

16th October Notes

Creating and Reshaping Arrays

Create Array from 1 to 12

```
import numpy as np
```

```
arr = np.arange(1, 13)
```

```
arr
```

Reshape Examples

```
a = arr.reshape(4, 3)
```

```
a
```

```
temp = arr.reshape(2, 6)
```

```
temp
```

Convert Back to 1D

```
temp.reshape(12)
```

Convert to Column Vector

```
a.reshape(12, 1)
```

Flatten Array (Column-wise)

```
temp.flatten('F')
```

'F' (Fortran order) means flatten column-wise instead of the default row-wise ('C').

Extracting Elements from Array

Example Matrix

```
temp
```

Extract Element “11”

```
temp[1, 4]
```

Extract Elements by Index

```
temp[1, [0, 1, 2]] # Elements from 2nd row, columns 0,1,2
```

Extract Entire Row

```
temp[1]
```

Extract Entire Column

```
temp[:, 0]
```

Matrix Operations

Define Matrices

```
arr1 = np.array([[1, 2],  
                 [8, 9]])
```

```
arr2 = np.array([[3, 4],  
                 [8, 9]])
```

Horizontal Stack (Side-by-Side)

```
np.hstack((arr1, arr2))
```

Vertical Stack (One Below Another)

```
np.vstack((arr1, arr2))
```

Transpose of a Matrix

```
temp.T
```

Matrix Multiplication

Dimensions:

- $(3 \times 2) \times (2 \times 4)$
- $(2 \times 3) \times (4 \times 2)$

Matrix multiplication is possible only when **columns of the first = rows of the second**

Matrix Multiplication

`np.matmul(arr1, arr2)`

or using the shorthand:

`arr1 @ arr2`

In [5]:
#16/10/25
import pandas as pd
df=pd.read_csv(r"C:\Users\Agnel Sharon Jerald\OneDrive\Desktop\Machine learning\sales_data_sample.csv")
df

Out[5]:

	TransactionID	Date	Region	SalesPerson	Product	Quantity	UnitPrice	Discount	Category	CustomerType
0	1	2025-01-01	North	Rahul	Laptop	2	60000	5	Electronics	Regular
1	2	2025-01-02	South	Meena	Mouse	5	800	0	Accessories	New
2	3	2025-01-03	East	Ajay	Keyboard	3	1200	10	Accessories	Regular
3	4	2025-01-04	West	Priya	Laptop	1	62000	0	Electronics	Regular
4	5	2025-01-05	North	Karan	Mobile	4	25000	5	Electronics	New
5	6	2025-01-06	East	Rahul	Tablet	2	18000	0	Electronics	Regular
6	7	2025-01-07	South	Meena	Charger	6	500	0	Accessories	New
7	8	2025-01-08	North	Priya	Laptop	3	58000	10	Electronics	Regular
8	9	2025-01-09	West	Ajay	Mobile	2	27000	5	Electronics	Regular
9	10	2025-01-10	East	Karan	Keyboard	5	1000	0	Accessories	New
10	11	2025-01-11	South	Rahul	Laptop	1	61000	0	Electronics	Regular
11	12	2025-01-12	North	Meena	Mouse	4	850	5	Accessories	New
12	13	2025-01-13	East	Priya	Tablet	3	19000	10	Electronics	Regular
13	14	2025-01-14	West	Ajay	Laptop	2	63000	0	Electronics	New
14	15	2025-01-15	North	Karan	Charger	5	550	0	Accessories	Regular
15	16	2025-01-16	South	Rahul	Keyboard	4	1100	0	Accessories	New
16	17	2025-01-17	East	Meena	Mobile	3	26000	5	Electronics	Regular
17	18	2025-01-18	West	Priya	Laptop	2	60000	5	Electronics	New
18	19	2025-01-19	North	Ajay	Tablet	4	18500	10	Electronics	Regular
19	20	2025-01-20	South	Karan	Mouse	6	750	0	Accessories	New

In [6]:
Calculate Total Sales Amount for each transaction
df["TotalSales"] = df["Quantity"] * df["UnitPrice"] * (1 - df["Discount"] / 100)
df["TotalSales"]

Out[6]:

0	114000.0
1	4000.0
2	3240.0
3	62000.0
4	95000.0
5	36000.0
6	3000.0
7	156600.0
8	51300.0
9	5000.0
10	61000.0
11	3230.0
12	51300.0
13	126000.0
14	2750.0
15	4400.0
16	74100.0
17	114000.0
18	66600.0
19	4500.0

Name: TotalSales, dtype: float64

In [7]:
Find total sales by each Region
total_sales_by_region = df.groupby("Region")["TotalSales"].sum().reset_index()
total_sales_by_region

Out[7]:

	Region	TotalSales
0	East	169640.0
1	North	438180.0
2	South	76900.0
3	West	353300.0

In [8]:
df.groupby("Region")["Quantity"].sum().reset_index()

Out[8]:

	Region	Quantity
0	East	16
1	North	22
2	South	22
3	West	7

In [10]:
Display all transactions where Discount is greater than 5%
discount_gt_5 = df[df["Discount"] > 5]
discount_gt_5

Out[10]:

	TransactionID	Date	Region	SalesPerson	Product	Quantity	UnitPrice	Discount	Category	CustomerType	TotalSales
2	3	2025-01-03	East	Ajay	Keyboard	3	1200	10	Accessories	Regular	3240.0
7	8	2025-01-08	North	Priya	Laptop	3	58000	10	Electronics	Regular	156600.0
12	13	2025-01-13	East	Priya	Tablet	3	19000	10	Electronics	Regular	51300.0
18	19	2025-01-19	North	Ajay	Tablet	4	18500	10	Electronics	Regular	66600.0

In [11]:
Group data by Category and find average UnitPrice for each
avg_unitprice_by_category = df.groupby("Category")["UnitPrice"].mean().reset_index()
avg_unitprice_by_category

Out[11]:

	Category	UnitPrice
0	Accessories	843.750000
1	Electronics	41458.333333

In [13]:
Find top 2 SalesPersons with highest total sales amount
top2_salespersons = (
df.groupby("SalesPerson")["TotalSales"]
 .sum()
 .nlargest(2)
 .reset_index()
)
top2_salespersons

Out[13]:

	SalesPerson	TotalSales
0	Priya	383900.0
1	Ajay	247140.0

In [15]:
Filter records for only Electronics category where sales are above ₹50,000
electronics_above_50k = df[(
df["Category"] == "Electronics") & (df["TotalSales"] > 50000)]
electronics_above_50k

Out[15]:

	TransactionID	Date	Region	SalesPerson	Product	Quantity	UnitPrice	Discount	Category	CustomerType	TotalSales
0	1	2025-01-01	North	Rahul	Laptop	2	60000	5	Electronics	Regular	114000.0
3	4	2025-01-04	West	Priya	Laptop	1	62000	0	Electronics	Regular	62000.0
4	5	2025-01-05	North	Karan	Mobile	4	25000	5	Electronics	New	95000.0
7	8	2025-01-08	North	Priya	Laptop	3	58000	10	Electronics	Regular	156600.0
8	9	2025-01-09	West	Ajay	Mobile	2	27000	5	Electronics	Regular	51300.0
10	11	2025-01-11	South	Rahul	Laptop	1	61000	0	Electronics	Regular	61000.0
12	13	2025-01-13	East	Priya	Tablet	3	19000	10	Electronics	Regular	51300.0
13	14	2025-01-14	West	Ajay	Laptop	2	63000	0	Electronics	New	126000.0
16	17	2025-01-17	East	Meena	Mobile	3	26000	5	Electronics	Regular	74100.0
17	18	2025-01-18	West	Priya	Laptop	2	60000	5	Electronics	New	114000.0
18	19	2025-01-19	North	Ajay	Tablet	4	18500	10	Electronics	Regular	66600.0

In [16]:
Sort the dataset by Date and SalesPerson in ascending order
df_sorted = df.sort_values(by=["Date", "SalesPerson"], ascending=[True, True])
df_sorted

Out[16]:

	TransactionID	Date	Region	SalesPerson	Product	Quantity	UnitPrice	Discount	Category	CustomerType	TotalSales
0	1	2025-01-01	North	Rahul	Laptop	2	60000	5	Electronics	Regular	114000.0
1	2	2025-01-02	South	Meena	Mouse	5	800	0	Accessories	New	4000.0
2	3	2025-01-03	East	Ajay	Keyboard	3	1200	10	Accessories	Regular	3240.0
3	4	2025-01-04	West	Priya	Laptop	1	62000	0	Electronics	Regular	62000.0
4	5	2025-01-05	North	Karan	Mobile	4	25000	5	Electronics	New	95000.0
5	6	2025-01-06	East	Rahul	Tablet	2	18000	0	Electronics	Regular	36000.0
6	7	2025-01-07	South	Meena	Charger	6	500	0	Accessories	New	3000.0
7	8	2025-01-08	North	Priya	Laptop	3	58000	10	Electronics	Regular	156600.0
8	9	2025-01-09	West	Ajay	Mobile	2	27000	5	Electronics	Regular	51300.0
9	10	2025-01-10	East	Karan	Keyboard	5	1000	0	Accessories	New	5000.0
10	11	2025-01-11	South	Rahul	Laptop	1	61000	0	Electronics	Regular	61000.0
11	12	2025-01-12	North	Meena	Mouse	4	850	5	Accessories	New	3230.0
12	13	2025-01-13	East	Priya	Tablet	3	19000	10	Electronics	Regular	51300.0
13	14	2025-01-14	West	Ajay	Laptop	2	63000	0	Electronics	New	126000.0
14	15	2025-01-15	North	Karan	Charger	5	550	0	Accessories	Regular	2750.0
15	16	2025-01-16	South	Rahul	Keyboard	4	1100	0	Accessories	New	4400.0
16	17	2025-01-17	East	Meena	Mobile	3	26000	5	Electronics	Regular	74100.0
17	18	2025-01-18	West	Priya	Laptop	2	60000	5	Electronics	New	114000.0
18	19	2025-01-19	North	Ajay	Tablet	4	18500	10	Electronics	Regular	66600.0
19	20	2025-01-20	South	Karan	Mouse	6	750	0	Accessories	New	4500.0

In [17]:
Find number of transactions for each CustomerType (Regular vs New)
transactions_by_customertype = (
df["CustomerType"].value_counts().reset_index()
)
transactions_by_customertype.columns = ["CustomerType", "TransactionCount"]
transactions_by_customertype

Out[17]:

	CustomerType	TransactionCount
0	Regular	11
1	New	9

In [18]:
Get the most frequently sold product in each region
most_frequent_product = (
df.groupby(["Region", "Product"])
 .size()
 .reset_index(name="Count")
)
most_frequent_product = most_frequent_product.loc[
most_frequent_product.groupby("Region")["Count"].idxmax()
]
most_frequent_product

Out[18]:

	Region	Product	Count
0	East	Keyboard	2
4	North	Laptop	2
11	South	Mouse	2
12	West	Laptop	3

In [20]:
Pivot Table: Total sales by Region and Category
pivot_sales_region_category = pd.pivot_table(
df,
 values="TotalSales",
 index="Region",
 columns="Category",
 aggfunc="sum",
 fill_value=0
)
pivot_sales_region_category

Out[20]:

	Category	Accessories	Electronics
	Region		
	East	8240.0	161400.0
	North	5980.0	432200.0
	South	15900.0	61000.0

West	0.0	353300.0
------	-----	----------

In []: