In [7]:
```
import pandas as pd
df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")
df
```

Out[7]:

	Sr	Unnamed: 0	Company Name	Location	Datum	Detail	Status Rocket	Rocket	S M
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	Su
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	Su
2	2	2	SpaceX	Pad A, Boca Chica, Texas, USA	Tue Aug 04, 2020 23:57 UTC	Starship Prototype | 150 Meter Hop	StatusActive	NaN	Su
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	Su
4	4	4	ULA	SLC-41, Cape Canaveral AFS, Florida, USA	Thu Jul 30, 2020 11:50 UTC	Atlas V 541 | Perseverance	StatusActive	145	Su
...	...	...	...	...	...	...	...	...	
4319	4319	4319	US Navy	LC-18A, Cape Canaveral AFS, Florida, USA	Wed Feb 05, 1958 07:33 UTC	Vanguard | Vanguard TV3BU	StatusRetired	NaN	F
4320	4320	4320	AMBA	LC-26A, Cape Canaveral AFS, Florida, USA	Sat Feb 01, 1958 03:48 UTC	Juno I | Explorer 1	StatusRetired	NaN	Su
4321	4321	4321	US Navy	LC-18A, Cape Canaveral AFS, Florida, USA	Fri Dec 06, 1957 16:44 UTC	Vanguard | Vanguard TV3	StatusRetired	NaN	F
4322	4322	4322	RVSN USSR	Site 1/5, Baikonur Cosmodrome, Kazakhstan	Sun Nov 03, 1957 02:30 UTC	Sputnik 8K71PS | Sputnik-2	StatusRetired	NaN	Su

	Sr	Unnamed: 0	Company Name	Location	Datum	Detail	Status Rocket	Rocket	S M
4323	4323	4323	RVSN USSR	Site 1/5, Baikonur Cosmodrome, Kazakhstan	Fri Oct 04, 1957 19:28 UTC	Sputnik 8K71PS | Sputnik-1	StatusRetired	NaN	Su

4324 rows × 9 columns

In [3]:
```
df.nunique()
```

Out[3]:
```
Unnamed: 0         4324
Unnamed: 0.1       4324
Company Name         56
Location            137
Datum              4319
Detail             4278
Status Rocket         2
 Rocket              56
Status Mission        4
dtype: int64
```

In [4]:
```
df = df.drop(['Unnamed: 0', 'Unnamed: 0.1'], axis = 1)
df.head()
```

Out[4]:

	Company Name	Location	Datum	Detail	Status Rocket	Rocket	Status Mission
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	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	SpaceX	Pad A, Boca Chica, Texas, USA	Tue Aug 04, 2020 23:57 UTC	Starship Prototype | 150 Meter Hop	StatusActive	NaN	Success
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
4	ULA	SLC-41, Cape Canaveral AFS, Florida, USA	Thu Jul 30, 2020 11:50 UTC	Atlas V 541 | Perseverance	StatusActive	145	Success

In [5]:
```
df.describe()
```

Out[5]:

	Company Name	Location	Datum	Detail	Status Rocket	Rocket	Status Mission
count	4324	4324	4324	4324	4324	964	4324
unique	56	137	4319	4278	2	56	4
top	RVSN USSR	Site 31/6, Baikonur Cosmodrome, Kazakhstan	Tue Jun 26, 1973	Cosmos-3MRB (65MRB) | BOR-5 Shuttle	StatusRetired	450	Success
freq	1777	235	2	6	3534	136	3879

In [6]:
```
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           4324 non-null   object
 3   Detail          4324 non-null   object
 4   Status Rocket   4324 non-null   object
 5    Rocket         964 non-null    object
 6   Status Mission  4324 non-null   object
dtypes: object(7)
memory usage: 236.6+ KB
```

In [8]:
```
ds = df["Company Name"].value_counts().reset_index()[:28]
ds
```

Out[8]:

	index	Company Name
0	RVSN USSR	1777
1	Arianespace	279
2	General Dynamics	251
3	CASC	251
4	NASA	203
5	VKS RF	201
6	US Air Force	161
7	ULA	140
8	Boeing	136
9	Martin Marietta	114
10	SpaceX	100
11	MHI	84
12	Northrop	83
13	Lockheed	79
14	ISRO	76
15	Roscosmos	55
16	ILS	46
17	Sea Launch	36
18	ISAS	30
19	Kosmotras	22
20	US Navy	17
21	ESA	13
22	ISA	13
23	Eurockot	13
24	Rocket Lab	13
25	Blue Origin	12
26	IAI	11
27	ExPace	10

1) Bar Graph

In [18]:
```
pip install plotly
```

```
Collecting plotly
  Downloading plotly-5.18.0-py3-none-any.whl (15.6 MB)
Requirement already satisfied: packaging in c:\users\roari\anaconda3\lib\site
-packages (from plotly) (20.9)
Collecting tenacity>=6.2.0
  Downloading tenacity-8.2.3-py3-none-any.whl (24 kB)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\roari\anaconda3\l
ib\site-packages (from packaging->plotly) (2.4.7)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.18.0 tenacity-8.2.3
Note: you may need to restart the kernel to use updated packages.
```

In [23]:
```
import pandas as pd

data = {
    'Company Name': ['RVSN USSR', 'Arianespace', 'General Dynamics', 'CASC', '
    'Successful': [1000, 250, 120, 200, 180, 50, 120, 90, 80, 60, 100, 40, 50,
    'Unsuccessful': [777, 29, 131, 51, 23, 151, 41, 50, 56, 54, 0, 44, 33, 19,
}

df = pd.DataFrame(data)

print(df)
```

```
        Company Name  Successful  Unsuccessful
0          RVSN USSR        1000           777
1        Arianespace         250            29
2   General Dynamics         120           131
3               CASC         200            51
4               NASA         180            23
5             VKS RF          50           151
6       US Air Force         120            41
7                ULA          90            50
8             Boeing          80            56
9    Martin Marietta          60            54
10           SpaceX         100             0
11              MHI          40            44
12         Northrop          50            33
13         Lockheed          60            19
14             ISRO          70             6
```

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt

data = {
    'Company Name': ['RVSN USSR', 'Arianespace', 'General Dynamics', 'CASC', '
    'Successful': [1000, 250, 120, 200, 180, 50, 120, 90, 80, 60, 100, 40, 50,
    'Unsuccessful': [777, 29, 131, 51, 23, 151, 41, 50, 56, 54, 0, 44, 33, 19,
}

df = pd.DataFrame(data)

df.set_index('Company Name', inplace=True)

success_color = 'salmon'
failure_color = 'palegreen'

plt.figure(figsize=(12, 8))
df.plot(kind='bar', stacked=True, width=0.8, color=[success_color, failure_col
plt.xlabel('Company Name')
plt.ylabel('Number of Missions')
plt.title('Successful and Unsuccessful Missions by Companies')
plt.xticks(rotation=45)
plt.legend(['Successful', 'Unsuccessful'], loc='upper right')
plt.tight_layout()

plt.show()
```

```
<Figure size 864x576 with 0 Axes>
```

Conclusion

It can be concluded from the the above stack graph that Rassia uSSR have highest no. of successful and unsuccessful mission . Also Space X have negligible unsuccessful missions. The above graph shows that top 15 companies out of 56 companies in world where No. of

count of successful mission by top companies shown by "Salmon color" while unsuccessful mission count shown by "Pale green color". It is shown in stack bar graph. Now it can easily be concluded that Out of top 15 companies most mission are done by russia also it have highest no. of successful and unsuccessful mission. While From figure also concluded that Space x have negligeble amount of unsuccessful missions

Purpose of Bar Graph:

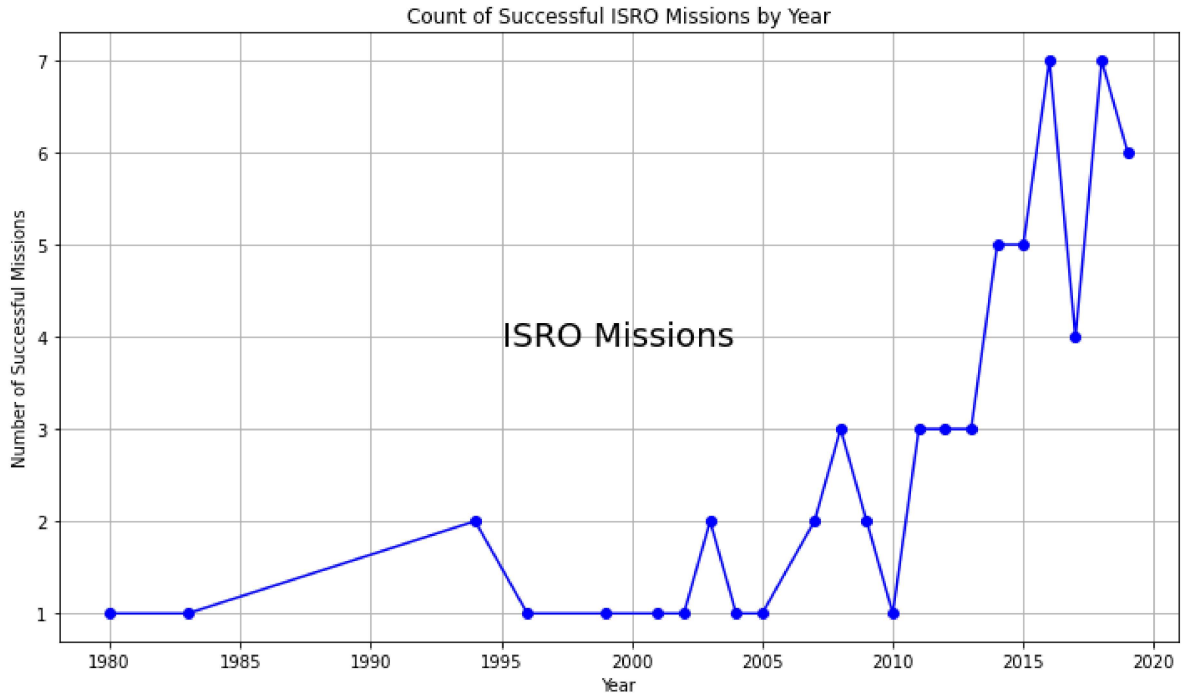
Comparison: The bar graph aims to compare the count of successful and unsuccessful missions among the top 15 space companies, showcasing their performance in terms of mission outcomes. Insights Gained:

Mission Volume Overview: The graph provides an overview of mission volumes conducted by top space companies, highlighting Russia's dominance in the number of missions conducted compared to other companies. Success-Failure Ratios: It showcases SpaceX's significantly lower count of unsuccessful missions relative to its total missions, indicating its reliability and efficiency in conducting successful space missions.

2) Line chart

In [46]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Datum'] = pd.to_datetime(df['Datum'], errors='coerce', utc=True)

isro_successful = df[(df['Company Name'] == 'ISRO') & (df['Status Mission'] ==

isro_successful['Year'] = isro_successful['Datum'].dt.year

isro_successful_by_year = isro_successful['Year'].value_counts().sort_index()

plt.figure(figsize=(10, 6))
plt.plot(isro_successful_by_year.index, isro_successful_by_year.values, marker
plt.xlabel('Year')
plt.ylabel('Number of Successful Missions')
plt.title('Count of Successful ISRO Missions by Year')
plt.grid(True)
plt.tight_layout()
plt.text(0.5, 0.5, 'ISRO Missions', ha='center', va='center', transform=plt.gc
plt.show()
```

```
<ipython-input-46-a7fbacf858aa>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  isro_successful['Year'] = isro_successful['Datum'].dt.year
```



In [47]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Datum'] = pd.to_datetime(df['Datum'], errors='coerce', utc=True)

isro_successful = df[(df['Company Name'] == 'ISRO') & (df['Status Mission'] ==

isro_successful['Year'] = isro_successful['Datum'].dt.year

isro_successful_by_year = isro_successful['Year'].value_counts().sort_index()

isro_successful_table = pd.DataFrame({'Year': isro_successful_by_year.index, '
print(isro_successful_table)
```

```
    Year  Successful Missions
0   1980                    1
1   1983                    1
2   1994                    2
3   1996                    1
4   1999                    1
5   2001                    1
6   2002                    1
7   2003                    2
8   2004                    1
9   2005                    1
10  2007                    2
11  2008                    3
12  2009                    2
13  2010                    1
14  2011                    3
15  2012                    3
16  2013                    3
17  2014                    5
18  2015                    5
19  2016                    7
20  2017                    4
21  2018                    7
22  2019                    6

<ipython-input-47-3a3272cd39e9>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  isro_successful['Year'] = isro_successful['Datum'].dt.year
```

Conclusion

From above line chart it can be conclude that ISRO have highest no. of successful mission in 2016 and 2018.

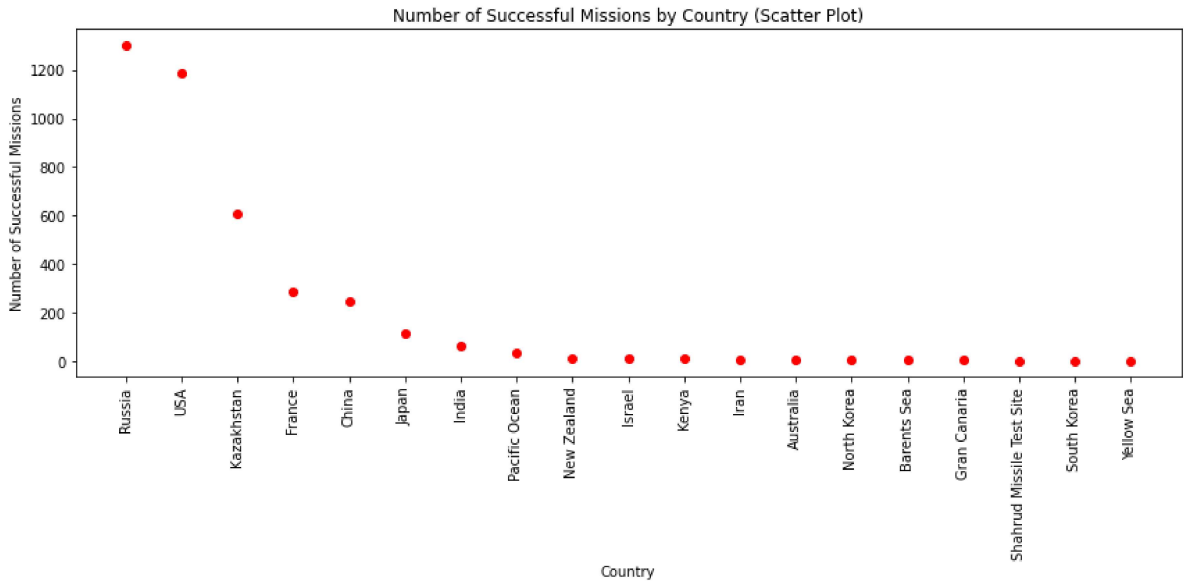
Purpose of Line Graph:

Temporal Trends: The line graph serves to illustrate the temporal trends of successful missions over the years, specifically for ISRO, providing a year-on-year perspective of their success rates. Insights Gained:

Yearly Performance: It highlights ISRO's outstanding performance in 2016 and 2018, showing peaks in the number of successful missions during these years. Comparative Success: By focusing on ISRO's success rates across different years, the graph implies variations in their mission success, identifying specific years of exceptional performance.

3) Scatter plot

In [53]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Country'] = df['Location'].apply(lambda x: x.split(",")[-1].strip())

successful_missions = df[df['Status Mission'] == 'Success']

success_count_by_country = successful_missions['Country'].value_counts()

df_success_count = success_count_by_country.reset_index()
df_success_count.columns = ['Country', 'Successful Missions']

plt.figure(figsize=(12, 6))
plt.scatter(df_success_count['Country'], df_success_count['Successful Missions
plt.xlabel('Country')
plt.ylabel('Number of Successful Missions')
plt.title('Number of Successful Missions by Country (Scatter Plot)')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



Conclusion

In above scatter plot , It can be concluded that Russia have highest no. of successful mission and thus act as Outlier in my dataset

Purpose of Scatter Plot:

Outlier Identification: The scatter plot aims to visualize the distribution of successful missions across different countries, with a particular focus on identifying any outliers or extreme values within the dataset. Insights Gained:

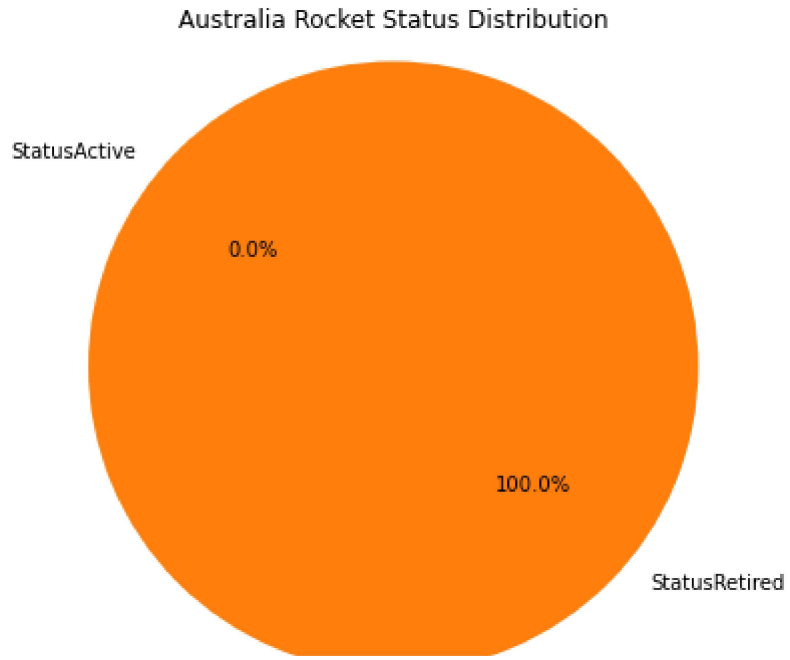
Identification of Outlier: The scatter plot reveals Russia as an outlier due to its significantly higher number of successful missions compared to other countries. This highlights Russia's exceptional performance and its distinct position within the dataset, acting as an outlier due to its exceptionally high success rate in missions.

4) Pie chart

In [38]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Country'] = df['Location'].apply(lambda x: x.split(",")[-1].strip())

country_status_counts = df.groupby(['Country', 'Status Rocket']).size().unstac
print(country_status_counts)
```

```
Status Rocket                  StatusActive  StatusRetired
Country
Australia                                 0              6
Barents Sea                               0              3
Brazil                                    3              0
China                                   223             45
France                                  113            190
Gran Canaria                              2              0
India                                    50             26
Iran                                      9              4
Israel                                    5              6
Japan                                    38             88
Kazakhstan                               44            657
Kenya                                     0              9
New Mexico                                4              0
New Zealand                              13              0
North Korea                               3              2
Pacific Missile Range Facility            1              0
Pacific Ocean                            36              0
Russia                                   36           1359
Shahrud Missile Test Site                 1              0
South Korea                               0              3
USA                                     208           1136
Yellow Sea                                1              0
```

In [39]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Country'] = df['Location'].apply(lambda x: x.split(",")[-1].strip())

country_status_counts = df.groupby(['Country', 'Status Rocket']).size().unstac

for country, counts in country_status_counts.iterrows():
    plt.figure(figsize=(6, 6))
    plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=140)
    plt.title(f'{country} Rocket Status Distribution')
    plt.axis('equal')
    plt.show()
```



From above data we can easily conclude Australiya , BarentSeas, kenya , South korea have 100% Retire status while brazil , new zealand, Gran Canaria , New mexico, Pacific region and Yellow sea region have 100% active status. Also China and Usa have highest count of active rocket status while Russia and Usa have highest retired status of rocket

Purpose of Pie Chart:

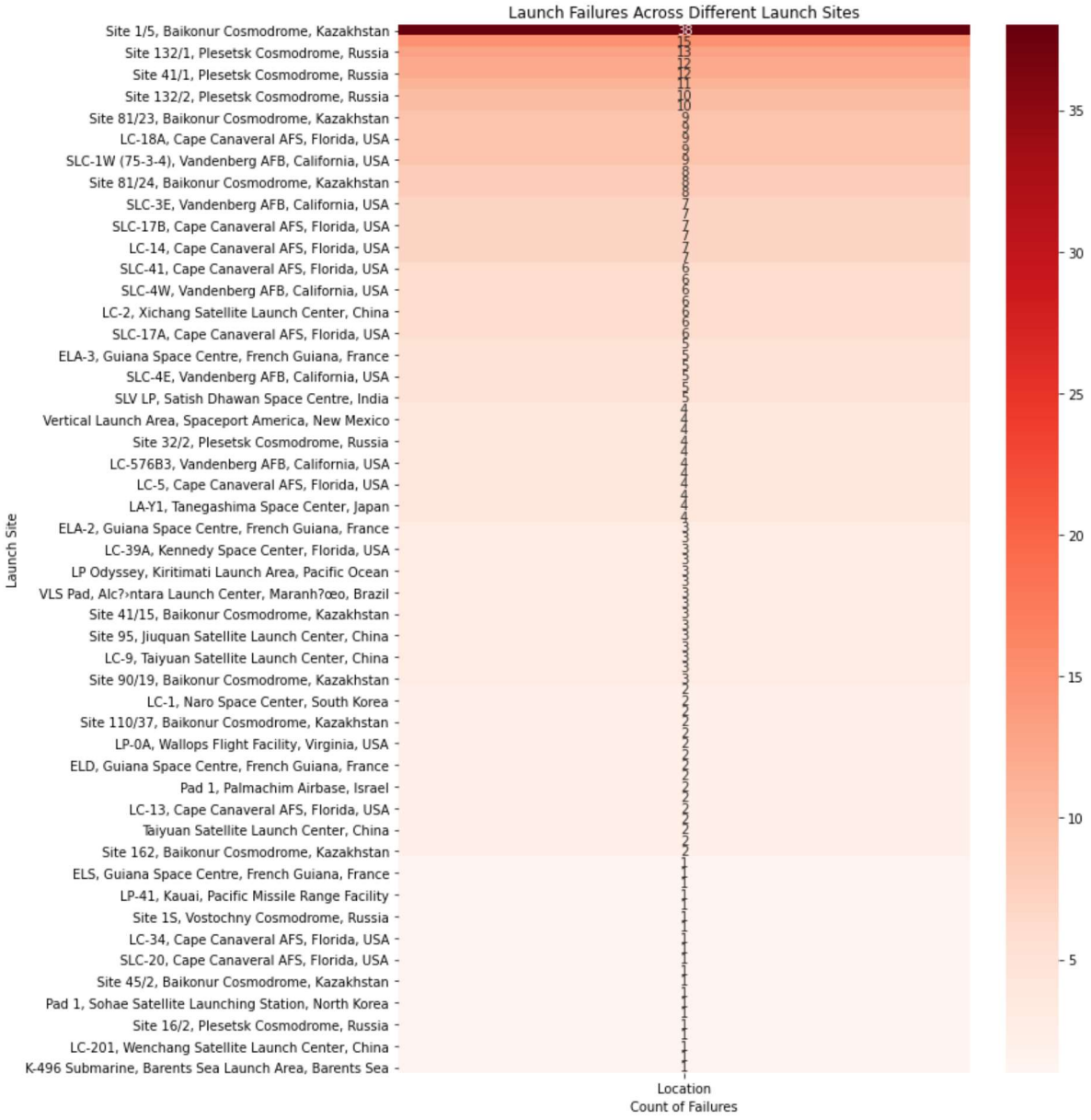
Status Distribution: The pie chart aims to represent the distribution of rocket status (active or retired) across different regions or countries in the dataset. It serves to visualize the proportion of rockets that are active or retired in various locations. Insights Gained:

Percentage Distribution: The pie chart illustrates that specific regions like Australiya, BarentSeas, Kenya, and South Korea exclusively have retired rockets, accounting for 100% of their status. Conversely, regions like Brazil, New Zealand, Gran Canaria, New Mexico, Pacific region, and Yellow Sea region possess rockets that are all active, representing a 100% active status. Comparative Analysis: Furthermore, the chart indicates that countries like China and the

USA have the highest count of active rocket statuses. Simultaneously, Russia and the USA have the highest count of retired rocket statuses, suggesting a comparative analysis between

5) Heatmap

In [58]:
```
failures_by_location = failed_missions['Location'].value_counts()

plt.figure(figsize=(10, 15))
sns.heatmap(pd.DataFrame(failures_by_location), cmap='Reds', annot=True, fmt='
plt.title('Launch Failures Across Different Launch Sites')
plt.xlabel('Count of Failures')
plt.ylabel('Launch Site')
plt.show()
```



Conclusion

It can be concluded that Barents Sea, , LC-201 Launch site of chaina and site 16/72 of Russia have lowest launch failureThe heatmap created from the code snippet you provided visualizes the count of launch failures across different launch sites. Here's an explanation: Y-Axis (Launch Site): Each row on the y-axis represents a launch site where missions have taken place. X-Axis (Count of Failures): The x-axis denotes the count of failures for each launch site. Color Intensity: The color intensity (in the chosen colormap 'Reds') represents the count of failures at each launch site. The darker shades indicate higher counts of failures, while lighter shades or no color indicate lower or no failures.

Purpose of Heat Map:

Launch Failure Comparison: The heat map aims to compare the count of launch failures across various launch sites. It provides a visual representation of the number of failures associated with each launch site, offering an easy-to-understand overview of failure counts. Insights Gained:

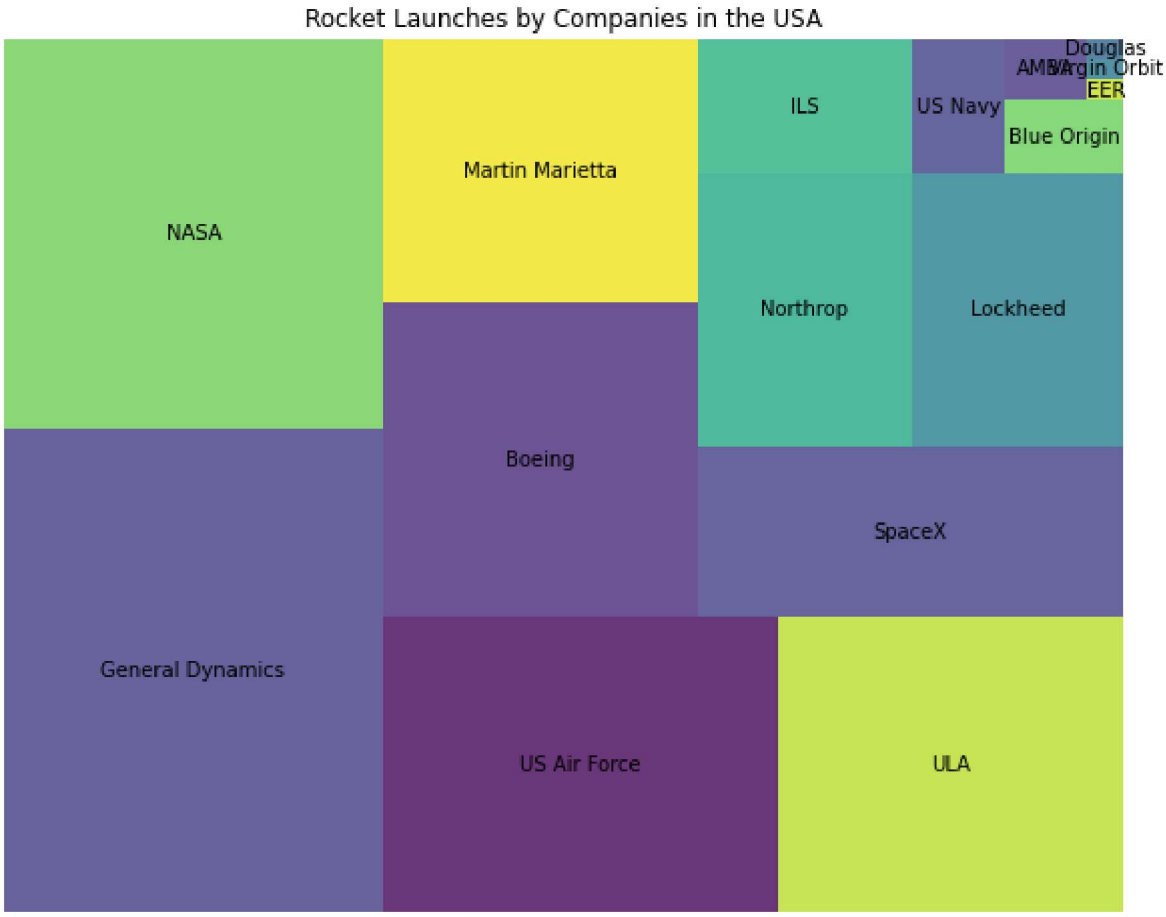
Identifying Low Failure Sites: The heatmap indicates that launch sites such as Barents Sea, LC-201 (China), and site 16/72 (Russia) have the lowest count of launch failures. These sites exhibit lighter shades or no color, signifying fewer or no failures compared to other sites. Relative Failure Rates: By observing the color intensity, darker shades represent higher counts of failures, helping to distinguish launch sites with more frequent failure occurrences. Conversely, lighter shades or the absence of color indicate launch sites with lower failure rates, enabling comparative analysis among different launch locations based on their failure counts.

6)Tree Map

In [2]:
```
pip install squarify
```

```
Collecting squarify
  Downloading squarify-0.4.3-py3-none-any.whl (4.3 kB)
Installing collected packages: squarify
Successfully installed squarify-0.4.3
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt
import squarify

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Country'] = df['Location'].apply(lambda x: x.split(', ')[-1])

usa_launches = df[df['Country'] == 'USA']

company_counts_usa = usa_launches['Company Name'].value_counts().reset_index()
company_counts_usa.columns = ['Company', 'Launch Count']

plt.figure(figsize=(10, 8))
squarify.plot(sizes=company_counts_usa['Launch Count'], label=company_counts_u
plt.axis('off')
plt.title("Rocket Launches by Companies in the USA")
plt.show()
```



Conclusion

The aboving Treemap shows that in USA , General Dynamics is the one who launch most of the rockets. Also tree map shows all the compies who launch rocket from USA location

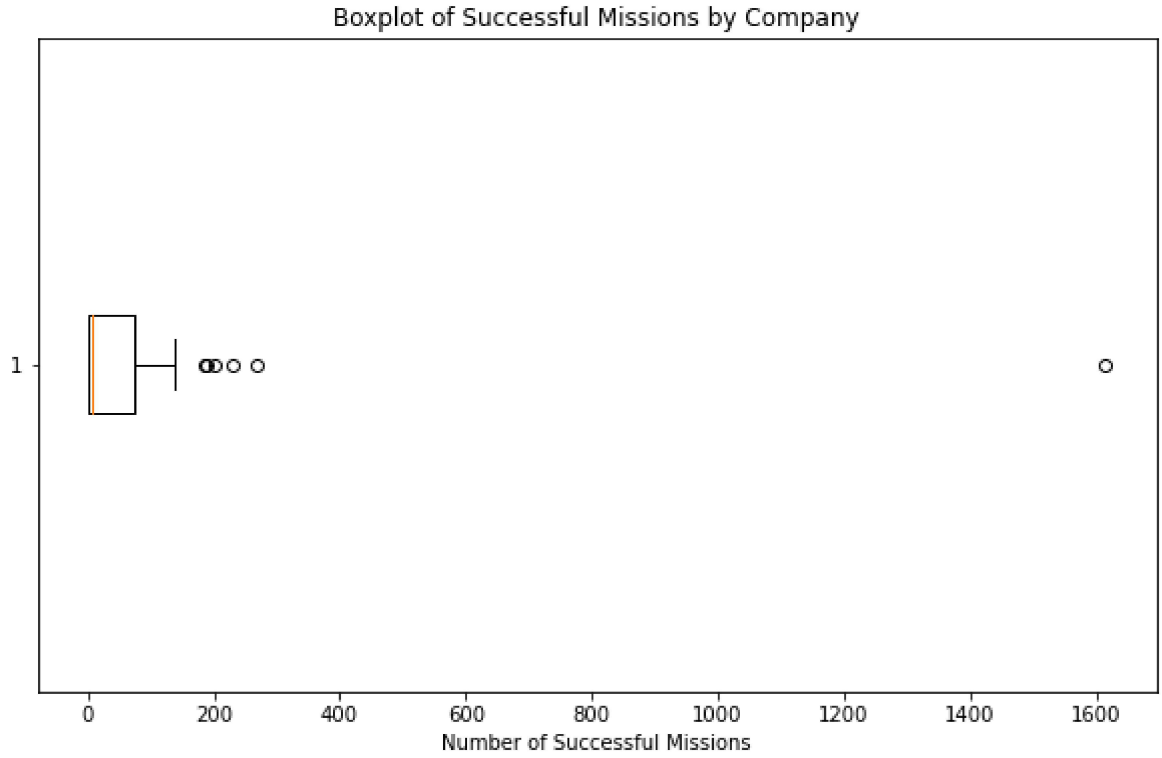
Purpose of Treemap:

Launch Distribution by Company: The treemap aims to visualize the distribution of rocket launches among various companies specifically within the USA. It provides an intuitive representation of the proportion of launches conducted by different companies within the USA. Insights Gained:

Launch Dominance by General Dynamics: The larger size of the block representing General Dynamics indicates that it has conducted the highest number of rocket launches among the listed companies within the USA. This highlights General Dynamics' significant contribution to the launches from this location. Company Comparison: The treemap offers a comparative view of various companies operating within the USA, showcasing the relative scale of their rocket launch activities. Smaller blocks represent companies with comparatively fewer launches, while larger blocks denote companies with more significant launch volumes.

7) Box plots

In [19]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

successful_missions = df[df['Status Mission'] == 'Success']


success_count_by_company = successful_missions.groupby('Company Name').size()

plt.figure(figsize=(10, 6))
plt.boxplot(success_count_by_company, vert=False)
plt.xlabel('Number of Successful Missions')
plt.title('Boxplot of Successful Missions by Company')
plt.show()
```



In [21]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

successful_missions = df[df['Status Mission'] == 'Success']

success_count_by_company = successful_missions['Company Name'].value_counts().
success_count_by_company.columns = ['Company Name', 'Successful Missions']

print(success_count_by_company)
```

	Company Name	Successful Missions
0	RVSN USSR	1614
1	Arianespace	269
2	CASC	231
3	General Dynamics	203
4	VKS RF	188
5	NASA	186
6	ULA	139
7	Boeing	131
8	US Air Force	129
9	Martin Marietta	100
10	SpaceX	94
11	MHI	80
12	Lockheed	74
13	Northrop	74
14	ISRO	63
15	Roscosmos	51
16	ILS	45
17	Sea Launch	33
18	ISAS	26
19	Kosmotras	21
20	Eurockot	12
21	Blue Origin	12
22	Rocket Lab	11
23	ESA	9
24	ExPace	9
25	IAI	9
26	ASI	9
27	Land Launch	6
28	CNES	6
29	JAXA	6
30	MITT	6
31	ISA	4
32	AMBA	4
33	Arm??e de l'Air	3
34	CASIC	3
35	Yuzhmash	2
36	US Navy	2
37	OKB-586	2
38	SRC	2
39	KCST	2
40	Khrunichev	1
41	Starsem	1
42	UT	1
43	RAE	1
44	IRGC	1
45	KARI	1
46	CECLES	1
47	Douglas	1
48	i-Space	1

Conclusion

This code will generate a boxplot showing the distribution of successful missions by company. From Another code it can be counted that By USSR there are 1614 which shows that its outlier of our dataset

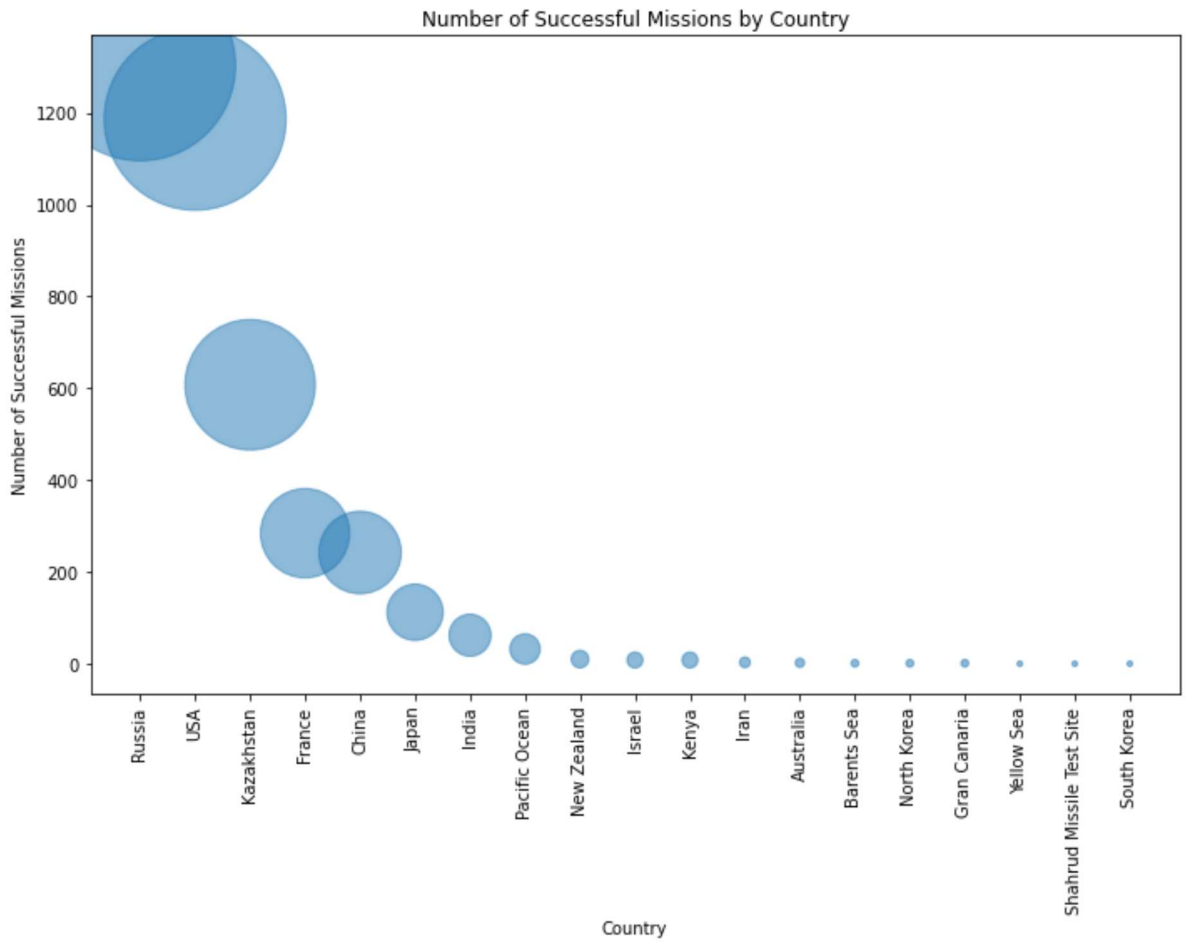
Purpose of Boxplots:

Distribution Comparison: Boxplots are utilized to showcase the distribution of a numerical dataset, specifically the successful mission counts by different companies. It helps in comparing the central tendency, spread, and potential outliers across multiple categories or groups.
Insights Gained:

Outlier Identification - USSR: The generated boxplot indicates that the USSR has an outlier status in terms of the count of successful missions. This conclusion is further supported by another code snippet that specifically counts 1614 missions by the USSR, highlighting its substantial and exceptional contribution compared to other companies. The outlier status suggests the USSR's significant dominance or exceptional activity compared to the rest of the companies in the dataset.

8) Bubble chart

In [25]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

df['Country'] = df['Location'].apply(lambda x: x.split(",")[-1].strip())

successful_missions = df[df['Status Mission'] == 'Success']

success_count_by_country = successful_missions['Country'].value_counts().reset
success_count_by_country.columns = ['Country', 'Successful Missions']

plt.figure(figsize=(10, 8))
plt.scatter(
    success_count_by_country['Country'],
    success_count_by_country['Successful Missions'],
    s=success_count_by_country['Successful Missions'] * 10,  # Bubble size bas
    alpha=0.5,
    cmap='viridis',
)
plt.xlabel('Country')
plt.ylabel('Number of Successful Missions')
plt.title('Number of Successful Missions by Country')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



Conclusion

From above data it can be concluded that Rassia and USA have most no of successful mission while above bubble graph shows South korea have lowes no. of successful mission

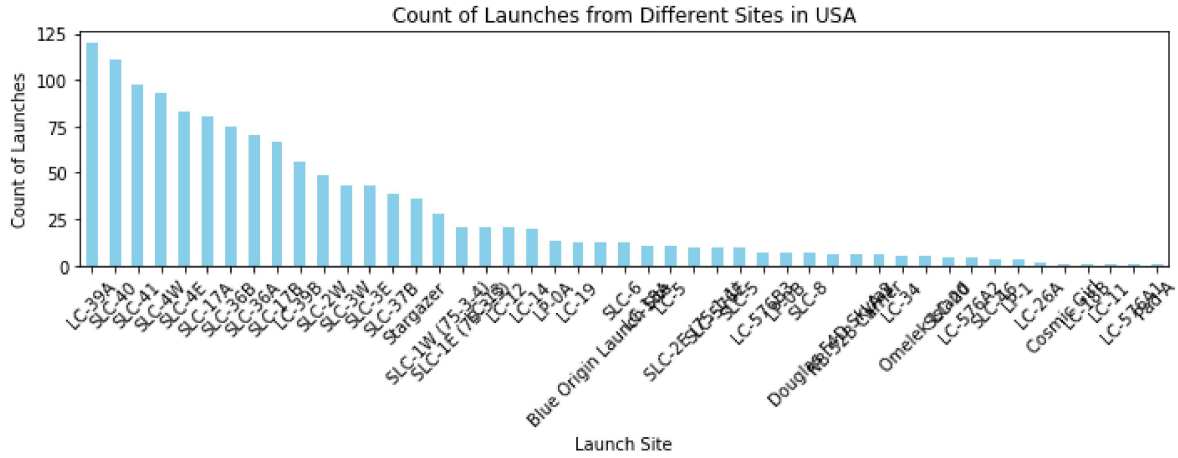
Purpose of Bubble Chart:

Comparison and Visualization: Bubble charts are utilized to represent data points in three dimensions: x-axis, y-axis, and the size of bubbles. They help compare multiple entities across two variables, showcasing the relationships between these variables and the frequency or magnitude of a third variable indicated by the size of the bubbles. Insights Gained:

Comparative Analysis - Russia and USA: The bubble chart indicates that Russia and USA have the highest count of successful missions, showcasing their dominance or significant contributions in the dataset. Relative Comparison - South Korea: Conversely, the size of the bubble for South Korea is notably smaller, suggesting a lower count of successful missions compared to Russia and the USA. This implies a comparative difference in their contributions to successful missions in the dataset.

9) Histogram

In [30]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\roari\Downloads\Space_Corrected.csv")

usa_launches = df[df['Location'].str.contains('USA', case=False, na=False)]
usa_sites = usa_launches['Location'].apply(lambda x: x.split(', ')[0])

plt.figure(figsize=(10, 4))
usa_sites.value_counts().plot(kind='bar', color='skyblue')
plt.xlabel('Launch Site')
plt.ylabel('Count of Launches')
plt.title('Count of Launches from Different Sites in USA')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Conclusion

Above Histogram which is type of bar graph shows that LC-394 have highest no of Launch

Purpose of Histogram:

Distribution Analysis: Histograms are used to display the distribution or frequency of numerical data within intervals or bins. They visualize the spread and concentration of data points within specific ranges, providing insights into the dataset's distribution. Insights Gained:

Identifying the Highest Count: In this case, the histogram highlights that LC-394, presumably a launch site or location, holds the highest count of launches compared to other locations. This information is crucial in understanding the frequency or concentration of launches from different sites within the dataset.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: