

✓ Introduction



This dataset was scraped from nextspaceflight.com and includes all the space missions since the beginning of Space Race between the USA and the Soviet Union in 1957!

✓ Install Package with Country Codes

```
# %pip install iso3166
```

✓ Upgrade Plotly

Run the cell below if you are working with Google Colab.

```
# %pip install --upgrade plotly
```

✓ Import Statements

```
import numpy as np
import pandas as pd
```

```
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns

# These might be helpful:
from iso3166 import countries
from datetime import datetime, timedelta
```

▼ Notebook Presentation

```
pd.options.display.float_format = '{:,.2f}'.format
```


▼ Load the Data

```
df_data = pd.read_csv('mission_launches.csv')
```

▼ Preliminary Data Exploration

- What is the shape of df_data ?
- How many rows and columns does it have?
- What are the column names?
- Are there any NaN values or duplicates?

```
df_data.head()
```



	Unnamed: 0.1	Unnamed: 0	Organisation	Location	Date	Detail	Rocket_Status	Price	Mission_Status
0	0	0	SpaceX	LC-39A, Kennedy Space Center, Florida, USA	Fri Aug 07, 2020 05:12 UTC	Falcon 9 Block 5 Starlink V1 L9 & BlackSky	StatusActive	50	Success
1	1	1	CASC	Site 9401 (SLS-2), Jiuquan Satellite Launch Ce...	Thu Aug 06, 2020 04:01 UTC	Long March 2D Gaofen-9 04 & Q-SAT	StatusActive	29.75	Success
2	2	2	SpaceX	Pad A, Boca Chica, Texas, USA	Tue Aug 04, 2020 23:57 UTC	Starship Prototype 150 Meter Hop	StatusActive	NaN	Success
3	3	3	Roscosmos	Site 200/39, Baikonur Cosmodrome, Kazakhstan	Thu Jul 30, 2020 21:25 UTC	Proton-M/Briz-M Ekspress-80 & Ekspress-103	StatusActive	65	Success

```
# Rows and columns:
df_data.shape
```

↵ (4324, 9)

```
# Retrieving only column names:  
df_data.columns
```

↵ Index(['Unnamed: 0.1', 'Unnamed: 0', 'Organisation', 'Location', 'Date',
 'Detail', 'Rocket_Status', 'Price', 'Mission_Status'],
 dtype='object')

```
# Checking of NaN values:  
df_data.isna().values.any()
```

↵ True

```
# Checking of duplicates:  
df_data.duplicated().values.any()
```

↵ False

✓ Data Cleaning - Check for Missing Values and Duplicates

Consider removing columns containing junk data.

```
# Cleaning of NaN values:  
df_data.dropna(inplace = True)
```

```
df_data.shape
```

↵ (964, 9)

✓ Descriptive Statistics

```
df_data.describe()
```



Unnamed: 0.1 Unnamed: 0

count	964.00	964.00
mean	858.49	858.49
std	784.21	784.21
min	0.00	0.00
25%	324.75	324.75
50%	660.50	660.50
75%	1,112.00	1,112.00
max	4,020.00	4,020.00

✓ Number of Launches per Company

Create a chart that shows the number of space mission launches by organisation.

```
organizations = df_data.groupby("Organisation").count()["Mission_Status"]
organizations
```



```
Organisation
Arianespace      96
Boeing             7
CASC            158
EER                1
ESA                1
Eurockot          13
ExPace            1
ILS               13
ISRO              67
JAXA               3
Kosmotras         22
Lockheed           8
MHI               37
Martin Marietta   9
NASA            149
Northrop          83
RVSN USSR         2
Rocket Lab        13
Roscosmos         23
Sandia            1
SpaceX           99
ULA              98
US Air Force      26
VKS RF           33
Virgin Orbit      1
Name: Mission_Status, dtype: int64
```

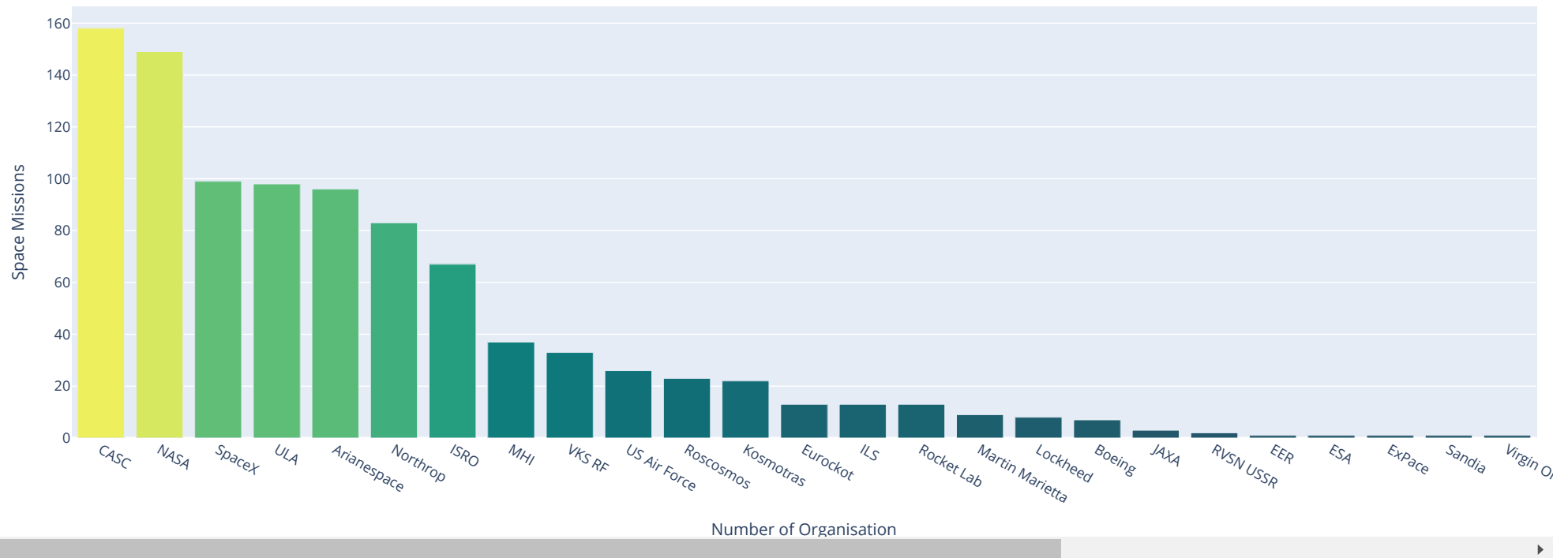
```
# Bar chart:
hbar = px.bar(organizations,
               x = organizations.index,
               y = organizations.values,
               color = organizations.values,
               color_continuous_scale = "Aggrnyl")

hbar.update_layout(title = "Organisation X Space Missions",
                   xaxis_title = "Number of Organisation",
                   yaxis_title = "Space Missions",
                   xaxis = {'categoryorder':'total descending'},
                   coloraxis_showscale = False)

hbar.show()
```



Organisation X Space Missions



✓ Number of Active versus Retired Rockets

How many rockets are active compared to those that are decommissioned?

```
df_data.groupby("Rocket_Status").count()
```



Unnamed: 0.1 Unnamed: 0 Organisation Location Date Detail Price Mission_Status

Rocket_Status

StatusActive	586	586	586	586	586	586	586	586
StatusRetired	378	378	378	378	378	378	378	378

✓ Distribution of Mission Status

How many missions were successful? How many missions failed?

```
df_data.groupby("Mission_Status").count()
```



Unnamed: 0.1 Unnamed: 0 Organisation Location Date Detail Rocket_Status Price

Mission_Status

Failure	36	36	36	36	36	36	36	36
Partial Failure	17	17	17	17	17	17	17	17
Prelaunch Failure	1	1	1	1	1	1	1	1
Success	910	910	910	910	910	910	910	910

✓ How Expensive are the Launches?

Create a histogram and visualise the distribution. The price column is given in USD millions (careful of missing values).

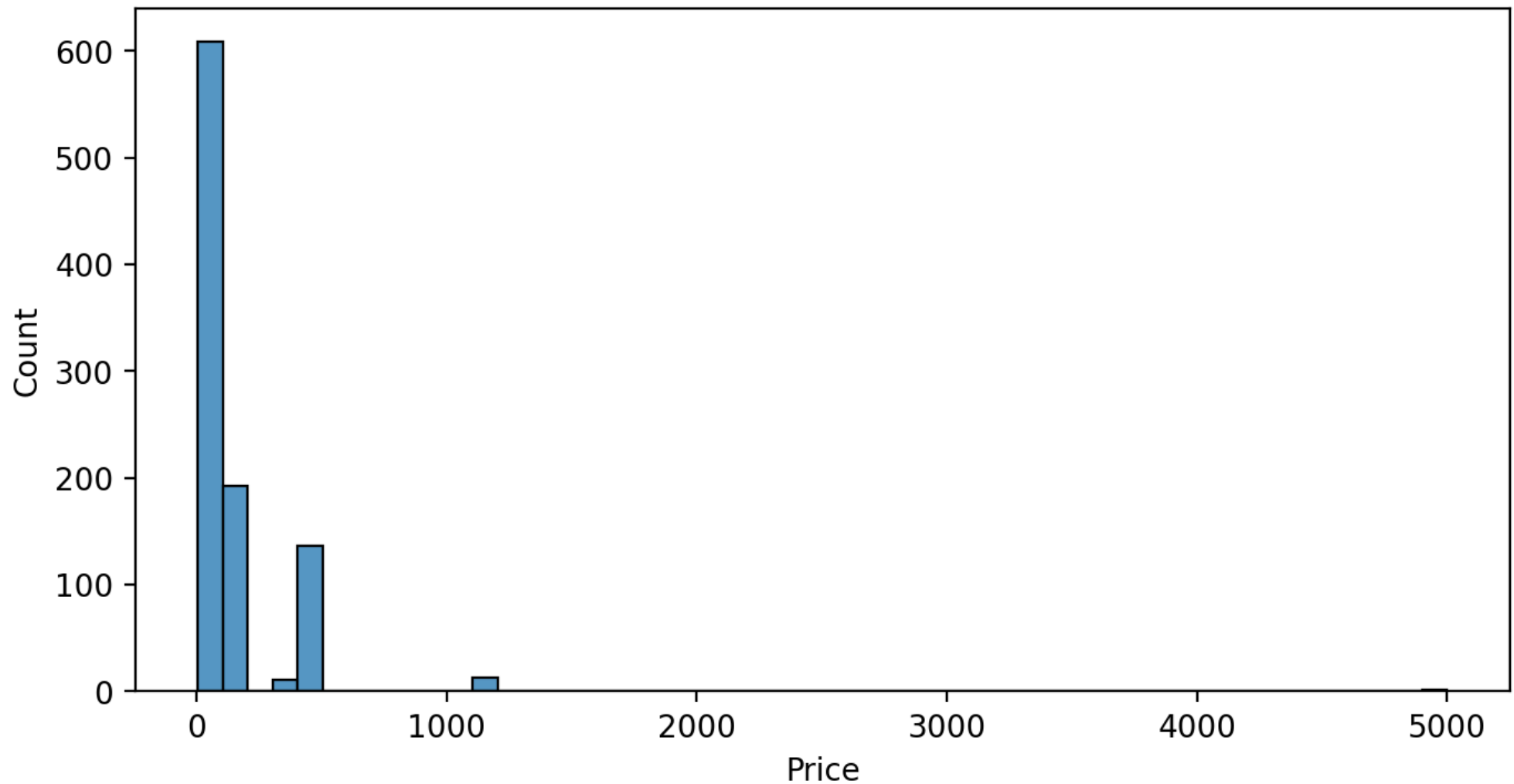
Histogram:

```
df_data.Price = df_data.Price.str.replace(",","").astype(float)
```

```
plt.figure(figsize = (8,4),dpi = 200)
plt.title("Price Distribution of Rockets")
sns.histplot(data = df_data,x = "Price",bins = 50)
plt.show()
```



Price Distribution of Rockets



✓ Use a Choropleth Map to Show the Number of Launches by Country

- Create a choropleth map using [the plotly documentation](#)
- Experiment with [plotly's available colours](#). I quite like the sequential colour matter on this map.
- You'll need to extract a `country` feature as well as change the country names that no longer exist.

Wrangle the Country Names

You'll need to use a 3 letter country code for each country. You might have to change some country names.

- Russia is the Russian Federation
- New Mexico should be USA
- Yellow Sea refers to China
- Shahrud Missile Test Site should be Iran
- Pacific Missile Range Facility should be USA
- Barents Sea should be Russian Federation
- Gran Canaria should be USA

You can use the iso3166 package to convert the country names to Alpha3 format.

```
# df_data.loc[930].Location.split()[-1]
all_countries = [data.split()[-1] for data in df_data.Location]

# Getting country codes:
codes = []
for country in all_countries:
    try:
        code = countries.get(country).alpha3
    except Exception:
        if country == "Russia":
            code = countries.get("Russian Federation").alpha3
        elif country == "Zealand":
            code = countries.get("New Zealand").alpha3
        elif country == "Sea":
            code = countries.get("China").alpha3
        elif country == "Facility":
            code = countries.get("USA").alpha3
        else:
            code = countries.get("USA").alpha3
    codes.append(code)

# Adding all country code:
df_data["Country_code"] = codes

launches = df_data.groupby("Country_code").count()["Mission_Status"]
launches = launches.reset_index()

# Adding Choropleth map:
cmap = px.choropleth(launches,
                     locations = "Country_code",
                     color = "Mission_Status",
```



```

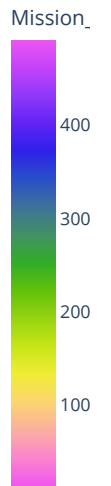
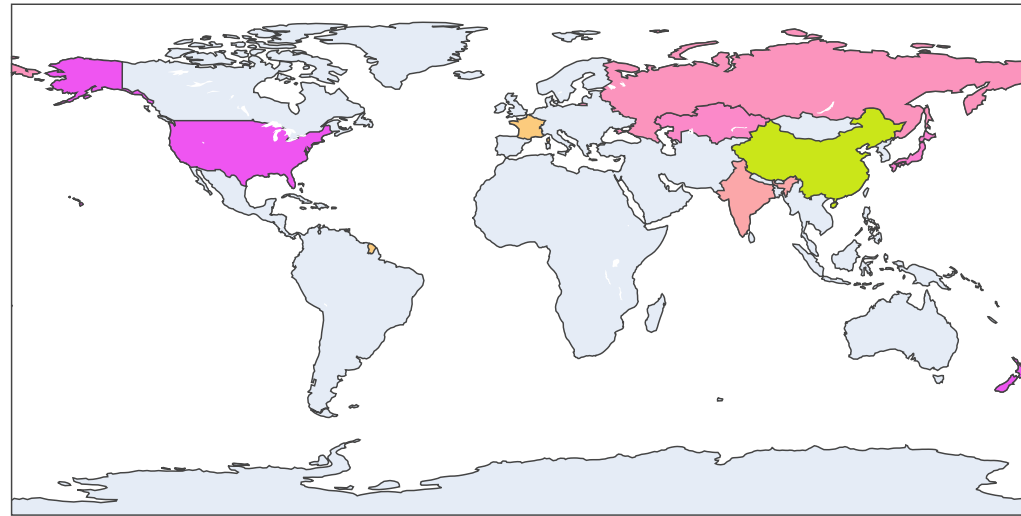
hover_name = "Country_code",
color_continuous_scale = "mygbm",
title = "Number of launches of countries")

cmap.show()

```



Number of launches of countries



✓ Use a Choropleth Map to Show the Number of Failures by Country

```

# Getting launches of failures:
failures = df_data[df_data.Mission_Status == "Failure"]
mission_failed = failures.groupby("Country_code")["Mission_Status"].count().reset_index()
mission_failed = mission_failed.rename(columns = {"Mission_Status":"Failures"})

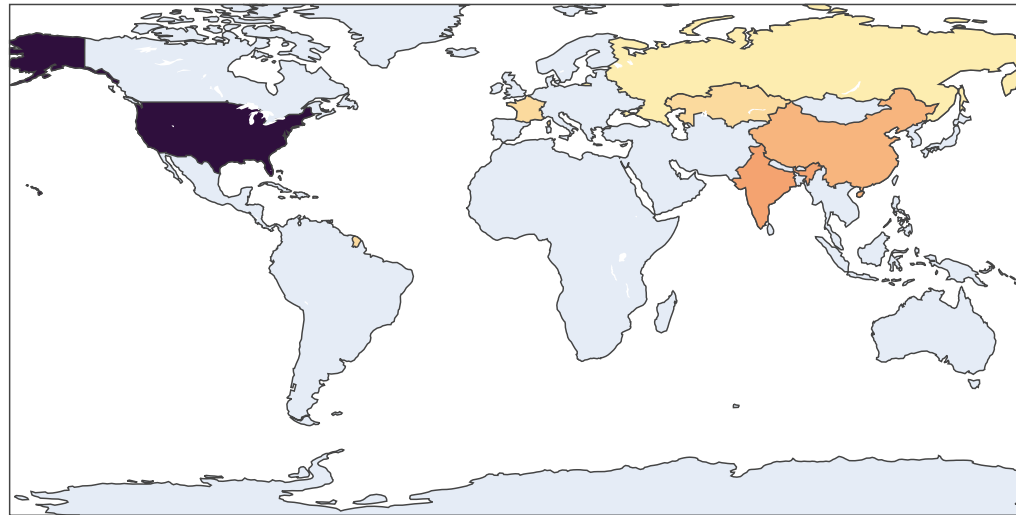
# Maps for failed launches of country:
failed_map = px.choropleth(mission_failed,
                           locations = "Country_code",
                           color = "Failures",
                           hover_name = "Country_code",
                           color_continuous_scale = "matter",
                           title = "Failures of launches per country")

```

```
failed_map.show()
```



Failures of launches per country



F

✓ Create a Plotly Sunburst Chart of the countries, organisations, and mission status.

```
# Sunburst Chart:  
df_data.head()
```



	Unnamed: 0.1	Unnamed: 0	Organisation	Location	Date	Detail	Rocket_Status	Price	Mission_Status	Country_code
0	0	0	SpaceX	LC-39A, Kennedy Space Center, Florida, USA	Fri Aug 07, 2020 05:12 UTC	Falcon 9 Block 5 Starlink V1 L9 & BlackSky	StatusActive	50.00	Success	USA
1	1	1	CASC	Site 9401 (SLS-2), Jiuquan Satellite Launch Ce...	Thu Aug 06, 2020 04:01 UTC	Long March 2D Gaofen-9 04 & Q-SAT	StatusActive	29.75	Success	CHN
3	3	3	Roscosmos	Site 200/39, Baikonur Cosmodrome, Kazakhstan	Thu Jul 30, 2020 21:25 UTC	Proton-M/Briz-M Ekspress-80 & Ekspress-103	StatusActive	65.00	Success	KAZ
4	4	4	ULA	SLC-41, Cape Canaveral AFS, Florida, USA	Thu Jul 30, 2020 11:50 UTC	Atlas V 541 Perseverance	StatusActive	145.00	Success	USA
5	5	5	CASC	LC-9, Taiyuan Satellite Launch Center, China	Sat Jul 25, 2020 03:13 UTC	Long March 4B Ziyuan-3 03, Apocalypse-10 & N...	StatusActive	64.68	Success	CHN

```
status = df_data.groupby(["Country_code","Organisation","Mission_Status"],as_index = False).agg({"Rocket_Status":pd.Series.count})
```

```
status.head()
```



	Country_code	Organisation	Mission_Status	Rocket_Status
0	CHN	CASC	Failure	3
1	CHN	CASC	Partial Failure	3
2	CHN	CASC	Success	152
3	CHN	ExPace	Failure	1
4	FRA	Arianespace	Failure	2

```
# Designing a sunburst chart:
burst = px.sunburst(status,
    path = ["Country_code","Organisation","Mission_Status"],
    values = "Rocket_Status",
    title = "Countries, Organizations and Mission Status")

burst.show()
```



Countries, Organizations and Mission Status



✓ Analyse the Total Amount of Money Spent by Organisation on Space Missions

```
df_data.groupby("Organisation").agg({"Price":pd.Series.sum}).sort_values(by = "Price",ascending = False)
```



Organisation	Price
NASA	76,280.00
Arianespace	16,345.00
ULA	14,798.00
RVSN USSR	10,000.00
CASC	6,340.26
SpaceX	5,444.00
Northrop	3,930.00
MHI	3,532.50
ISRO	2,177.00
US Air Force	1,550.92
VKS RF	1,548.90
ILS	1,320.00
Boeing	1,241.00
Roscosmos	1,187.50
Martin Marietta	721.40
Kosmotras	638.00
Eurockot	543.40
Lockheed	280.00
JAXA	168.00
Rocket Lab	97.50
ESA	37.00
ExPace	28.30
EER	20.00
Sandia	15.00
Virgin Orbit	12.00

✓ Analyse the Amount of Money Spent by Organisation per Launch

```
df_data.groupby(["Organisation", "Mission_Status"]).agg({"Price": pd.Series.sum})
```



Organisation	Mission_Status	Price
Arianespace	Failure	237.00
	Partial Failure	200.00
	Success	15,908.00
Boeing	Partial Failure	350.00
	Success	891.00
CASC	Failure	158.51
	Partial Failure	128.60
	Success	6,053.15
EER	Failure	20.00
ESA	Success	37.00
Eurockot	Failure	41.80
	Success	501.60
ExPace	Failure	28.30
ILS	Success	1,320.00
ISRO	Failure	197.00
	Partial Failure	119.00
	Success	1,861.00
JAXA	Success	168.00
Kosmotras	Failure	29.00
	Success	609.00
Lockheed	Success	280.00
MHI	Success	3,532.50
Martin Marietta	Failure	171.60
	Success	549.80
NASA	Failure	900.00
	Partial Failure	1,160.00
	Success	74,220.00
Northrop	Failure	335.00
	Partial Failure	80.00

	Success	3,515.00
RVSN USSR	Success	10,000.00
Rocket Lab	Failure	15.00
	Success	82.50
Roscosmos	Partial Failure	48.50
	Success	1,139.00
Sandia	Failure	15.00
SpaceX	Failure	77.50
	Partial Failure	59.50
	Prelaunch Failure	62.00
	Success	5,245.00
ULA	Partial Failure	109.00
	Success	14,689.00
US Air Force	Failure	122.23
	Partial Failure	59.00
	Success	1,369.69
VKS RF	Failure	65.00
	Partial Failure	83.60
	Success	1,400.30
Virgin Orbit	Failure	12.00

✓ Chart the Number of Launches per Year

```
df_data['Date'] = pd.to_datetime(df_data['Date'],errors='coerce',utc = True)
df_data["Year"] = df_data['Date'].dt.year
```

```
launches_by_year = df_data.groupby("Year",as_index = False).agg({"Mission_Status":pd.Series.count}).sort_values(by = "Year")
launches_by_year.head()
```



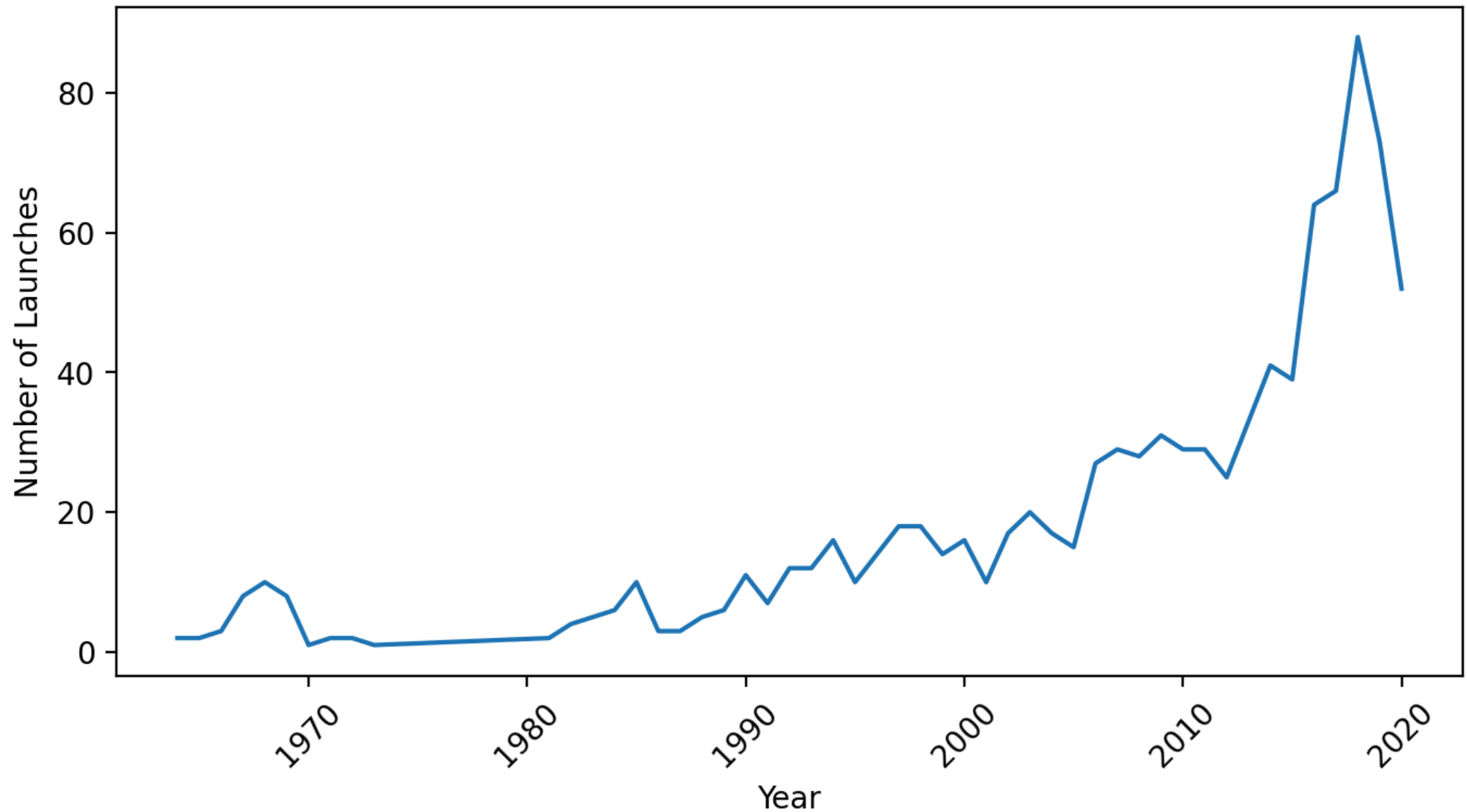
	Year	Mission_Status
0	1964	2
1	1965	2
2	1966	3
3	1967	8
4	1968	10

Plotting chart:

```
plt.figure(figsize = (8,4),dpi = 200)
plt.title("Number of Launches per Year")
plt.plot(launches_by_year.Year,launches_by_year.Mission_Status)
plt.xlabel("Year")
plt.xticks(rotation = 45)
plt.ylabel("Number of Launches")
plt.show()
```



Number of Launches per Year



✓ Chart the Number of Launches Month-on-Month until the Present

Which month has seen the highest number of launches in all time? Superimpose a rolling average on the month on month time series chart.

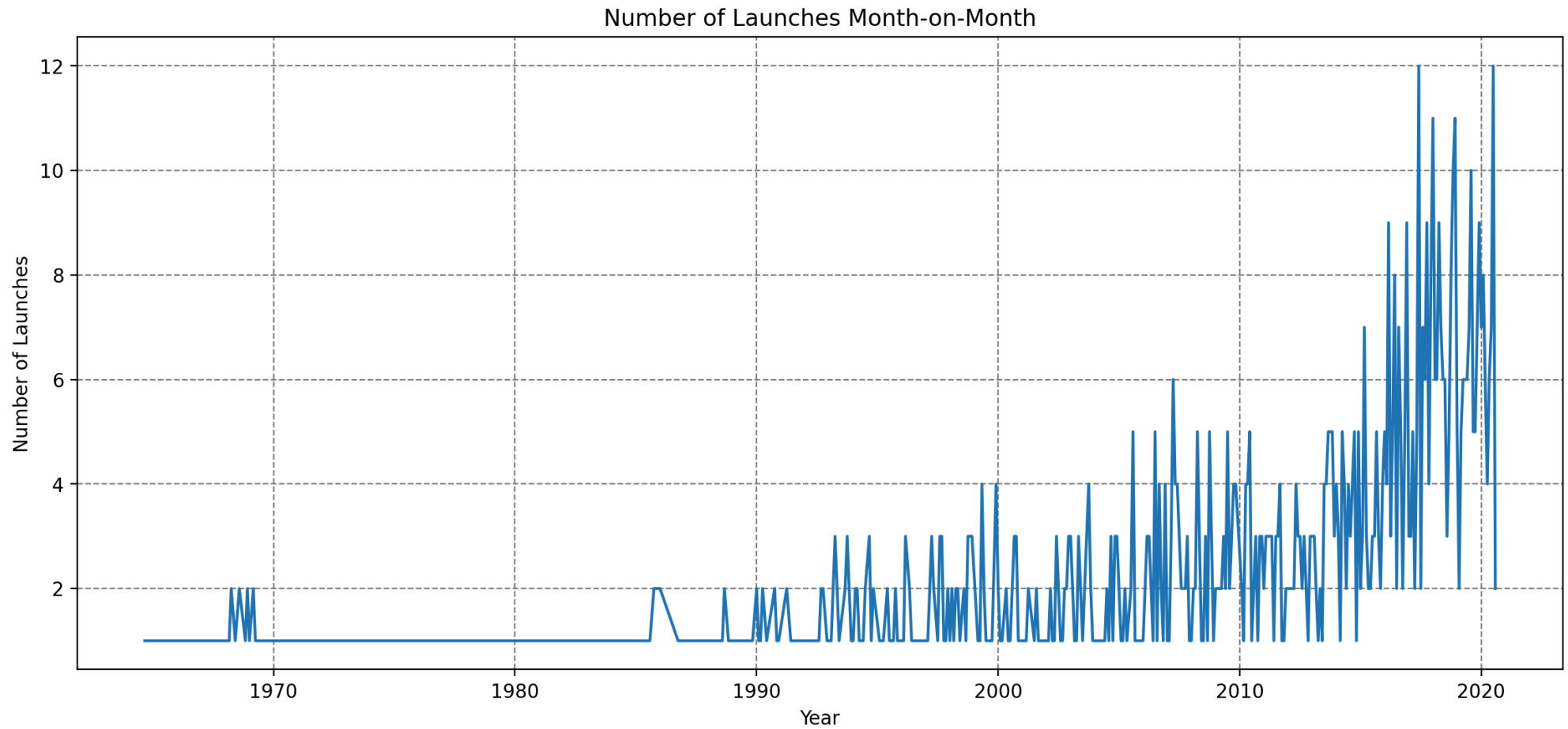
```
monthly_launches = df_data.groupby(df_data["Date"].dt.to_period("M")).agg({"Mission_Status":pd.Series.count})
monthly_launches.head()
```

↗ C:\Users\91937\AppData\Local\Temp\ipykernel_15704\1488531555.py:1: UserWarning:

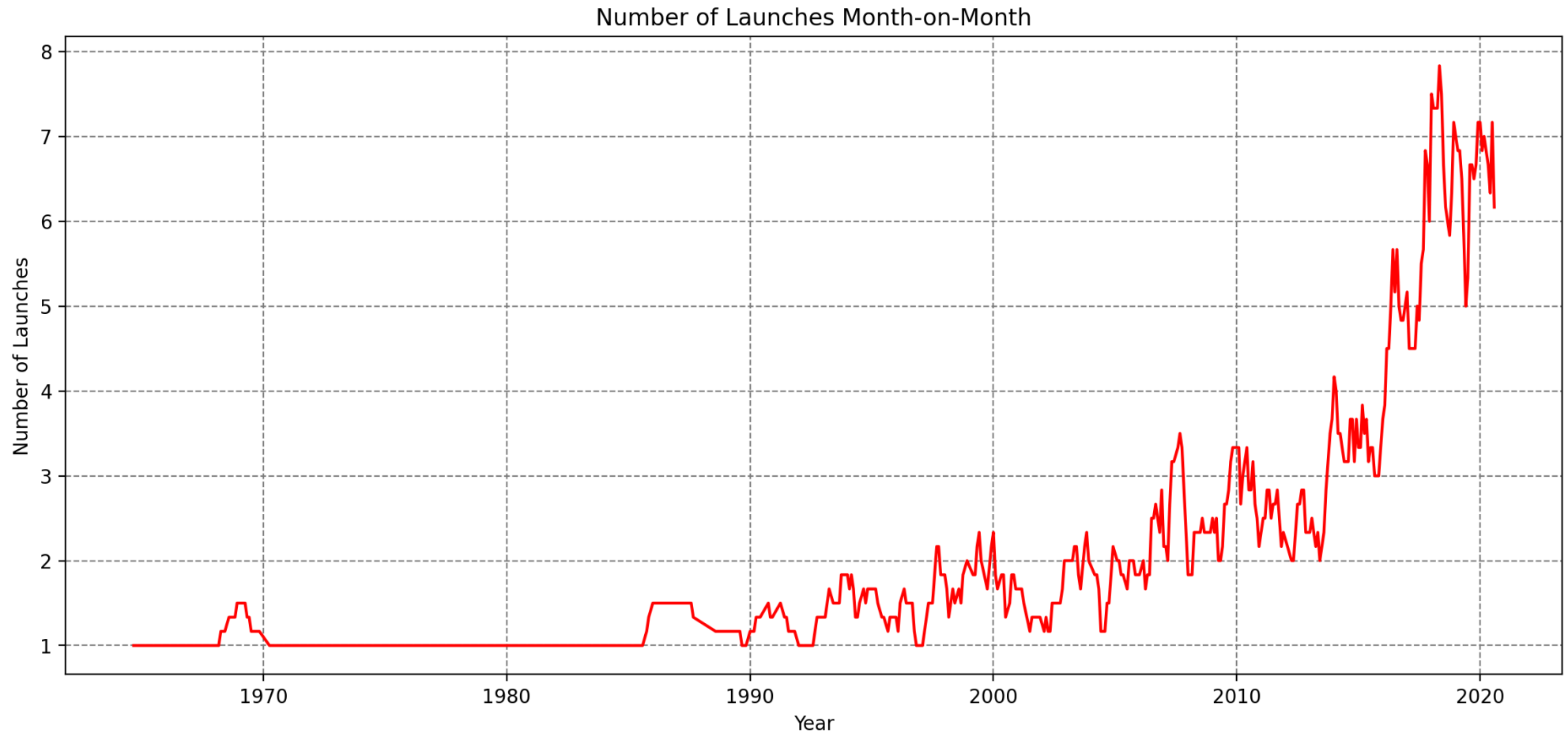
Converting to PeriodArray/Index representation will drop timezone information.

Mission_Status	
Date	
1964-09	1
1964-12	1
1965-02	1
1965-05	1
1966-07	1

```
# Plotting the number of launches on month-on-month:
plt.figure(figsize = (14,6),dpi = 200)
plt.plot(monthly_launches.index.to_timestamp(),monthly_launches.Mission_Status)
plt.title("Number of Launches Month-on-Month")
plt.xlabel("Year")
plt.ylabel("Number of Launches")
plt.grid(linestyle = "--",color = "grey")
plt.show()
```



```
# Calculating rolling mean and thereby smoothing:
smooth_launches = monthly_launches.rolling(window = 6,min_periods = 1).mean()
plt.figure(figsize = (14,6),dpi = 200)
plt.plot(smooth_launches.index.to_timestamp(),smooth_launches.Mission_Status,color = "red")
plt.title("Number of Launches Month-on-Month")
plt.xlabel("Year")
plt.ylabel("Number of Launches")
plt.grid(linestyle = "--",color = "grey")
plt.show()
```



✓ Launches per Month: Which months are most popular and least popular for launches?

Some months have better weather than others. Which time of year seems to be best for space missions?

```
monthly_launches.reset_index(inplace = True)
```

```
monthly_launches.sort_values(by = "Mission_Status",ascending = False)[:3] # July, June, Jan popular for launches:
```



	Date	Mission_Status
386	2020-07	12
349	2017-06	12
356	2018-01	11

✓ How has the Launch Price varied Over Time?

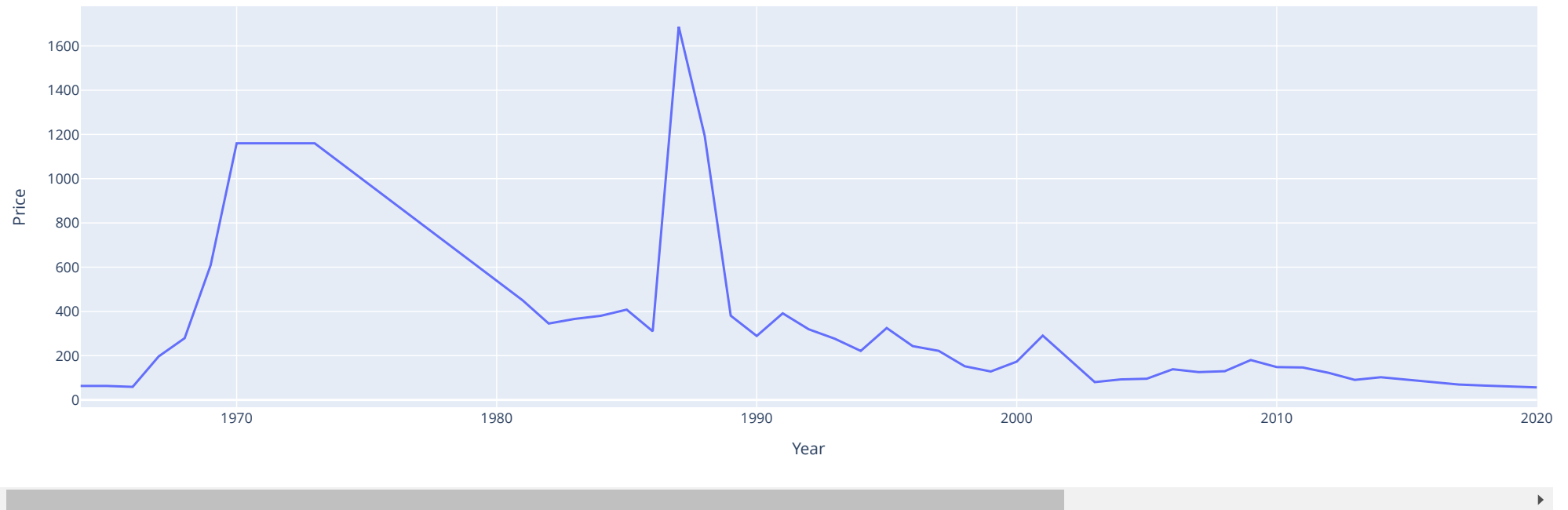
Create a line chart that shows the average price of rocket launches over time.

```
# Line chart:
avg_prices = df_data.groupby("Year", as_index = False).agg({"Price": pd.Series.mean})

line_chart = px.line(avg_prices,
                      x = "Year",
                      y = "Price",
                      title = "Average Price of Rocket Launches over time")
line_chart.show()
```



Average Price of Rocket Launches over time



✓ Chart the Number of Launches over Time by the Top 10 Organisations.

How has the dominance of launches changed over time between the different players?

```
# Getting all organizations:
all_orgs = df_data.groupby(["Organisation", "Year"], as_index = False).agg({"Mission_Status": pd.Series.count})

# Top 10 from dataset:
top10 = df_data.groupby("Organisation", as_index = False).agg({"Mission_Status": pd.Series.count})[:10]

# Filtering out:
filtered_data = all_orgs[all_orgs["Organisation"].isin(top10["Organisation"])]

# Line chart:
top10_chart = px.line(filtered_data,
                      x = "Year",
                      y = "Mission_Status",
```

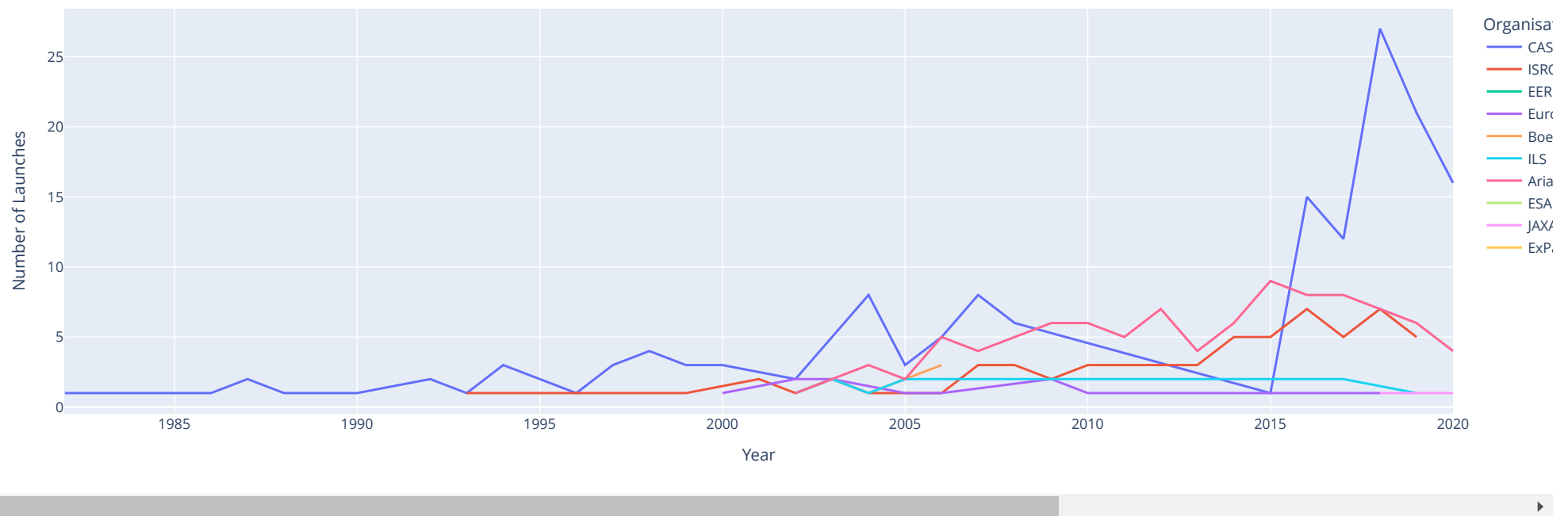
```
color = "Organisation",
hover_name = "Organisation",
title = "Number of Launches over time")
```

```
top10_chart.update_layout(yaxis_title = "Number of Launches")
```

```
top10_chart.show()
```



Number of Launches over time



✓ Cold War Space Race: USA vs USSR

The cold war lasted from the start of the dataset up until 1991.

```
cold_war = df_data[df_data.Year <= 1991].sort_values(by = "Year")
```

```
cold_war.head()
```




	Unnamed: 0.1	Unnamed: 0	Organisation	Location	Date	Detail	Rocket_Status	Price	Mission_Status	Country_code	Year
4020	4020	4020	US Air Force	SLC-20, Cape Canaveral AFS, Florida, USA	1964-09-01 15:00:00+00:00	Titan IIIA Transtage 1	StatusRetired	63.23	Failure	USA	1964
4000	4000	4000	US Air Force	SLC-20, Cape Canaveral AFS, Florida, USA	1964-12-10 16:52:00+00:00	Titan IIIA Transtage 2	StatusRetired	63.23	Success	USA	1964
3971	3971	3971	US Air Force	SLC-20, Cape Canaveral AFS, Florida, USA	1965-05-06 15:00:00+00:00	Titan IIIA LES 2 & LCS 1	StatusRetired	63.23	Success	USA	1965
3993	3993	3993	US Air Force	SLC-20, Cape Canaveral AFS, Florida, USA	1965-02-11 15:19:00+00:00	Titan IIIA LES 1	StatusRetired	63.23	Success	USA	1965
3855	3855	3855	US Air Force	SLC-4W, Vandenberg AFB, California, USA	1966-07-29 18:43:00+00:00	Titan IIIB KH-8	StatusRetired	59.00	Success	USA	1966

✓ Create a Plotly Pie Chart comparing the total number of launches of the USSR and the USA

Hint: Remember to include former Soviet Republics like Kazakhstan when analysing the total number of launches.

```
cold_war_countries = df_data.query("Country_code == 'USA' or Country_code == 'KAZ' or Country_code == 'RUS'")

distribution = cold_war_countries.groupby(["Country_code", "Year"], as_index = False).agg({"Mission_Status": pd.Series.count})

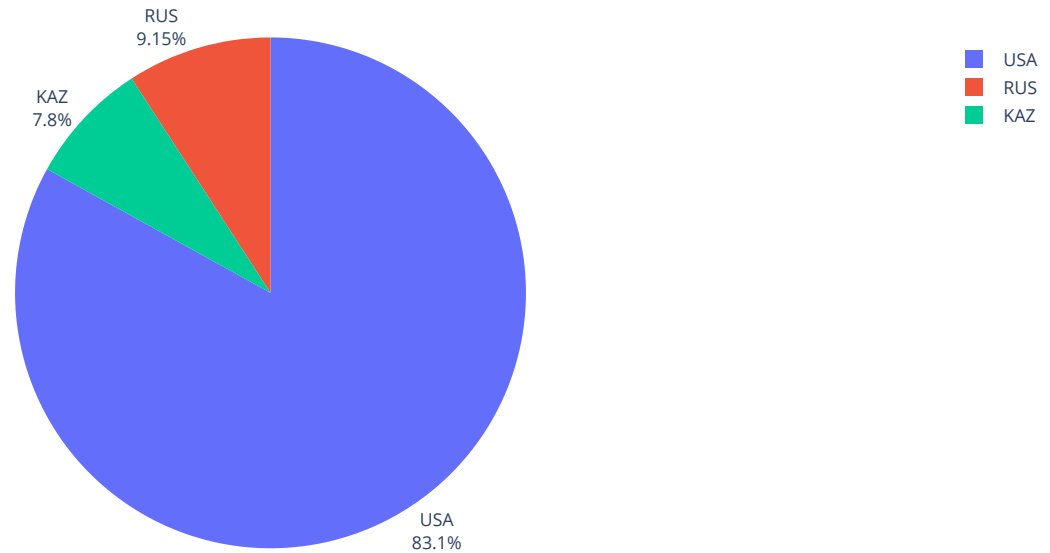
# Pie chart:
pie_chart = px.pie(
    values = distribution.Mission_Status,
    names = distribution.Country_code,
    title = "USA Vs USSR Launches")

pie_chart.update_traces(textposition = "outside", textinfo = "percent+label")

pie_chart.show()
```



USA Vs USSR Launches



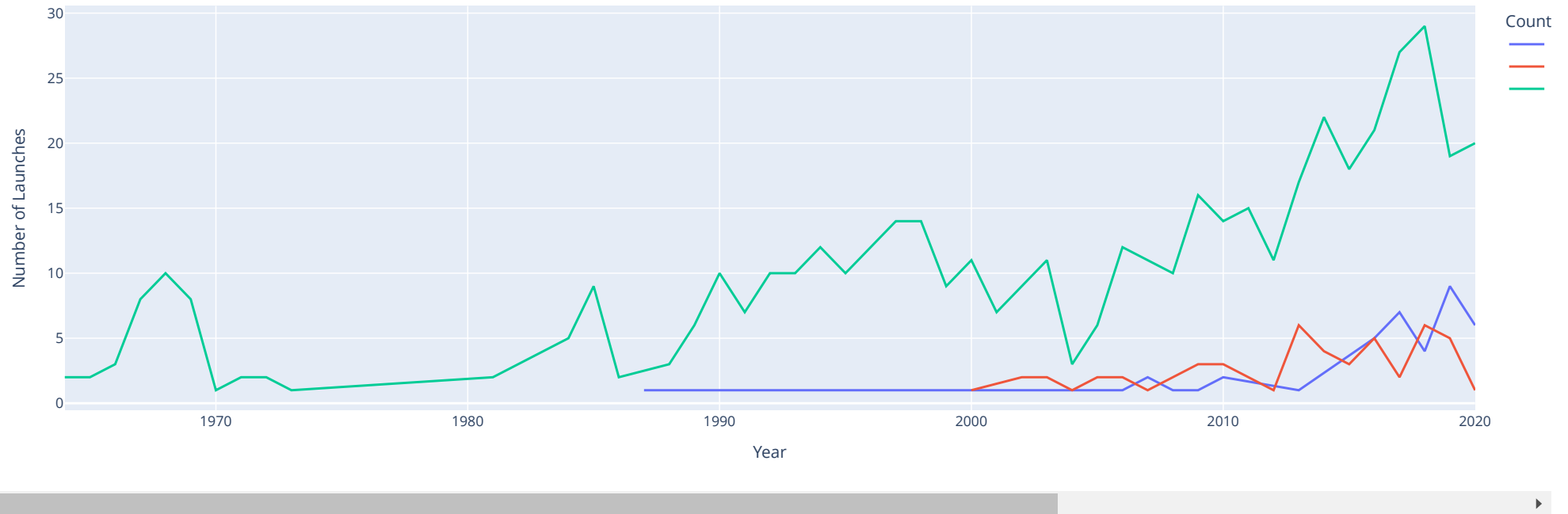
✓ Create a Chart that Shows the Total Number of Launches Year-On-Year by the Two Superpowers

```
# Line chart:
cold_chart = px.line(distribution,
                     x = "Year",
                     y = "Mission_Status",
                     color = "Country_code",
                     title = "Number of Launches USA Vs USSR")

cold_chart.update_layout(yaxis_title = "Number of Launches")
cold_chart.show()
```



Number of Launches USA Vs USSR



✓ Chart the Total Number of Mission Failures Year on Year.

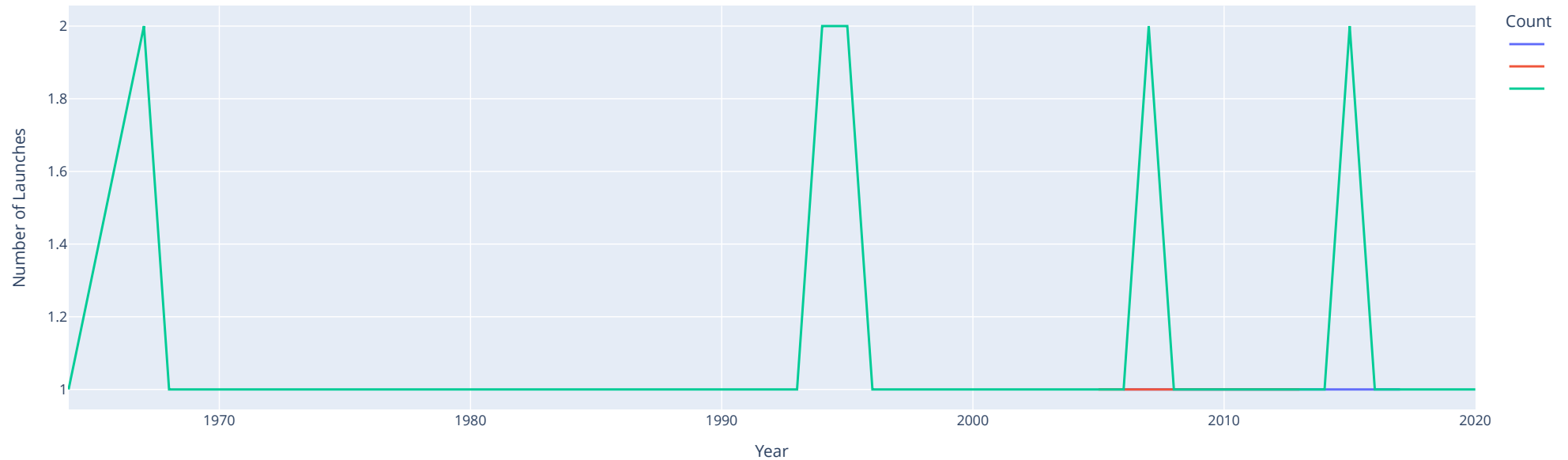
```
# Failures:
failures = cold_war_countries.query(" Mission_Status != 'Success' ")
failed = failures.groupby(["Country_code", "Year"], as_index = False).agg({"Mission_Status": pd.Series.count})

# Line chart:
failure_chart = px.line(failed,
                        x = "Year",
                        y = "Mission_Status",
                        color = "Country_code",
                        title = "Launch Failures USA Vs USSR")

failure_chart.update_layout(yaxis_title = "Number of Launches")
failure_chart.show()
```



Launch Failures USA Vs USSR



✓ Chart the Percentage of Failures over Time

Did failures go up or down over time? Did the countries get better at minimising risk and improving their chances of success over time?

```
# Getting failure percentage:
merged_data = pd.merge(distribution,failed, on = ["Country_code","Year"])
merged_data = merged_data.rename(columns = {'Mission_Status_x':'Total_Launches','Mission_Status_y':'Failures'})

merged_data['Per_failure'] = merged_data.Failures/merged_data.Total_Launches * 100
merged_data.head()
```



	Country_code	Year	Total_Launches	Failures	Per_failure
0	KAZ	2006	1	1	100.00