Project management Analysis with Python

Problem Statement The project management dataset encompasses a wide range of project-related details, from project names and descriptions to costs, benefits, and statuses. Cleaning and preprocessing of data were initially undertaken to ensure data accuracy, followed by exploratory data analysis to derive insights.

Key analyses included the performance of project managers, project trends over time, project complexity, and the distribution of projects across different phases and departments. These analyses provide a data-driven foundation for enhanced project tracking and management, aiding decision-making and improving overall project outcomes.

Import Library

In [1]:
```
import pandas as pd
```

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWar
ning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Sc
iPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Uploading Csv fle

In [3]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Desktop\Project Management Dataset.csv")
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

In [4]:
```
df.head()
```

Out[4]:

Project Description	Project Type	Project Manager	Region	Department	Project Cost	Project Benefit	Complexity	Sta
Associations Now Is A Casual Game To Teach You...	INCOME GENERATION	Yael Wilcox	North	Admin & BI	3648615	8443980	High	Progr
Is A Fully Managed Content Marketing Software ...	INCOME GENERATION	Brenda Chandler	West	eCommerce	4018835	9012225	High	Cance
Most Content Marketers Know The Golden Rule: Y...	INCOME GENERATION	Nyasia Hunter	North	Warehouse	4285483	9078339	High	Comple
Utilize And Utilizes (Verb Form) The Open, Inc...	PROCESS IMPROVEMENT	Brenda Chandler	East	Sales and Marketing	5285864	8719006	High	Cance
Is A Solution For Founders Who Want To Win At ...	WORKING CAPITAL IMPROVEMENT	Jaylyn Mckenzie	East	eCommerce	5785601	8630148	High	Comple

.tail()

tail is used to show rows by Ascending order

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

Out[5]:

Project escription	Project Type	Project Manager	Region	Department	Project Cost	Project Benefit	Complexity	Statu
as Built To Founders Create nized Co...	WORKING CAPITAL IMPROVEMENT	Nyasia Hunter	South	Supply Chain	5259436	8817917	Medium	On - Hol
In This cosystem, ssociation Content Is Simp...	INCOME GENERATION	Kamari Norris	North	Warehouse	4790417	8872443	Medium	In Progres
th 15 Five, e Take The swork Out Of Con...	PROCESS IMPROVEMENT	Yael Wilcox	West	Supply Chain	4283481	8895152	Low	Complete
s Founded To Help nders And preneurs...	COST REDUCTION	Jaylyn Mckenzie	East	eCommerce	4606575	8658343	High	In Progres
elcome To Future Of Content tion. The...	WORKING CAPITAL IMPROVEMENT	Nyasia Hunter	West	Sales and Marketing	5054482	8422578	High	In Progres

.shape

It show the total no of rows & Column in the dataset

In [6]:
```
df.shape
```

Out[6]:
```
(99, 16)
```

.Columns

It show the no of each Column

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Project Name', 'Project Description', 'Project Type',
       'Project Manager', 'Region', 'Department', ' Project Cost ',
       ' Project Benefit ', 'Complexity', 'Status', 'Completion%', 'Phase',
       'Year', 'Month', 'Start Date', 'End Date'],
      dtype='object')
```

.dtypes

This Attribute show the data type of each column

In [8]:
```
df.dtypes
```

Out[8]:
```
Project Name           object
Project Description    object
Project Type           object
Project Manager        object
Region                 object
Department             object
 Project Cost           int64
 Project Benefit        int64
Complexity             object
Status                 object
Completion%            object
Phase                  object
Year                    int64
Month                   int64
Start Date             object
End Date               object
dtype: object
```

.unique()

In a column, It show the unique value of specific column.

In [9]:
```
df["Department"].unique()
```

Out[9]:
```
array(['Admin & BI', 'eCommerce', 'Warehouse', 'Sales and Marketing',
       'Supply Chain'], dtype=object)
```

.nuique()

It will show the total no of unique value from whole data frame

In [10]:
```
df.nunique()
```

Out[10]:
```
Project Name           99
Project Description    95
Project Type            4
Project Manager         7
Region                  4
Department              5
 Project Cost          99
 Project Benefit       99
Complexity              3
Status                  4
Completion%            22
Phase                   5
Year                    5
Month                  12
Start Date             49
End Date               43
dtype: int64
```

.describe()

It show the Count, mean , median etc

In [11]:
```
df.describe()
```

Out[11]:

	Project Cost	Project Benefit	Year	Month
count	9.900000e+01	9.900000e+01	99.000000	99.000000
mean	4.156649e+06	8.828178e+06	2022.747475	7.151515
std	1.076544e+06	2.164019e+05	1.402210	3.211471
min	2.418301e+06	8.422578e+06	2021.000000	1.000000
25%	3.251948e+06	8.656248e+06	2022.000000	4.500000
50%	4.172827e+06	8.846243e+06	2022.000000	7.000000
75%	5.063288e+06	9.019234e+06	2024.000000	10.000000
max	5.974815e+06	9.165877e+06	2025.000000	12.000000

.value_counts

It Shows all the unique values with their count

In [12]:
```
df["Department"].value_counts()
```

Out[12]:
```
Supply Chain           24
Warehouse              23
eCommerce              20
Admin & BI             18
Sales and Marketing    14
Name: Department, dtype: int64
```

.isnull()

It shows the how many null values

In [13]:
```
df.isnull()
```

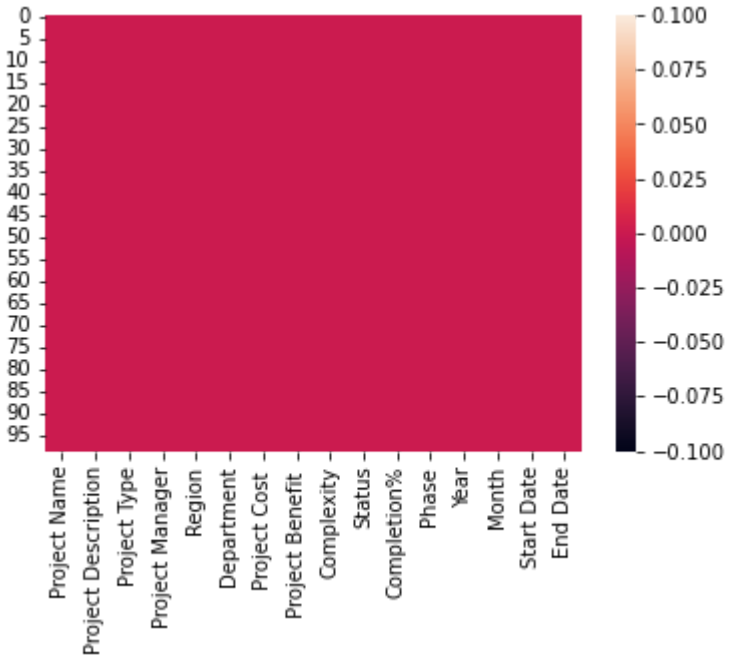
Out[13]:

	Project Name	Project Description	Project Type	Project Manager	Region	Department	Project Cost	Project Benefit	Complexity	St
0	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	
94	False	False	False	False	False	False	False	False	False	F
95	False	False	False	False	False	False	False	False	False	F
96	False	False	False	False	False	False	False	False	False	F
97	False	False	False	False	False	False	False	False	False	F
98	False	False	False	False	False	False	False	False	False	F

99 rows × 16 columns

In [14]:
```
sns.heatmap(df.isnull())
```

Out[14]: <AxesSubplot:>

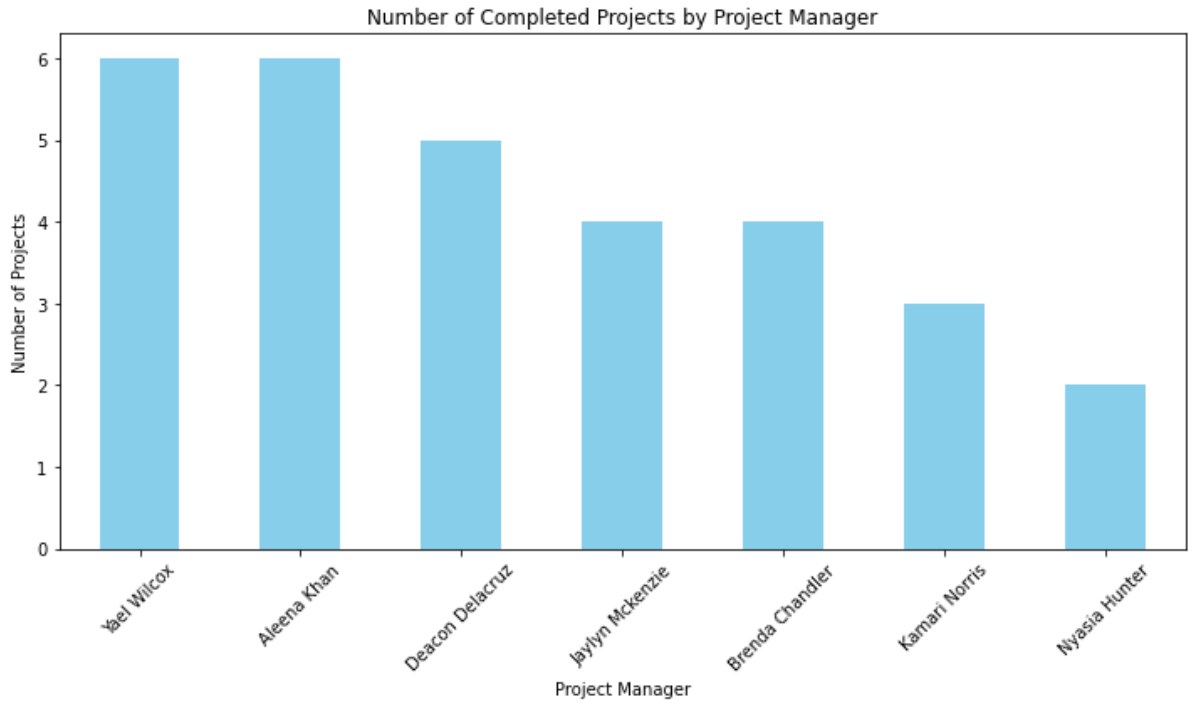


In [15]:
```
df.duplicated().sum()
```

Out[15]: 0

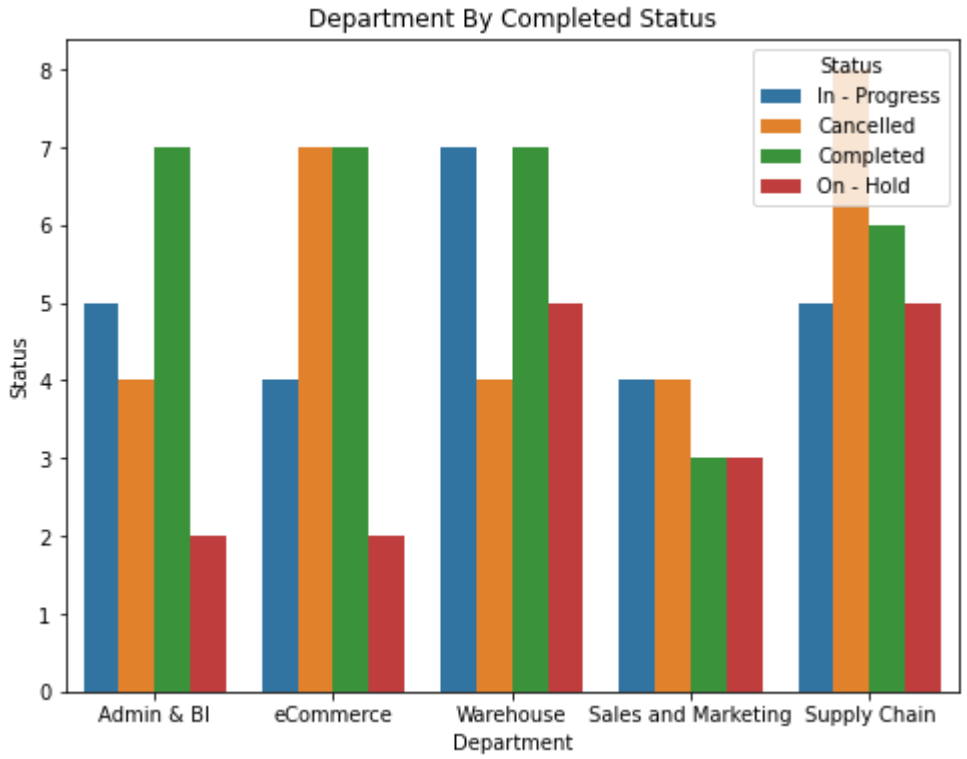
What is the total project cost for each project type?

In [16]:
```
completed_projects = df[df['Status'] == 'Completed']
project_managers = completed_projects['Project Manager'].value_counts()

# Create a bar chart
plt.figure(figsize=(10, 6))
project_managers.plot(kind='bar', color='skyblue')
plt.title('Number of Completed Projects by Project Manager')
plt.xlabel('Project Manager')
plt.ylabel('Number of Projects')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



How many projects are in progress or on hold in each department?

In [17]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Department', hue='Status')
plt.xlabel('Department')
plt.ylabel('Status')
plt.title('Department By Completed Status')
plt.show()
```



What is the average project benefit by region?

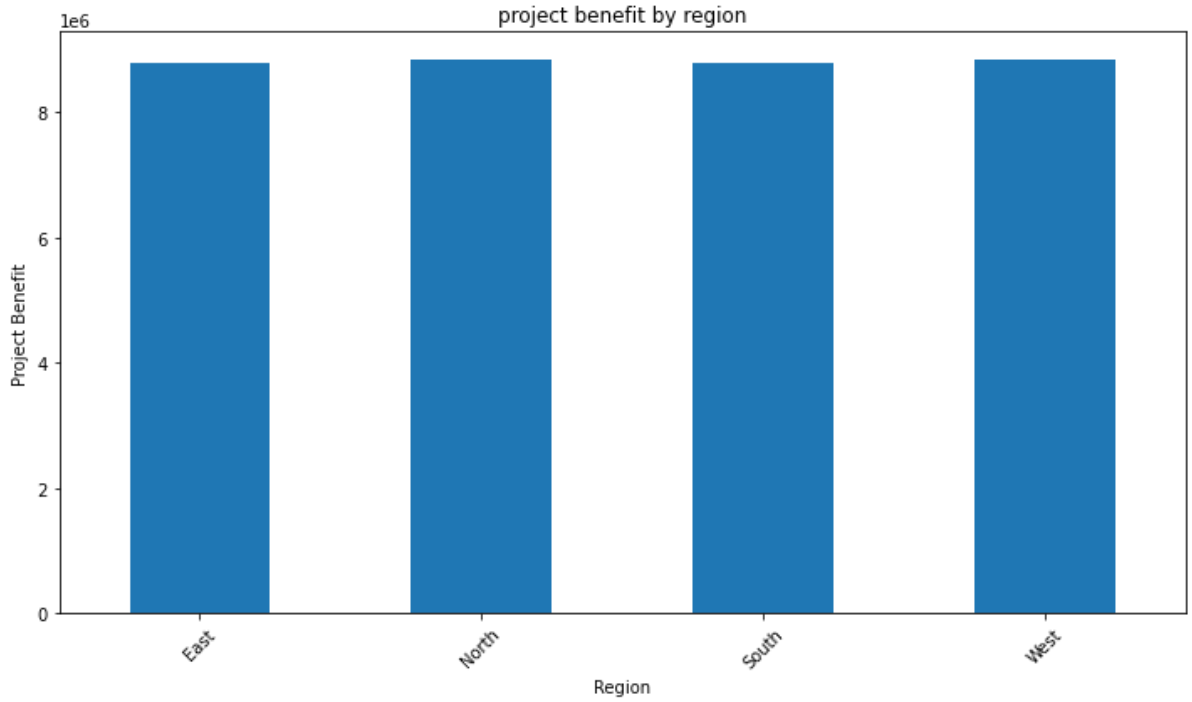
Why we Use Str.strip .....??

This ensures that any extra whitespaces in the column names are removed.

In [18]:
```
df.columns = df.columns.str.strip()
average_benefit_by_region = df.groupby('Region')['Project Benefit'].mean()
average_benefit_by_region
```

Out[18]:
```
Region
East     8.801634e+06
North    8.849363e+06
South    8.796756e+06
West     8.847781e+06
Name: Project Benefit, dtype: float64
```

In [19]:
```
# Create a bar chart
plt.figure(figsize=(10, 6))
average_benefit_by_region.plot(kind='bar')
plt.title('project benefit by region')
plt.xlabel('Region')
plt.ylabel('Project Benefit')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```

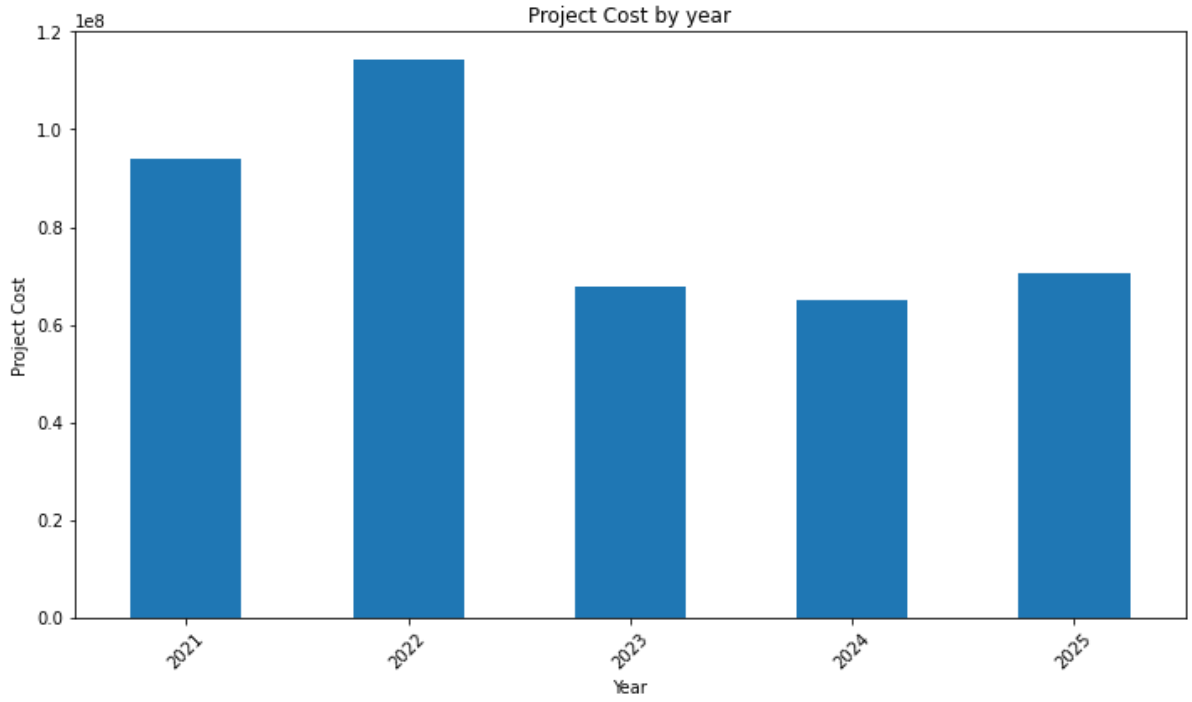


How has the project cost and benefit evolved over the years?

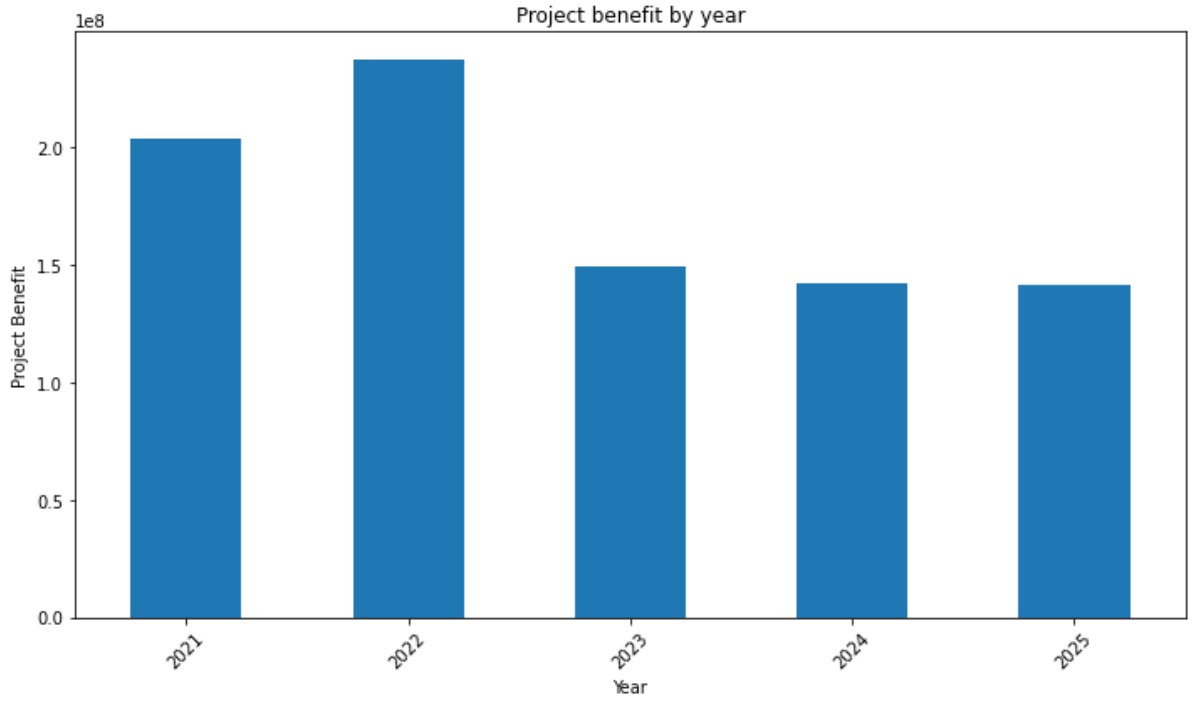
In [20]:
```
df.columns = df.columns.str.strip()
project_cost_by_year = df.groupby('Year')['Project Cost'].sum()
project_benefit_by_year = df.groupby('Year')['Project Benefit'].sum()
project_cost_by_year
```

Out[20]:
```
Year
2021     93941527
2022    114304574
2023     67860451
2024     64856107
2025     70545628
Name: Project Cost, dtype: int64
```

In [21]:
```
# Create a bar chart
plt.figure(figsize=(10, 6))
project_cost_by_year.plot(kind='bar')
plt.title('Project Cost by year')
plt.xlabel('Year')
plt.ylabel('Project Cost')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



In [22]:
```
# Create a bar chart
plt.figure(figsize=(10, 6))
project_benefit_by_year.plot(kind='bar')
plt.title('Project benefit by year')
plt.xlabel('Year')
plt.ylabel('Project Benefit')
plt.xticks(rotation=45)
plt.tight_layout()

# Show the plot
plt.show()
```



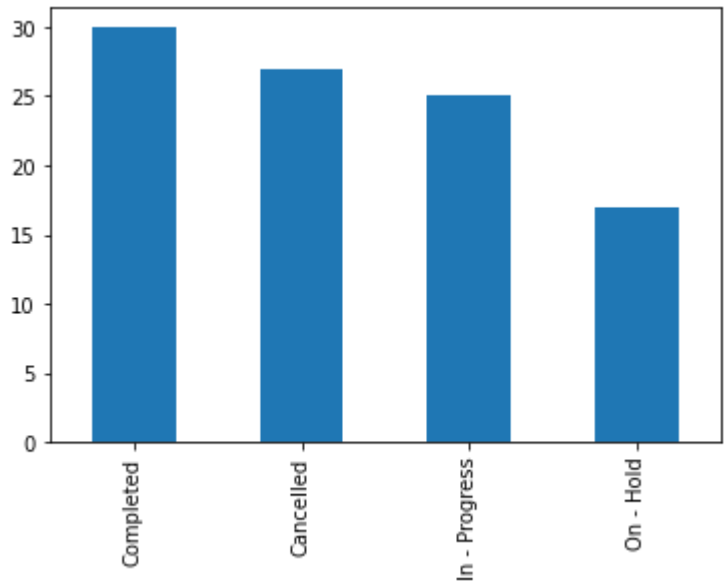
How many projects were completed each month?

In [23]:
```
completed_projects = df[df['Status'] == 'Completed']
projects_completed_by_month = completed_projects['Month'].value_counts().sort_
projects_completed_by_month
```

Out[23]:
```
1     1
2     1
3     2
4     5
5     2
6     2
7     3
8     2
9     1
10    2
11    6
12    3
Name: Month, dtype: int64
```

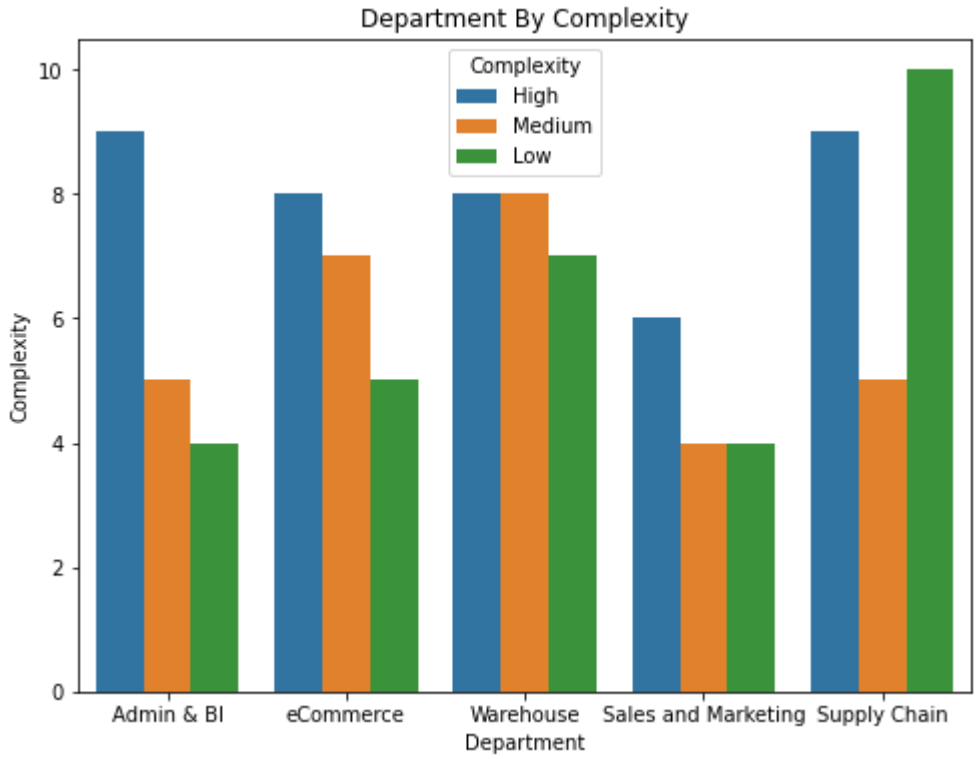
How many projects are in each status category?

In [24]:
```
project_status_counts = df['Status'].value_counts().plot( kind = "bar")
```



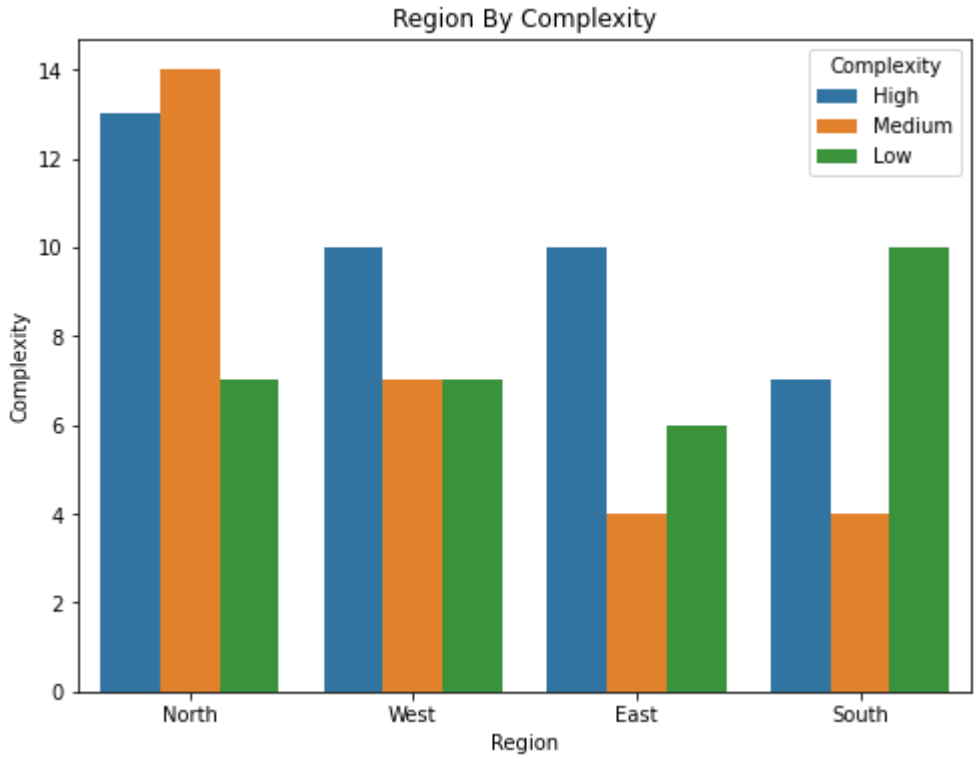
Department By Complexity

In [25]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Department', hue='Complexity')
plt.xlabel('Department')
plt.ylabel('Complexity')
plt.title('Department By Complexity')
plt.show()
```



Region By Complexity

In [26]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Region', hue='Complexity')
plt.xlabel('Region')
plt.ylabel('Complexity')
plt.title('Region By Complexity')
plt.show()
```



Project type wise Cost & Benefit..

In [29]:
```
Project_cost_by_type = df.groupby('Project Type')['Project Cost', 'Project Bene
Project_cost_by_type
```

```
C:\Users\Syed Arif\AppData\Local\Temp\ipykernel_11576\683233896.py:1: FutureW
arning: Indexing with multiple keys (implicitly converted to a tuple of keys)
will be deprecated, use a list instead.
  Project_cost_by_type = df.groupby('Project Type')['Project Cost', 'Project
Benefit'].sum().reset_index()
```

Out[29]:

	Project Type	Project Cost	Project Benefit
0	COST REDUCTION	93387098	194574043
1	INCOME GENERATION	105591918	237933332
2	PROCESS IMPROVEMENT	102249635	222234293
3	WORKING CAPITAL IMPROVEMENT	110279636	219247967

In [ ]: