ESSENTIALS OF DATA SCIENCE All DIVISIONS

Theory Activity No. 1

NAME : KETAN SUNIL RAKHE
DIVISION : ET2
ROLL NO : ET2-02
PRN : 202401070040
SUBJECT : EDS

sales_data_sample.csv

20 Problem Statements + Solutions using Pandas and NumPy

No	Problem Statement	Pandas/NumPy Methods Used
1	What is the total sales amount for the entire dataset?	df['Total'].sum()
2	Find the average quantity of products sold per transaction.	df['Quantity'].mean()
3	Identify the product with the highest total revenue.	df.groupby('Product')['Total'].sum().idxmax()
4	Count how many transactions occurred in each region.	df['Region'].value_counts()
5	Find the most commonly used payment method.	df['Payment Method'].mode()[0]

No	Problem Statement	Pandas/NumPy Methods Used
6	Show all sales made on or after 1st Jan 2024.	df[df['Date'] >= '2024-01-01']
7	Calculate total revenue generated by each salesperson.	df.groupby('Salesperson')['Total'].sum()
8	What is the standard deviation of unit prices?	df['Unit Price'].std()
9	List the top 5 highest-value invoices by total amount.	df.sort_values('Total', ascending=False).head(5)
10	Count the number of unique products sold.	df['Product'].nunique()
11	Filter the transactions where quantity sold is greater than 10.	df[df['Quantity'] > 10]
12	Group sales data by region and show average unit price.	df.groupby('Region')['Unit Price'].mean()
13	Add a new column showing tax (18%) for each transaction.	df['Tax'] = df['Total'] * 0.18
14	Create a column showing discounted price (10% off total).	df['Discounted'] = df['Total'] * 0.90
15	Find the correlation between quantity and total sales.	df[['Quantity','Total']].corr()
16	Identify transactions with missing customer information.	df[df['Customer'].isnull()]
17	Replace null customer names with "Unknown".	df['Customer'].fillna('Unknown')
18	Extract transactions where payment was done via	df[df['Payment Method'] == 'Credit Card']

No	Problem Statement	Pandas/NumPy Methods Used
	Credit Card.	
19	Sort transactions by date.	df.sort_values('Date')
20	Get the earliest and latest transaction dates.	df['Date'].min(), df['Date'].max()

Code:

```
import pandas as pd

import numpy as np


# Load the dataset from the correct path

df = pd.read_csv("/content/sample_data/sales_data_sample.csv",
encoding='ISO-8859-1')


# 1. Total sales revenue

total_sales = df['SALES'].sum()

print("1. Total Sales Revenue:", total_sales)


# 2. Average price per item sold

average_price = df['PRICEEACH'].mean()

print("2. Average Price Each:", average_price)


# 3. Total quantity sold per product line

quantity_per_productline =
df.groupby('PRODUCTLINE')['QUANTITYORDERED'].sum()

print("3. Quantity Sold per Product Line:\n", quantity_per_productline)


# 4. Total revenue by country
```

```
revenue_by_country = df.groupby('COUNTRY')['SALES'].sum()

print("4. Revenue by Country:\n", revenue_by_country)


# 5. Most popular product line (by quantity)

most_popular_productline =
df.groupby('PRODUCTLINE')['QUANTITYORDERED'].sum().idxmax()

print("5. Most Popular Product Line:", most_popular_productline)


# 6. Number of orders per year

orders_per_year = df['YEAR_ID'].value_counts()

print("6. Orders per Year:\n", orders_per_year)


# 7. Unique number of products sold

unique_products = df['PRODUCTCODE'].nunique()

print("7. Unique Products Sold:", unique_products)


# 8. Highest sales per order

highest_sale = df['SALES'].max()

print("8. Highest Sale Value:", highest_sale)


# 9. Order with the highest quantity

max_quantity_order = df[df['QUANTITYORDERED'] ==
df['QUANTITYORDERED'].max()]

print("9. Max Quantity Order:\n", max_quantity_order[['ORDERNUMBER',
'QUANTITYORDERED', 'PRODUCTLINE']])


# 10. Count orders by deal size

deal_size_counts = df['DEALSIZE'].value_counts()

print("10. Deal Size Counts:\n", deal_size_counts)


# 11. Average sale per order line

average_sale_per_order = df['SALES'].mean()
```

```
print("11. Average Sale per Order Line:", average_sale_per_order)

# 12. Correlation between quantity ordered and total sales
correlation_quantity_sales = df[['QUANTITYORDERED', 'SALES']].corr()
print("12. Correlation Between Quantity and Sales:\n",
correlation_quantity_sales)

# 13. Earliest order date
df['ORDERDATE'] = pd.to_datetime(df['ORDERDATE'], errors='coerce')
earliest_order_date = df['ORDERDATE'].min()
print("13. Earliest Order Date:", earliest_order_date)

# 14. Count of orders per status
orders_per_status = df['STATUS'].value_counts()
print("14. Orders per Status:\n", orders_per_status)

# 15. Total revenue by status
revenue_by_status = df.groupby('STATUS')['SALES'].sum()
print("15. Revenue by Status:\n", revenue_by_status)

# 16. Discounted sales (10% off)
df['DISCOUNTED_SALE'] = df['SALES'] * 0.9
print("16. Discounted Sales Sample:\n", df[['SALES',
'DISCOUNTED_SALE']].head())

# 17. Profit assuming MSRP - PriceEach
df['PROFIT'] = (df['MSRP'] - df['PRICEEACH']) * df['QUANTITYORDERED']
print("17. Profit Sample:\n", df[['PRICEEACH', 'MSRP', 'QUANTITYORDERED',
'PROFIT']].head())

# 18. Country with highest average sales
highest_avg_country = df.groupby('COUNTRY')['SALES'].mean().idxmax()
```

```
print("18. Country with Highest Avg Sales:", highest_avg_country)

# 19. Customers per Country
customers_per_country = df['COUNTRY'].value_counts()
print("19. Customers per Country:\n", customers_per_country)

# 20. Orders with negative profit
negative_profit_orders = df[df['PROFIT'] < 0]
print("20. Negative Profit Orders:\n",
negative_profit_orders[['ORDERNUMBER', 'PROFIT', 'PRODUCTLINE']])
```

Output:

```
1. Total Sales Revenue: 10032628.85

2. Average Price Each: 83.65854410201914

3. Quantity Sold per Product Line:
 PRODUCTLINE
Classic Cars        33992
Motorcycles         11663
Planes              10727
Ships                8127
Trains               2712
Trucks and Buses    10777
Vintage Cars        21069
Name: QUANTITYORDERED, dtype: int64

4. Revenue by Country:
 COUNTRY
```

```
Australia       630623.10
Austria         202062.53
Belgium         108412.62
Canada          224078.56
Denmark         245637.15
Finland         329581.91
France         1110916.52
Germany         220472.09
Ireland          57756.43
Italy           374674.31
Japan           188167.81
Norway          307463.70
Philippines      94015.73
Singapore       288488.41
Spain          1215686.92
Sweden          210014.21
Switzerland     117713.56
UK              478880.46
USA            3627982.83
Name: SALES, dtype: float64
```

5. Most Popular Product Line: Classic Cars

6. Orders per Year:

```
 YEAR_ID
2004    1345
2003    1000
2005     478
Name: count, dtype: int64
```

7. Unique Products Sold: 109

8. Highest Sale Value: 14082.8

9. Max Quantity Order:

	ORDERNUMBER	QUANTITYORDERED	PRODUCTLINE
418	10405	97	Classic Cars

10. Deal Size Counts:

DEALSIZE

Medium 1384

Small 1282

Large 157

Name: count, dtype: int64

11. Average Sale per Order Line: 3553.889071909316

12. Correlation Between Quantity and Sales:

	QUANTITYORDERED	SALES
QUANTITYORDERED	1.000000	0.551426
SALES	0.551426	1.000000

13. Earliest Order Date: 2003-01-06 00:00:00

14. Orders per Status:

STATUS

Shipped 2617

Cancelled 60

Resolved 47

On Hold 44

```
In Process      41
Disputed        14
Name: count, dtype: int64
```

15. Revenue by Status:

```
 STATUS
Cancelled      194487.48
Disputed        72212.86
In Process     144729.96
On Hold        178979.19
Resolved       150718.28
Shipped       9291501.08
Name: SALES, dtype: float64
```

16. Discounted Sales Sample:

	SALES	DISCOUNTED_SALE
0	2871.00	2583.900
1	2765.90	2489.310
2	3884.34	3495.906
3	3746.70	3372.030
4	5205.27	4684.743

17. Profit Sample:

	PRICEEACH	MSRP	QUANTITYORDERED	PROFIT
0	95.70	95	30	-21.00
1	81.35	95	34	464.10
2	94.74	95	41	10.66
3	83.26	95	45	528.30
4	100.00	95	49	-245.00

18. Country with Highest Avg Sales: Denmark

19. Customers per Country:

COUNTRY	
USA	1004
Spain	342
France	314
Australia	185
UK	144
Italy	113
Finland	92
Norway	85
Singapore	79
Canada	70
Denmark	63
Germany	62
Sweden	57
Austria	55
Japan	52
Belgium	33
Switzerland	31
Philippines	26
Ireland	16

Name: count, dtype: int64

20. Negative Profit Orders:

	ORDERNUMBER	PROFIT	PRODUCTLINE
0	10107	-21.00	Motorcycles
4	10159	-245.00	Motorcycles
5	10168	-59.76	Motorcycles

7	10188	-240.00	Motorcycles
8	10201	-78.54	Motorcycles
...	...	...	...
2818	10350	-920.00	Ships
2819	10373	-1334.00	Ships
2820	10386	-1978.00	Ships
2821	10397	-280.16	Ships
2822	10414	-541.44	Ships

[843 rows x 3 columns]