



```
print("\nQues 4:")  
print("Median Amount Paid Each Year by Payment Method:")  
print(median_payments)
```



Ques 4:

Median Amount Paid Each Year by Payment Method:

	year	payment_method	amount_paid
0	2018	Bank Transfer	281.65
1	2018	Check	216.60
2	2018	Credit Card	229.15
3	2019	Bank Transfer	184.20
4	2019	Check	410.20
5	2019	Credit Card	401.90
6	2020	Bank Transfer	225.10
7	2020	Check	413.10
8	2020	Credit Card	285.25
9	2021	Bank Transfer	255.30
10	2021	Check	435.10
11	2021	Credit Card	208.70
12	2022	Bank Transfer	196.50
13	2022	Check	275.50
14	2022	Credit Card	326.20

```

#Ques4: What is the median amount paid each year for all payment methods?

# Extract year
payment_df["year"] = payment_df["payment_date"].dt.year

# Group by year and payment method, then compute median
median_payments = payment_df.groupby(["year", "payment_method"])["amount_paid"].median().reset_index()

print("\nQues 4:")
print("Median Amount Paid Each Year by Payment Method:")
print(median_payments)

```



```

Ques 4:
Median Amount Paid Each Year by Payment Method:
   year payment_method  amount_paid
0  2018   Bank Transfer    281.65
1  2018         Check    216.60
2  2018   Credit Card    229.15
3  2019   Bank Transfer    184.20
4  2019         Check    410.20
5  2019   Credit Card    401.90
6  2020   Bank Transfer    325.10

```

```

#Ques3:- What was the average inflation rate when their subscriptions were renewed?
# Filter only renewed subscriptions
renewed_subs = subscription_df[subscription_df["renewed"] == True]

# Function to find matching inflation rate
def find_inflation(date):
    row = finance_df[(finance_df["start_date"] <= date) & (finance_df["end_date"] >= date)]
    return row.iloc[0]["inflation_rate"] if not row.empty else None

# Apply function
renewed_subs["inflation_rate"] = renewed_subs["end_date"].apply(find_inflation)

# Average inflation
avg_inflation = renewed_subs["inflation_rate"].mean()

print("\nQues 3:")
print("Average Inflation Rate at Time of Renewals:", round(avg_inflation, 2))

```

```

Ques 3:
Average Inflation Rate at Time of Renewals: 4.31
<ipython-input-29-29772861c581>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.

```



```
#Question 2:Which industry in the organization has the highest renewal rate?
# Merge to get industry + renewal status
merged_df = pd.merge(industry_df, subscription_df, on="client_id")

# Calculate renewal rates by industry
renewal_rates = merged_df.groupby("industry")["renewed"].mean().sort_values(ascending=False)

# Top industry by renewal rate
highest_renewal_industry = renewal_rates.idxmax()
highest_rate = renewal_rates.max()

print("\nQues 2:")
print("Industry with Highest Renewal Rate:", highest_renewal_industry)
print("Renewal Rate:", round(highest_rate * 100, 2), "%")
```



```
Ques 2:
Industry with Highest Renewal Rate: Gaming
Renewal Rate: 72.73 %
```



```
# Question 1: How many finance lending and blockchain clients does the organization have?
# Filter the industries
finance_lending_clients = industry_df[industry_df["industry"] == "Finance Lending"]
blockchain_clients = industry_df[industry_df["industry"] == "Block Chain"]

# Count unique clients, to remove duplicate values
num_finance_lending = finance_lending_clients["client_id"].nunique()
num_blockchain = blockchain_clients["client_id"].nunique()

print("Ques 1:")
print("Finance Lending Clients:", num_finance_lending)
print("Blockchain Clients:", num_blockchain)
```

Close



```
Ques 1:
Finance Lending Clients: 22
Blockchain Clients: 25
```

```
# Importing files from the system
from google.colab import files
import pandas as pd
```

```
uploaded = files.upload()
```

```
# Reading files
```

```
finance_df = pd.read_csv("finanical_information.csv")
industry_df = pd.read_csv("industry_client_details.csv")
payment_df = pd.read_csv("payment_information.csv")
subscription_df = pd.read_csv("subscription_information (1).csv")
```

```
# Converting date fields will help us to analyse data easily.
```

```
finance_df["start_date"] = pd.to_datetime(finance_df["start_date"])
finance_df["end_date"] = pd.to_datetime(finance_df["end_date"])
subscription_df["end_date"] = pd.to_datetime(subscription_df["end_date"])
payment_df["payment_date"] = pd.to_datetime(payment_df["payment_date"])
```



Choose Files 4 files

- **finanical_information.csv**(text/csv) - 770 bytes, last modified: 4/21/2025 - 100% done
- **industry_client_details.csv**(text/csv) - 3842 bytes, last modified: 4/21/2025 - 100% done
- **payment_information.csv**(text/csv) - 3898 bytes, last modified: 4/21/2025 - 100% done
- **subscription_information (1).csv**(text/csv) - 4658 bytes, last modified: 4/21/2025 - 100% done