

DATA INGESTION

TASK-1

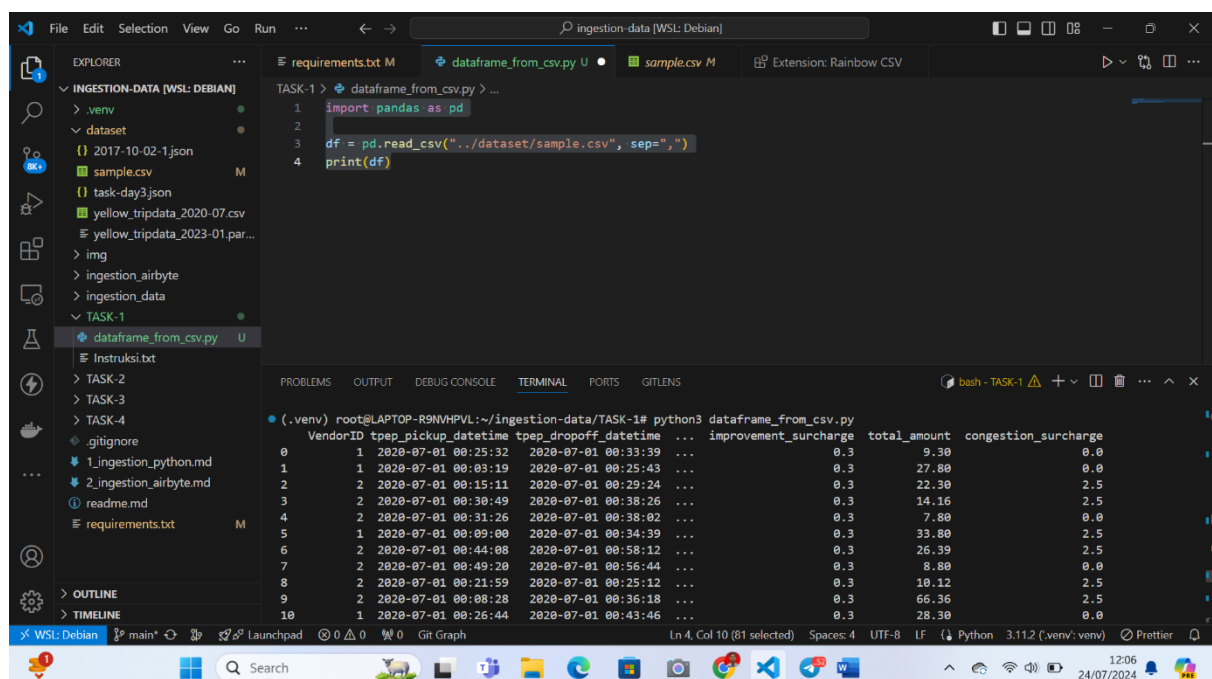
1. We have already learned how to create DataFrame from files [here](#). Now, we are going to create a DataFrame from a larger [csv file](#) on our [datasets](#).

Jawaban :

```
import pandas as pd

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

Screenshot :



The screenshot shows a VS Code editor window with the file explorer on the left. The file explorer shows a project named 'INGESTION-DATA [WSL: DEBIAN]' with subfolders like '.venv', 'dataset', and 'TASK-1'. The 'TASK-1' folder contains a file 'dataframe_from_csv.py'. The editor window shows the following code in 'dataframe_from_csv.py':

```
1 import pandas as pd
2
3 df = pd.read_csv("../dataset/sample.csv", sep=",")
4 print(df)
```

The terminal at the bottom shows the output of running the script:

```
(.venv) root@LAPTOP-R9MHPVL:~/ingestion-data/TASK-1# python3 dataframe_from_csv.py
VendorID tpep_pickup_datetime tpep_dropoff_datetime ... improvement_surcharge total_amount congestion_surcharge
0 1 2020-07-01 00:25:32 2020-07-01 00:33:39 ... 0.3 9.30 0.0
1 1 2020-07-01 00:03:19 2020-07-01 00:25:43 ... 0.3 27.80 0.0
2 2 2020-07-01 00:15:11 2020-07-01 00:29:24 ... 0.3 22.30 2.5
3 2 2020-07-01 00:30:49 2020-07-01 00:38:26 ... 0.3 14.16 2.5
4 2 2020-07-01 00:31:26 2020-07-01 00:38:02 ... 0.3 7.80 0.0
5 1 2020-07-01 00:09:00 2020-07-01 00:34:39 ... 0.3 33.80 2.5
6 2 2020-07-01 00:44:08 2020-07-01 00:58:12 ... 0.3 26.39 2.5
7 2 2020-07-01 00:49:20 2020-07-01 00:56:44 ... 0.3 8.80 0.0
8 2 2020-07-01 00:21:59 2020-07-01 00:25:12 ... 0.3 10.12 2.5
9 2 2020-07-01 00:08:28 2020-07-01 00:36:18 ... 0.3 66.36 2.5
10 1 2020-07-01 00:26:44 2020-07-01 00:43:46 ... 0.3 28.30 0.0
```

2. Rename all the columns with snake_case format.

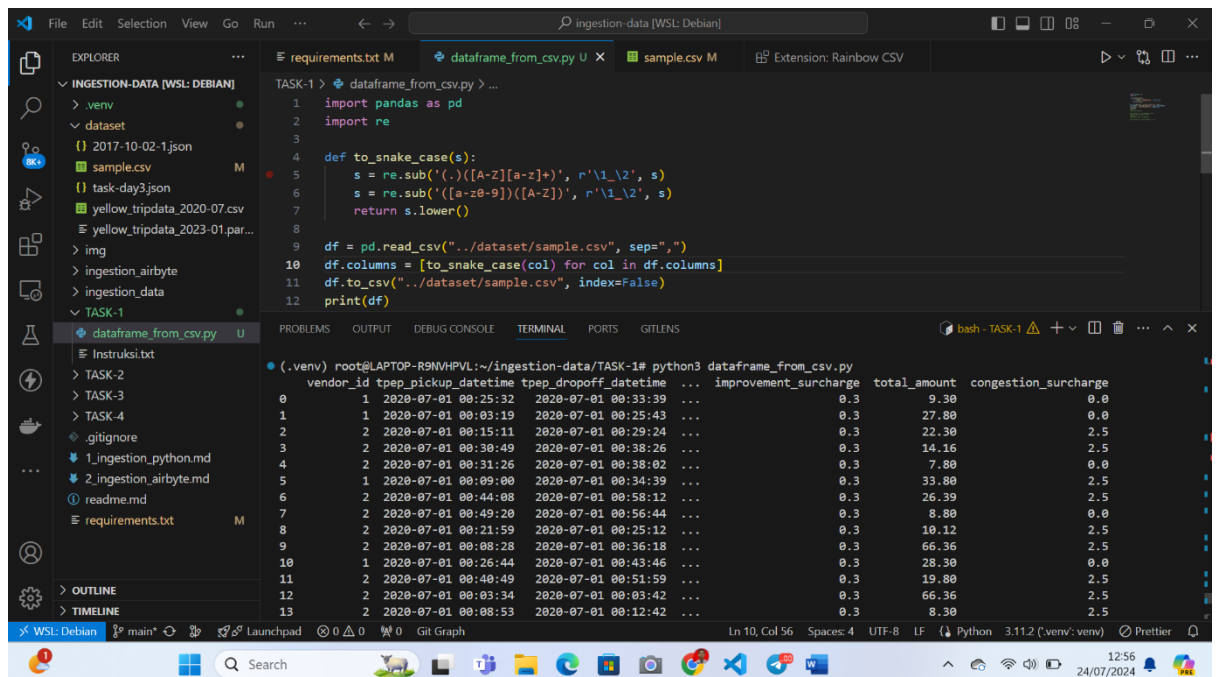
Jawaban :

```
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 :

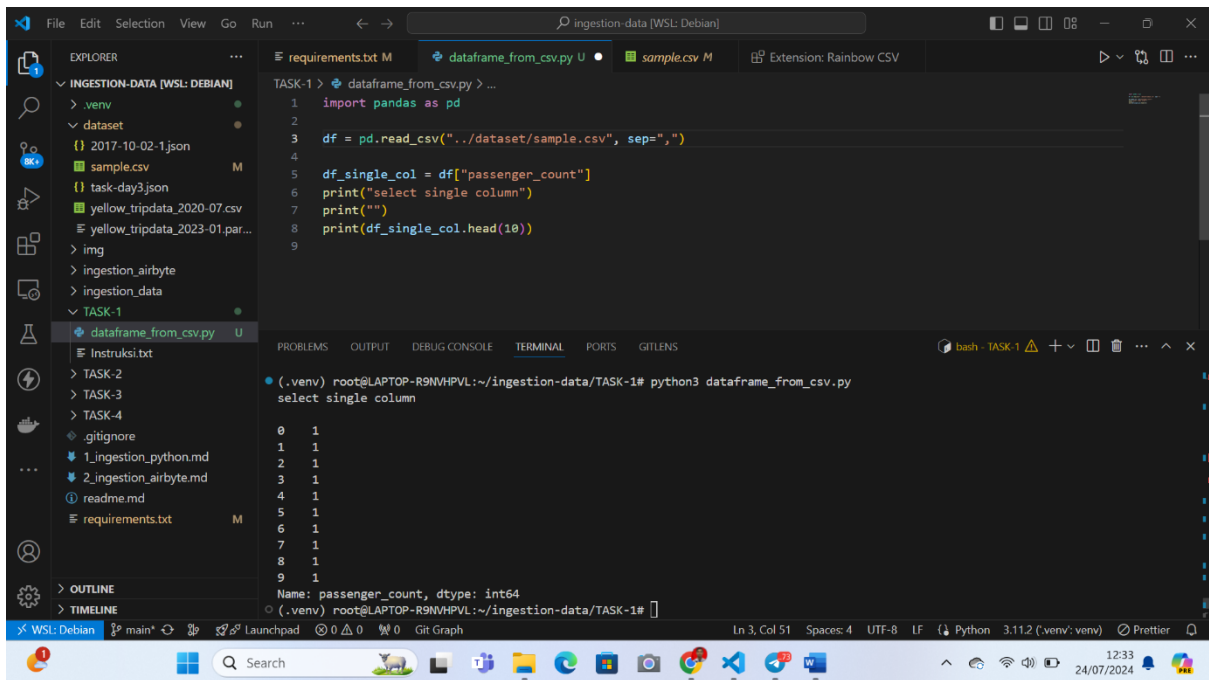
- Select only 10 top of highest number of passenger_count

```
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 :



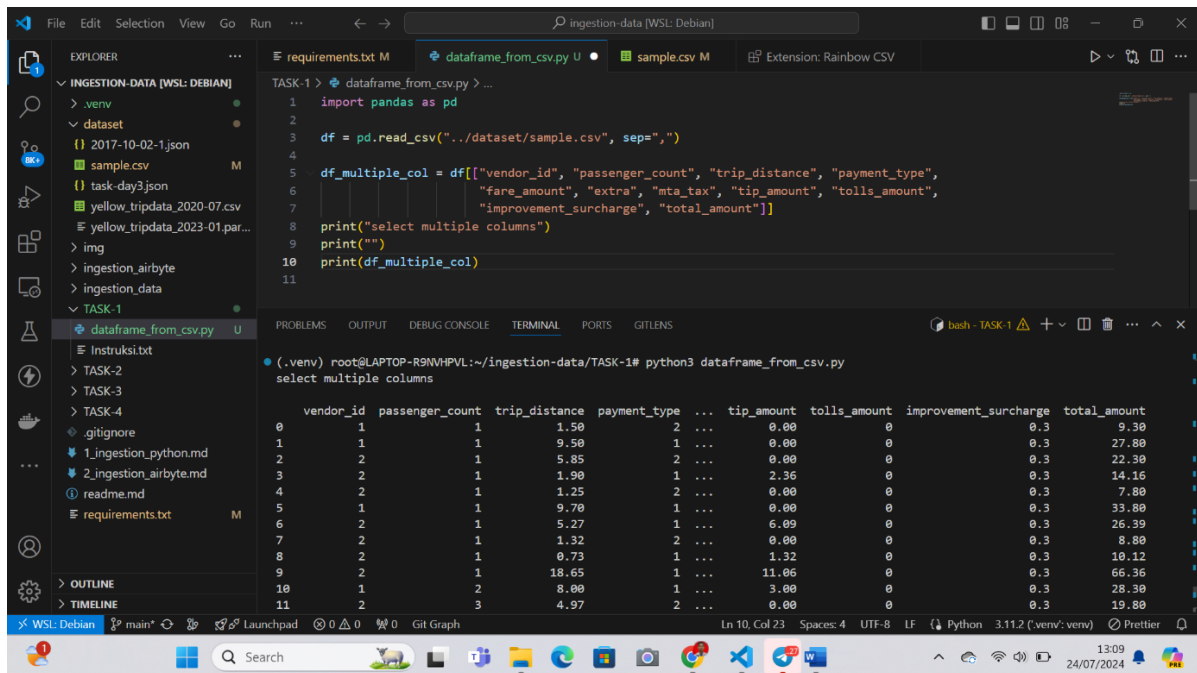
- show only columns `vendor_id`, `passenger_count`, `trip_distance`, `payment_type`, `fare_amount`, `extra`, `mta_tax`, `tip_amount`, `tolls_amount`, `improvement_surcharge`, `total_amount`, `congestion_surcharge` from the DataFrame.

```
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.

```
5. import pandas as pd
6.
7. df = pd.read_csv("../dataset/sample.csv", sep=",")
8.
9. df['vendor_id'] = df['vendor_id'].astype(int)
10. df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'])
11. df['tpep_dropoff_datetime'] =
    pd.to_datetime(df['tpep_dropoff_datetime'])
12. df['passenger_count'] = df['passenger_count'].astype(int)
13. df['trip_distance'] = df['trip_distance'].astype(float)
14. df['ratecode_id'] = df['ratecode_id'].astype(int)
15. df['store_and_fwd_flag'] = df['store_and_fwd_flag'].astype('category')
16. df['pu_location_id'] = df['pu_location_id'].astype(int)
17. df['do_location_id'] = df['do_location_id'].astype(int)
18. df['payment_type'] = df['payment_type'].astype('category')
19. df['fare_amount'] = df['fare_amount'].astype(float)
20. df['extra'] = df['extra'].astype(float)
21. df['mta_tax'] = df['mta_tax'].astype(float)
22. df['tip_amount'] = df['tip_amount'].astype(float)
23. df['tolls_amount'] = df['tolls_amount'].astype(float)
24. df['improvement_surcharge'] = df['improvement_surcharge'].astype(float)
25. df['total_amount'] = df['total_amount'].astype(float)
26. df['congestion_surcharge'] = df['congestion_surcharge'].astype(float)
27. print(df.dtypes)
```

Screenshot :

The screenshot shows a VS Code editor window with the following components:

- EXPLORER:** Displays the file structure of the 'INGESTION-DATA [WSL: DEBIAN]' project, including files like `.venv`, `dataset`, `sample.csv`, `task-day3.json`, `yellow_tripdata_2020-07.csv`, `yellow_tripdata_2023-01.par...`, `img`, `ingestion_airbyte`, `ingestion_data`, `TASK-1`, `dataframe_from_csv.py`, `requirements.txt`, `Instruksi.txt`, `TASK-2`, `TASK-3`, `TASK-4`, `.gitignore`, `1_ingestion_python.md`, `2_ingestion_airbyte.md`, `readme.md`, `OUTLINE`, and `TIMELINE`.
- EDITOR:** Shows the `dataframe_from_csv.py` file with the following code:

```
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)
46
```
- TERMINAL:** Displays the output of running the script:

```
(.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
```