DATA INGESTION

TASK-1

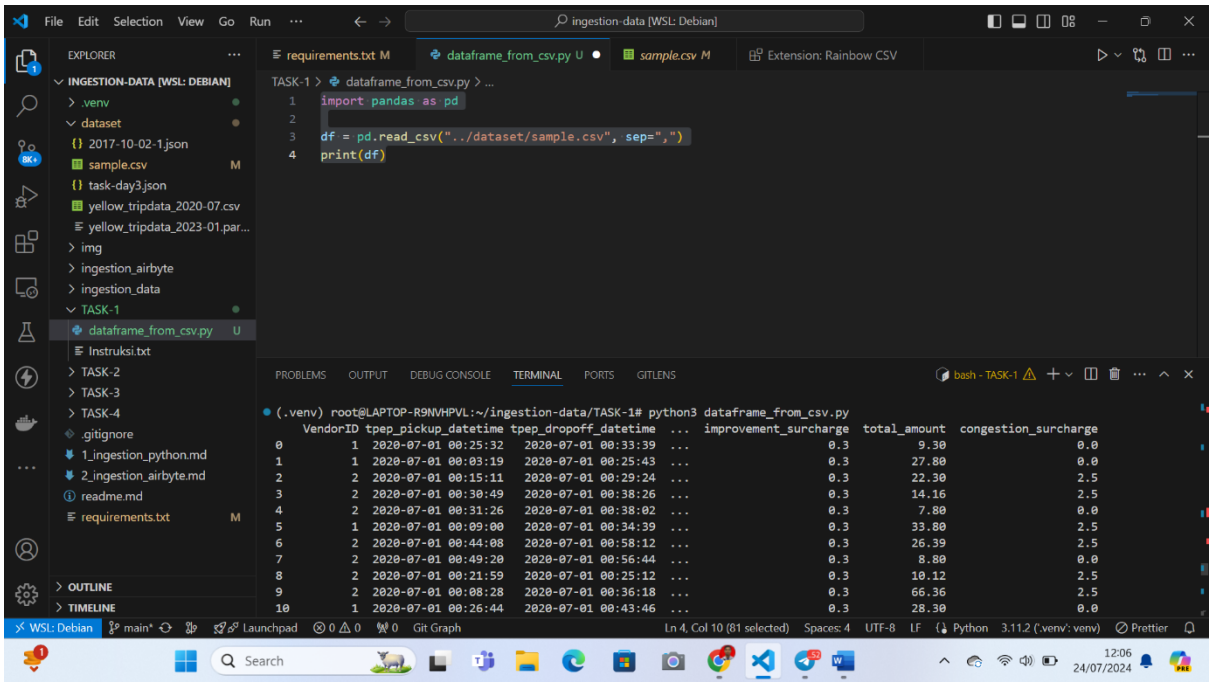
1. **We have already learned how to create DataFrame from files here. Now, we are going to create a DataFrame from a larger csv file on our datasets.**

   Jawaban :

```
import pandas as pd

df = pd.read_csv("../dataset/sample.csv", sep=",")
print(df)
```

   Screenshot :



2. **Rename all the columns with snake_case format.**

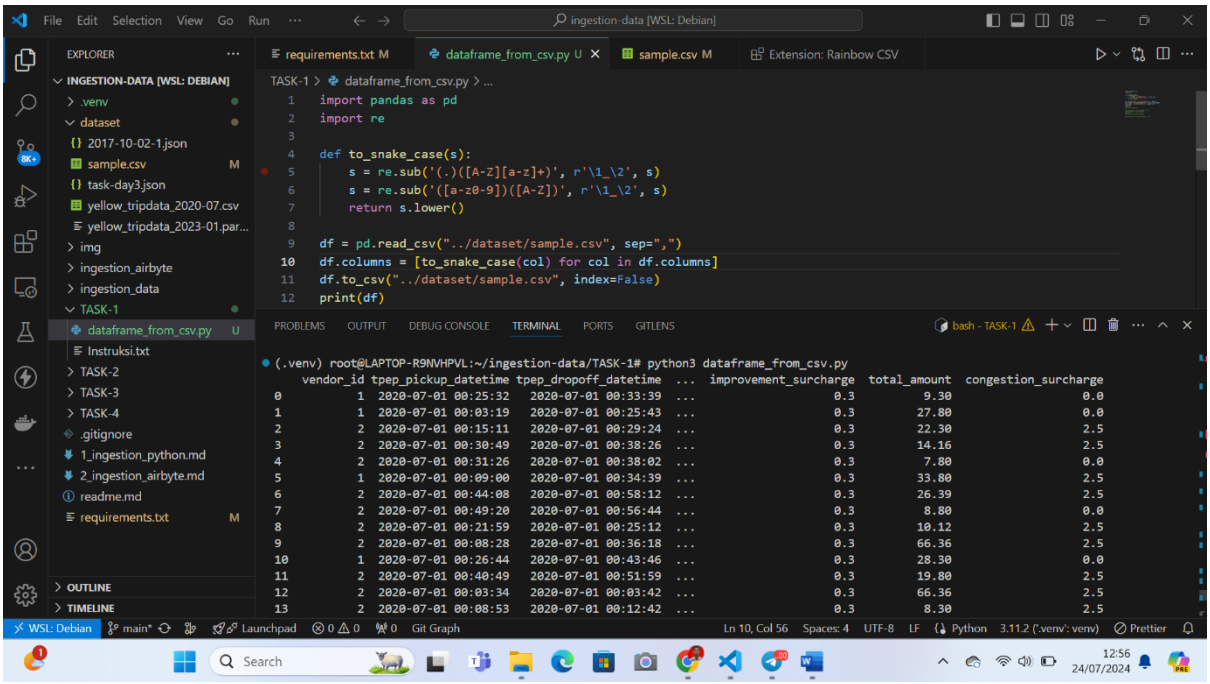
   Jawaban :

```
import pandas as pd
import re

def to_snake_case(s):
    s = re.sub('(.)([A-Z][a-z]+)', r'\1_\2', s)
    s = re.sub('([a-z0-9])([A-Z])', r'\1_\2', s)
    return s.lower()

df = pd.read_csv("../dataset/sample.csv", sep=",")
df.columns = [to_snake_case(col) for col in df.columns]
df.to_csv("../dataset/sample.csv", index=False)
print(df)
```

Screenshot :



3. **Select only 10 top of highest number of `passenger_count`, show only columns `vendor_id`, `passenger_count`, `trip_distance`, `payment_type`, `fare_amount`, `extra`, `mta_tax`, `tip_amount`, `tolls_amount`, `improvement_surcharge`, `total_amount`, `congestion_surcharge` from the DataFrame.**

Jawaban :

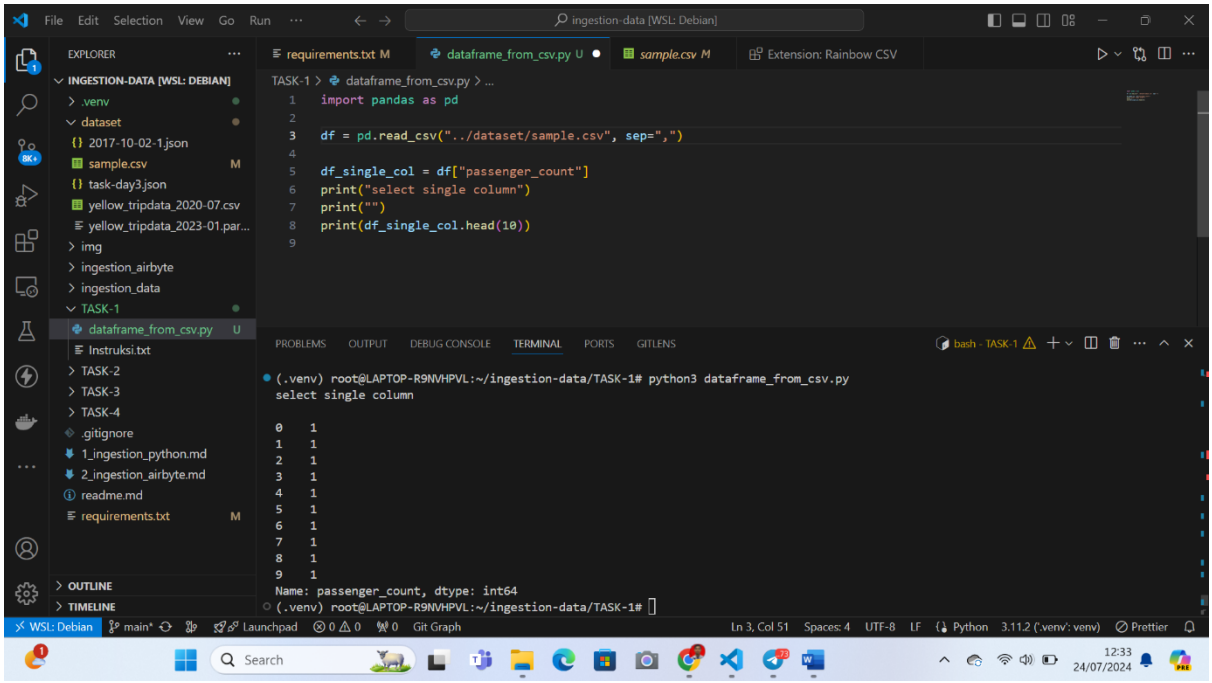
- Select only 10 top of highest number of `passenger_count`

```
import pandas as pd

df = pd.read_csv("../dataset/sample.csv", sep=",")

df_single_col = df["passenger_count"]
print("select single column")
print("")
print(df_single_col.head(10))
```

Screenshot :



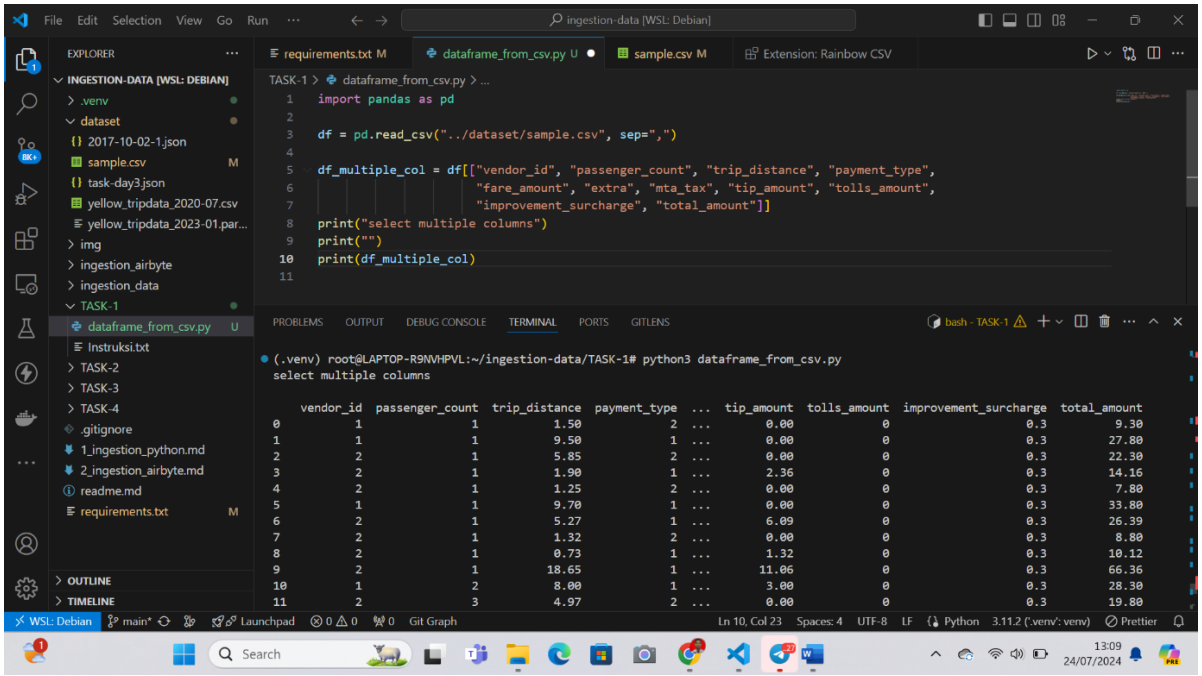
- show only columns `vendor_id, passenger_count, trip_distance, payment_type, fare_amount, extra, mta_tax, tip_amount, tolls_amount, improvement_surcharge, total_amount, congestion_surcharge` from the DataFrame.

```
import pandas as pd

df = pd.read_csv("../dataset/sample.csv", sep=",")

df_multiple_col = df[["vendor_id", "passenger_count", "trip_distance",
"payment_type", "fare_amount", "extra", "mta_tax", "tip_amount",
"tolls_amount", "improvement_surcharge", "total_amount"]]
print("select multiple columns")
print("")
print(df_multiple_col)
```

Screenshot :



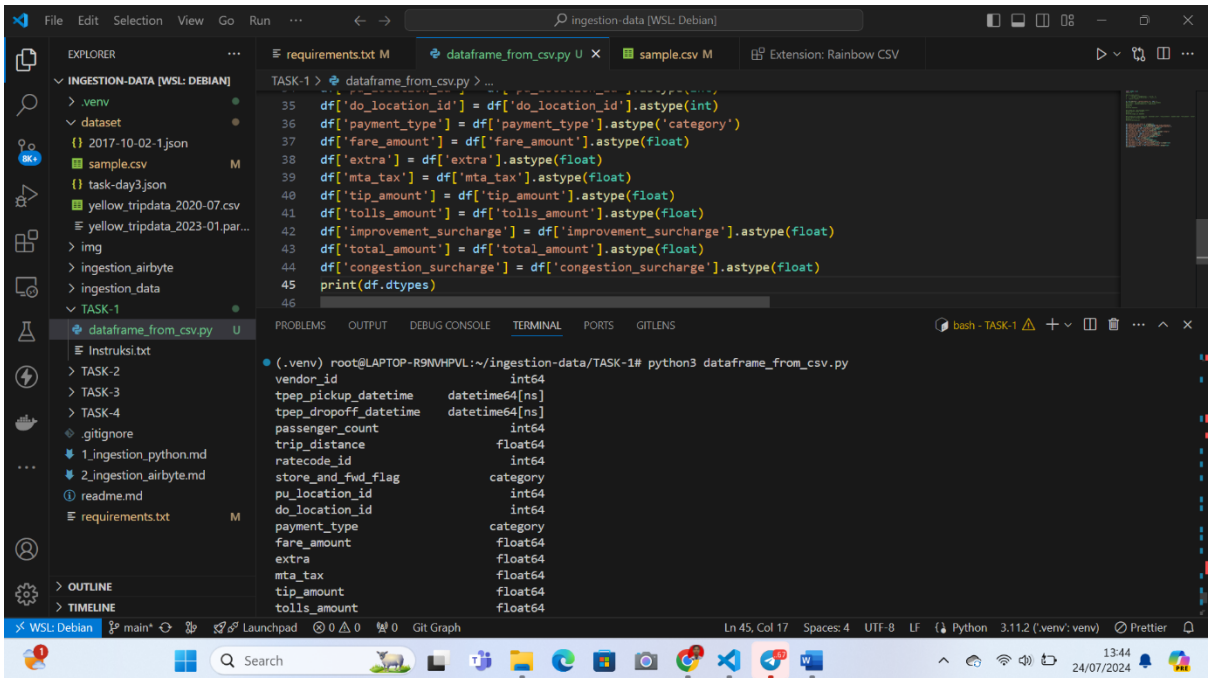
4. **[Extra] Cast the data type to the appropriate value.**

```
import pandas as pd

df = pd.read_csv("../dataset/sample.csv", sep=",")

df['vendor_id'] = df['vendor_id'].astype(int)
df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'])
df['tpep_dropoff_datetime'] =
pd.to_datetime(df['tpep_dropoff_datetime'])
df['passenger_count'] = df['passenger_count'].astype(int)
df['trip_distance'] = df['trip_distance'].astype(float)
df['ratecode_id'] = df['ratecode_id'].astype(int)
df['store_and_fwd_flag'] = df['store_and_fwd_flag'].astype('category')
df['pu_location_id'] = df['pu_location_id'].astype(int)
df['do_location_id'] = df['do_location_id'].astype(int)
df['payment_type'] = df['payment_type'].astype('category')
df['fare_amount'] = df['fare_amount'].astype(float)
df['extra'] = df['extra'].astype(float)
df['mta_tax'] = df['mta_tax'].astype(float)
df['tip_amount'] = df['tip_amount'].astype(float)
df['tolls_amount'] = df['tolls_amount'].astype(float)
df['improvement_surcharge'] = df['improvement_surcharge'].astype(float)
df['total_amount'] = df['total_amount'].astype(float)
df['congestion_surcharge'] = df['congestion_surcharge'].astype(float)
print(df.dtypes)
```

Screenshot :


ingestion-data [WSL: Debian]
requirements.txt M
dataframe_from_csv.py U
sample.csv M
Extension: Rainbow CSV
TASK-1 > dataframe_from_csv.py > ...
35 df['do_location_id'] = df['do_location_id'].astype(int)
36 df['payment_type'] = df['payment_type'].astype('category')
37 df['fare_amount'] = df['fare_amount'].astype(float)
38 df['extra'] = df['extra'].astype(float)
39 df['mta_tax'] = df['mta_tax'].astype(float)
40 df['tip_amount'] = df['tip_amount'].astype(float)
41 df['tolls_amount'] = df['tolls_amount'].astype(float)
42 df['improvement_surcharge'] = df['improvement_surcharge'].astype(float)
43 df['total_amount'] = df['total_amount'].astype(float)
44 df['congestion_surcharge'] = df['congestion_surcharge'].astype(float)
45 print(df.dtypes)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
(.venv) root@LAPTOP-R9NVHPVL:~/ingestion-data/TASK-1# python3 dataframe_from_csv.py
vendor_id int64
tpep_pickup_datetime datetime64[ns]
tpep_dropoff_datetime datetime64[ns]
passenger_count int64
trip_distance float64
ratecode_id int64
store_and_fwd_flag category
pu_location_id int64
do_location_id int64
payment_type category
fare_amount float64
extra float64
mta_tax float64
tip_amount float64
tolls_amount float64