

| Activity No. 6.2 GETTING STARTED WITH PANDAS | |
|---|-------------------------------|
| Course Code: CPE 310 | Program: BSIE |
| Course Title: FUNDAMENTALS OF DATA SCIENCE | Date Performed:10/01/25 |
| Section: IE22S1 | Date Submitted:10/02/25 |
| Name:Franco,Adrian Faye H. | Instructor: Mr. Roman Richard |
| 1. Discussion | |
| <p>Discuss here the relevant concepts of the activity in your own words.</p> <p>In this activity, I worked with real-world datasets using Python and the pandas library to explore, summarize, and analyze structured data. I completed several exercises that involved reading data from CSV files, understanding its structure, and extracting meaningful insights. The first dataset I worked with was a New York City Yellow Taxi trip dataset, and the second was a student performance dataset from the UCI Machine Learning Repository. Each step of the activity helped reinforce important data analysis concepts and built my confidence in handling data programmatically. I then computed summary statistics for the academic performance features in the student dataset and for important numeric columns like fare_amount, tip_amount, and total_amount in the taxi dataset using.describe(). I learned about averages, ranges, and outliers from this, which is helpful for figuring out trends and the quality of the data.</p> <p>To isolate particular rows with the greatest values, such as the student with the highest admission grade or the longest taxi ride, I also used.idxmax() and.loc[]. These methods were useful for locating noteworthy records and comprehending their background.</p> <p>All things considered, the exercise improved my comfort using pandas to work with actual datasets and helped me apply fundamental data analysis abilities.</p> | |
| 2. Materials and Equipment | |
| <p>What materials did you use? Explain in detail.</p> <p>I performed data analysis for this task using Python programming and the pandas module. It was easy to use on my iPad because the work was completed in Google Colab, a browser-based notebook environment that enables file uploads and code execution. The "Predict Students' Dropout and Academic Success" dataset from the UCI Machine Learning Repository and the 2019 New York City Yellow Taxi Trip data were the two real-world datasets I worked with. I loaded the datasets using pandas, using methods like .head() and .shape to investigate their structure, .describe() to create summary statistics, and .loc[] and .idxmax() to isolate particular rows. These resources and technologies offered a useful and efficient method for analyzing structured data and deriving insightful conclusions.</p> | |
| 3. Procedure | |
| <p>What are the procedures that you performed?</p> <p>To finish the activity, I used Google Colab to do a number of data analysis operations with Python and Pandas. I started by uploading the dataset files and used the read_csv() function to import them into pandas DataFrames. I then used.head() to inspect the first few rows of the data and.shape to verify the dataset size. I then used the.describe() method to create summary statistics for important numerical columns in order to have a better understanding of the distribution of the data. Additionally, I used.idxmax() to get the row with the greatest value in a column and.loc[] to extract pertinent information in order to identify individual records. I was able to successfully analyze, examine, and interpret real-world data by repeating these procedures for both datasets—the student performance data and the taxi trip data.</p> | |
| 4. Output | |
| Screenshot of your outputs based on the procedures. | |

CommandsCodeTextRun all

Files

2019_Yellow_Taxi_Trip_Data.csv

```
import pandas as pd

# Load the CSV (adjust the filename if needed)
df = pd.read_csv("2019_Yellow_Taxi_Trip_Data.csv")

# Show dimensions
print(f"The dataset has {df.shape[0]} rows and {df.shape[1]} columns.")
```

The dataset has 10000 rows and 18 columns.

How can I install Python libraries?

Load data from Google Drive

Show an e

What can I help you build?

8:51 PMPython 3

Run cell (⌘+Enter)
cell executed since last change
executed by ADRIAN FAYE FRANCO
9:00 PM (0 minutes ago)
executed in 0.165s

```
import pandas as pd

df = pd.read_csv("2019_Yellow_Taxi_Trip_Data.csv")

# Show the first 5 rows
```

1 to 5 of 5 entries

| index | vendorid | tpep_pickup_datetime | tpep_dropoff_datetime | passenger_count | trip_distance | ratecodeid | store_and_fwd_flag | pu_locationid | do_locationid |
|-------|----------|-------------------------|-------------------------|-----------------|---------------|------------|--------------------|---------------|---------------|
| 0 | 2 | 2019-10-23T16:39:42.000 | 2019-10-23T17:14:10.000 | 1 | 7.93 | 1 | N | 138 | 170 |
| 1 | 1 | 2019-10-23T16:32:08.000 | 2019-10-23T16:45:26.000 | 1 | 2.0 | 1 | N | 11 | 26 |
| 2 | 2 | 2019-10-23T16:08:44.000 | 2019-10-23T16:21:11.000 | 1 | 1.36 | 1 | N | 163 | 162 |
| 3 | 2 | 2019-10-23T16:22:44.000 | 2019-10-23T16:43:26.000 | 1 | 1.0 | 1 | N | 170 | 163 |
| 4 | 2 | 2019-10-23T16:45:11.000 | 2019-10-23T16:58:49.000 | 1 | 1.96 | 1 | N | 163 | 236 |

Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

Next steps:

Generate code with df

How can I install Python libraries?

Load data from Google Drive

Show an e

What can I help you build?

9:00 PMPython 3

CommandsCodeTextRun all

[5] ✓ Os

```
# Select the specific columns
columns_of_interest = ['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']

# Generate summary statistics
df[columns_of_interest].describe()
```

| | fare_amount | tip_amount | tolls_amount | total_amount |
|-------|--------------|--------------|--------------|--------------|
| count | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| mean | 15.106313 | 2.634494 | 0.623447 | 22.564659 |
| std | 13.954762 | 3.409800 | 6.437507 | 19.209255 |
| min | -52.000000 | 0.000000 | -6.120000 | -65.920000 |
| 25% | 7.000000 | 0.000000 | 0.000000 | 12.375000 |
| 50% | 10.000000 | 2.000000 | 0.000000 | 16.300000 |
| 75% | 16.000000 | 3.250000 | 0.000000 | 22.880000 |
| max | 176.000000 | 43.000000 | 612.000000 | 671.800000 |

How can I install Python libraries?

Load data from Google Drive

Show an e

What can I help you build?

VariablesTerminal

9:04 PMPython 3

CommandsCodeTextRun all

[6] ✓ Os

```
# Find the row with the maximum trip_distance
longest_trip = df.loc[df['trip_distance'].idxmax()]

# Print the required columns for that trip
trip_details = longest_trip[['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']]

# Print the result
longest_trip_details
```

| | 8338 |
|--------------|--------|
| fare_amount | 176.0 |
| tip_amount | 18.29 |
| tolls_amount | 6.12 |
| total_amount | 201.21 |

dtype: object

How can I install Python libraries?

Load data from Google Drive

Show an e

What can I help you build?

VariablesTerminal

9:05 PMPython 3

CommandsCodeTextRun all

RAMDisk

df2.head()

| | Marital Status | Application mode | Application order | Course | Daytime/evening attendance | Previous qualification | qua |
|---|----------------|------------------|-------------------|--------|----------------------------|------------------------|-----|
| 0 | 1 | 17 | 5 | 171 | 1 | 1 | |
| 1 | 1 | 15 | 1 | 9254 | 1 | 1 | |
| 2 | 1 | 1 | 5 | 9070 | 1 | 1 | |
| 3 | 1 | 17 | 2 | 9773 | 1 | 1 | |
| 4 | 2 | 39 | 1 | 8014 | 0 | 1 | |

5 rows x 37 columns

You can find numeric columns automatically
numeric_cols = df2.select_dtypes(include='number').columns.tolist()

Or pick some features, e.g. "Admission grade", "Curricular units 1st sem (gra
df2[numeric_cols].describe()

| | Marital Status | Application mode | Application order | Course | Daytime/evening attendance | Pre | qualific |
|-------|----------------|------------------|-------------------|-------------|----------------------------|--------|----------|
| count | 4424.000000 | 4424.000000 | 4424.000000 | 4424.000000 | 4424.000000 | 4424.0 | |
| mean | 1.178571 | 18.669078 | 1.727848 | 8856.642631 | 0.890823 | 4.5 | |
| std | 0.605747 | 17.484682 | 1.313793 | 2063.566416 | 0.311897 | 10.2 | |
| min | 1.000000 | 1.000000 | 0.000000 | 33.000000 | 0.000000 | 1.0 | |
| 25% | 1.000000 | 1.000000 | 1.000000 | 9085.000000 | 1.000000 | 1.0 | |
| 50% | 1.000000 | 17.000000 | 1.000000 | 9238.000000 | 1.000000 | 1.0 | |
| 75% | 1.000000 | 39.000000 | 2.000000 | 9556.000000 | 1.000000 | 1.0 | |
| max | 6.000000 | 57.000000 | 9.000000 | 9991.000000 | 1.000000 | 43.0 | |





8 rows x 36 columns

What can I help you build?

max_admission_index = df2['Admission grade'].idxmax()


VariablesTerminal


9:25 PMPython 3



```
max_admission_index = df2['Admission grade'].idxmax()
student_max_admission = df2.loc[max_admission_index]

# Show some key columns
cols_of_interest = ['Admission grade', 'Curricular units 1st sem (grade)',
                    'Curricular units 2nd sem (grade)', 'Target']
student_max_admission[cols_of_interest]
```





| | |
|----------------------------------|---------|
| | 702 |
| Admission grade | 190.0 |
| Curricular units 1st sem (grade) | 13.875 |
| Curricular units 2nd sem (grade) | 12.5 |
| Target | Dropout |

dtype: object

5. Supplementary Activity

Include here screenshots of the module completion test.

6. Assessment Rubric