CAN	124
242	Canada	Americas	1962	71.300	18985849	13462.48555	CAN	124
243	Canada	Americas	1967	72.130	20819767	16076.58803	CAN	124
244	Canada	Americas	1972	72.880	22284500	18970.57086	CAN	124
245	Canada	Americas	1977	74.210	23796400	22090.88306	CAN	124
246	Canada	Americas	1982	75.760	25201900	22898.79214	CAN	124
247	Canada	Americas	1987	76.860	26549700	26626.51503	CAN	124
248	Canada	Americas	1992	77.950	28523502	26342.88426	CAN	124
249	Canada	Americas	1997	78.610	30305843	28954.92589	CAN	124
250	Canada	Americas	2002	79.770	31902268	33328.96507	CAN	124
251	Canada	Americas	2007	80.653	33390141	36319.23501	CAN	124

In [26]:

```
# Filter the data based on the country of choice
fig = px.bar(canada_df, x='year', y = 'pop', labels = {'pop':'Population of Canada'})
```

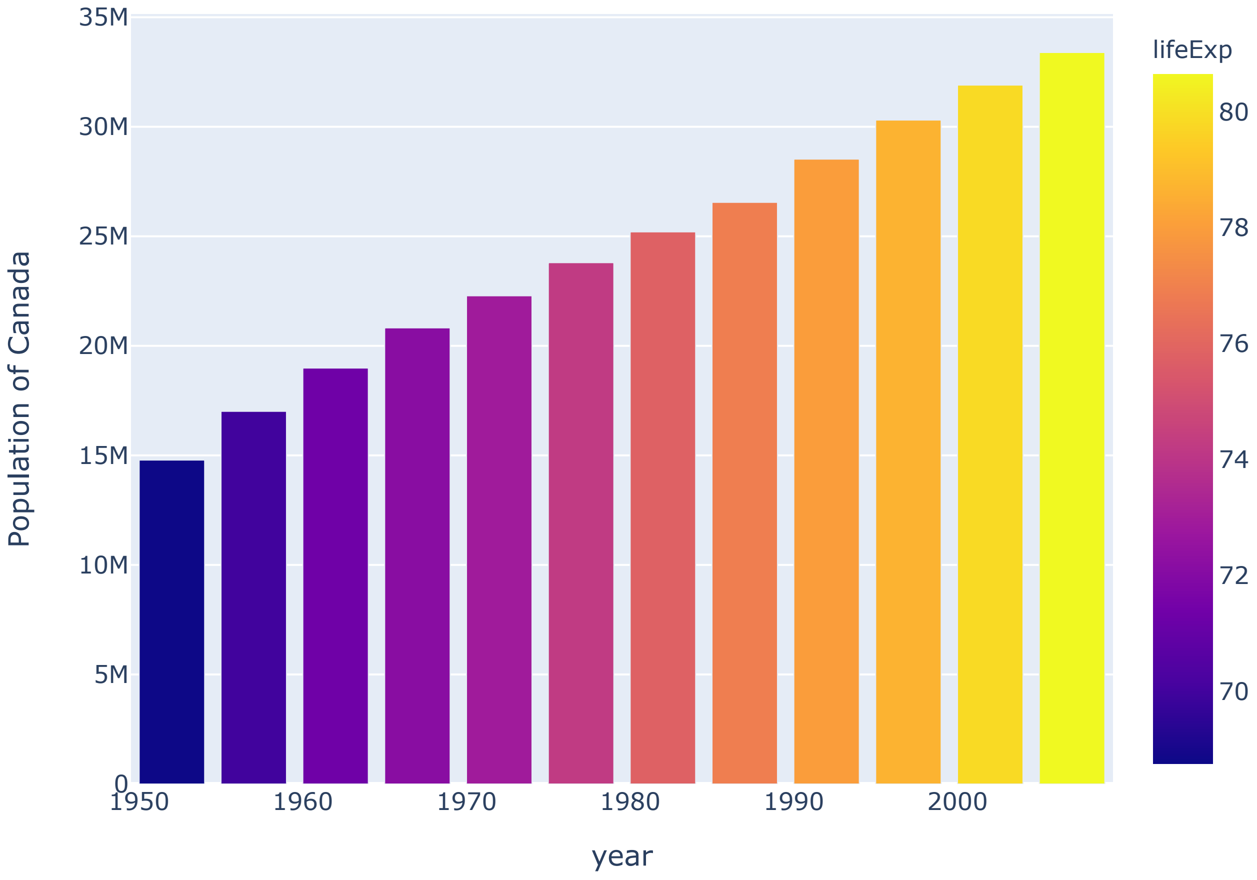
In [27]:

```
fig.show()
```



In [28]:

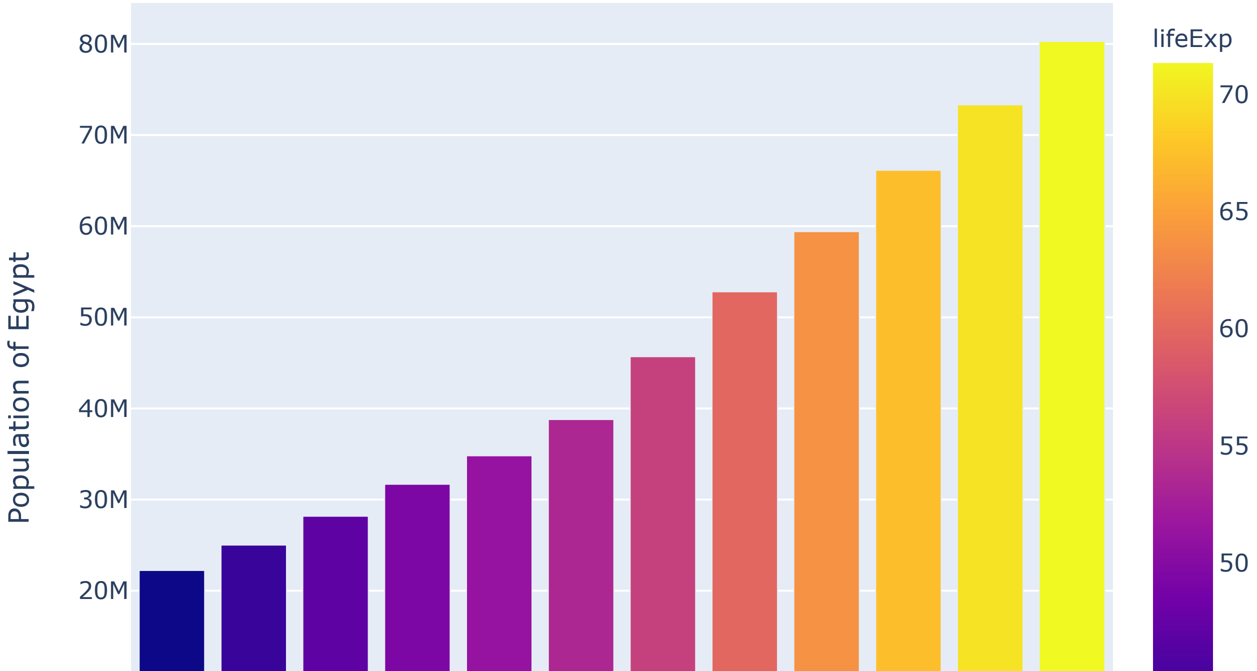
```
# You can add hoverdata and color (third dimension) as follows:
fig = px.bar(canada_df, x='year', y = 'pop', labels = {'pop':'Population of Canada'}, co
fig.show()
```

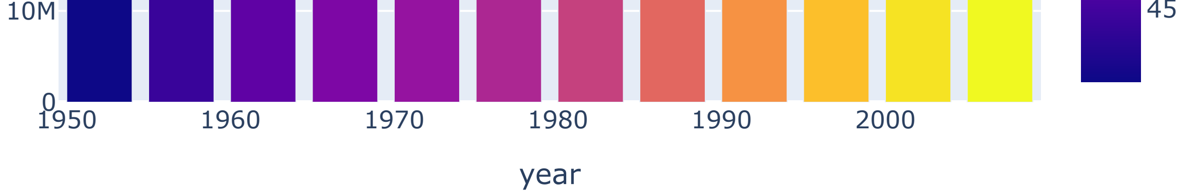


MINI CHALLENGE #6:

- **Plot similar plot for Egypt instead of Canada**

In [29]:
```
egypt_df = data[data.country == 'Egypt']

fig = px.bar(egypt_df, x='year', y = 'pop', labels = {'pop':'Population of Egypt'}, colo
fig.show()
```





TASK #7: PLOT INTERACTIVE GANTT CHART

In [30]:

```
# Define Job #1
job_1 = {'Task' : 'Develop Content', 'Start': '2020-10-10', 'Finish':'2021-01-01'}
job_1
```

Out[30]:

```
{'Task': 'Develop Content', 'Start': '2020-10-10', 'Finish': '2021-01-01'}
```

In [31]:

```
# Define Job #2
# Define Job #1
job_2 = {'Task' : 'Film Videos', 'Start': '2021-01-01', 'Finish':'2021-03-03'}
job_2
```

Out[31]:

```
{'Task': 'Film Videos', 'Start': '2021-01-01', 'Finish': '2021-03-03'}
```

In [32]:

```
# Define Job #3
# Define Job #1
job_3 = {'Task' : 'Edit Video', 'Start': '2021-03-06', 'Finish':'2021-04-04'}
job_3
```

Out[32]:

```
{'Task': 'Edit Video', 'Start': '2021-03-06', 'Finish': '2021-04-04'}
```

In [33]:

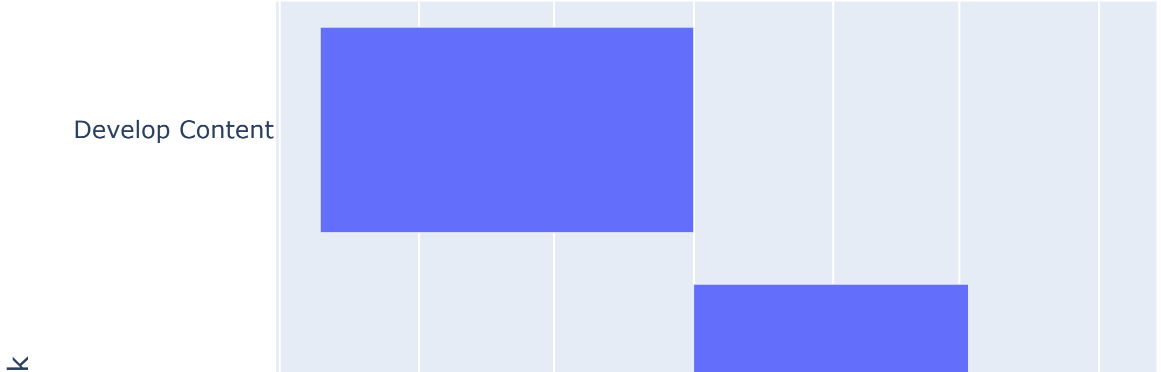
```
project_df = pd.DataFrame([job_1,job_2,job_3])
project_df
```

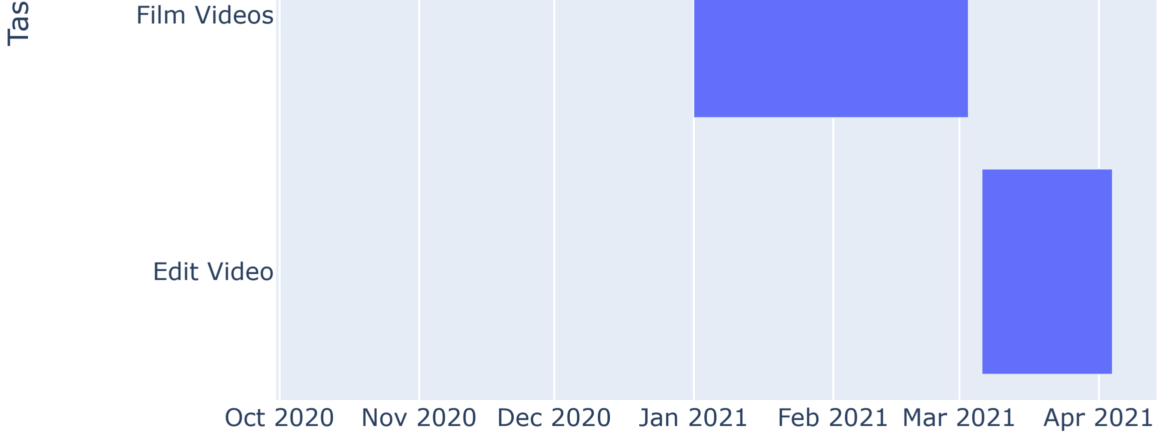
Out[33]:

	Task	Start	Finish
0	Develop Content	2020-10-10	2021-01-01
1	Film Videos	2021-01-01	2021-03-03
2	Edit Video	2021-03-06	2021-04-04

In [34]:

```
fig = px.timeline(project_df, x_start = 'Start', x_end = 'Finish', y = 'Task')
fig.update_yaxes(autorange = 'reversed')
fig.show()
```



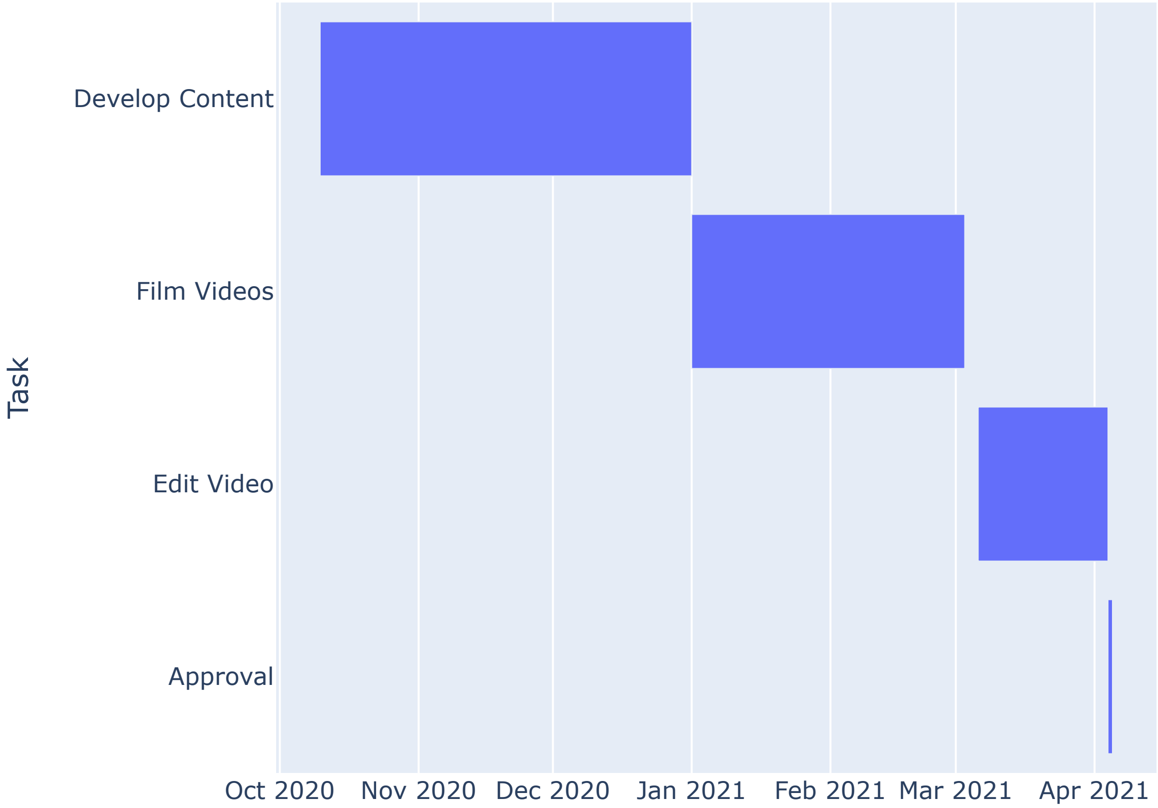


MINI CHALLENGE #7:

- **Add an additional task of "Send the course for approval" that lasts for 1 day and that starts immediately after "Edit Videos & Upload Content"**

In [35]:

```
job_4 = {'Task' : 'Approval', 'Start': '2021-04-04', 'Finish':'2021-04-05'}

project_df = pd.DataFrame([job_1,job_2,job_3, job_4])
fig = px.timeline(project_df, x_start = 'Start', x_end = 'Finish', y = 'Task')
fig.update_yaxes(autorange = 'reversed')
fig.show()
```



TASK #8: PLOT INTERACTIVE SUNBURST

In [36]:

```
# A sunburst plot represents hierarchial data as sectors laid out over several levels of

restaurant_df = pd.read_csv(r"C:\Users\tashf\Desktop\Python Data Visualiazation with plo
```

In [37]:

```
restaurant_df
```

Out[37]:

	Unnamed: 0	total_bill	tip	sex	smoker	day	time	size
0	0	16.99	1.01	Female	No	Sun	Dinner	2
1	1	10.34	1.66	Male	No	Sun	Dinner	3
2	2	21.01	3.50	Male	No	Sun	Dinner	3
3	3	23.68	3.31	Male	No	Sun	Dinner	2
4	4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...	...
239	239	29.03	5.92	Male	No	Sat	Dinner	3
240	240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	242	17.82	1.75	Male	No	Sat	Dinner	2
243	243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 8 columns

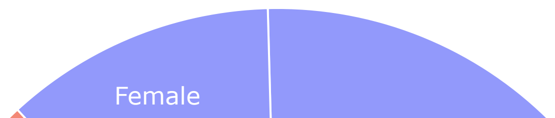
MINI CHALLENGE #8:

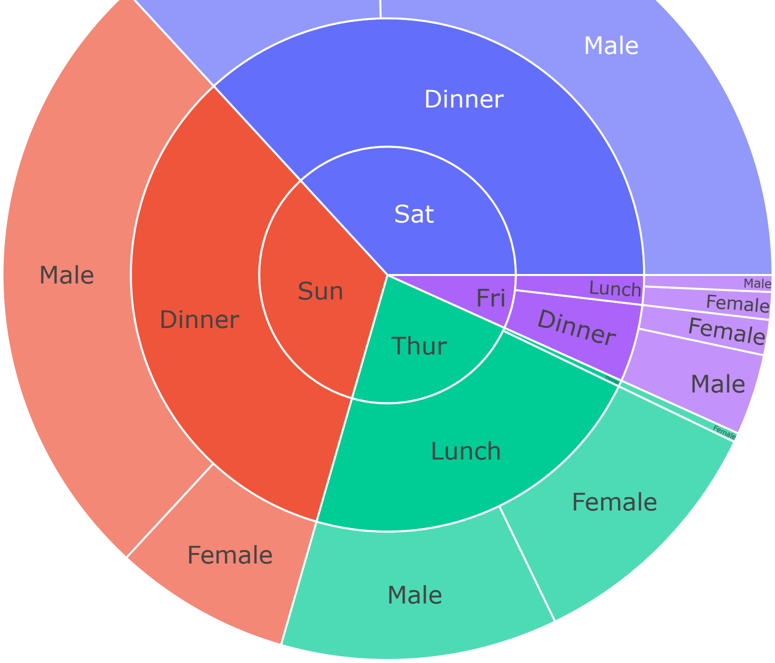
- **Import the full restaurant dataset "restaurant.csv"**
- **Plot the sunburst plot using the following: path = [day, time, sex] and value = [total bill]**

In [38]:

```
fig = px.sunburst(restaurant_df, path = ['day', 'time', 'sex'], values = 'total_bill')
fig.show()
```

```
C:\Users\tashf\anaconda3\lib\site-packages\plotly\express\_core.py:1594: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future versio
n. Use pandas.concat instead.

C:\Users\tashf\anaconda3\lib\site-packages\plotly\express\_core.py:1594: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future versio
n. Use pandas.concat instead.

C:\Users\tashf\anaconda3\lib\site-packages\plotly\express\_core.py:1594: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future versio
n. Use pandas.concat instead.
```





In [42]:

```
!pip install nbconvert pip install pyppeteer

!pip install jinja2==3.0.3
```

```
Requirement already satisfied: nbconvert in c:\users\tashf\anaconda3\lib\site-packages
(6.4.4)
Requirement already satisfied: pip in c:\users\tashf\anaconda3\lib\site-packages (22.2.
2)
Requirement already satisfied: install in c:\users\tashf\anaconda3\lib\site-packages (1.
3.5)
Requirement already satisfied: pyppeteer in c:\users\tashf\anaconda3\lib\site-packages
(1.0.2)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\tashf\anaconda3\lib\site
-packages (from nbconvert) (1.5.0)
Requirement already satisfied: beautifulsoup4 in c:\users\tashf\anaconda3\lib\site-packa
ges (from nbconvert) (4.11.1)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\tashf\anaconda3\lib\site-pa
ckages (from nbconvert) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\tashf\anaconda3\lib\site-p
ackages (from nbconvert) (0.4)
Requirement already satisfied: bleach in c:\users\tashf\anaconda3\lib\site-packages (fro
m nbconvert) (4.1.0)
Requirement already satisfied: jinja2>=2.4 in c:\users\tashf\anaconda3\lib\site-packages
(from nbconvert) (2.11.3)
Requirement already satisfied: testpath in c:\users\tashf\anaconda3\lib\site-packages (f
rom nbconvert) (0.6.0)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\tashf\anaconda3\lib\si
te-packages (from nbconvert) (0.5.13)
Requirement already satisfied: nbformat>=4.4 in c:\users\tashf\anaconda3\lib\site-packag
es (from nbconvert) (5.5.0)
Requirement already satisfied: jupyterlab-pygments in c:\users\tashf\anaconda3\lib\site-
packages (from nbconvert) (0.1.2)
Requirement already satisfied: traitlets>=5.0 in c:\users\tashf\anaconda3\lib\site-packa
ges (from nbconvert) (5.1.1)
Requirement already satisfied: defusedxml in c:\users\tashf\anaconda3\lib\site-packages
(from nbconvert) (0.7.1)
Requirement already satisfied: jupyter-core in c:\users\tashf\anaconda3\lib\site-package
s (from nbconvert) (4.11.1)
Requirement already satisfied: pygments>=2.4.1 in c:\users\tashf\anaconda3\lib\site-pack
```

```
ages (from nbconvert) (2.11.2)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\tashf\anaconda3\lib\sit
e-packages (from pyppeteer) (1.4.4)
Requirement already satisfied: pyee<9.0.0,>=8.1.0 in c:\users\tashf\anaconda3\lib\site-p
ackages (from pyppeteer) (8.2.2)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\tashf\anaconda3\lib\site-
packages (from pyppeteer) (4.64.1)
Requirement already satisfied: certifi>=2021 in c:\users\tashf\anaconda3\lib\site-packag
es (from pyppeteer) (2022.12.7)
Requirement already satisfied: websockets<11.0,>=10.0 in c:\users\tashf\anaconda3\lib\si
te-packages (from pyppeteer) (10.4)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\tashf\anaconda3\lib\s
ite-packages (from pyppeteer) (4.11.3)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\tashf\anaconda3\lib\si
te-packages (from pyppeteer) (1.26.11)
Requirement already satisfied: zipp>=0.5 in c:\users\tashf\anaconda3\lib\site-packages
(from importlib-metadata>=1.4->pyppeteer) (3.8.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\tashf\anaconda3\lib\site-pac
kages (from jinja2>=2.4->nbconvert) (2.0.1)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\users\tashf\anaconda3\lib\sit
e-packages (from nbclient<0.6.0,>=0.5.0->nbconvert) (7.3.4)
Requirement already satisfied: nest-asyncio in c:\users\tashf\anaconda3\lib\site-package
s (from nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.5)
Requirement already satisfied: jsonschema>=2.6 in c:\users\tashf\anaconda3\lib\site-pack
ages (from nbformat>=4.4->nbconvert) (4.16.0)
Requirement already satisfied: fastjsonschema in c:\users\tashf\anaconda3\lib\site-packa
ges (from nbformat>=4.4->nbconvert) (2.16.2)
Requirement already satisfied: colorama in c:\users\tashf\anaconda3\lib\site-packages (f
rom tqdm<5.0.0,>=4.42.1->pyppeteer) (0.4.5)
Requirement already satisfied: soupsieve>1.2 in c:\users\tashf\anaconda3\lib\site-packag
es (from beautifulsoup4->nbconvert) (2.3.1)
Requirement already satisfied: six>=1.9.0 in c:\users\tashf\anaconda3\lib\site-packages
(from bleach->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in c:\users\tashf\anaconda3\lib\site-package
s (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: packaging in c:\users\tashf\anaconda3\lib\site-packages
(from bleach->nbconvert) (21.3)
Requirement already satisfied: pywin32>=1.0 in c:\users\tashf\anaconda3\lib\site-package
s (from jupyter-core->nbconvert) (302)
Requirement already satisfied: attrs>=17.4.0 in c:\users\tashf\anaconda3\lib\site-packag
es (from jsonschema>=2.6->nbformat>=4.4->nbconvert) (21.4.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users
\tashf\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=4.4->nbconvert) (0.1
8.0)
Requirement already satisfied: pyzmq>=23.0 in c:\users\tashf\anaconda3\lib\site-packages
(from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (23.2.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\tashf\anaconda3\lib\si
te-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: tornado>=6.0 in c:\users\tashf\anaconda3\lib\site-package
s (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\tashf\anaconda3\lib
\site-packages (from packaging->bleach->nbconvert) (3.0.9)
Collecting jinja2==3.0.3
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
     ------------------------------------ 133.6/133.6 kB 785.9 kB/s eta 0:00:00
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\tashf\anaconda3\lib\site-pack
ages (from jinja2==3.0.3) (2.0.1)
Installing collected packages: jinja2
  Attempting uninstall: jinja2
    Found existing installation: Jinja2 2.11.3
    Uninstalling Jinja2-2.11.3:
      Successfully uninstalled Jinja2-2.11.3
Successfully installed jinja2-3.0.3
```

```
In [ ]:
```