In [1]:
```
# Requests allows us to make HTTP requests which we will use to get data from an
import requests
# Pandas is a software library written for the Python programming language for da
import pandas as pd
# NumPy is a library for the Python programming language, adding support for larg
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all collumns of a dataframe
pd.set_option('display.max_columns', None)
# Setting this option will print all of the data in a feature
pd.set_option('display.max_colwidth', None)
```

In [2]:
```
# Takes the dataset and uses the rocket column to call the API and append the dat
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).
        BoosterVersion.append(response['name'])
```

In [3]:
```
# Takes the dataset and uses the launchpad column to call the API and append the
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

In [4]:
```
# Takes the dataset and uses the payloads column to call the API and append the d
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).j
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

In [5]:
```
# Takes the dataset and uses the cores column to call the API and append the data
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+co
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_typ
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

In [7]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"sh
ips":[]},"links":{"patch":{"small":"https://images2.imgbox.com/3c/0e/T8iJcSN3
_o.png","large":"https://images2.imgbox.com/40/e3/GypSkayF_o.png"},"reddit":
{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"smal
l":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watc
h?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/
2196-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://
en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.0
00Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d
0d95eda69955f709d1eb","success":false,"failures":[{"time":33,"altitude":nul
l,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds
and loss of vehicle","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b5
b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":
1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":114323
9400,"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcomi
ng":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":1,"gridfins":f
alse,"legs":false,"reused":false,"landing_attempt":false,"landing_success":nu
ll,"landing_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launc
h_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":fal
```

In [8]:
```
'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-S
```

In [9]:
```
response.status_code
```

Out[9]:
```
200
```

In [10]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

In [11]:
```
# Get the head of the dataframe
data.head()
```

Out[11]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	(p sh
2	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	t(a
3	2008-09-20T00:00:00.000Z	1.221869e+09	False	0.0	5e9d0d95eda69955f709d1eb	True	

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success
4	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	True

In [12]:
```
# Lets take a subset of our dataframe keeping only the features we want and the f
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_u

# We will remove rows with multiple cores because those are falcon rockets with 2
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single va
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

In [13]:
```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

In [14]:
```
BoosterVersion
```

Out[14]:
```
[]
```

In [15]:
```
# Call getBoosterVersion
getBoosterVersion(data)
```

In [16]:
```
BoosterVersion[0:5]
```

Out[16]:
```
['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

In [17]:
```
# Call getLaunchSite
getLaunchSite(data)
```

In [18]:
```
# Call getPayloadData
getPayloadData(data)
```

In [19]:
```
# Call getCoreData
getCoreData(data)
```

In [20]:
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

In [21]:
```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

In [22]:
```
# Show the head of the dataframe
data
```

Out[22]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	Grid
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	F
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	F
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	F
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	F
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	F
...	...	...	...	...	...	...	...	...	
89	102	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	
90	103	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	
91	104	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	
92	105	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	
93	106	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	

94 rows × 17 columns

In [23]:
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
data_falcon9
```

Out[23]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	Grid
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	F
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	F
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	F
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	F
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	F
...	...	...	...	...	...	...	...	...	
89	102	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	
90	103	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	
91	104	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	
92	105	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	
93	106	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	

90 rows × 17 columns

In [24]:
```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9.shape
```

```
C:\Users\HP\anaconda3\lib\site-packages\pandas\core\indexing.py:1773: SettingWi
thCopyWarning:
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/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  self._setitem_single_column(ilocs[0], value, pi)
```

Out[24]:
```
(90, 17)
```

In [25]:
```
data_falcon9.isnull().sum()
```

Out[25]:
```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

In [26]:
```
# Calculate the mean value of PayloadMass column
Mean_PayloadMass = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_Pa
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_8264/1566627384.py:4: SettingWithCopyW
arning:
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/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mea
n_PayloadMass)
```

In [27]:
```
data_falcon9.isnull().sum()
```

Out[27]:
```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

In [28]:
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```