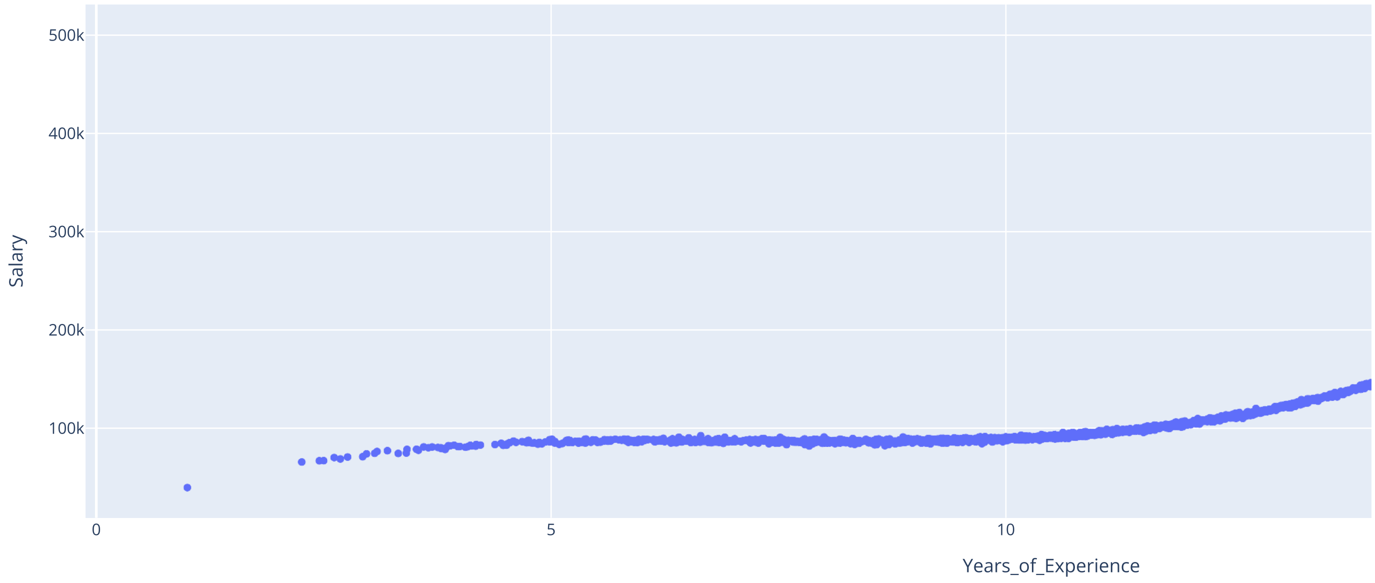
Experiments with Plotly

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

```
salary_df = pd.read_csv('/content/employee_salaries.csv')
salary_df
```

	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

```
#Plot years of experience vs Salary
fig = px.scatter(salary_df, x='Years_of_Experience', y='Salary')
fig.show()
```

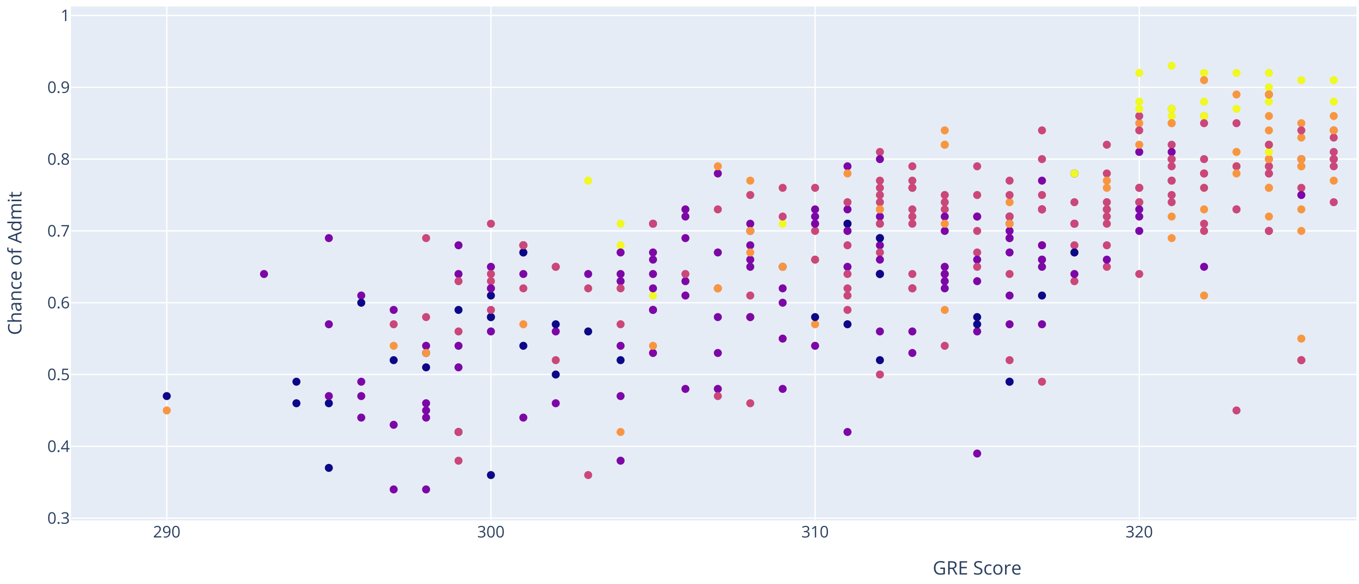


```
admission_df = pd.read_csv('/content/university_admission.csv')
admission_df
```

	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

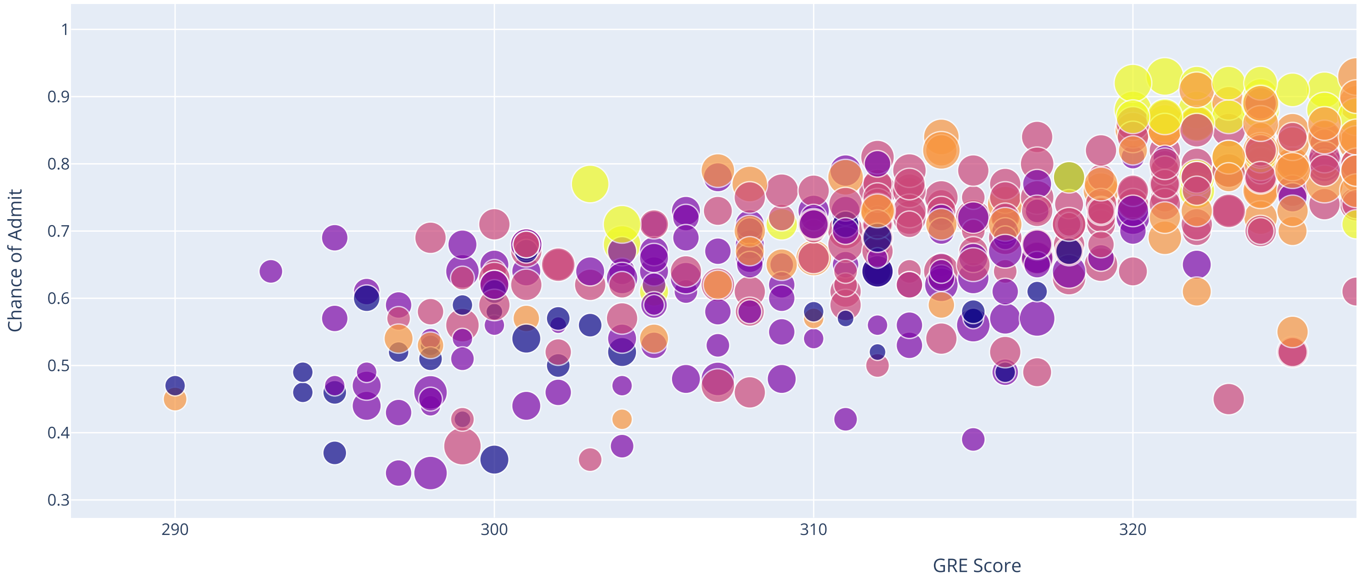
500 rows × 9 columns

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

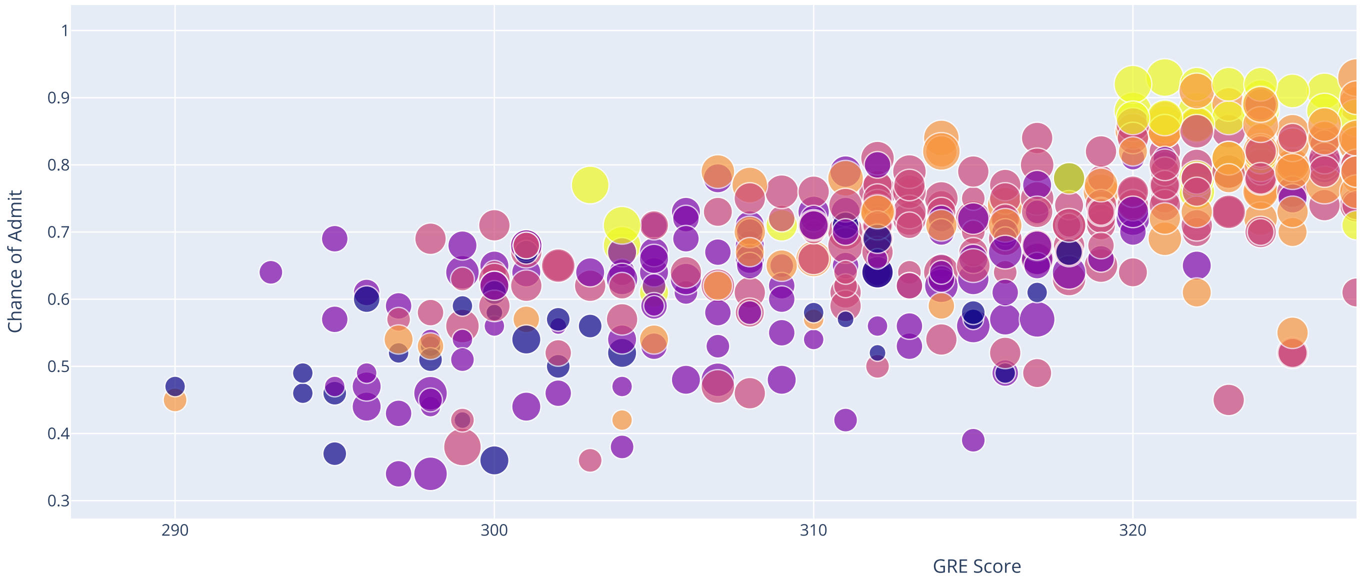


Interactive Bubble Chart

```
#lets plot the bubble plot with size varying wit the SOP
fig = px.scatter(admission_df, x='GRE Score', y='Chance of Admit', color='University Rating', size='SOP')
fig.show()
```



```
#Adding additional hover data "LOR"
fig = px.scatter(admission_df, x='GRE Score', y='Chance of Admit', color='University Rating', size='SOP', hover_data='LOR')
fig.show()
```



```
#crypto prices
crypto_df = pd.read_csv('/content/crypto_prices.csv')
crypto_df
```

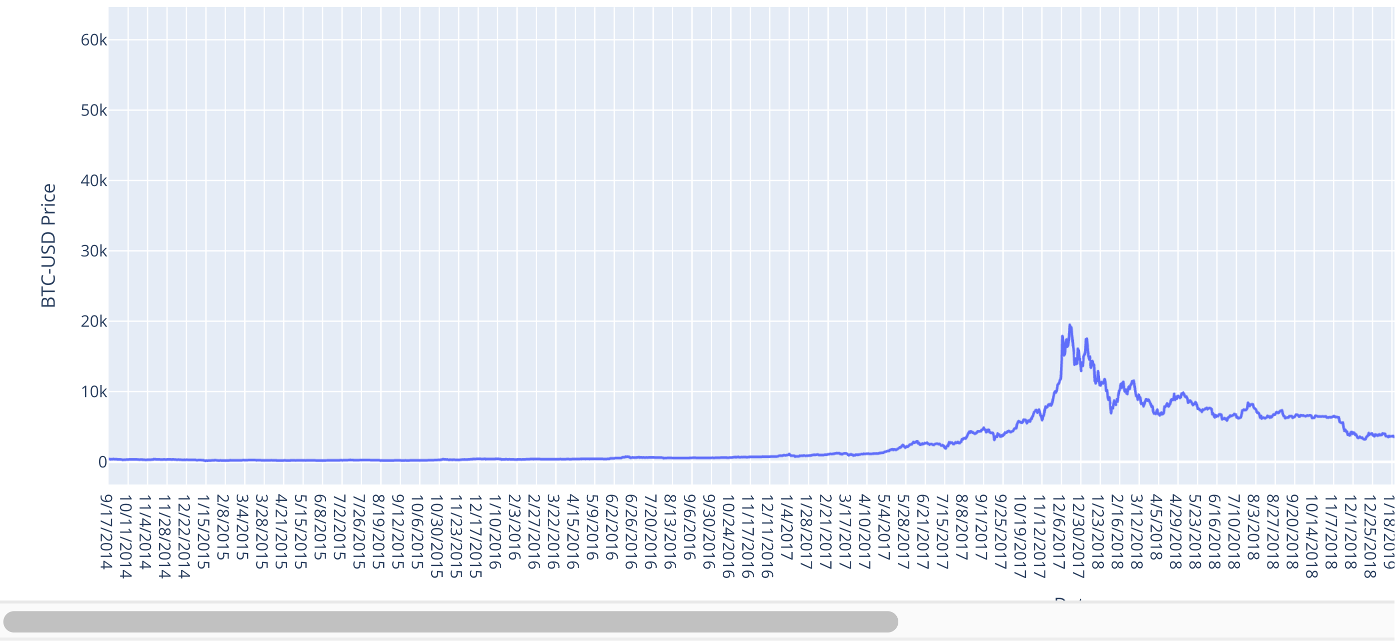
	Date	BTC-USD Price	ETH-USD Price	LTC-USD Price
0	9/17/2014	457.334015	NaN	5.058550
1	9/18/2014	424.440002	NaN	4.685230
2	9/19/2014	394.795990	NaN	4.327770
3	9/20/2014	408.903992	NaN	4.286440
4	9/21/2014	398.821014	NaN	4.245920
...	...	...	...	...
2380	3/28/2021	55950.746090	1691.355957	185.028488
2381	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

```
#show all the columns in crypto_df
crypto_df.columns
```

```
Index(['Date', 'BTC-USD Price', 'ETH-USD Price', 'LTC-USD Price'], dtype='object')
```

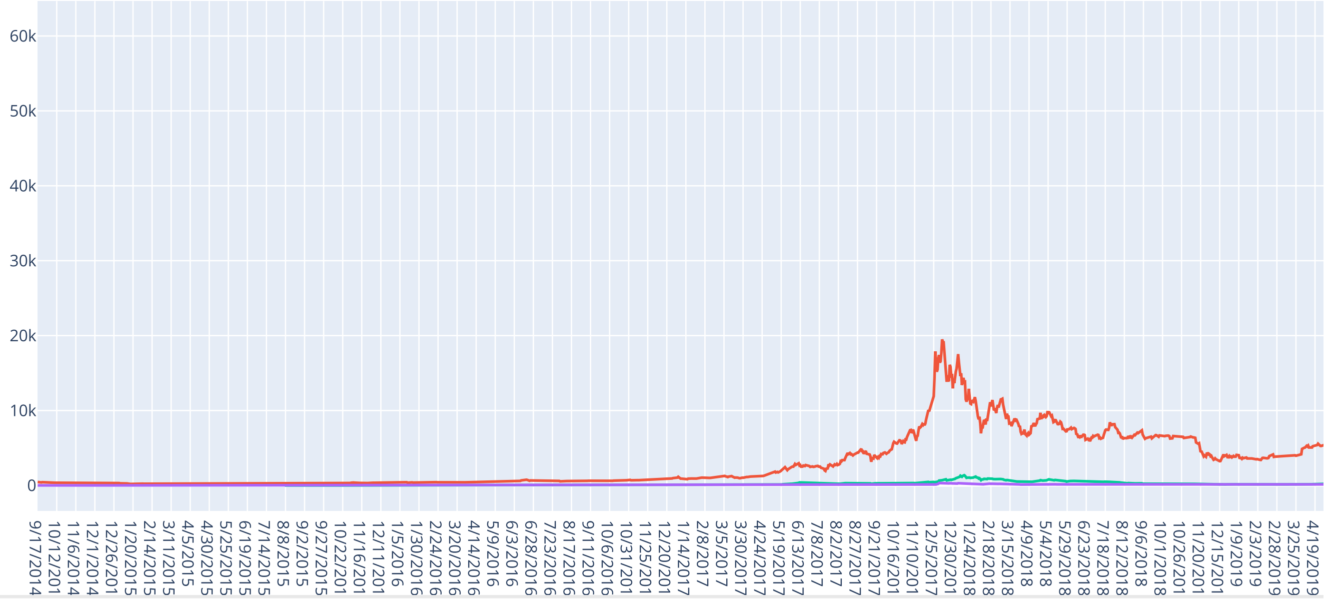
```
#Line plot
fig = px.line(crypto_df, x='Date', y='BTC-USD Price')
fig.show()
```



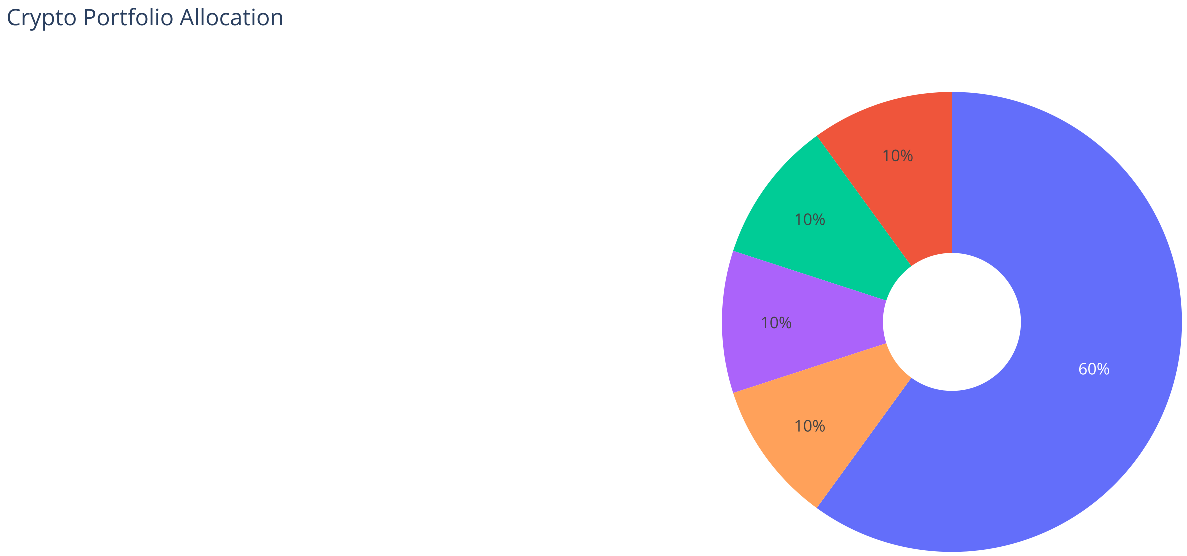
```
crypto_df[crypto_df['BTC-USD Price'] == crypto_df['BTC-USD Price'].max()]

crypto_df[crypto_df['Date'] == '3/13/2021']
```

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

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



```
my_dict =  {'allocation %': [10, 10, 10, 60, 10]}
# explode = (0, 0, 0, 0.2, 0)

crypto_df_allocate = pd.DataFrame(data = my_dict, index = ['BTC', 'ETH', 'LTC', 'XRP', 'ADA'])

# Use Plotly Express to plot a pie chart
fig = px.pie(crypto_df_allocate, values = 'allocation %', names = ['BTC', 'ETH', 'LTC', 'XRP', 'ADA'], title = 'Crypto Portfolic
fig.show()
```



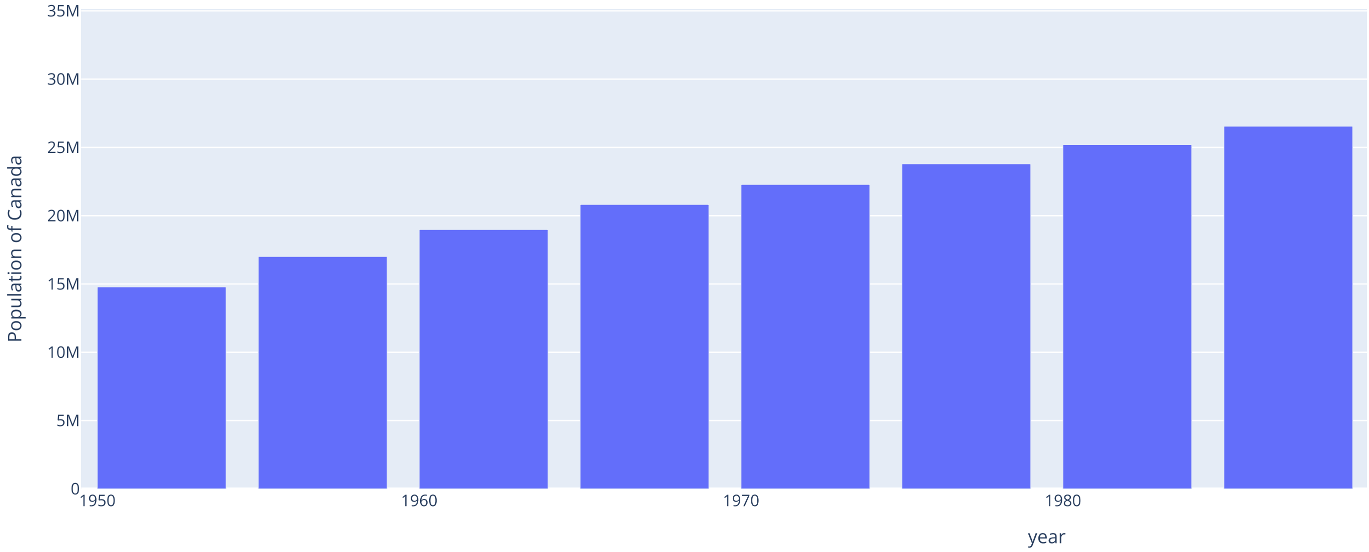
```
data = pd.read_csv('/content/gapminder.csv')
data
```

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

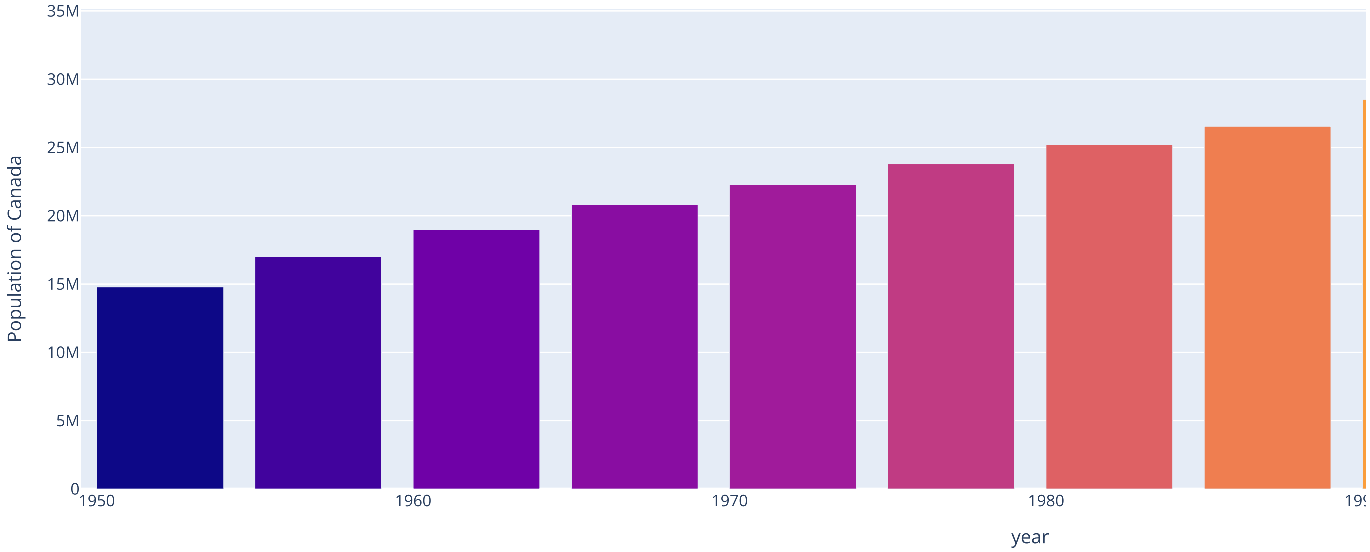
```
canada_df=data[data['country'] == 'Canada']
canada_df
```

	Unnamed: 0	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
240	240	Canada	Americas	1952	68.750	14785584	11367.16112	CAN	124
241	241	Canada	Americas	1957	69.960	17010154	12489.95006	CAN	124
242	242	Canada	Americas	1962	71.300	18985849	13462.48555	CAN	124
243	243	Canada	Americas	1967	72.130	20819767	16076.58803	CAN	124
244	244	Canada	Americas	1972	72.880	22284500	18970.57086	CAN	124
245	245	Canada	Americas	1977	74.210	23796400	22090.88306	CAN	124
246	246	Canada	Americas	1982	75.760	25201900	22898.79214	CAN	124
247	247	Canada	Americas	1987	76.860	26549700	26626.51503	CAN	124
248	248	Canada	Americas	1992	77.950	28523502	26342.88426	CAN	124
249	249	Canada	Americas	1997	78.610	30305843	28954.92589	CAN	124
250	250	Canada	Americas	2002	79.770	31902268	33328.96507	CAN	124

```
#Bar chart to show population growth in Canada
fig = px.bar(canada_df, x='year', y='pop', labels={'pop':'Population of Canada'}, height=500)
fig.show()
```



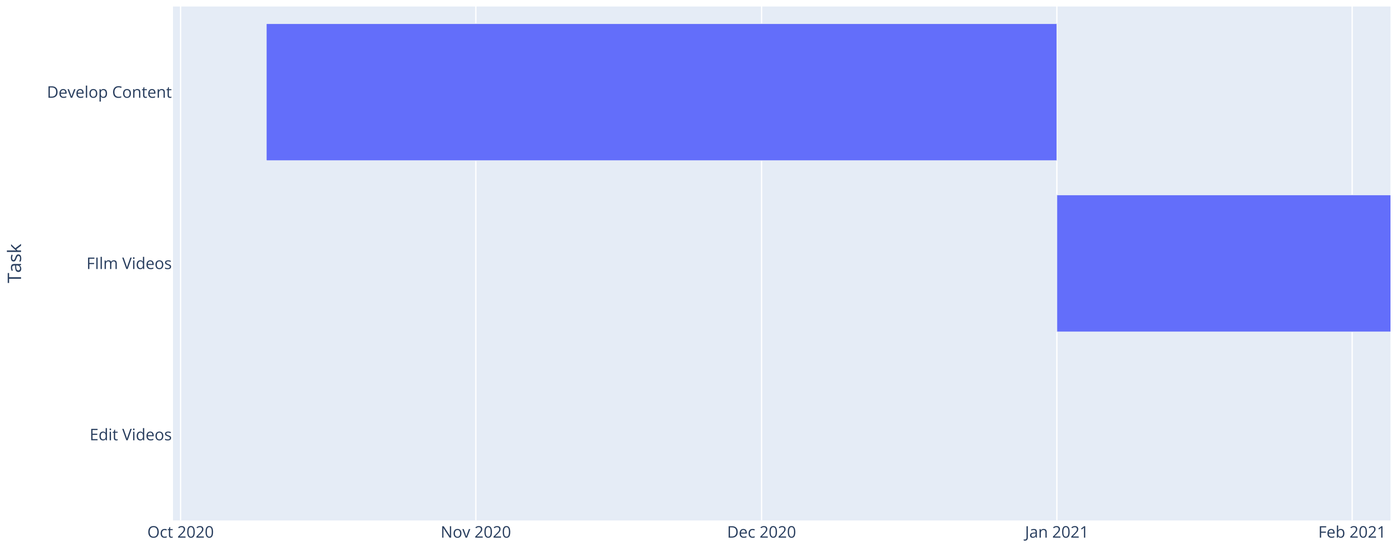
```
#Adding color and hover data
fig = px.bar(canada_df, x='year', y='pop', labels={'pop':'Population of Canada'}, height=500, color='lifeExp', hover_data=['gdpP
fig.show()
```



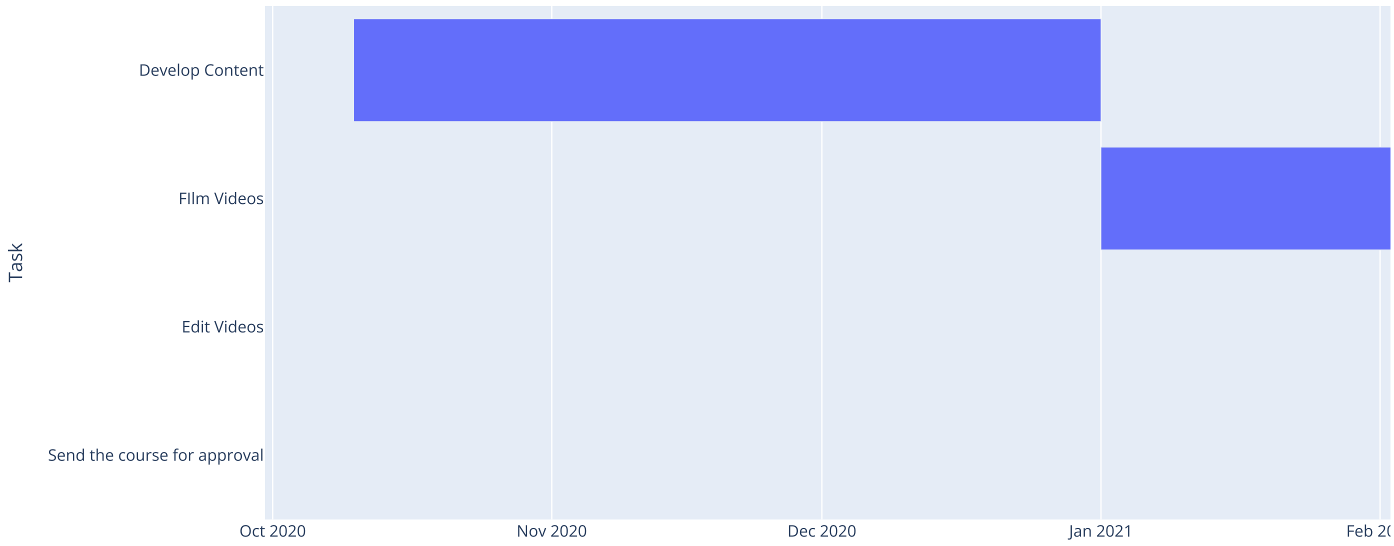
```
#Interactive Gantt Chart
job_1={'Task':'Develop Content', 'Start':'2020-10-10','Finish':'2021-01-01'}

job_2={'Task':'FIlm Videos', 'Start':'2021-01-01','Finish':'2021-03-03'}

job_3={'Task':'Edit Videos', 'Start':'2021-03-06','Finish':'2021-04-04'}

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

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

```
fig = px.timeline(project_df, x_start = "Start", x_end = "Finish", y = "Task")
fig.update_yaxes(autorange = "reversed") # otherwise tasks are listed from the bottom up
fig.show()
```



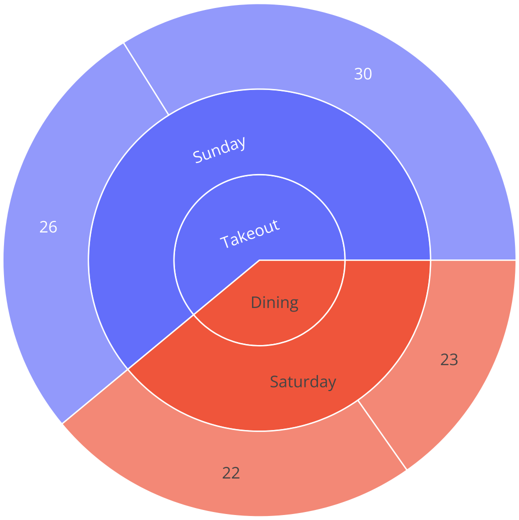
```
# Define Job #4
job_4 = {'Task':"Send the course for approval", 'Start':'2021-04-04', 'Finish':'2021-04-05'}
job_4

project_df = pd.DataFrame([job_1, job_2, job_3, job_4])
project_df

fig = px.timeline(project_df, x_start = "Start", x_end = "Finish", y = "Task")
fig.update_yaxes(autorange = "reversed") # otherwise tasks are listed from the bottom up
fig.show()
```



```
#Interactive Sunburst
restaurant_df = pd.read_csv('/content/restaurant_mini.csv'
```

	Customer ID	Day	Dining or Takeout	Age	Invoic
0	1	Saturday	Dining	23	45
1	2	Saturday	Dining	22	70
2	3	Sunday	Takeout	26	80
3	4	Sunday	Takeout	30	100

```
fig = px.sunburst(restaurant_df, path=['Dining or Takeout', 'Day','Age'], values='Invoice')
fig.show()
```



```
restaurant_df = pd.read_csv('/content/restaurant.csv')
restaurant_df
```

	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