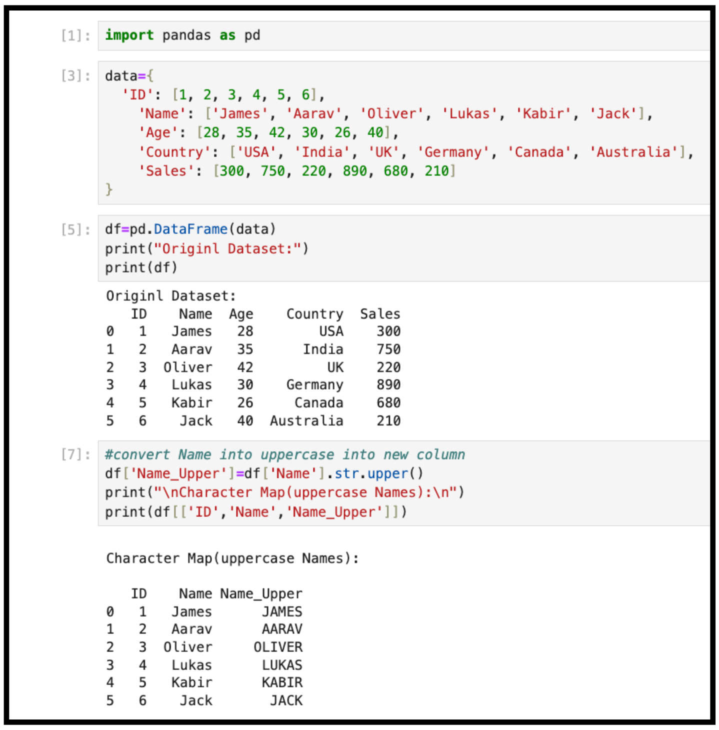


```
[1]: import pandas as pd
```

```
[3]: data={
       'ID': [1, 2, 3, 4, 5, 6],
         'Name': ['James', 'Aarav', 'Oliver', 'Lukas', 'Kabir', 'Jack'],
         'Age': [28, 35, 42, 30, 26, 40],
         'Country': ['USA', 'India', 'UK', 'Germany', 'Canada', 'Australia'],
         'Sales': [300, 750, 220, 890, 680, 210]
     }
```

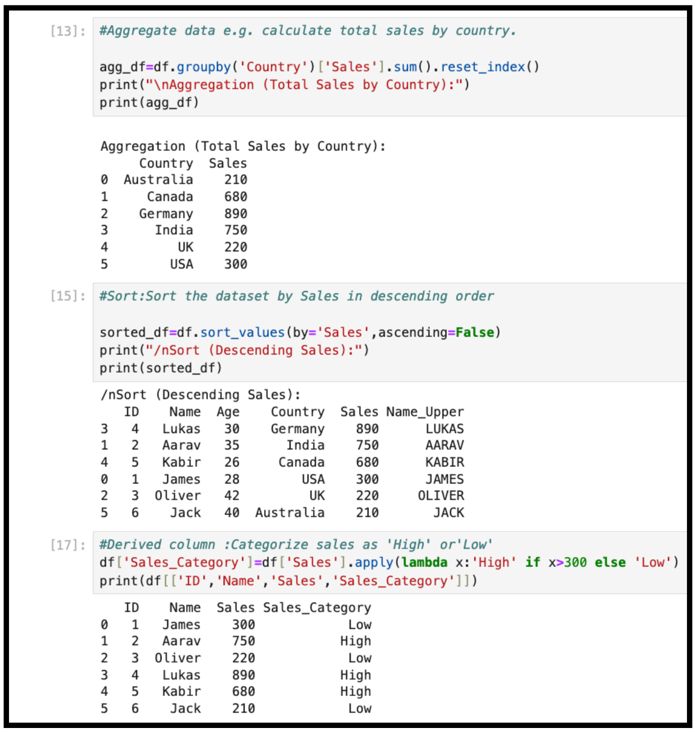
```
[5]: df=pd.DataFrame(data)
     print("Originl Dataset:")
     print(df)
```

```
Originl Dataset:
   ID    Name  Age    Country  Sales
0   1   James   28        USA    300
1   2   Aarav   35      India    750
2   3  Oliver   42         UK    220
3   4   Lukas   30    Germany    890
4   5   Kabir   26     Canada    680
5   6    Jack   40  Australia    210
```

```
[7]: #convert Name into uppercase into new column
     df['Name_Upper']=df['Name'].str.upper()
     print("\nCharacter Map(uppercase Names):\n")
     print(df[['ID','Name','Name_Upper']])
```

```
Character Map(uppercase Names):

   ID    Name Name_Upper
0   1   James      JAMES
1   2   Aarav      AARAV
2   3  Oliver     OLIVER
3   4   Lukas      LUKAS
4   5   Kabir      KABIR
5   6    Jack       JACK
```



```
[13]: #Aggregate data e.g. calculate total sales by country.

      agg_df=df.groupby('Country')['Sales'].sum().reset_index()
      print("\nAggregation (Total Sales by Country):")
      print(agg_df)
```

```
Aggregation (Total Sales by Country):
     Country  Sales
0  Australia    210
1     Canada    680
2    Germany    890
3      India    750
4         UK    220
5        USA    300
```

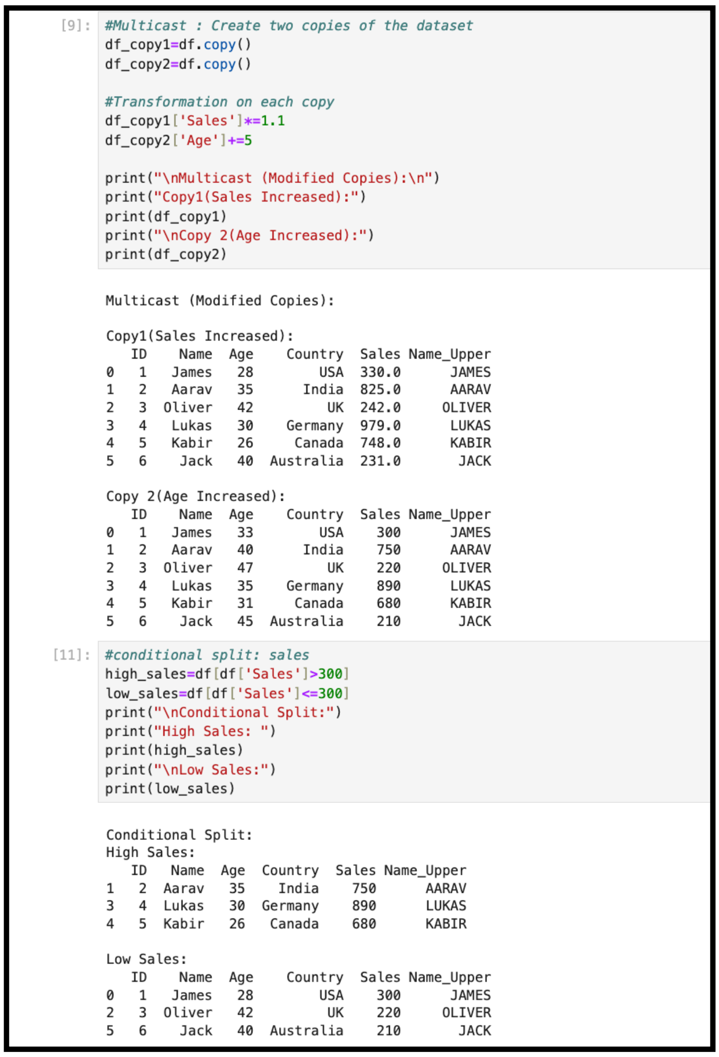
```
[15]: #Sort:Sort the dataset by Sales in descending order

      sorted_df=df.sort_values(by='Sales',ascending=False)
      print("/nSort (Descending Sales):")
      print(sorted_df)
```

```
/nSort (Descending Sales):
   ID    Name  Age    Country  Sales Name_Upper
3   4   Lukas   30    Germany    890      LUKAS
1   2   Aarav   35      India    750      AARAV
4   5   Kabir   26     Canada    680      KABIR
0   1   James   28        USA    300      JAMES
2   3  Oliver   42         UK    220     OLIVER
5   6    Jack   40  Australia    210       JACK
```

```
[17]: #Derived column :Categorize sales as 'High' or'Low'
      df['Sales_Category']=df['Sales'].apply(lambda x:'High' if x>300 else 'Low')
      print(df[['ID','Name','Sales','Sales_Category']])
```

```
   ID    Name  Sales Sales_Category
0   1   James    300            Low
1   2   Aarav    750           High
2   3  Oliver    220            Low
3   4   Lukas    890           High
4   5   Kabir    680           High
5   6    Jack    210            Low
```



```
[9]: #Multicast : Create two copies of the dataset
     df_copy1=df.copy()
     df_copy2=df.copy()

     #Transformation on each copy
     df_copy1['Sales']*=1.1
     df_copy2['Age']+=5

     print("\nMulticast (Modified Copies):\n")
     print("Copy1(Sales Increased):")
     print(df_copy1)
     print("\nCopy 2(Age Increased):")
     print(df_copy2)
```

```
Multicast (Modified Copies):

Copy1(Sales Increased):
   ID    Name  Age    Country  Sales Name_Upper
0   1   James   28        USA  330.0      JAMES
1   2   Aarav   35      India  825.0      AARAV
2   3  Oliver   42         UK  242.0     OLIVER
3   4   Lukas   30    Germany  979.0      LUKAS
4   5   Kabir   26     Canada  748.0      KABIR
5   6    Jack   40  Australia  231.0       JACK

Copy 2(Age Increased):
   ID    Name  Age    Country  Sales Name_Upper
0   1   James   33        USA    300      JAMES
1   2   Aarav   40      India    750      AARAV
2   3  Oliver   47         UK    220     OLIVER
3   4   Lukas   35    Germany    890      LUKAS
4   5   Kabir   31     Canada    680      KABIR
5   6    Jack   45  Australia    210       JACK
```

```
[11]: #conditional split: sales
      high_sales=df[df['Sales']>300]
      low_sales=df[df['Sales']<=300]
      print("\nConditional Split:")
      print("High Sales: ")
      print(high_sales)
      print("\nLow Sales:")
      print(low_sales)
```

```
Conditional Split:
High Sales:
   ID   Name  Age  Country  Sales Name_Upper
1   2  Aarav   35    India    750      AARAV
3   4  Lukas   30  Germany    890      LUKAS
4   5  Kabir   26   Canada    680      KABIR

Low Sales:
   ID    Name  Age    Country  Sales Name_Upper
0   1   James   28        USA    300      JAMES
2   3  Oliver   42         UK    220     OLIVER
5   6    Jack   40  Australia    210       JACK
```