In [5]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [6]:
```
df=pd.read_csv("C:\Users\ASUS\Desktop\dataset\Data set Part 2 - propulsion_module.csv")
```

In [7]:
```
df
```

Out[7]:

	Parameter	Specifications
0	Lunar Polar Orbit	From 170 x 36500 km to lunar polar orbit
1	Mission life	Carrying Lander Module & Rover upto ~100 x 100...
2	Structure	Modified version of 3.5 K
3	Dry Mass	448.62 kg (including pressurant)
4	Propellant Mass	1696.19 kg
5	Total PM Mass	2145.01 kg
6	Power Generation	738 W, Summer solstices and with bias
7	Communication	S-Band Transponder (TTC) 46° with EDFA
8	Attitude Sensors	CASS, DIAN' Micro star sensor
9	Propulsion System	Bi-Propellant Propulsion System (MMH + MON3)
10	undefined	undefined
11	# Lander Module dataframe	undefined
12	undefined	undefined
13	Parameter	Specifications
14	-	-
15	Mission life	1 Lunar day (14 Earth days)
16	Mass	1749.86 kg including Rover
17	Power	738 W (Winter solstice)
18	Payloads	3
19	Dimensions (mm3)	2000 x 2000 x 1166
20	Communication	ISDN, Ch-2 Orbiter, Rover
21	Landing site	69.367621 S, 32.348126 E
22	undefined	undefined
23	# Rover dataframe	undefined
24	undefined	undefined
25	Parameter	Specifications
26	-	-
27	Mission Life	1 Lunar day
28	Mass	26 kg
29	Power	50 W
30	Payloads	2
31	Dimensions (mm3)	917 x 750 x 397
32	Communication	Lander

In [8]:
```
data={
    "parameters":[
        "Lunar Polar Orbit",
        "Mission life",
        "Structure",
        "Dry Mass",
        "Propellant Mass",
        "Total PM Mass",
        "Power Generation",
        "Communication",
        "Attitude Sensors",
        "Propulsion System"
    ],
    "Specifications":[
        "From 170 x 36500 km to lunar polar orbit",
        "Carrying Lander Module & Rover upto ~100 x 100 km launch injection. Subsequently, operation of experimental payload for a period of 3 to 6 months.",
        "Modified version of 3.5 K",
        "448.62 kg (including pressurant)",
        "1696.19 kg",
        "2145.01 kg",
        "738 W, Summer solstices and with bias",
        "S-Band Transponder (TTC) 46° with EDFA",
        "CASS, DIAN' Micro star sensor",
        "Bi-Propellant Propulsion System (MMH + MON3)"
    ]
}
```

In [9]:
```
df=pd.DataFrame(data)
```

In [10]:
```
df
```

Out[10]:

	parameters	Specifications
0	Lunar Polar Orbit	From 170 x 36500 km to lunar polar orbit
1	Mission life	Carrying Lander Module & Rover upto ~100 x 100...
2	Structure	Modified version of 3.5 K
3	Dry Mass	448.62 kg (including pressurant)
4	Propellant Mass	1696.19 kg
5	Total PM Mass	2145.01 kg
6	Power Generation	738 W, Summer solstices and with bias
7	Communication	S-Band Transponder (TTC) 46° with EDFA
8	Attitude Sensors	CASS, DIAN' Micro star sensor
9	Propulsion System	Bi-Propellant Propulsion System (MMH + MON3)

In [11]:
```
data={
    "parameters":[
        "Mission life",
        "Mass",
        "Power",
        "Payloads",
        "Dimensions (mm3)",
        "Communication",
        "Landing site"
    ],
    "Specifications":[
        "1 Lunar day (14 Earth days)",
        "1749.86 kg including Rover",
        "738 W (Winter solstice)",
        "3",
        "2000 x 2000 x 1166",
        "ISDN, Ch-2 Orbiter, Rover",
        "69.367621 S, 32.348126 E"
    ]
}
```

In [12]:
```
lander_df=pd.DataFrame(data)
```

In [13]:
```
lander_df
```

Out[13]:

	Parameters	Specifications
0	Mission life	1 Lunar day (14 Earth days)
1	Mass	1749.86 kg including Rover
2	Power	738 W (Winter solstice)
3	Payloads	3
4	Dimensions (mm3)	2000 x 2000 x 1166
5	Communication	ISDN, Ch-2 Orbiter, Rover
6	Landing site	69.367621 S, 32.348126 E

In [14]:
```
data={
    "Parameters":[
        "Mission Life",
        "Mass",
        "Power",
        "Payloads",
        "Dimensions (mm3)",
        "Communication"
    ],
    "Specifications":[
        "1 Lunar day",
        "26 kg",
        "50 W",
        "2",
        "917 x 750 x 397",
        "Lander"
    ]
}
```

In [15]:
```
rover_df=pd.DataFrame(data)
```

In [16]:
```
rover_df
```

Out[16]:

	Parameters	Specifications
0	Mission Life	1 Lunar day
1	Mass	26 kg
2	Power	50 W
3	Payloads	2
4	Dimensions (mm3)	917 x 750 x 397
5	Communication	Lander

In [17]:
```
import re
```

In [18]:
```
def extract_numerical_value(spec):
    numeric_pattern = r'(\d+\.?\d*)'
    range_pattern = r'(\d+\.?\d*)\s*[-~]\s*(\d+\.?\d*)'
    combined_pattern = r'(' + range_pattern + r')|(' + numeric_pattern + r')'
    matches = re.findall (combined_pattern, spec)
    if matches:
        return float (matches[0][0])
    else:
        return None
```

In [19]:
```
df["numerical value"]=df["Specifications"].apply(extract_numerical_value)
```

In [20]:
```
df
```

Out[20]:

	parameters	Specifications	numerical value
0	Lunar Polar Orbit	From 170 x 36500 km to lunar polar orbit	170.0
1	Mission life	Carrying Lander Module & Rover upto ~100 x 100...	100.0
2	Structure	Modified version of 3.5 K	NaN
3	Dry Mass	448.62 kg (including pressurant)	NaN
4	Propellant Mass	1696.19 kg	NaN
5	Total PM Mass	2145.01 kg	NaN
6	Power Generation	738 W, Summer solstices and with bias	NaN
7	Communication	S-Band Transponder (TTC) 46° with EDFA	NaN
8	Attitude Sensors	CASS, DIAN' Micro star sensor	NaN
9	Propulsion System	Bi-Propellant Propulsion System (MMH + MON3)	NaN

In [21]:
```
lander_df["numerical value"]=lander_df["Specifications"].apply(extract_numerical_value)
```

In [22]:
```
lander_df
```

Out[22]:

	Parameters	Specifications	numerical value
0	Mission life	1 Lunar day (14 Earth days)	NaN
1	Mass	1749.86 kg including Rover	NaN
2	Power	738 W (Winter solstice)	NaN
3	Payloads	3	NaN
4	Dimensions (mm3)	2000 x 2000 x 1166	2000.000000
5	Communication	ISDN, Ch-2 Orbiter, Rover	NaN
6	Landing site	69.367621 S, 32.348126 E	32.348126

In [23]:
```
rover_df["numerical value"]=rover_df["Specifications"].apply(extract_numerical_value)
```

In [24]:
```
rover_df
```

Out[24]:

	Parameters	Specifications	numerical value
0	Mission Life	1 Lunar day	NaN
1	Mass	26 kg	NaN
2	Power	50 W	NaN
3	Payloads	2	NaN
4	Dimensions (mm3)	917 x 750 x 397	750.0
5	Communication	Lander	NaN

In [25]:
```
import math
```

In [26]:
```
rover_mass = 26
lander_dry_mass = 1749.86
total_mass = rover_mass + lander_dry_mass
delta_v_required = 2000
isp_lander_engine = 300
propellant_mass_required = total_mass * math.exp(delta_v_required / (isp_lander_engine) - total_mass
propellant_mass_required = round(propellant_mass_required, 2)
```

In [27]:
```
rover_power_requirement=50
lander_battery_capacity = 2000
rover_operating_time_hours = lander_battery_capacity / rover_power_requirement
```

In [32]:
```
print("Mass Budget:")
print (f"Lander mass: {lander_dry_mass} kg")
print (f"Rover mass: {rover_mass} kg")
print (f"Propellant mass required: {propellant_mass_required} kg (matches value in lander DataFrame)")
print("\nPower Budget:")
print (f"Rover power requirement: {rover_power_requirement} W")
print (f"Lander battery capacity: {lander_battery_capacity} Wh")
print (f"Rover can operate for {rover_operating_time_hours: .2f} hours on stored power")
print("\nMobility Assessment:")
print("Rover's mass of the rover allows for mobility on uneven lunar surface")
print("\nNumber of payloads for science measurements is 2")
```

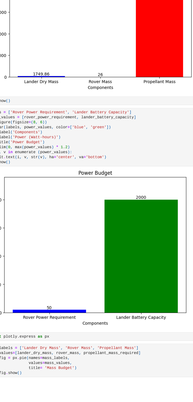
```
Mass Budget:
Lander mass: 1749.86 kg
Rover mass: 26 kg
Propellant mass required: 261785.13 kg (matches value in lander DataFrame)

Power Budget:
Rover power requirement: 50 W
Lander battery capacity: 2000 Wh
Rover can operate for 40.00 hours on stored power

Mobility Assessment:
Rover's mass of the rover allows for mobility on uneven lunar surface

Number of payloads for science measurements is 2
```

In [33]:
```
import matplotlib.pyplot as plt
labels = ['Lander Dry Mass', 'Rover Mass', 'Propellant Mass']
mass_values = [lander_dry_mass, rover_mass, propellant_mass_required]
plt.figure(figsize=(8, 6))
plt.bar(labels, mass_values, color=['blue', 'green', 'red'])
plt.xlabel('Components')
plt.ylabel('Mass (kg)')
plt.title('Mass Budget')
for i, v in enumerate(mass_values):
    plt.text(i, v, str(v), ha='center', va='bottom')
```



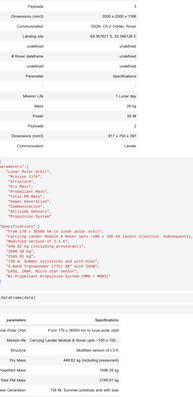
In [75]:
```
plt.show()
```

In [102]:
```
labels = ['Rover Power Requirement', 'Lander Battery Capacity']
power_values = [rover_power_requirement, lander_battery_capacity]
plt.figure(figsize=(8, 6))
plt.bar(labels, power_values, color=['blue', 'green'])
plt.xlabel('Components')
plt.ylabel('Power (watt-hours)')
plt.title('Power Budget')
plt.ylim(0, max(power_values) * 1.2)
for i, v in enumerate(power_values):
    plt.text(i, v, str(v), ha='center', va='bottom')
plt.show()
```

In [36]:
```
import plotly.express as px
```

In [37]:
```
mass_labels = ['Lander Dry Mass', 'Rover Mass', 'Propellant Mass']
mass_values = [lander_dry_mass, rover_mass, propellant_mass_required]
mass_fig = px.bar(x=mass_labels,
                  y=mass_values,
                  title='Mass Budget')

mass_fig.show()
```



In [42]:
```
labels = ['Rover Power Requirement', 'Lander Battery Capacity']
power_values = [rover_power_requirement, lander_battery_capacity]
power_fig = px.bar(x=labels,
                   y=power_values,
                   title='Power Budget')

power_fig.show()
```