

Exploring NASA's Data Universe

```
In [1]: 
#-- Import pandas ---
import pandas as pd

#-- Read in dataset---
df=pd.read_csv('nasa.csv')

#-- Data discovery ---
df.head()
```

	Name	Year	GroupNum	Status	Birth_Date	Birth_Place	Gender	Alma_Mater	Undergraduate_Major	
0	Alan B. Shepard Jr.	1959	1	Deceased	18-11-1923	East Derry, NH	Male	US Naval Academy	Naval Sciences	
1	Alan G. Poindexter	1998	17	Deceased	1961-05-11	Pasadena, CA	Male	Georgia Institute of Technology; US Naval Post...	Aerospace Engineering	
2	Alan L. Bean	1963	3	Deceased	15-03-1932	Wheeler, TX	Male	University of Texas	Aeronautical Engineering	
3	Albert Sacco Jr.	1963	3	Retired	1949-03-05	Boston, MA	Male	Northeastern University; MIT	Chemical Engineering	
4	Alfred M. Worden	1966	5	Retired	1932-07-02	Jackson, MI	Male	US Military Academy; University of Michigan	Military Science	

Exploring data completeness

```
In [2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 357 entries, 0 to 356
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                  357 non-null    object
1   Year                  357 non-null    int64
2   GroupNum              357 non-null    int64
3   Status                357 non-null    object
4   Birth_Date            357 non-null    object
5   Birth_Place           357 non-null    object
6   Gender                357 non-null    object
7   Alma_Mater            356 non-null    object
8   Undergraduate_Major   357 non-null    object
9   Graduate_Major        357 non-null    object
10  Military_Rank          357 non-null    object
11  Military_Branch        357 non-null    object
12  Space_Flights          357 non-null    int64
13  Space_Flight_hr        357 non-null    int64
14  Space_Walks            357 non-null    int64
15  Space_Walks_hr         357 non-null    float64
16  Missions               357 non-null    object
```

17 Death_Date 54 non-null object
18 Death_Mission 48 non-null object
dtypes: float64(1), int64(5), object(13)
memory usage: 53.1+ KB

In [3]: df.describe()

Out[3]:

	Year	GroupNum	Space_Flights	Space_Flight_hr	Space_Walks	Space_Walks_hr
count	357.000000	357.000000	357.000000	357.000000	357.000000	357.000000
mean	1984.739496	11.249300	2.364146	1249.266106	1.246499	7.707283
std	13.347850	5.221179	1.428700	1896.759857	2.056989	13.367973
min	1959.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	1978.000000	8.000000	1.000000	289.000000	0.000000	0.000000
50%	1987.000000	12.000000	2.000000	590.000000	0.000000	0.000000
75%	1996.000000	16.000000	3.000000	1045.000000	2.000000	12.000000
max	2009.000000	20.000000	7.000000	12818.000000	10.000000	67.000000

In [4]: df.tail()

Out[4]:

	Name	Year	GroupNum	Status	Birth_Date	Birth_Place	Gender	Alma_Mater	Undergraduate_Maj
352	William M. Shepherd	1984	10	Retired	26-07-1949	Oak Ridge, TN	Male	US Naval Academy; MIT	Aerospace Engineeri
353	William R. Pogue	1966	5	Retired	23-01-1930	Okemah, OK	Male	Oklahoma Baptist University; Oklahoma State Un...	Educati
354	William S. McArthur Jr.	1990	3	Management	26-07-1951	Laurinburg, NC	Male	US Military Academy; Georgia Institute of Tech...	Applied Science Engineeri
355	Winston E. Scott	1992	14	Retired	1950-06-08	Miami, FL	Male	Florida State University; US Naval Postgraduat...	Mu
356	Yvonne D. Cagle	1996	16	Management	24-04-1959	West Point, NY	Female	San Francisco State University	Biochemis

In [5]: num_rows,num_cols=df.shape

In [6]: num_rows

Out[6]: 357

In [7]: num_cols

Out[7]: 19

```
In [8]: # ---to evaluate whether any value is missing in a dataframe ---'true' means some values  
df.isnull().any()
```

```
Out[8]: Name                False  
Year                False  
GroupNum            False  
Status              False  
Birth_Date          False  
Birth_Place         False  
Gender              False  
Alma_Mater           True  
Undergraduate_Major False  
Graduate_Major       False  
Military_Rank        False  
Military_Branch      False  
Space_Flights        False  
Space_Flight_hr      False  
Space_Walks          False  
Space_Walks_hr       False  
Missions            False  
Death_Date           True  
Death_Mission        True  
dtype: bool
```

```
In [9]: # --to determine how many missing values exist in each column  
df.isnull().sum()
```

```
Out[9]: Name                0  
Year                0  
GroupNum            0  
Status              0  
Birth_Date          0  
Birth_Place         0  
Gender              0  
Alma_Mater           1  
Undergraduate_Major  0  
Graduate_Major       0  
Military_Rank        0  
Military_Branch      0  
Space_Flights        0  
Space_Flight_hr      0  
Space_Walks          0  
Space_Walks_hr       0  
Missions            0  
Death_Date           303  
Death_Mission        309  
dtype: int64
```

```
In [10]: #-- to find duplicate rows--  
# since there is no 'true' it means there is no duplicate row exist  
df.duplicated()
```

```
Out[10]: 0      False  
1      False  
2      False  
3      False  
4      False  
...  
352    False  
353    False  
354    False  
355    False  
356    False  
Length: 357, dtype: bool
```

Data Refinement for Astronaut data

1. Handling missing rows

```
In [11]: # --Since 'Alma-Mater' column consists of one missing row, missing row in that column is  
df.dropna(subset=['Alma_Mater'], inplace=True)
```

```
In [12]: df.isnull().sum()
```

```
Out[12]: Name                0  
Year                0  
GroupNum            0  
Status              0  
Birth_Date          0  
Birth_Place         0  
Gender              0  
Alma_Mater          0  
Undergraduate_Major 0  
Graduate_Major      0  
Military_Rank        0  
Military_Branch      0  
Space_Flights        0  
Space_Flight_hr      0  
Space_Walks          0  
Space_Walks_hr       0  
Missions             0  
Death_Date          302  
Death_Mission        308  
dtype: int64
```

```
In [13]: df.shape
```

```
Out[13]: (356, 19)
```

```
In [14]: # If there is Death_Date, there should be Death_Mission, a row with death_date but no de  
# then that particular row can be removed  
  
# So first, filtering the rows where 'death mission' shows null values but 'death date'  
filtered_rows=df[(df['Death_Mission'].isnull()) & (~df['Death_Date'].isnull())]
```

```
In [15]: # there were 6 rows with null values in 'death mission' and not null values in 'death da  
filtered_rows.shape
```

```
Out[15]: (6, 19)
```

```
In [16]: df.drop(filtered_rows.index, inplace=True)
```

```
In [17]: df.shape
```

```
Out[17]: (350, 19)
```

```
In [18]: df.isnull().sum()
```

```
Out[18]: Name                0  
Year                0  
GroupNum            0  
Status              0  
Birth_Date          0  
Birth_Place         0  
Gender              0
```

```

Alma_Mater          0
Undergraduate_Major 0
Graduate_Major      0
Military_Rank        0
Military_Branch      0
Space_Flights        0
Space_Flight_hr      0
Space_Walks          0
Space_Walks_hr       0
Missions            0
Death_Date          302
Death_Mission        302
dtype: int64

```

2. Datatype conversion

```
In [19]: df.dtypes
```

```

Out[19]: Name          object
Year            int64
GroupNum        int64
Status          object
Birth_Date      object
Birth_Place     object
Gender          object
Alma_Mater      object
Undergraduate_Major object
Graduate_Major  object
Military_Rank    object
Military_Branch  object
Space_Flights    int64
Space_Flight_hr  int64
Space_Walks      int64
Space_Walks_hr   float64
Missions        object
Death_Date       object
Death_Mission    object
dtype: object

```

```
In [20]: ## Both birth_date and death_date are object. it has to be converted to datetime format
df['Birth_Date'] = pd.to_datetime(df['Birth_Date'], format=None, errors='coerce')
```

C:\Users\bthan\AppData\Local\Temp\ipykernel_29708\1558921612.py:2: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.

```
df['Birth_Date'] = pd.to_datetime(df['Birth_Date'], format=None, errors='coerce')
```

```
In [21]: df['Death_Date'] = pd.to_datetime(df['Death_Date'], format=None, errors='coerce')
```

C:\Users\bthan\AppData\Local\Temp\ipykernel_29708\1290535510.py:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consistent parsing.

```
df['Death_Date'] = pd.to_datetime(df['Death_Date'], format=None, errors='coerce')
```

```
In [22]: df.dtypes
```

```

Out[22]: Name          object
Year            int64
GroupNum        int64
Status          object
Birth_Date      datetime64[ns]
Birth_Place     object
Gender          object
Alma_Mater      object
Undergraduate_Major object

```

Graduate_Major	object
Military_Rank	object
Military_Branch	object
Space_Flights	int64
Space_Flight_hr	int64
Space_Walks	int64
Space_Walks_hr	float64
Missions	object
Death_Date	datetime64[ns]
Death_Mission	object
dtype:	object

```
In [23]: # nasa data cleaned-now it has to be prepared for sql analysis
#-before that the data has to be transferred to the relational database mysql
df.to_csv('astronauts.csv',index=False) #If you want the auto-index to be saved in the
```

Preparing data for SQL analysis

1. Load sql extension

```
In [24]: !pip3 install pymysql
!pip3 install ipython-sql
!pip3 install mysqlclient
```

```
Requirement already satisfied: pymysql in c:\users\bthan\anaconda3\lib\site-packages (1.1.0)
Requirement already satisfied: ipython-sql in c:\users\bthan\anaconda3\lib\site-packages (0.5.0)
Requirement already satisfied: prettytable in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (3.9.0)
Requirement already satisfied: ipython in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (8.12.0)
Requirement already satisfied: sqlalchemy>=2.0 in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (2.0.23)
Requirement already satisfied: sqlparse in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (0.4.4)
Requirement already satisfied: six in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in c:\users\bthan\anaconda3\lib\site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.2.0 in c:\users\bthan\anaconda3\lib\site-packages (from sqlalchemy>=2.0->ipython-sql) (4.7.1)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\bthan\anaconda3\lib\site-packages (from sqlalchemy>=2.0->ipython-sql) (2.0.1)
Requirement already satisfied: backcall in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: decorator in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (0.18.1)
Requirement already satisfied: matplotlib-inline in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (0.1.6)
Requirement already satisfied: pickleshare in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30 in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (3.0.36)
Requirement already satisfied: pygments>=2.4.0 in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (2.15.1)
Requirement already satisfied: stack-data in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: traitlets>=5 in c:\users\bthan\anaconda3\lib\site-packages (from ipython->ipython-sql) (5.7.1)
Requirement already satisfied: colorama in c:\users\bthan\anaconda3\lib\site-packages (f
```

rom ipython->ipython-sql) (0.4.6)
Requirement already satisfied: wcwidth in c:\users\bthan\anaconda3\lib\site-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\bthan\anaconda3\lib\site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: executing in c:\users\bthan\anaconda3\lib\site-packages (from stack-data->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: asttokens in c:\users\bthan\anaconda3\lib\site-packages (from stack-data->ipython->ipython-sql) (2.0.5)
Requirement already satisfied: pure-eval in c:\users\bthan\anaconda3\lib\site-packages (from stack-data->ipython->ipython-sql) (0.2.2)
Requirement already satisfied: mysqlclient in c:\users\bthan\anaconda3\lib\site-packages (2.2.0)

2. Connect to MySQL

In [25]: `%load_ext sql`

In [26]: `%sql mysql+mysqldb://root:Thanabanu%401@localhost/astro`

3. Explore data from astronauts table

In [27]: `%sql select * from astronauts;`

* mysql+mysqldb://root:***@localhost/astro
350 rows affected.

Out[27]:

	Name	Year	GroupNum	Status	Birth_Date	Birth_Place	Gender	Alma_Mater	Undergraduate_I
	Alan B. Shepard Jr.	1959	1	Deceased	1923-11-18	East Derry, NH	Male	US Naval Academy	Naval Sci
	Alan G. Poindexter	1998	17	Deceased	1961-05-11	Pasadena, CA	Male	Georgia Institute of Technology; US Naval Postgraduate School	Aerospace Engine
	Alan L. Bean	1963	3	Deceased	1932-03-15	Wheeler, TX	Male	University of Texas	Aerona Engine
	Albert Sacco Jr.	1963	3	Retired	1949-03-05	Boston, MA	Male	Northeastern University; MIT	Chemical Engine
	Alfred M. Worden	1966	5	Retired	1932-07-02	Jackson, MI	Male	US Military Academy; University of Michigan	Military Sc
	Alvin B. Drew Jr.	2000	18	Active	1962-05-11	Washington, DC	Male	US Air Force Academy; Embry-Riddle Aeronautical University	Physics & Astrona Engine
	Andrew J. Feustel	2000	18	Active	1965-08-25	Lancaster, PA	Male	Purdue University; Queenâ€™s University-Canada	Solid Earth Sci
	Andrew M. Allen	1987	12	Retired	1955-04-08	Philadelphia, PA	Male	Villanova University;	Mech Engine

Gregory				14	Academy; Columbia University; Troy State University			
William M. Shepherd	1984	10	Retired	1949-07-26	Oak Ridge, TN	Male	US Naval Academy; MIT	Aerospace Engineer
William R. Pogue	1966	5	Retired	1930-01-23	Okemah, OK	Male	Oklahoma Baptist University; Oklahoma State University	Education
William S. McArthur Jr.	1990	3	Management	1951-07-26	Laurinburg, NC	Male	US Military Academy; Georgia Institute of Technology	Applied Sciences Engineer
Winston E. Scott	1992	14	Retired	1950-06-08	Miami, FL	Male	Florida State University; US Naval Postgraduate School	
Yvonne D. Cagle	1996	16	Management	1959-04-24	West Point, NY	Female	San Francisco State University	Biochemistry

Astronauts Status: Counting the cosmic pioneers

```
In [28]: %%sql
SELECT Status, count(*) as Number
FROM astronauts
GROUP by status;

* mysql+mysqldb://root:***@localhost/astro
4 rows affected.
```

```
Out[28]:
```

Status	Number
Deceased	48
Retired	216
Active	50
Management	36

```
In [56]: !pip install matplotlib
!pip install mysql-connector-python

Requirement already satisfied: matplotlib in c:\users\bthan\anaconda3\lib\site-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\bthan\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cyclers>=0.10 in c:\users\bthan\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\bthan\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
```



```
'host': 'localhost',  
'user': 'root',  
'password': 'Thanabanu@1',  
'database': 'astro'  
}
```

```
In [59]: conn = mysql.connector.connect(**db_config)  
         cursor = conn.cursor()  
         query = "SELECT Status, count(*) as Number FROM astronauts GROUP by status"  
         cursor.execute(query)  
         results = cursor.fetchall()
```

```
In [60]: results
```

```
Out[60]: [('Deceased', 48), ('Retired', 216), ('Active', 50), ('Management', 36)]
```

```
In [61]: statuses=[row[0] for row in results]  
         numbers=[row[1] for row in results]
```

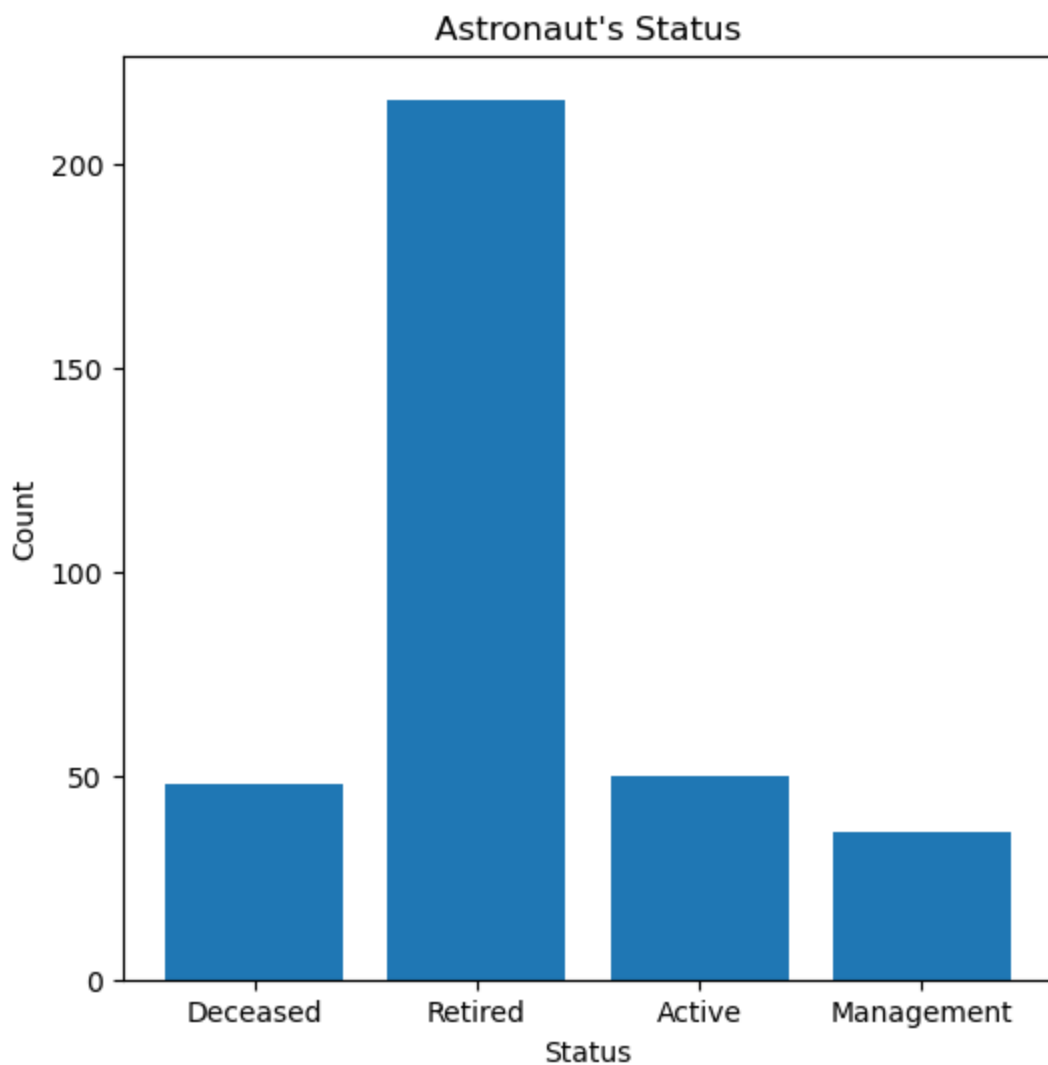
```
In [62]: statuses
```

```
Out[62]: ['Deceased', 'Retired', 'Active', 'Management']
```

```
In [63]: numbers
```

```
Out[63]: [48, 216, 50, 36]
```

```
In [84]: plt.figure(figsize=(6, 6))  
         plt.bar(statuses, numbers)  
         plt.xlabel('Status')  
         plt.ylabel('Count')  
         plt.title('Astronaut\'s Status')  
         #plt.xticks(rotation=45) # Rotate x-axis labels if needed  
         plt.show()
```



Exploring Astronauts Military branch diversity

```
In [70]: query1 = "SELECT Military_Branch, count(*) as Number FROM astronauts GROUP by Military_B  
cursor.execute(query1)  
results1 = cursor.fetchall()
```

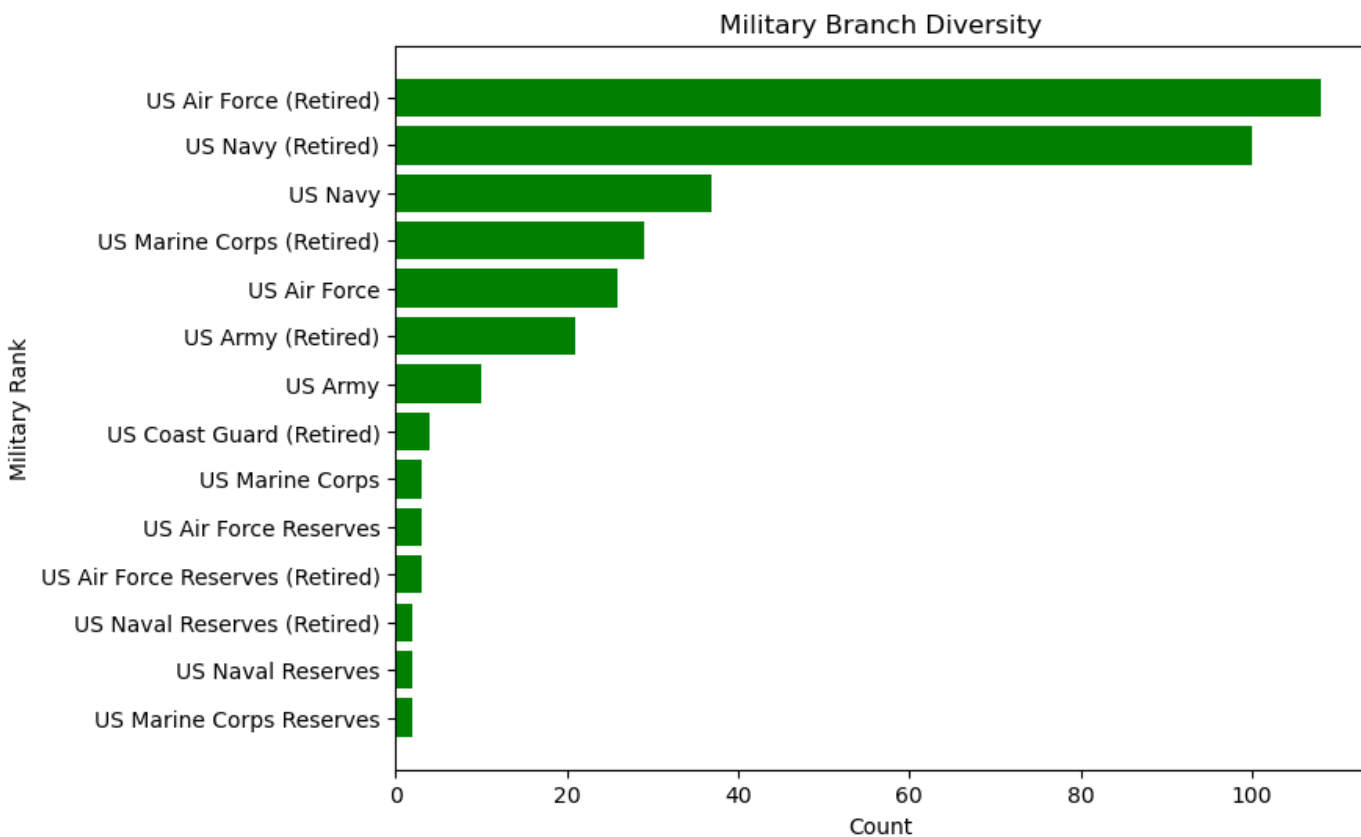
```
In [71]: results1
```

```
Out[71]: [('US Air Force (Retired)', 108),  
( 'US Navy (Retired)', 100),  
( 'US Navy', 37),  
( 'US Marine Corps (Retired)', 29),  
( 'US Air Force', 26),  
( 'US Army (Retired)', 21),  
( 'US Army', 10),  
( 'US Coast Guard (Retired)', 4),  
( 'US Marine Corps', 3),  
( 'US Air Force Reserves', 3),  
( 'US Air Force Reserves (Retired)', 3),  
( 'US Naval Reserves (Retired)', 2),  
( 'US Naval Reserves', 2),  
( 'US Marine Corps Reserves', 2)]
```

```
In [75]: Military_Branch=[row[0] for row in results1]  
Number=[row[1] for row in results1]
```

```
In [85]: plt.figure(figsize=(8, 6))  
plt.barh(Military_Branch, Number, color='green')
```

```
plt.xlabel('Count')
plt.ylabel('Military Rank')
plt.title('Military Branch Diversity')
plt.gca().invert_yaxis() # To display the highest rank at the top
plt.show()
```



In [29]:

```
%%sql
SELECT Military_Branch, count(*) as Number
FROM astronauts
GROUP by Military_Branch
ORDER BY Number DESC;

* mysql+mysqldb://root:***@localhost/astro
14 rows affected.
```

Out[29]:

Military_Branch	Number
US Air Force (Retired)	108
US Navy (Retired)	100
US Navy	37
US Marine Corps (Retired)	29
US Air Force	26
US Army (Retired)	21
US Army	10
US Coast Guard (Retired)	4
US Marine Corps	3
US Air Force Reserves	3
US Air Force Reserves (Retired)	3
US Naval Reserves (Retired)	2
US Naval Reserves	2

Top Military Ranks Among Astronauts

```
In [66]: %%sql
SELECT Military_Rank, count(*) as Number
FROM astronauts
GROUP by Military_Rank
ORDER BY Number DESC
limit 5;

* mysql+mysqldb://root:***@localhost/astro
5 rows affected.
```

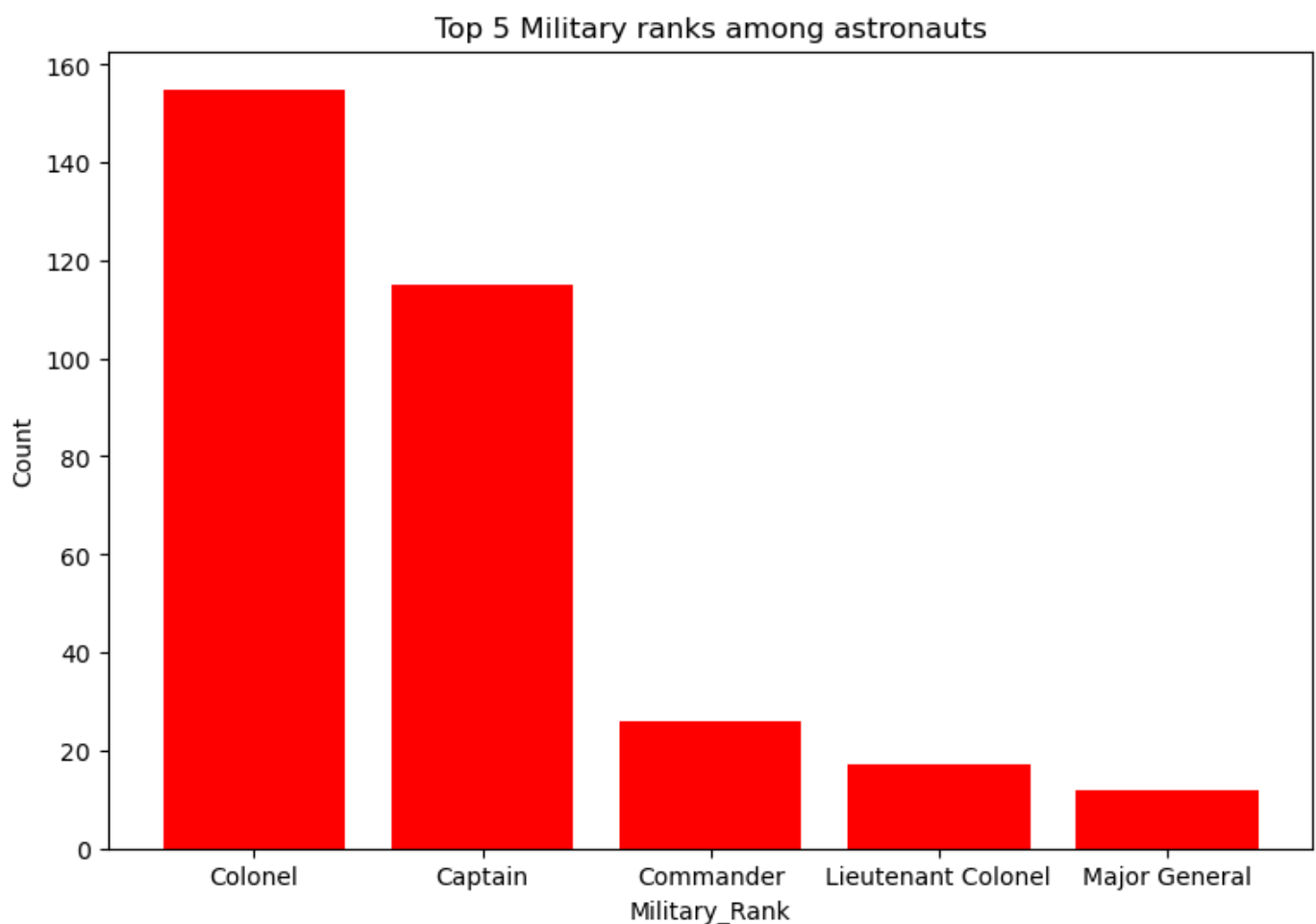
```
Out[66]:
```

Military_Rank	Number
Colonel	155
Captain	115
Commander	26
Lieutenant Colonel	17
Major General	12

```
In [83]: query2 = "SELECT Military_Rank, count(*) as Number FROM astronauts GROUP by Military_Ran
cursor.execute(query2)
results2 = cursor.fetchall()
```

```
In [87]: Military_Rank=[row[0] for row in results2]
Number=[row[1] for row in results2]
```

```
In [92]: plt.figure(figsize=(9, 6))
plt.bar(Military_Rank, Number, color='red')
plt.xlabel('Military_Rank')
plt.ylabel('Count')
plt.title('Top 5 Military ranks among astronauts')
#plt.xticks(rotation=45) # Rotate x-axis labels if needed
plt.show()
```



Exploring Astronaut Demographics: Gender Insights

```
In [31]: %%sql
SELECT Gender, count(*) as Number
FROM astronauts
GROUP by Gender;

* mysql+mysqldb://root:***@localhost/astro
2 rows affected.
```

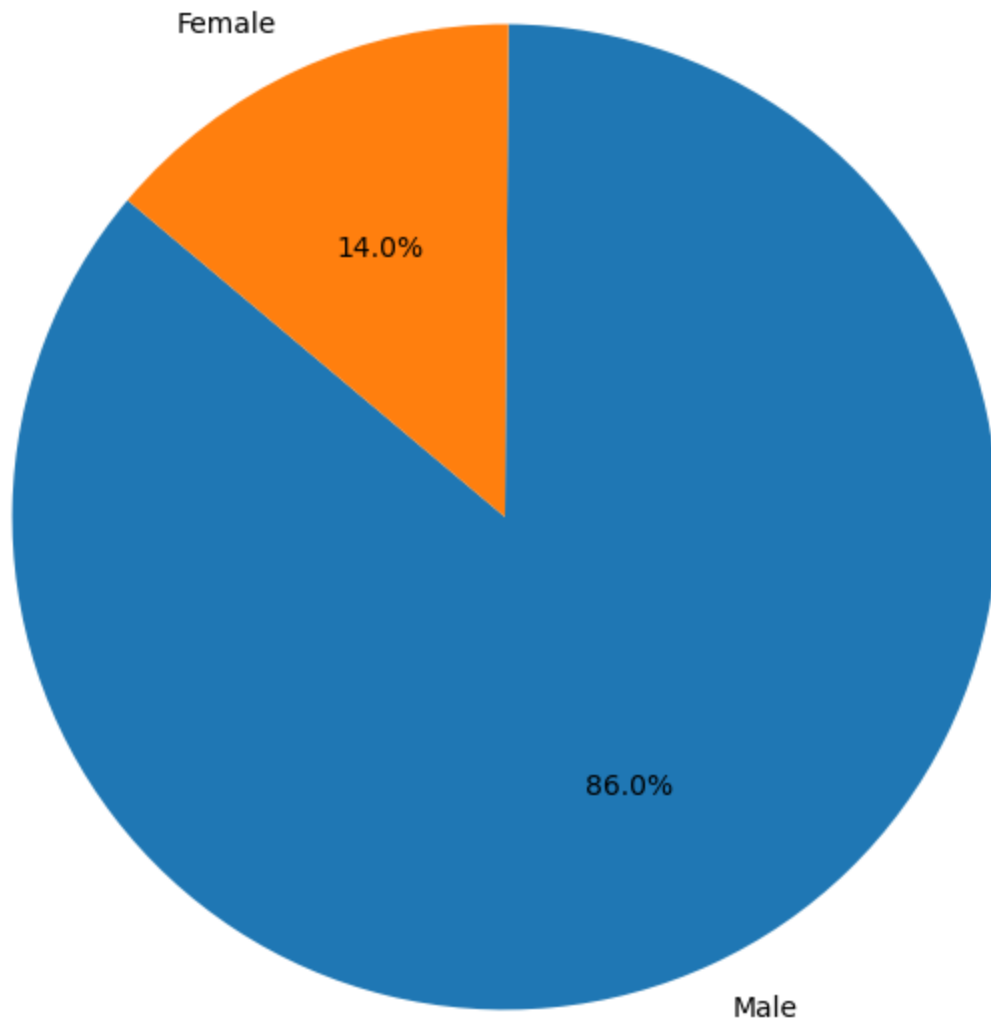
```
Out[31]: Gender  Number
-----
Male          301
Female         49
```

```
In [100... query3 = "SELECT Gender, count(*) as Number FROM astronauts GROUP by Gender"
cursor.execute(query3)
results3 = cursor.fetchall()
```

```
In [101... Gender=[row[0] for row in results3]
Counts=[row[1] for row in results3]
```

```
In [102... # Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(Counts, labels=Gender, autopct='%1.1f%%', startangle=140)
plt.title('Gender Demographics')
plt.show()
```

Gender Demographics



Astronaut's Life Expectancy : A Cosmic Perspective

```
In [32]: %%sql
SELECT * from astronauts;

* mysql:mysqldb://root:***@localhost/astro
350 rows affected.
```

	Name	Year	GroupNum	Status	Birth_Date	Birth_Place	Gender	Alma_Mater	Undergraduate_I
	Alan B. Shepard Jr.	1959	1	Deceased	1923-11-18	East Derry, NH	Male	US Naval Academy	Naval Sci
	Alan G. Poindexter	1998	17	Deceased	1961-05-11	Pasadena, CA	Male	Georgia Institute of Technology; US Naval Postgraduate School	Aerospace Engine
	Alan L. Bean	1963	3	Deceased	1932-03-15	Wheeler, TX	Male	University of Texas	Aerona Engine
	Albert Sacco	1963	3	Retired	1949-03-	Boston, MA	Male	Northeastern	Chemical Engine

McCool					23			Academy; University of Maryland; US Naval Postgraduate School	
William E. Thornton	1967	6	Retired	1929-04-14	Faison, NC	Male	University of North Carolina		Pl
William F. Fisher	1980	9	Retired	1946-01-04	Dallas, TX	Male	Stanford University; University of Houston; University of Florida		Pl
William F. Readdy	1987	12	Retired	1952-01-24	Quonset Point, RI	Male	US Naval Academy	Aerospace Engine	
William G. Gregory	1990	13	Retired	1957-05-14	Lockport, NY	Male	US Air Force Academy; Columbia University; Troy State University	Engineering Sc	
William M. Shepherd	1984	10	Retired	1949-07-26	Oak Ridge, TN	Male	US Naval Academy; MIT	Aerospace Engine	
William R. Pogue	1966	5	Retired	1930-01-23	Okemah, OK	Male	Oklahoma Baptist University; Oklahoma State University		Educ
William S. McArthur Jr.	1990	3	Management	1951-07-26	Laurinburg, NC	Male	US Military Academy; Georgia Institute of Technology	Applied Scien	Engine
Winston E. Scott	1992	14	Retired	1950-06-08	Miami, FL	Male	Florida State University; US Naval Postgraduate School		
Yvonne D. Cagle	1996	16	Management	1959-04-24	West Point, NY	Female	San Francisco State University		Biocher

In []:

To find the average life expectancy
First life expectancy has to be computed

In [45]:

%%sql
SELECT ROUND(**AVG**(s.life_expectancy)) **as** avg_life_expectancy
FROM (
 SELECT
 CASE
 WHEN Status = 'Deceased' **THEN** (**YEAR**(Death_Date)-**YEAR**(Birth_Date))

```

        ELSE (2023 - YEAR(Birth_Date))
    END AS life_expectancy
FROM astronauts
) as s;

```

```

* mysql+mysqldb://root:***@localhost/astro
1 rows affected.

```

Out[45]: **avg_life_expectancy**

69

```

In [53]: %%sql
SELECT ROUND(AVG(sm.male_life_expectancy)) as avg_male_life_expectancy
FROM (
    SELECT
        CASE
            WHEN Status = 'Deceased' AND Gender='Male' THEN (YEAR(Death_Date)-YEAR(Birth
            WHEN Gender = 'Male' THEN 2023 - YEAR(Birth_Date)
            ELSE NULL
        END AS male_life_expectancy
    FROM astronauts
) as sm;

```

```

* mysql+mysqldb://root:***@localhost/astro
1 rows affected.

```

Out[53]: **avg_male_life_expectancy**

70

```

In [54]: %%sql
SELECT ROUND(AVG(sf.female_life_expectancy)) as avg_female_life_expectancy
FROM (
    SELECT
        CASE
            WHEN Status = 'Deceased' AND Gender='Male' THEN (YEAR(Death_Date)-YEAR(Birth
            WHEN Gender = 'Female' THEN 2023 - YEAR(Birth_Date)
            ELSE NULL
        END AS female_life_expectancy
    FROM astronauts
) as sf;

```

```

* mysql+mysqldb://root:***@localhost/astro
1 rows affected.

```

Out[54]: **avg_female_life_expectancy**

60

Cosmic scholars : Top Graduate Majors of Astronauts

```

In [34]: %%sql
SELECT Graduate_Major, count(*) as Number
FROM astronauts
GROUP by Graduate_Major
ORDER BY Number DESC
limit 10;

```

```

* mysql+mysqldb://root:***@localhost/astro
10 rows affected.

```

Out[34]: **Graduate_Major Number**

Aeronautical Engineering	34
--------------------------	----

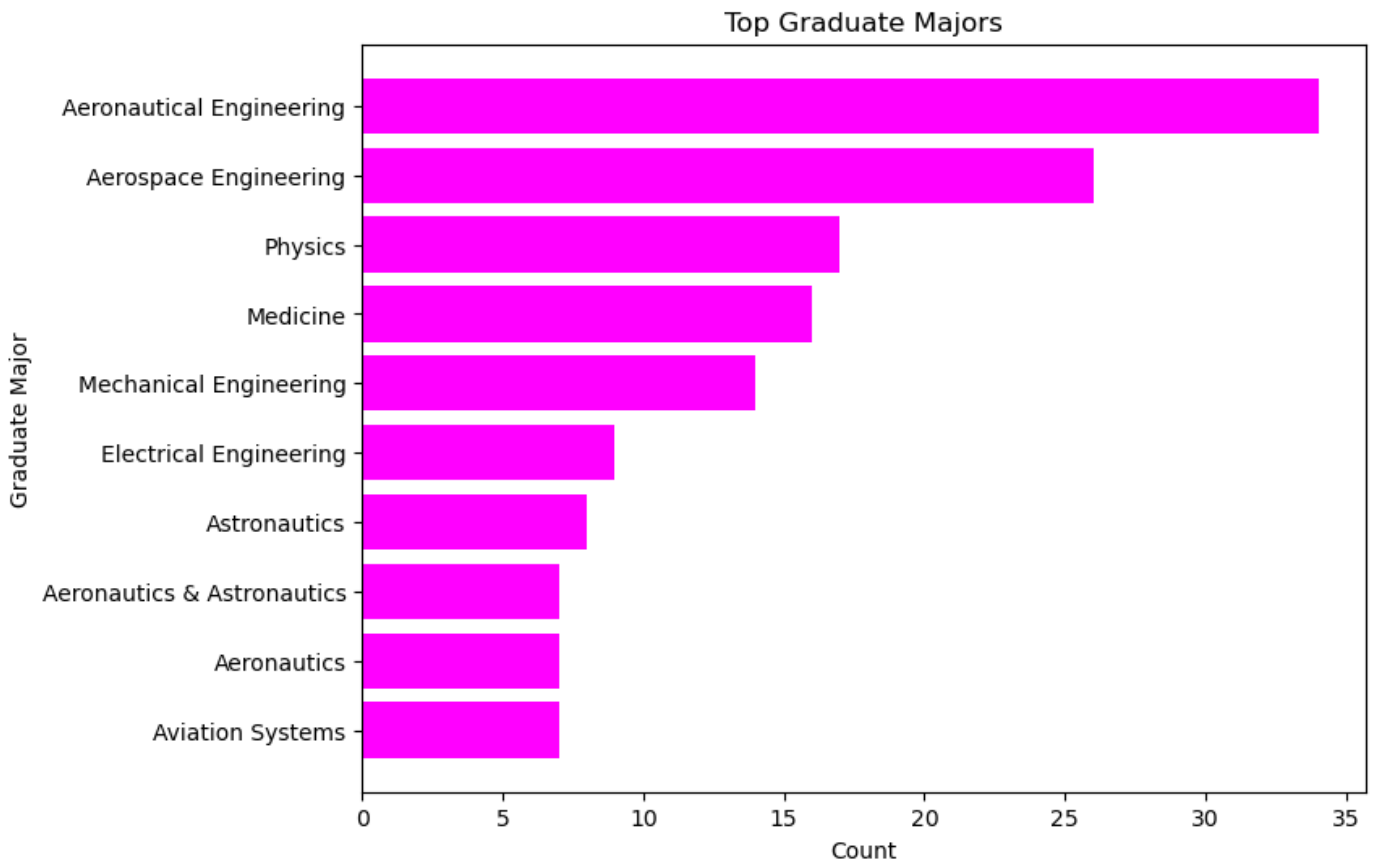
Aerospace Engineering	26
-----------------------	----

Physics	17
Medicine	16
Mechanical Engineering	14
Electrical Engineering	9
Astronautics	8
Aeronautics & Astronautics	7
Aeronautics	7
Aviation Systems	7

```
In [103... query4 = "SELECT Graduate_Major, count(*) as Number FROM astronauts GROUP by Graduate_Ma
cursor.execute(query4)
results4 = cursor.fetchall()
```

```
In [104... Graduate_Major=[row[0] for row in results4]
Counts=[row[1] for row in results4]
```

```
In [106... plt.figure(figsize=(8, 6))
plt.barh(Graduate_Major, Counts, color='magenta')
plt.xlabel('Count')
plt.ylabel('Graduate Major')
plt.title('Top Graduate Majors')
plt.gca().invert_yaxis() # To display the highest rank at the top
plt.show()
```



Educational Odyssey : Astronauts and their degrees

```
In [35]: %%sql
SELECT
    COUNT(*) as No_of_astronauts,
```

```
SUM(CASE WHEN Undergraduate_Major IS NOT NULL THEN 1 ELSE 0 END) as No_of_Undergrad_
SUM(CASE WHEN Graduate_Major IS NOT NULL THEN 1 ELSE 0 END) as No_of_Grad_Major
FROM astronauts;
```

```
* mysql+mysqldb://root:***@localhost/astro
1 rows affected.
```

Out[35]:

No_of_astronauts	No_of_Undergrad_Major	No_of_Grad_Major
350	350	350

Astronauts Birthplaces: Launching from Earth's Diverse Cities

In []: *# to extract the last two letters from the State and count the occurrence of each state*

In [94]:

```
%%sql
SELECT
    SUBSTRING_INDEX(Birth_Place, ',', -1) AS STATE,
    COUNT(*) AS Number
FROM astronauts
GROUP BY STATE
ORDER BY Number desc
LIMIT 5;
```

```
* mysql+mysqldb://root:***@localhost/astro
5 rows affected.
```

Out[94]:

STATE	Number
NY	30
CA	25
TX	23
OH	21
PA	19

Voyages Beyond: Astronaut's Space Flights and Spacewalks

In [43]:

```
%%sql
SELECT
    ROUND(AVG(Space_Flights),2) as avg_Space_flights,
    ROUND(AVG(Space_Walks),2) as avg_Space_Walks
FROM ASTRONAUTS;
```

```
* mysql+mysqldb://root:***@localhost/astro
1 rows affected.
```

Out[43]:

avg_Space_flights	avg_Space_Walks
2.40	1.27

In [108... !pip install -U notebook-as-pdf

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Requirement already satisfied: nbconvert in c:\users\bthan\anaconda3\lib\site-packages
(from notebook-as-pdf) (6.5.4)
Collecting pypeteer (from notebook-as-pdf)
  Downloading pypeteer-1.0.2-py3-none-any.whl (83 kB)
----- 0.0/83.4 kB ? eta -:--:--
----- 83.4/83.4 kB 4.9 MB/s eta 0:00:00
Collecting PyPDF2 (from notebook-as-pdf)
```

