

```
In [214... import pandas as pd
import glob
import os

folder_path = 'data'

csv_files = glob.glob(os.path.join(folder_path, 'Month-*.csv'))

observations = []

for file in csv_files:
    df = pd.read_csv(file)
    num_observations = len(df)
    observations.append((file, num_observations))

max_observation_file, max_observations = max(observations, key=lambda x: x[1])
min_observation_file, min_observations = min(observations, key=lambda x: x[1])

print(f"DataFrame with the most observations: {max_observation_file}, {max_observations} observations")
print(f"DataFrame with the least observations: {min_observation_file}, {min_observations} observations")
```

DataFrame with the most observations: data/Month-11.csv, 94315 observations  
DataFrame with the least observations: data/Month-02.csv, 44380 observations

```
In [215... dfs = [pd.read_csv(file) for file in csv_files]
df = pd.concat(dfs, ignore_index=True)

print(f"Shape of combined DataFrame:", df.shape)
print(f"Account_ID value for the last transaction:", df.iloc[-1]['Account_ID'])
```

Shape of combined DataFrame: (698159, 10)  
Account\_ID value for the last transaction: 495150

```
In [216... df
```

	Account_ID	Transaction_Timestamp	Factor_A	Factor_B	Factor_C	Factor_D	Factor_E	Response	Transaction_Status	Month	
	0	4	2009-08-17T14:20:06Z	8	23	AX	21	NaN	1020	Approved	Aug
	1	6	2009-08-31T17:33:42Z	2	6	VI	25	G2	1020	Approved	Aug
	2	22	2009-08-04T11:23:11Z	8	23	AX	26	NaN	1020	Approved	Aug
	3	22	2009-08-11T12:44:38Z	8	23	AX	26	NaN	1020	Approved	Aug
	4	32	2009-08-15T14:14:42Z	8	23	AX	21	NaN	1020	Approved	Aug
	...	...	...	...	...	...	...	...	...	...	
698154	495134	2009-11-30T23:55:08Z	2	6	VI	20	D	1020	Approved	Nov	
698155	495135	2009-11-30T23:57:00Z	7	15	MC	20	MCW	1020	Approved	Nov	
698156	495136	2009-11-30T23:57:33Z	2	6	VI	21	A	1020	Approved	Nov	
698157	495137	2009-11-30T23:57:39Z	7	15	MC	35	MPL	1020	Approved	Nov	
698158	495150	2009-11-30T23:58:32Z	2	6	VI	20	B	1020	Approved	Nov	

698159 rows x 10 columns

```
In [217... for col in df.columns:
    unique_count = df[col].nunique()
    print(f"Column '{col}' has {unique_count} unique values.")
```

Column 'Account\_ID' has 475413 unique values.  
Column 'Transaction\_Timestamp' has 686538 unique values.  
Column 'Factor\_A' has 7 unique values.  
Column 'Factor\_B' has 6 unique values.  
Column 'Factor\_C' has 4 unique values.  
Column 'Factor\_D' has 15 unique values.  
Column 'Factor\_E' has 62 unique values.  
Column 'Response' has 42 unique values.  
Column 'Transaction\_Status' has 2 unique values.  
Column 'Month' has 11 unique values.

```
In [218... def convert_to_qtr(month):
    if month in ['Jan', 'Feb', 'Mar']:
        return 'Q1'
    elif month in ['Apr', 'May', 'Jun']:
        return 'Q2'
    elif month in ['Jul', 'Aug', 'Sep']:
        return 'Q3'
    elif month in ['Oct', 'Nov', 'Dec']:
        return 'Q4'
df['Quarter'] = df['Month'].apply(convert_to_qtr)
```

```
In [237... df['Quarter'].value_counts()
```

Quarter  
Q3 205615  
Q4 174592  
Q2 165778  
Q1 152174  
Name: count, dtype: int64

```
In [220... df2 = [pd.read_csv(file) for file in csv_files if pd.read_csv(file).shape[0] >= 60000]
df2 = pd.concat(df2, ignore_index=True)
print('DF2 has:', df2.shape[0], 'observations.')
```

DF2 has: 380207 observations.

```
In [221... def get_data(files, min_req_obs):
    df_list = [pd.read_csv(file) for file in files if pd.read_csv(file).shape[0] >= min_req_obs]
    combined_df = pd.concat(df_list, ignore_index=True)
    return combined_df

df3 = get_data(csv_files, min_req_obs=75000)
print('DF3 has:', df3.shape[0], 'observations.')
```

DF3 has: 174592 observations.

```
In [222... values = [25000, 50000, 75000, 90000]
for min_req_obs in values:
    df4 = get_data(csv_files, min_req_obs=min_req_obs)
    num_observations = df4.shape[0]
    print(f"When min_req_obs = {min_req_obs}: The number of observations = {num_observations}")
```

When min\_req\_obs = 25000: The number of observations = 698159  
When min\_req\_obs = 50000: The number of observations = 653779  
When min\_req\_obs = 75000: The number of observations = 174592  
When min\_req\_obs = 90000: The number of observations = 94315

```
In [223... import random

def divisible(a, b):
    return a % b == 0

random.seed(123)
a = random.randint(10, 100)
b = random.randint(1, 10)

result = divisible(a, b)
print(f"Is {a} divisible by {b}? {result}")
```

Is 16 divisible by 5? False

```
In [224... def lucky_sum(*args):
    sum_result = 0
    skip = False

    for num in args:
        if num == 13:
            break
        if not skip:
            sum_result += num
            skip = (num != 13)
    return sum_result

random.seed(313)

numbers = random.choices(range(1, 14), k=26)
result = lucky_sum(*numbers)
print("Result of lucky_sum:", result)
```

Result of lucky\_sum: 8

```
In [ ]:
```