UNITED
UNIVERSITY

विद्या कल्पलतेव

SUBJECT NAME: DATA SCIENCE

PRACTICAL FILE

SESSION: 2025-26

SUBMITTED BY:

YASH KUMAR SRIVASTAVA

(UU24201010175)

SUBMITTED TO:

SAMARTH AMRUTE

(ASSISTANT PROFESSOR, CSE)

COURSE: BCA IBM

SEMESTER: 3rd

UNITED UNIVERSITY

RAWATPUR, PRAYAGRAJ, UTTAR PRADESH- 211012

Exploratory Data Analysis (EDA) on Space Missions Dataset

What is EDA?

Exploratory Data Analysis (EDA) is the process of exploring and understanding a dataset before applying any machine learning model.

It helps us:

- Understand the structure of data
- Detect missing values or errors
- Discover patterns and relationships
- Generate meaningful insights and observations

In this project, we perform EDA on the Space Missions Dataset, which contains historical information about global space launches conducted by multiple organizations such as:

- **SpaceX**
- **NASA**
- **Roscosmos**
- **CASC**
- **ISRO**
- **ULA**
- **Arianespace**
- **and many others**

We will explore:

1. Data loading and overview
2. Data cleaning & missing value handling
3. Descriptive statistics
4. Univariate and bivariate data visualization
5. Insight extraction from mission trends, company activity, rocket status, and outcomes

This EDA helps us understand:

- Global space mission trends
- Performance differences between companies
- Mission success and failure patterns
- How launch activity varies by country and year

About the Dataset

The **Space Missions Dataset** contains detailed historical information about global space launches conducted by multiple countries and space organizations over several decades. It is a rich dataset that helps us understand worldwide space activity, mission outcomes, and trends in the aerospace sector.

This dataset includes launch records from well-known organizations such as:

- **SpaceX**
- **NASA**
- **Roscosmos**
- **CASC (China)**
- **ISRO (India)**
- **ULA**
- **Arianespace**
- **Blue Origin**
- and many others

Each row of the dataset represents a **unique space mission**, with important attributes such as:

Company Name

The space agency or private organization responsible for the launch.

Location

The exact launch site including spaceports across the USA, Russia, China, India, Kazakhstan, and Europe.

Datum (Launch Date)

The date and time at which the mission was launched.

Detail

Short details about the mission, rocket type, payload information, and launch vehicle configuration.

Status Rocket

Indicates whether the rocket is **Active**, **Retired**, or used in multiple missions.

Status Mission

Represents the mission outcome:

- **Success**
- **Failure**
- **Partial Failure**
- **Prelaunch Failure**

This is one of the most important columns for analyzing mission performance.

1. Import Libraries

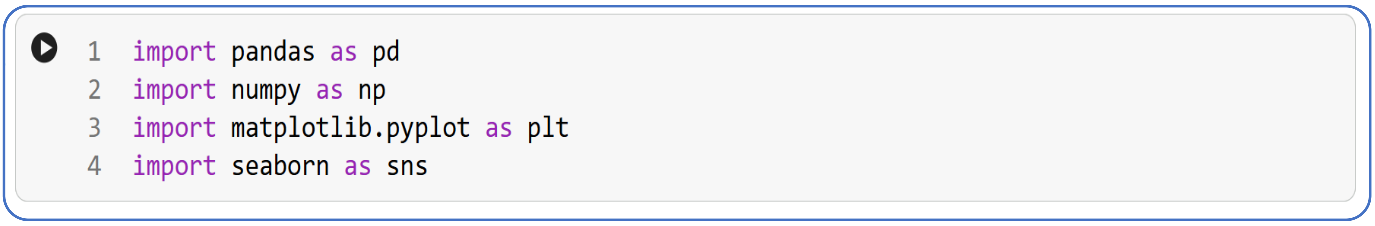
We will use the following libraries:

1. Pandas: Data manipulation and analysis

2. NumPy: Numerical operations and calculations

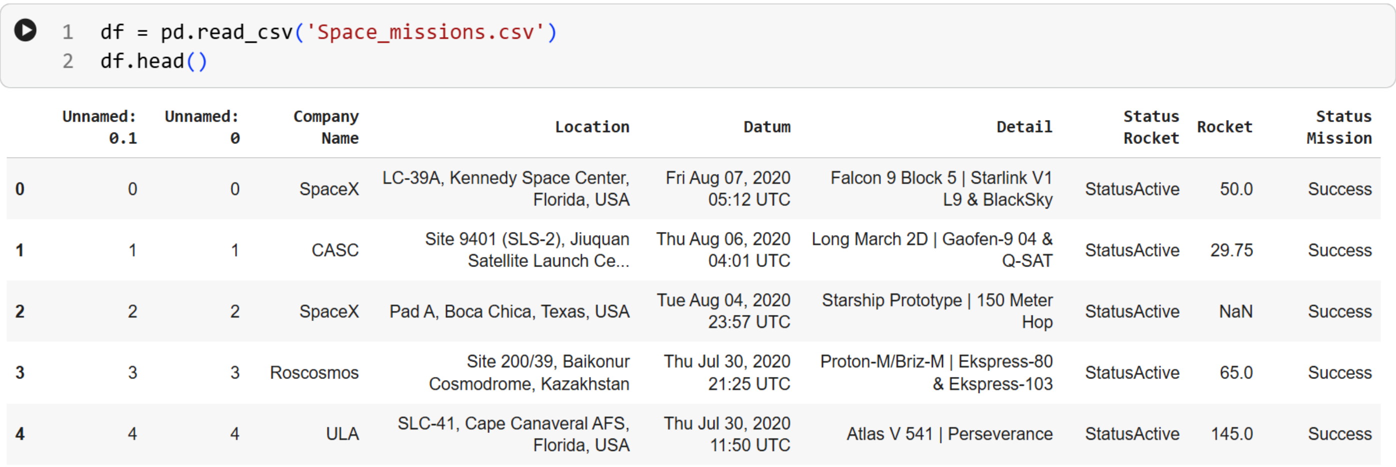
3. Matplotlib: Data visualization and plotting

4. Seaborn: Enhanced data visualization and statistical graphics



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Loading the Dataset



```python
df = pd.read_csv('Space_missions.csv')
df.head()
```

	Unnamed: 0.1	Unnamed: 0	Company Name	Location	Datum	Detail	Status Rocket	Rocket	Status Mission
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	50.0	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	65.0	Success
4	4	4	ULA	SLC-41, Cape Canaveral AFS, Florida, USA	Thu Jul 30, 2020 11:50 UTC	Atlas V 541 | Perseverance	StatusActive	145.0	Success

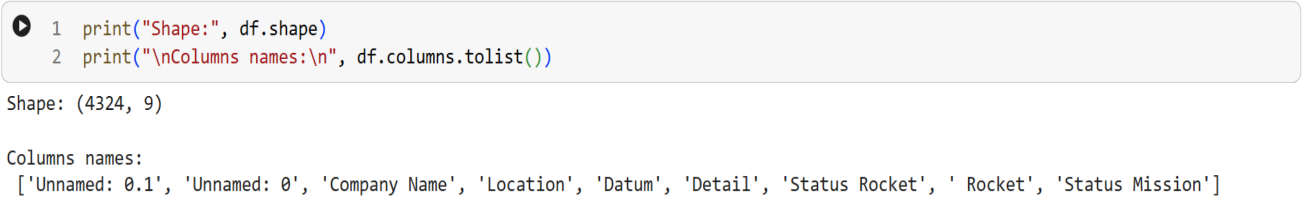
Explanation

`read_csv()`: loads our dataset

`head()`: displays the first 5 rows. This helps us understand the structure and columns of the dataset.

Initial Exploration

➢ Shape of Dataset



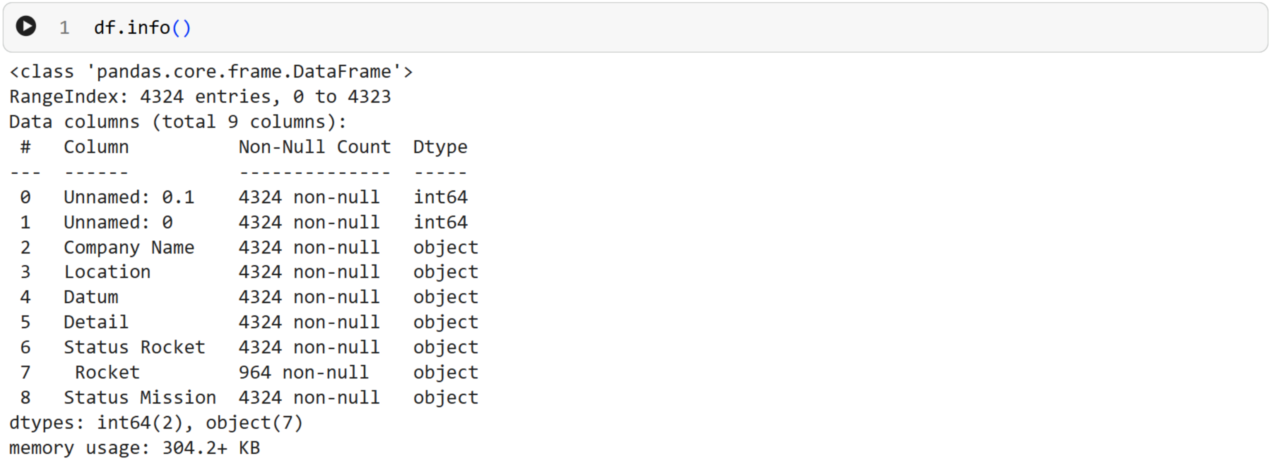
```python
print("Shape:", df.shape)
print("\nColumns names:\n", df.columns.tolist())
```

```
Shape: (4324, 9)

Columns names:
 ['Unnamed: 0.1', 'Unnamed: 0', 'Company Name', 'Location', 'Datum', 'Detail', 'Status Rocket', ' Rocket', 'Status Mission']
```

Explanation

- Tells total **rows** and **columns**.
- Shows all column names so we know what data is present.

➢ Info About Dataset



```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4324 entries, 0 to 4323
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Unnamed: 0.1    4324 non-null   int64 
 1   Unnamed: 0      4324 non-null   int64 
 2   Company Name    4324 non-null   object
 3   Location        4324 non-null   object
 4   Datum           4324 non-null   object
 5   Detail          4324 non-null   object
 6   Status Rocket   4324 non-null   object
 7    Rocket         964 non-null    object
 8   Status Mission  4324 non-null   object
dtypes: int64(2), object(7)
memory usage: 304.2+ KB
```

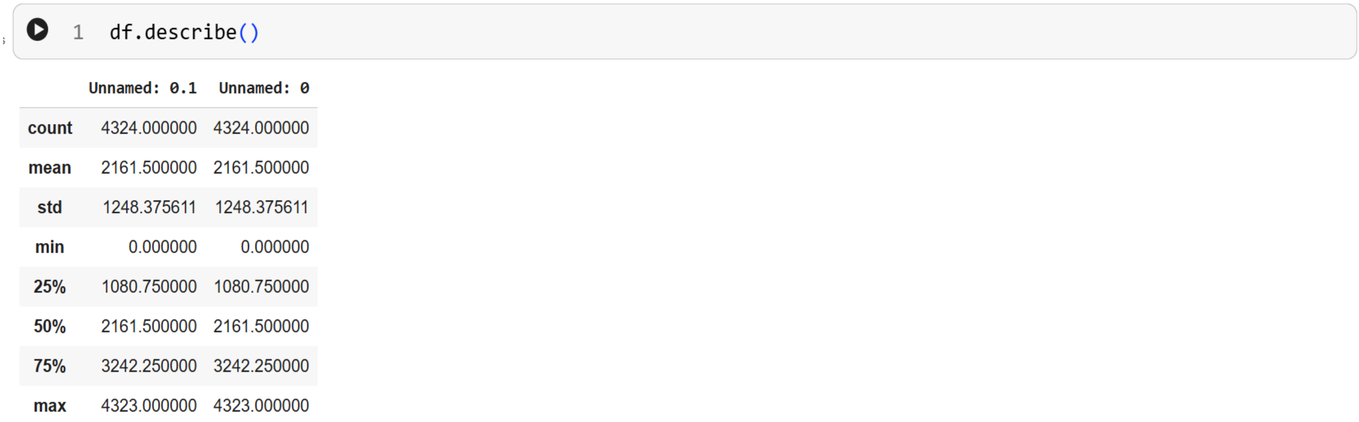
Explanation

Shows:

- Data types (int, float, object, datetime)
- Null values
- Memory usage

Helps prepare for cleaning.

➢ Summary Statistics



```python
df.describe()
```

	Unnamed: 0.1	Unnamed: 0
count	4324.000000	4324.000000
mean	2161.500000	2161.500000
std	1248.375611	1248.375611
min	0.000000	0.000000
25%	1080.750000	1080.750000
50%	2161.500000	2161.500000
75%	3242.250000	3242.250000
max	4323.000000	4323.000000

Explanation

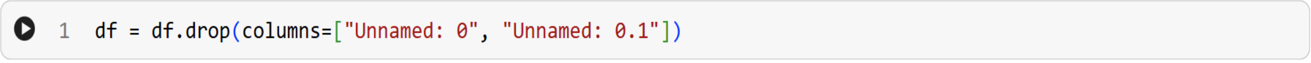
Gives:

- Mean
- Min / Max
- 25% / 50% / 75% values

Useful for understanding rocket costs and numeric fields.

Data Cleaning

Remove Unnecessary Columns

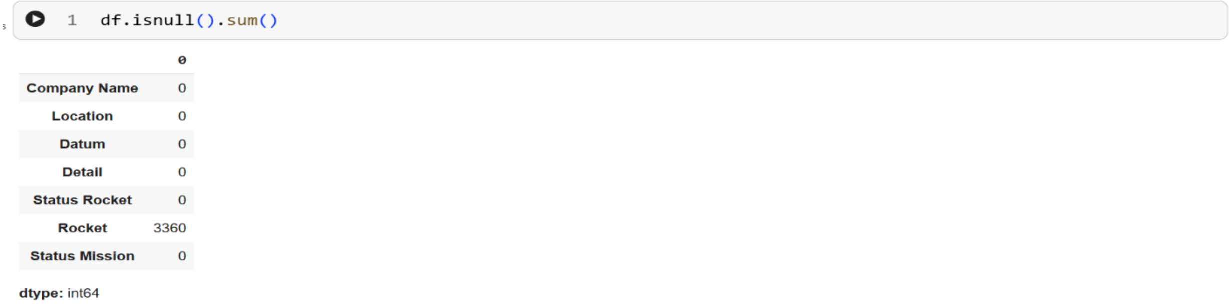


```python
df = df.drop(columns=["Unnamed: 0", "Unnamed: 0.1"])
```

Explanation

- These columns contain just row numbers created during Kaggle export. They do not help analysis, so we remove them.

Finding Missing Values



```python
df.isnull().sum()
```

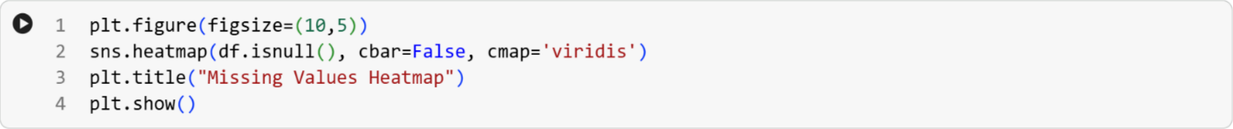
	0
Company Name	0
Location	0
Datum	0
Detail	0
Status Rocket	0
Rocket	3360
Status Mission	0

dtype: int64

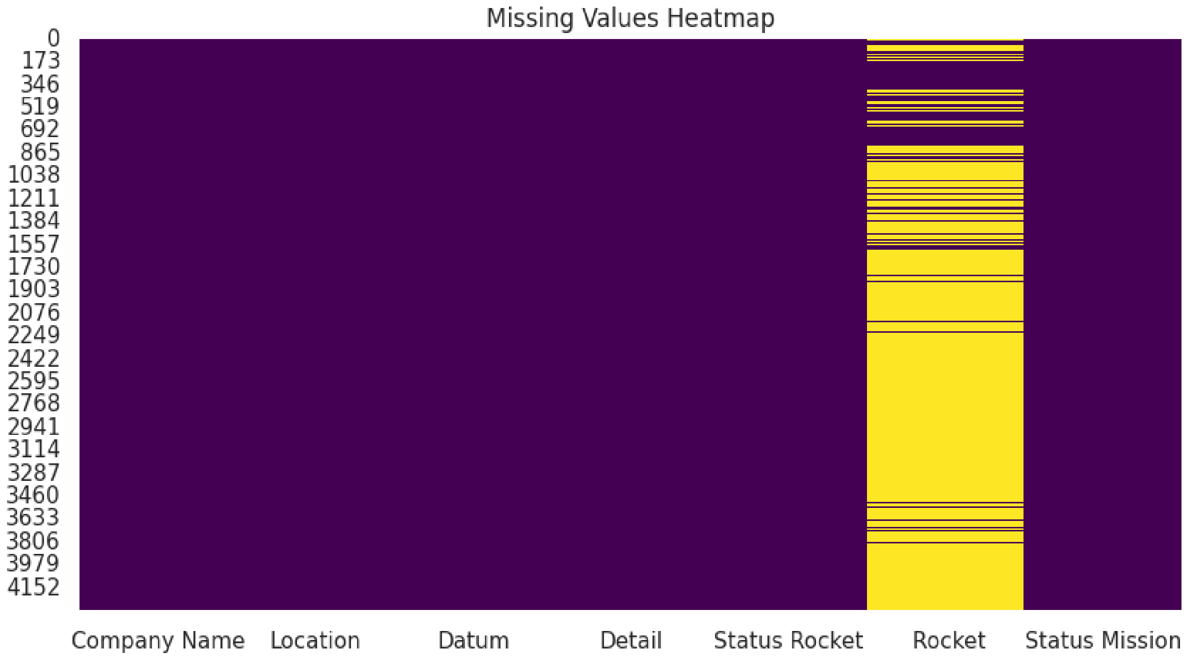
Explanation

Shows how many missing values are in each column.
This tells us where we need cleaning.

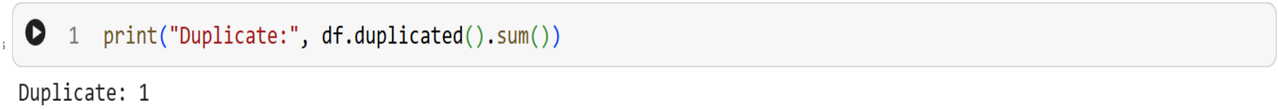
View Missing Values with Heatmap



```python
plt.figure(figsize=(10,5))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title("Missing Values Heatmap")
plt.show()
```



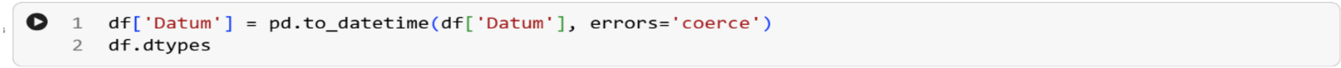
Check Duplicates



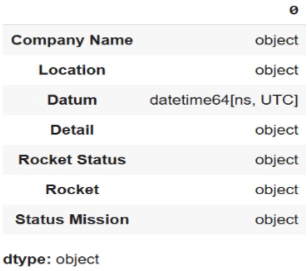
```python
print("Duplicate:", df.duplicated().sum())
```

```
Duplicate: 1
```

Convert Date Column



```python
df['Datum'] = pd.to_datetime(df['Datum'], errors='coerce')
df.dtypes
```



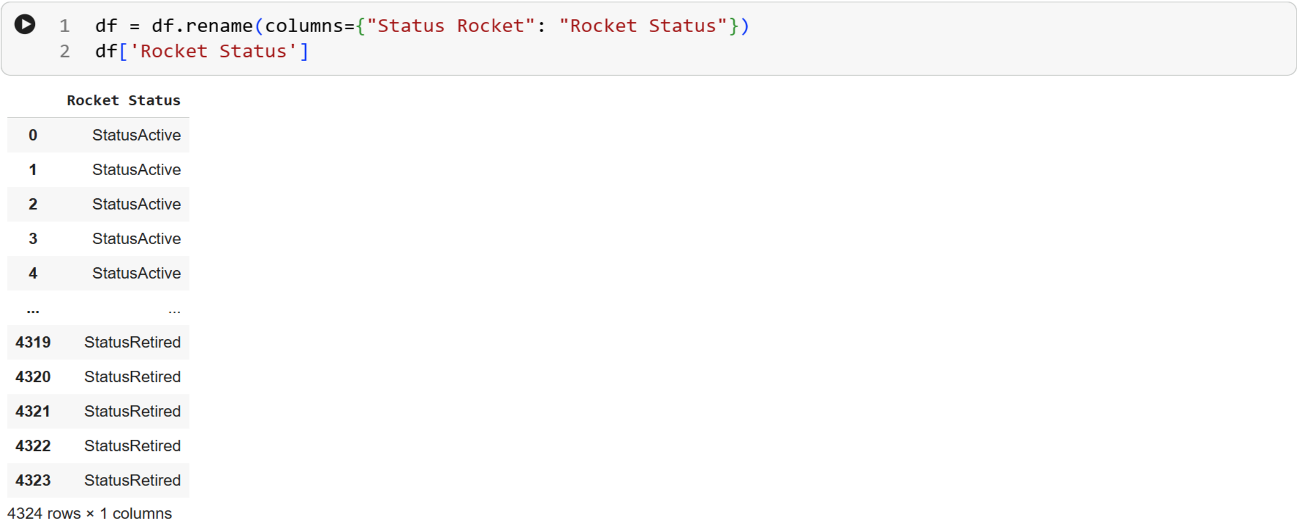
	0
Company Name	object
Location	object
Datum	datetime64[ns, UTC]
Detail	object
Rocket Status	object
Rocket	object
Status Mission	object

dtype: object

Explanation

- The "Datum" column is originally a string.
- To analyze trends by year/month, we need it as a datetime object.

Rename Column: Status Rocket to Rocket Status

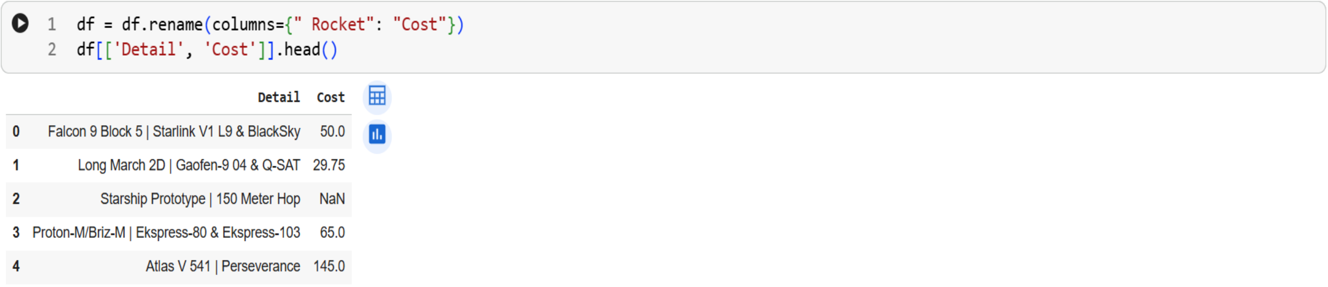


```
df = df.rename(columns={"Status Rocket": "Rocket Status"})
df['Rocket Status']
```

	Rocket Status
0	StatusActive
1	StatusActive
2	StatusActive
3	StatusActive
4	StatusActive
...	...
4319	StatusRetired
4320	StatusRetired
4321	StatusRetired
4322	StatusRetired
4323	StatusRetired

4324 rows × 1 columns

Rename Column: Rocket to Cost



```
df = df.rename(columns={" Rocket": "Cost"})
df[['Detail', 'Cost']].head()
```

	Detail	Cost
0	Falcon 9 Block 5 | Starlink V1 L9 & BlackSky	50.0
1	Long March 2D | Gaofen-9 04 & Q-SAT	29.75
2	Starship Prototype | 150 Meter Hop	NaN
3	Proton-M/Briz-M | Ekspress-80 & Ekspress-103	65.0
4	Atlas V 541 | Perseverance	145.0

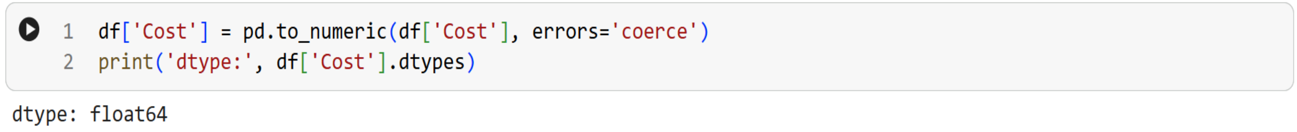
Explanations

In the Space Missions dataset, the rocket cost values were stored in a column named “ Rocket”, which contains a leading space in the column name.

To make the column easier to use and more meaningful, we renamed it to “Cost” using the following code:

Handle Missing Values:

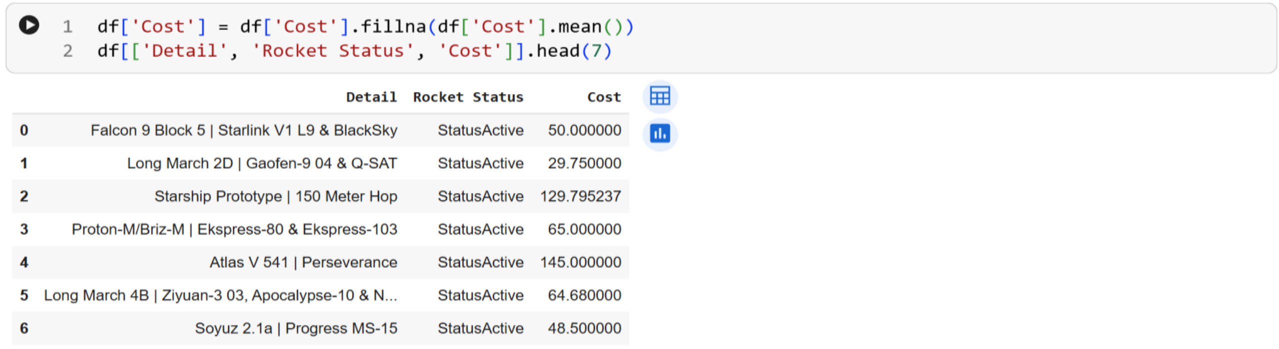
Convert Cost to Numeric



```
df['Cost'] = pd.to_numeric(df['Cost'], errors='coerce')
print('dtype:', df['Cost'].dtypes)
```

```
dtype: float64
```

fill missing values



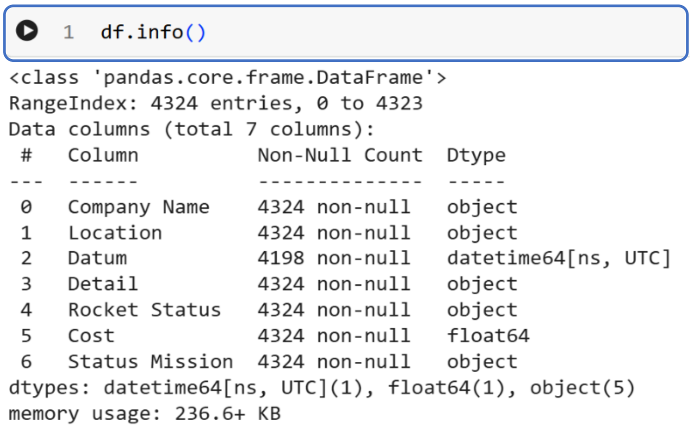
```
df['Cost'] = df['Cost'].fillna(df['Cost'].mean())
df[['Detail', 'Rocket Status', 'Cost']].head(7)
```

	Detail	Rocket Status	Cost
0	Falcon 9 Block 5 | Starlink V1 L9 & BlackSky	StatusActive	50.000000
1	Long March 2D | Gaofen-9 04 & Q-SAT	StatusActive	29.750000
2	Starship Prototype | 150 Meter Hop	StatusActive	129.795237
3	Proton-M/Briz-M | Ekspress-80 & Ekspress-103	StatusActive	65.000000
4	Atlas V 541 | Perseverance	StatusActive	145.000000
5	Long March 4B | Ziyuan-3 03, Apocalypse-10 & N...	StatusActive	64.680000
6	Soyuz 2.1a | Progress MS-15	StatusActive	48.500000

Explanation

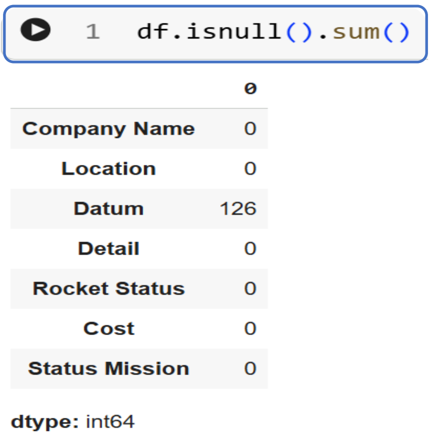
- The Cost column contained mixed data types (numbers and strings), which caused errors during numerical operations.
- To fix this, the column was converted into numeric format using pd.to_numeric() with errors='coerce', which safely replaces invalid values with NaN.
- After converting, missing values were filled using the mean cost value to maintain consistent analysis.

Re-check Info and Null values After Cleaning



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4324 entries, 0 to 4323
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Company Name    4324 non-null   object
 1   Location        4324 non-null   object
 2   Datum           4198 non-null   datetime64[ns, UTC]
 3   Detail          4324 non-null   object
 4   Rocket Status   4324 non-null   object
 5   Cost            4324 non-null   float64
 6   Status Mission  4324 non-null   object
dtypes: datetime64[ns, UTC](1), float64(1), object(5)
memory usage: 236.6+ KB
```



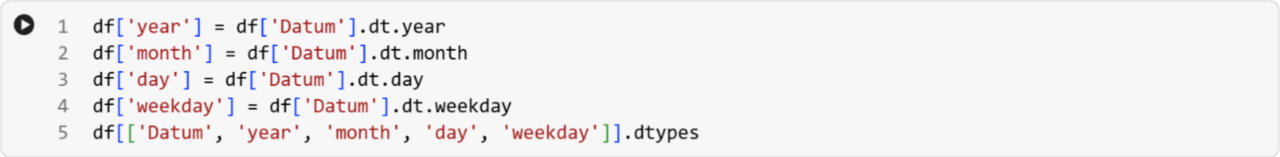
```
df.isnull().sum()
```

	0
Company Name	0
Location	0
Datum	126
Detail	0
Rocket Status	0
Cost	0
Status Mission	0

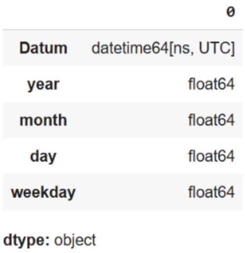
dtype: int64

Feature Engineering:

We create new features from the Datum column to analyze patterns.



```
df['year'] = df['Datum'].dt.year
df['month'] = df['Datum'].dt.month
df['day'] = df['Datum'].dt.day
df['weekday'] = df['Datum'].dt.weekday
df[['Datum', 'year', 'month', 'day', 'weekday']].dtypes
```



```
                           0
Datum    datetime64[ns, UTC]
year                 float64
month                float64
day                  float64
weekday              float64

dtype: object
```

Explanation of Each New Feature

• year

Extracts the year of the mission.
Helps analyze space mission trends over time.

• month

Extracts month number.
Useful for checking monthly mission frequency.

• day

Extracts day of the month.
Helps analyze daily patterns.

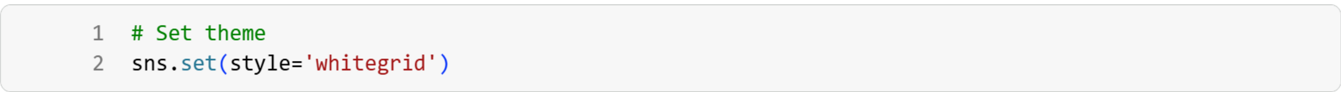
• weekday

0 = Monday, 6 = Sunday
Helps understand if certain days have higher mission counts.

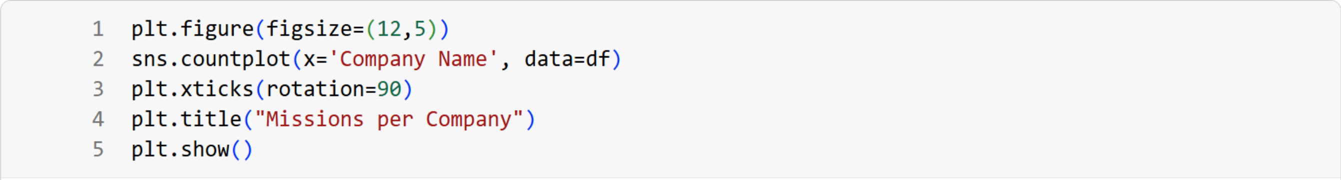
Univariate Analysis

Univariate = visualizing one column at a time

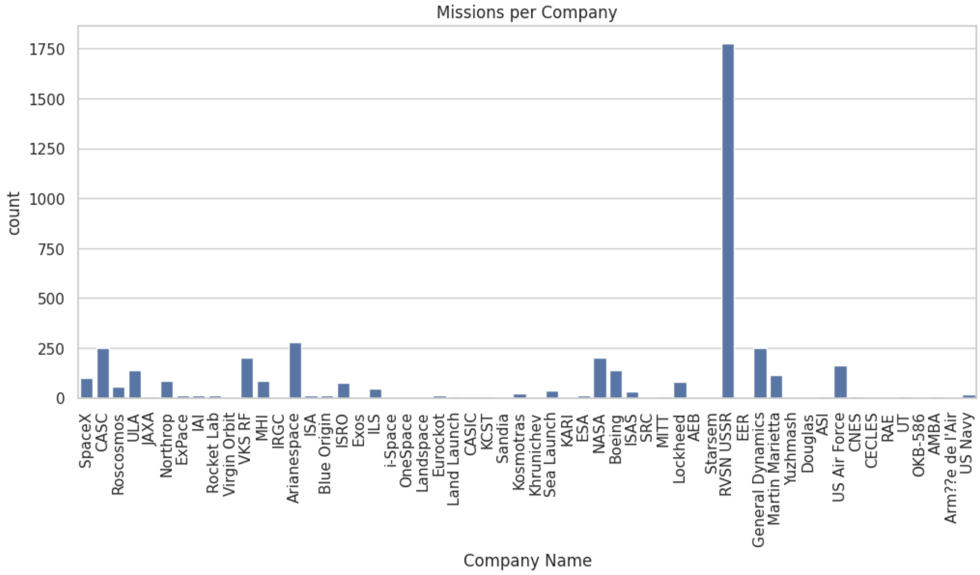
Missions Per Company



```
# Set theme
sns.set(style='whitegrid')
```

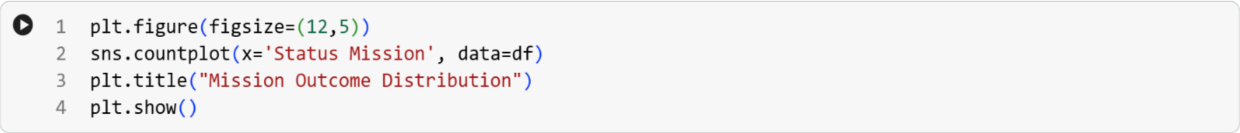


```
plt.figure(figsize=(12,5))
sns.countplot(x='Company Name', data=df)
plt.xticks(rotation=90)
plt.title("Missions per Company")
plt.show()
```

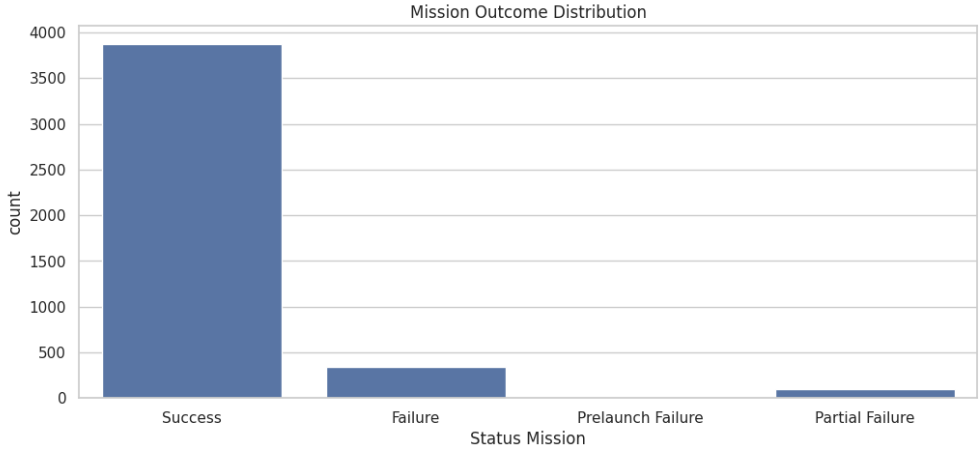


Shows the number of missions conducted by each space organization.

Mission Outcome Distribution

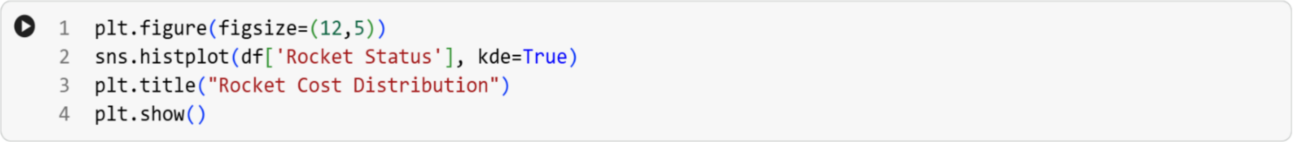


```
plt.figure(figsize=(12,5))
sns.countplot(x='Status Mission', data=df)
plt.title("Mission Outcome Distribution")
plt.show()
```

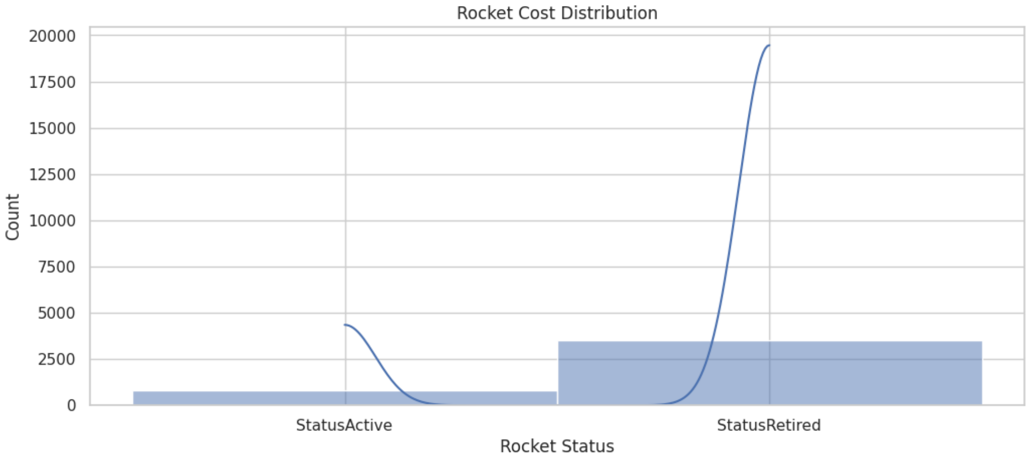


Shows how many missions succeeded, failed, or partially succeeded.

Rocket Cost Distribution



```python
plt.figure(figsize=(12,5))
sns.histplot(df['Rocket Status'], kde=True)
plt.title("Rocket Cost Distribution")
plt.show()
```

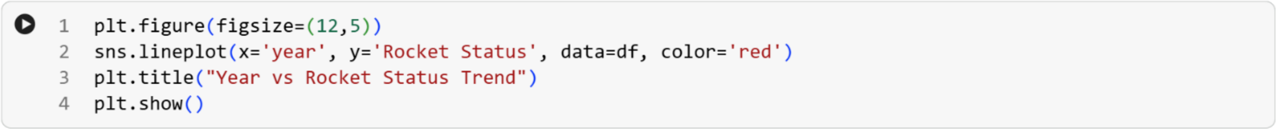


Shows how rocket costs are distributed across missions.

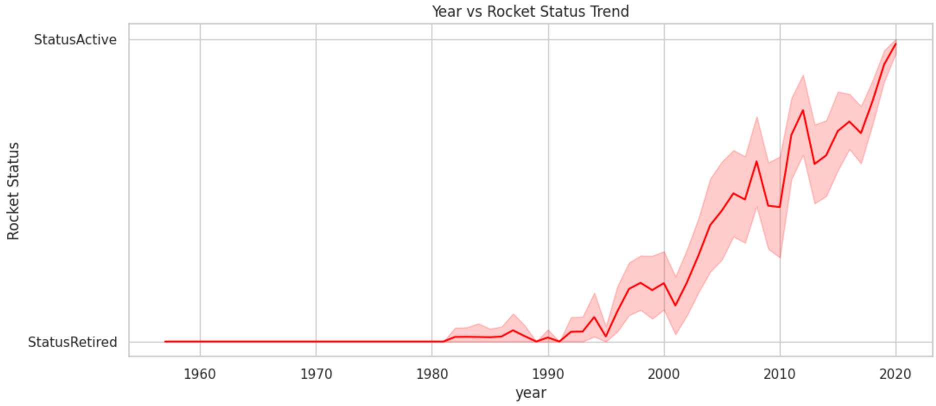
Bivariate Analysis

Bivariate = visualizing relationship between two variables

Year vs Rocket Status Trend

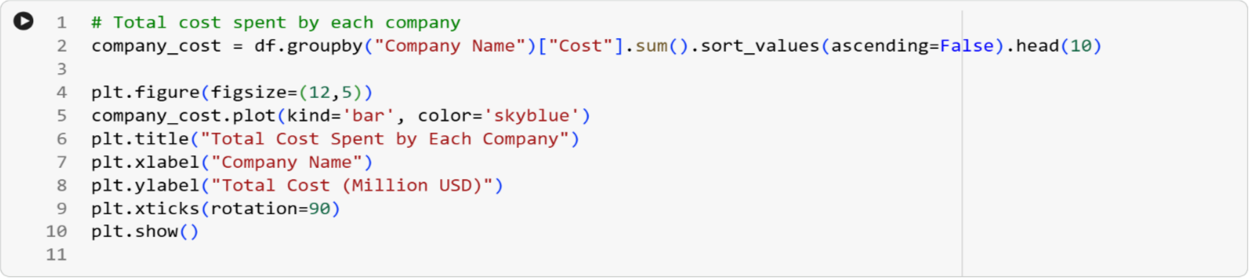


```python
plt.figure(figsize=(12,5))
sns.lineplot(x='year', y='Rocket Status', data=df, color='red')
plt.title("Year vs Rocket Status Trend")
plt.show()
```

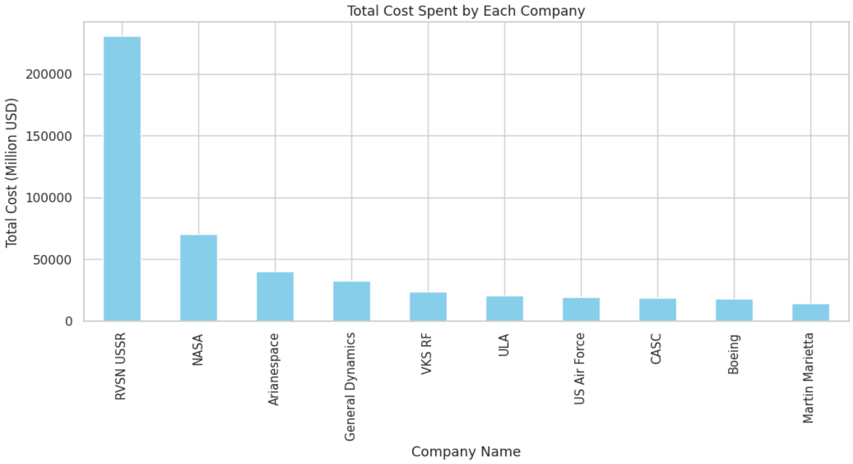


This line plot shows how the number of **active** and **retired rockets** has **changed over the years**.

Top 10 Companies by Total Cost

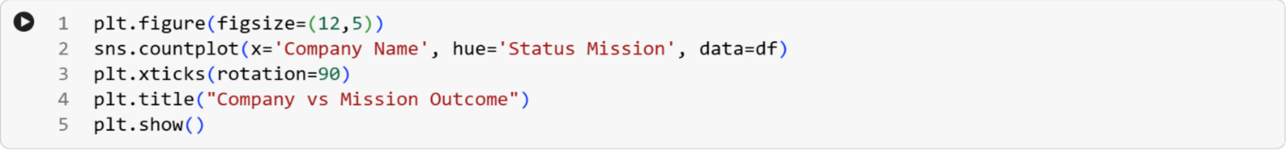


```python
# Total cost spent by each company
company_cost = df.groupby("Company Name")["Cost"].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(12,5))
company_cost.plot(kind='bar', color='skyblue')
plt.title("Total Cost Spent by Each Company")
plt.xlabel("Company Name")
plt.ylabel("Total Cost (Million USD)")
plt.xticks(rotation=90)
plt.show()

```

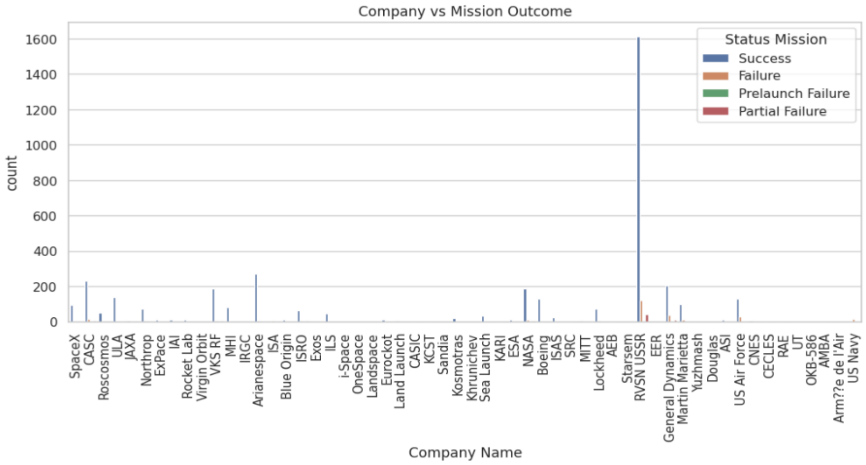


- We calculated the total **rocket cost** for each **company** by **grouping** the dataset using **Company Name** and **summing** the values of the **Cost column**.
- The **bar chart** shows which organizations have **spent the most on rocket launches**. This helps identify the most **active and high-budget space agencies.**

Company vs Mission Outcome



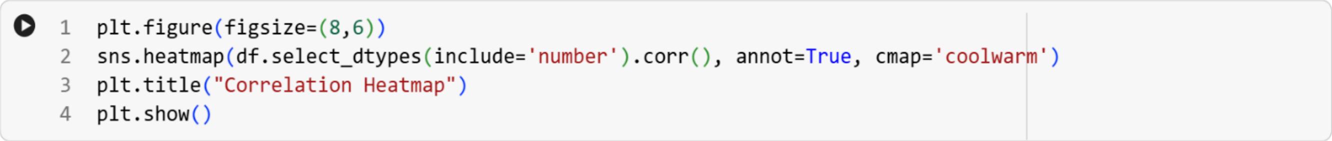
```python
plt.figure(figsize=(12,5))
sns.countplot(x='Company Name', hue='Status Mission', data=df)
plt.xticks(rotation=90)
plt.title("Company vs Mission Outcome")
plt.show()
```



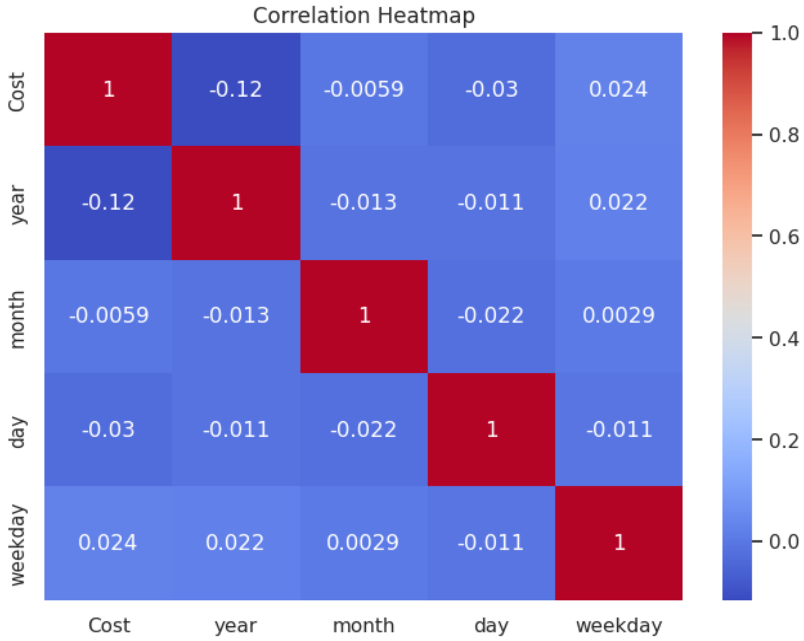
Explanations

- This chart shows how many missions each company has launched and what their outcomes were (Success, Failure, Prelaunch Failure, or Partial Failure).
- From the graph, we can clearly see that some companies have performed many more missions than others, and their mission results vary.

Correlation Heatmap



```
plt.figure(figsize=(8,6))
sns.heatmap(df.select_dtypes(include='number').corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



	Cost	year	month	day	weekday
Cost	1	-0.12	-0.0059	-0.03	0.024
year	-0.12	1	-0.013	-0.011	0.022
month	-0.0059	-0.013	1	-0.022	0.0029
day	-0.03	-0.011	-0.022	1	-0.011
weekday	0.024	0.022	0.0029	-0.011	1

This heatmap shows the **relationship between the numerical columns** in the dataset (Cost, Year, Month, Day, Weekday).
The values range from **-1 to +1**, where:

- **+1** means strong positive correlation
- **-1** means strong negative correlation
- **0** means no correlation

Insights:

1. Most space missions are successful.

Across all companies, successful missions dominate, showing strong reliability in global space launches.

2. A few companies launch the majority of missions.

Organizations like NASA, Roscosmos, ULA, SpaceX, and CASC contribute the highest number of launches compared to smaller companies.

3. Launch activity increased after 2000.

The number of missions rises sharply in the 2000s and 2010s, showing global growth in space exploration.

4. Rocket status shows technology improvements.

More rockets have become active in recent years, indicating advancements in reusable and modern rocket technology.

5. Cost does not strongly depend on year or date.

The correlation heatmap shows no strong link between cost and time-based features (year, month, day). Mission cost varies independently.

6. Most space missions belong to a small group of very active companies.

A few major agencies launch hundreds of missions, while many smaller ones have only a few launches in the entire dataset.

7. Missions with "Success" status are far more frequent than all types of failures combined.

The dataset shows that global mission reliability has become very strong overall.

8. Retired rockets appear mostly in older records, showing how launch vehicles evolve over time.

Newer missions rely on updated and active rocket models.

9. Some launch sites are used repeatedly for decades, confirming their importance as major global spaceports.

Examples include Cape Canaveral, Baikonur, and Jiuquan launch centers.

10. Costs vary heavily between missions, suggesting that different organizations follow different budget strategies.

High-budget missions usually belong to well-established companies.

11. Mission frequency increases sharply in recent years, showing rapid growth in global space activity.

The number of launches per year has significantly increased compared to early decades.

12. Companies with a long operational history show more variety in mission outcomes compared to new companies.

Older agencies have records of success, failure, and partial failure across decades.

Conclusion

This EDA gave us a clear and meaningful understanding of the global space mission landscape by exploring different factors such as mission outcomes, launch locations, rocket status, costs, and company-wise launch performance. Through various visualizations, we were able to identify which countries and organizations lead in space activities, how mission success rates differ across companies, and how launch patterns have changed over the years.

In real-world data science applications, such analysis plays an important role in identifying hidden patterns, supporting decision-making, and selecting the right features for building predictive models—such as predicting mission success, estimating costs, or analysing future launch trends.

Overall, this EDA provides a strong foundation for deeper analysis and gives valuable insights into the evolution and future direction of global space exploration.