

```
In [3]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
In [11]: import pandas as pd
```

Create a DataFrame by reading in the 2019\_Yellow\_Taxi\_Trip\_Data.csv file. Examine the first 5 rows.

```
In [19]: taxi_data = pd.read_csv('/content/drive/MyDrive/Copy of 2019_Yellow_Taxi_Trip_Data.csv')
print(taxi_data.head())
```

	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	
0	2	2019-10-23T16:39:42.000	2019-10-23T17:14:10.000	
1	1	2019-10-23T16:32:08.000	2019-10-23T16:45:26.000	
2	2	2019-10-23T16:08:44.000	2019-10-23T16:21:11.000	
3	2	2019-10-23T16:22:44.000	2019-10-23T16:43:26.000	
4	2	2019-10-23T16:45:11.000	2019-10-23T16:58:49.000	

  

	passenger_count	trip_distance	ratecodeid	store_and_fwd_flag	
0	1	7.93	1		N
1	1	2.00	1		N
2	1	1.36	1		N
3	1	1.00	1		N
4	1	1.96	1		N

  

	pulocationid	dolocationid	payment_type	fare_amount	extra	mta_tax	
0	138	170	1	29.5	1.0	0.5	
1	11	26	1	10.5	1.0	0.5	
2	163	162	1	9.5	1.0	0.5	
3	170	163	1	13.0	1.0	0.5	
4	163	236	1	10.5	1.0	0.5	

  

	tip_amount	tolls_amount	improvement_surcharge	total_amount	
0	7.98	6.12	0.3	47.90	
1	0.00	0.00	0.3	12.30	
2	2.00	0.00	0.3	15.80	
3	4.32	0.00	0.3	21.62	
4	0.50	0.00	0.3	15.30	

  

	congestion_surcharge
0	2.5
1	0.0
2	2.5
3	2.5
4	2.5

Find the dimensions (number of rows and number of columns) in the data.

```
In [20]: rows, cols = taxi_data.shape
```

```
print(f"Total Rows: {rows}")
print(f"Total Columns: {cols}")
```

Total Rows: 10000

Total Columns: 18

Using the data in the 2019\_Yellow\_Taxi\_Trip\_Data.csv file, calculate summary statistics for the fare\_amount, tip\_amount, tolls\_amount, and total\_amount columns.

```
In [21]: financial_cols = ['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']
summary = taxi_data[financial_cols].describe()
print(summary)
```

	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

Isolate the fare\_amount, tip\_amount, tolls\_amount, and total\_amount for the longest trip by distance (trip\_distance).

```
In [22]: longest_idx = taxi_data['trip_distance'].idxmax()
longest_trip_details = taxi_data.loc[longest_idx, ['fare_amount', 'tip_amount', 'tolls_amount', 'total_amount']]
print(longest_trip_details)
```

```
fare_amount    176.0
tip_amount     18.29
tolls_amount     6.12
total_amount   201.21
Name: 8338, dtype: object
```