How to lose 200.000 data points

(took 4 months to gather)

Mircea Ionescu

What is a data point

- A timestamp of one rotation

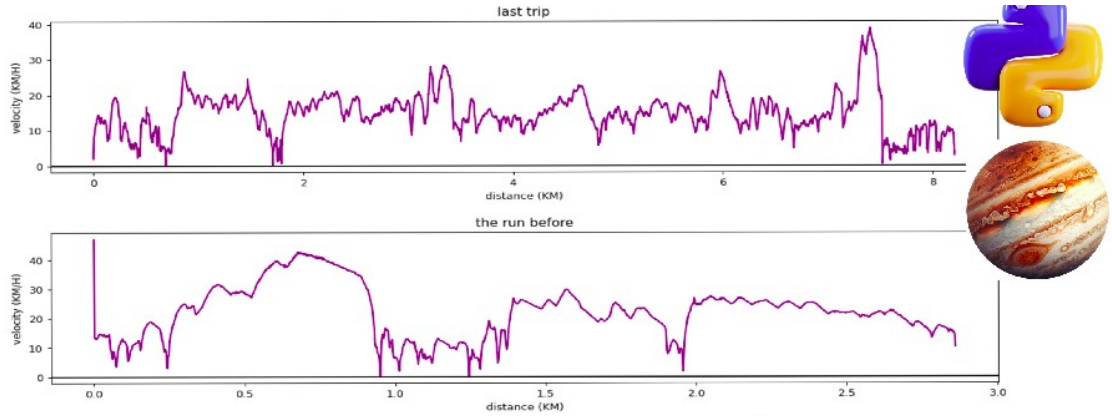
What is a data point

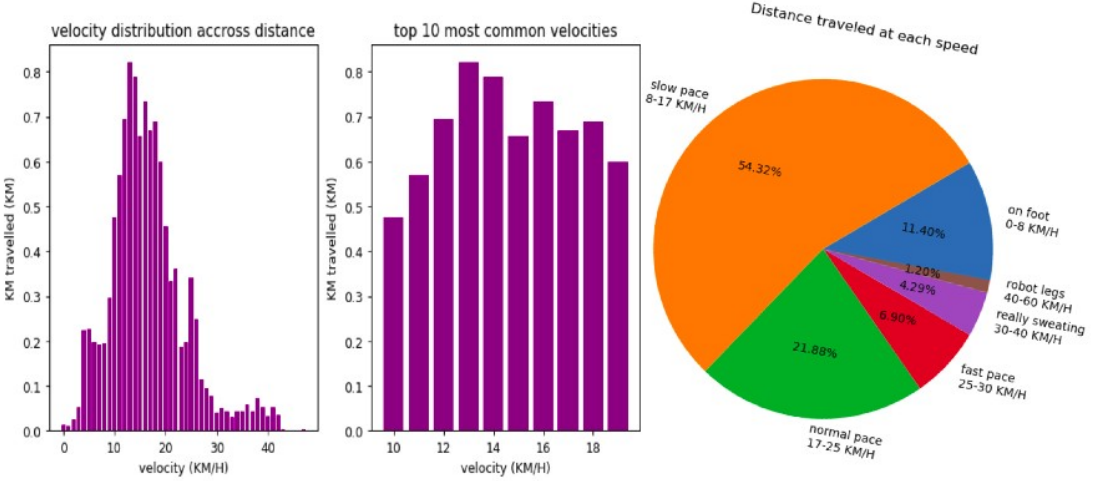
- A timestamp of one rotation
- Equivalent to 2.2 m

What is a data point

- A timestamp of one rotation
- Equivalent to 2.2 m
- Used to compute metrics (speed, distance, acceleration)

What is a data point



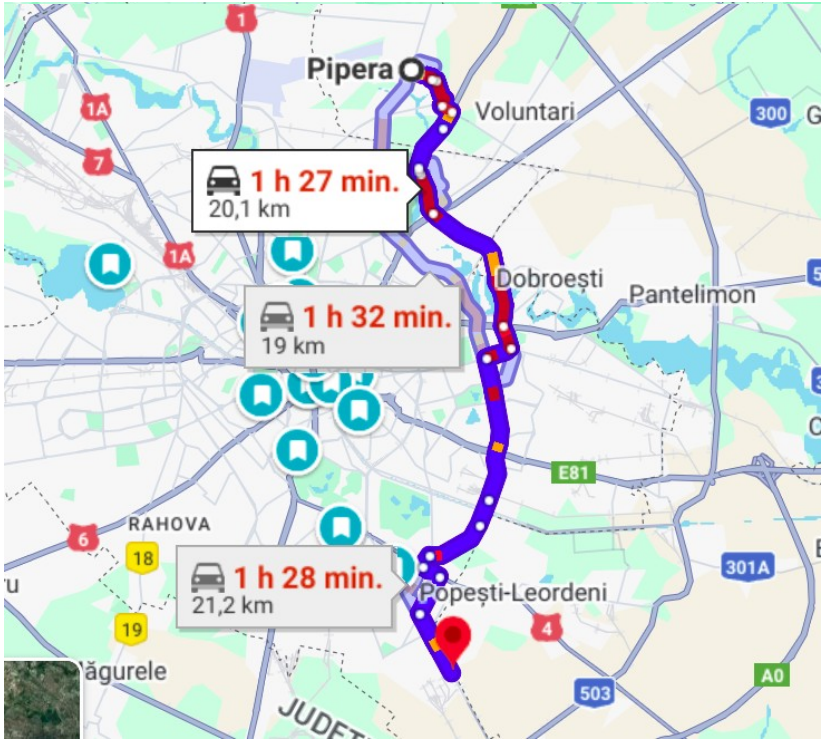


Equivalent to 500 KM of data

- Distance from here to Cluj

Equivalent to 500 KM of data

- Distance from here to Cluj
- Pipera-Berceni 25 times



Equivalent to 500 KM of data

- Distance from here to Cluj
- Pipera-Berceni 25 times
- London-Paris

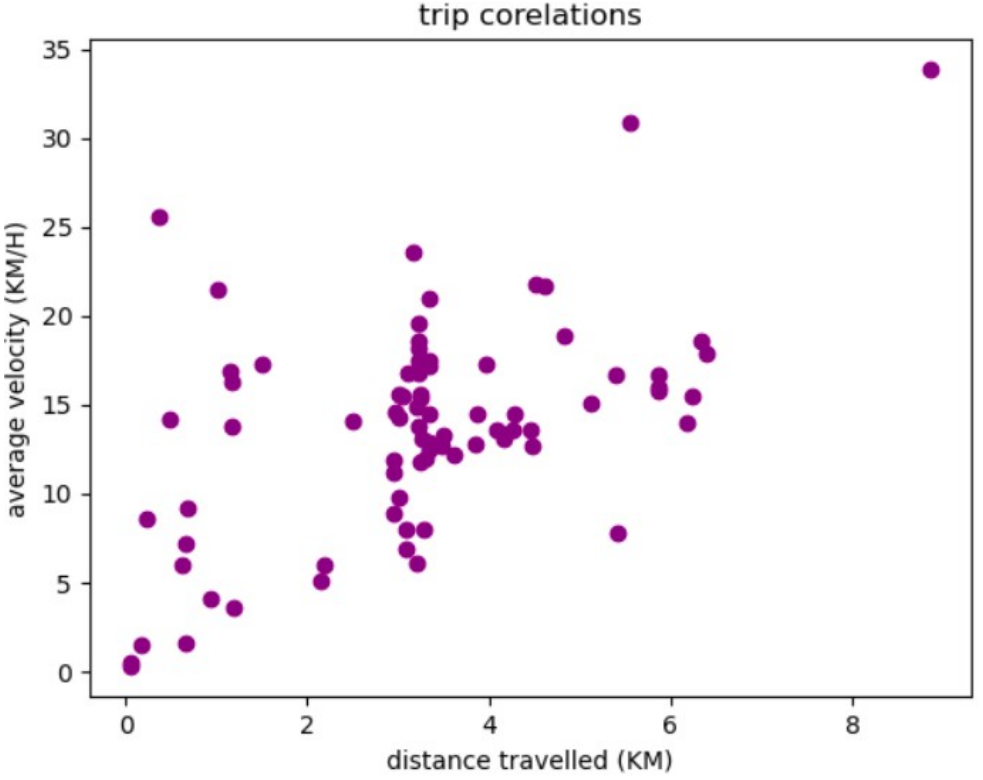
Equivalent to 500 KM of data

- Distance from here to Cluj
- Pipera-Berceni 25 times
- London-Paris

BY BIKE

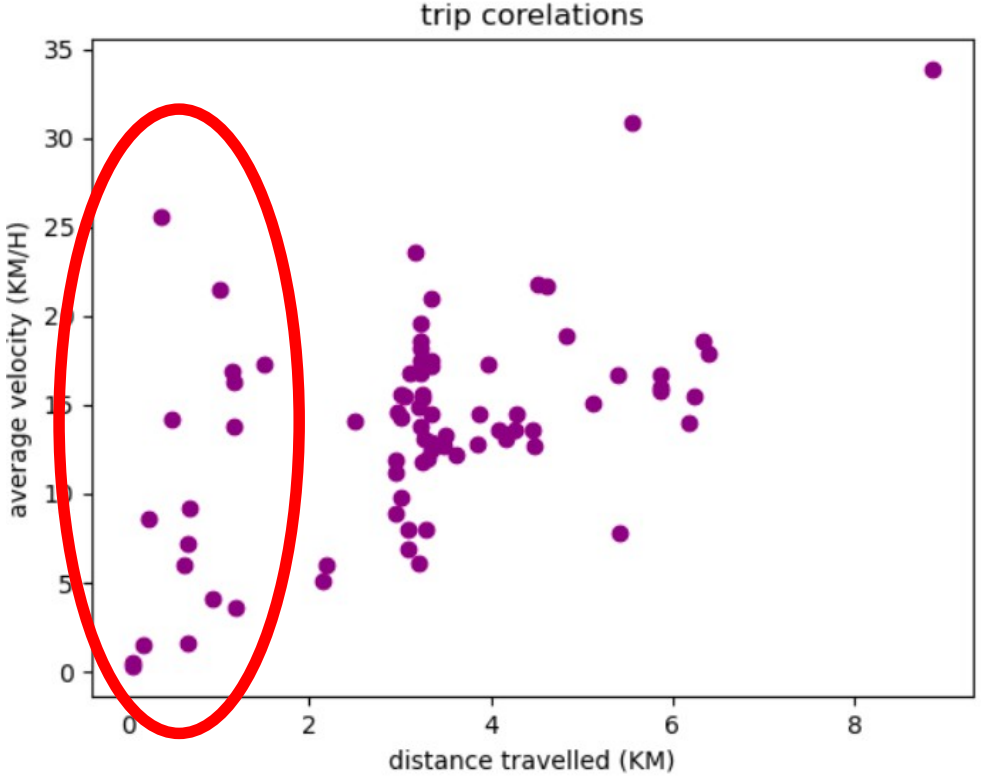
First 250 KM

- Started from an attempt to filter short trips



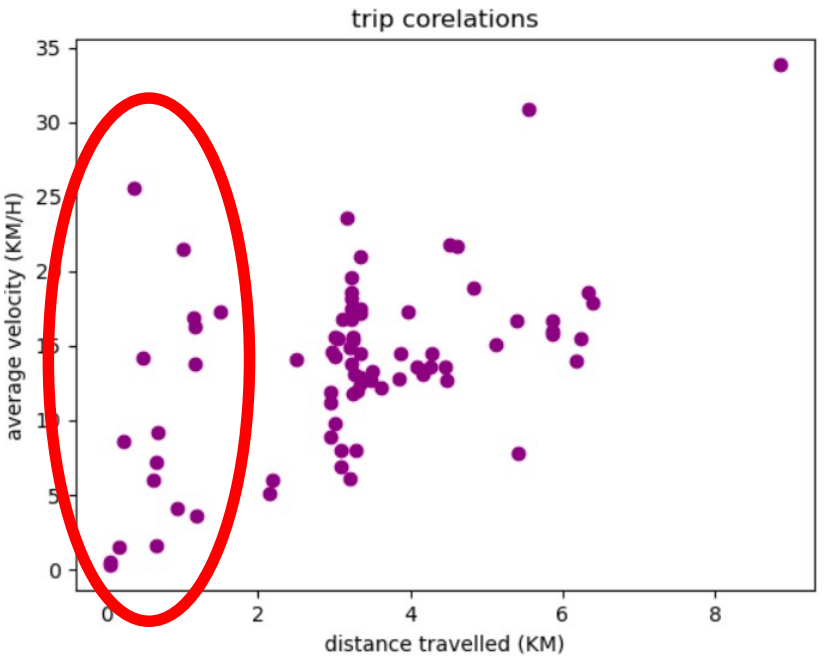
First 250 KM

- Started from an attempt to filter short trips



First 250 KM

- Started from an attempt to filter short trips
- Tried to delete them with os.remove(file)



First 250 KM

- Started from an attempt to filter short trips
- Tried to delete them with

  os.remove(file)

```
def delete_short_files(file_names):
    trip_list = []

    for file_name in file_names:
        trip_csv = pd.read_csv(TRIP_SAVE_LOCATION + file_name)
        file_lines = len(trip_list)

        if file_lines <= MIN_LINES_IN_FILE: # at least 100 lines
            os.remove(file_name)
```

First 250 KM

- Started from an attempt to filter short trips
- Tried to delete them with os.remove(file)

```
def delete_short_files(file_names):
    trip_list = []

    for file_name in file_names:
        trip_csv = pd.read_csv(TRIP_SAVE_LOCATION + file_name)

        trip_list = trip_csv['detection time micros']

        file_lines = len(trip_list)

        if file_lines <= MIN_LINES_IN_FILE: # at least 100 lines
            os.remove(file_name)
```

First 250 KM

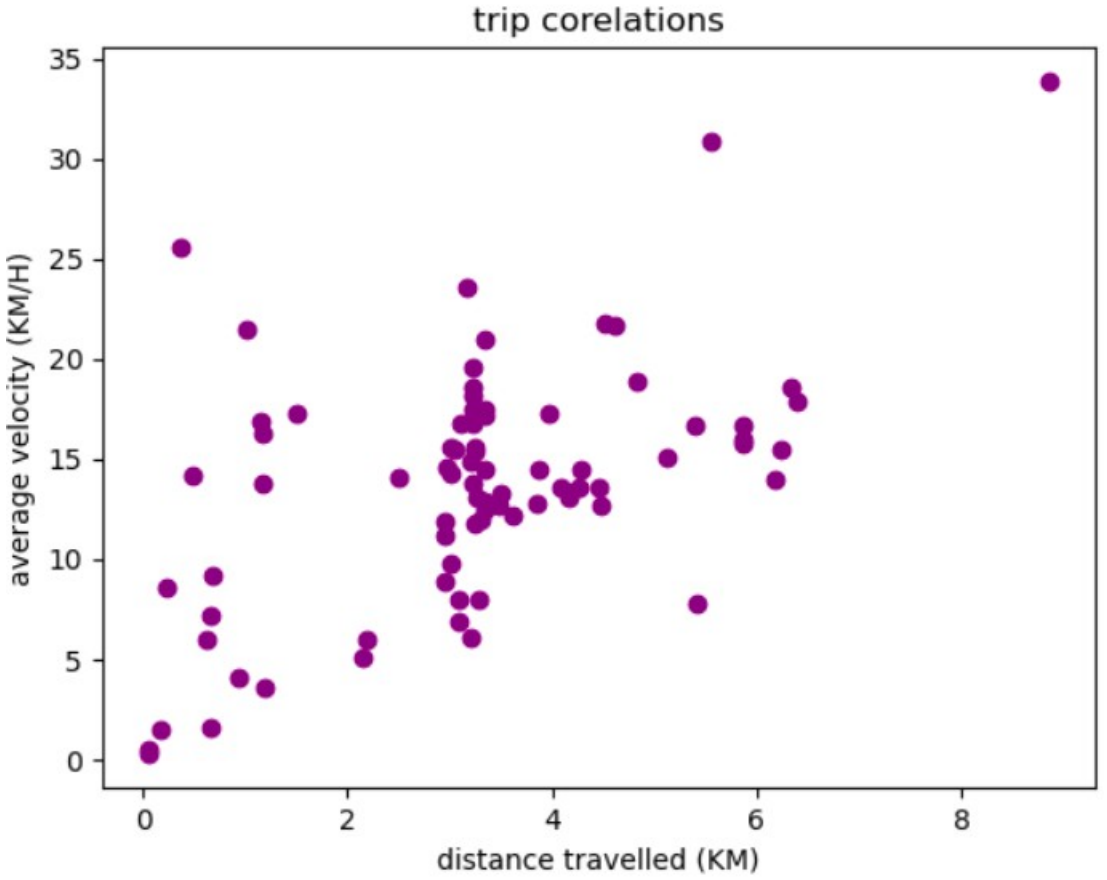
- Started from an attempt to filter short trips
- Tried to delete them with

  os.remove(file)

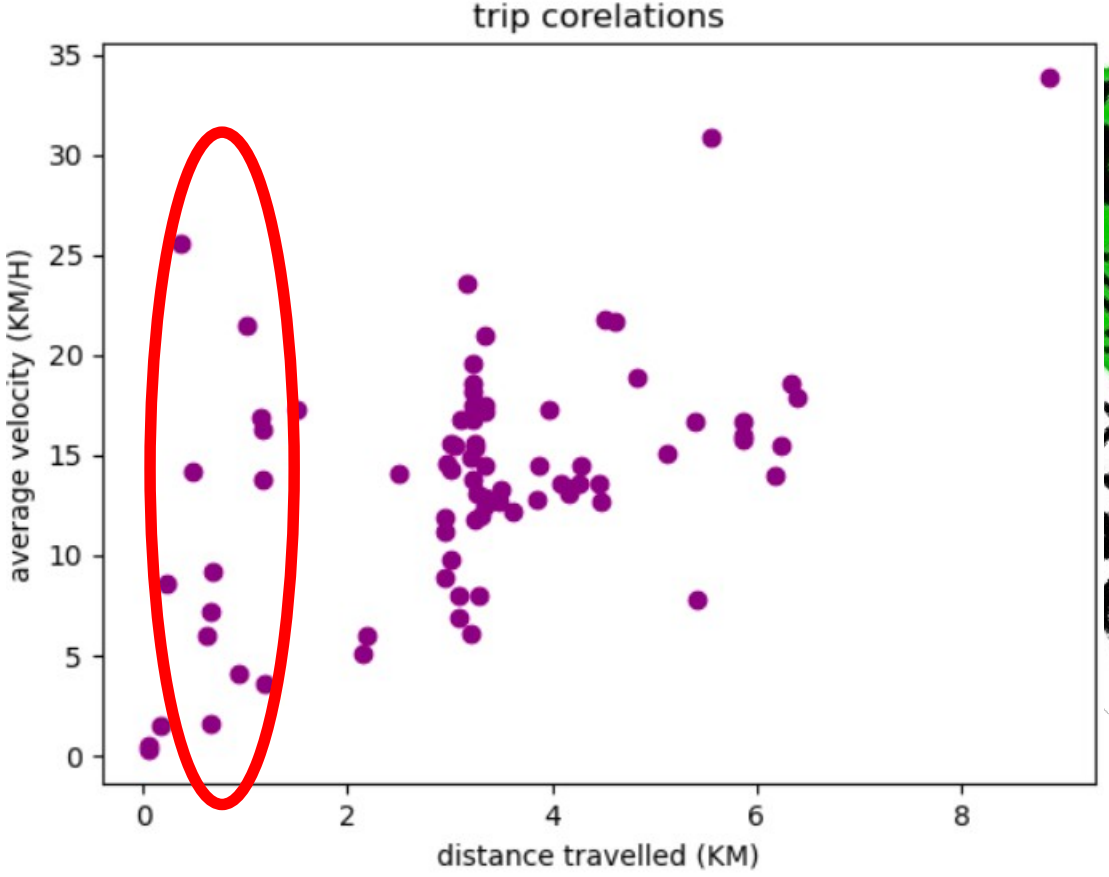
```
def delete_short_files(file_names):
    trip_list = []

    for file_name in file_names:
        trip_csv = pd.read_csv(TRIP_SAVE_LOCATION + file_name)

        trip_list = trip_csv['detection time micros']

        file_lines = len(trip_list)

        if file_lines <= MIN_LINES_IN_FILE: # at least 100 lines
            os.remove(file_name)
```

Another 250 KM later...

Do you remember this plot?

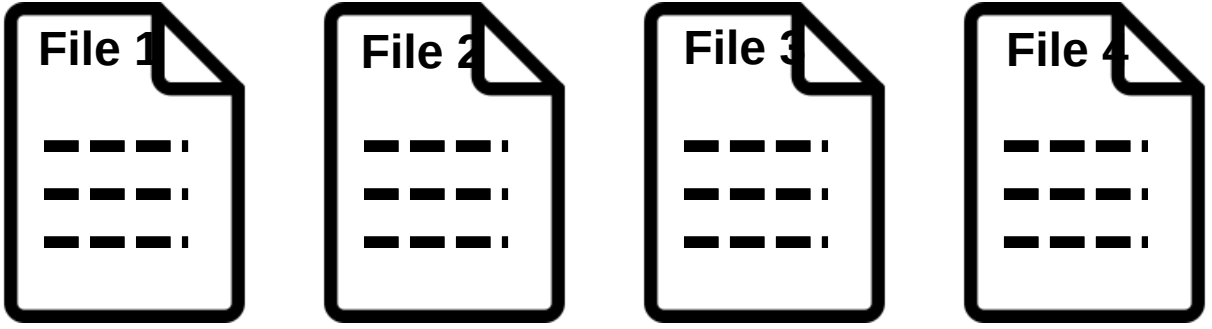


Something’s not right...



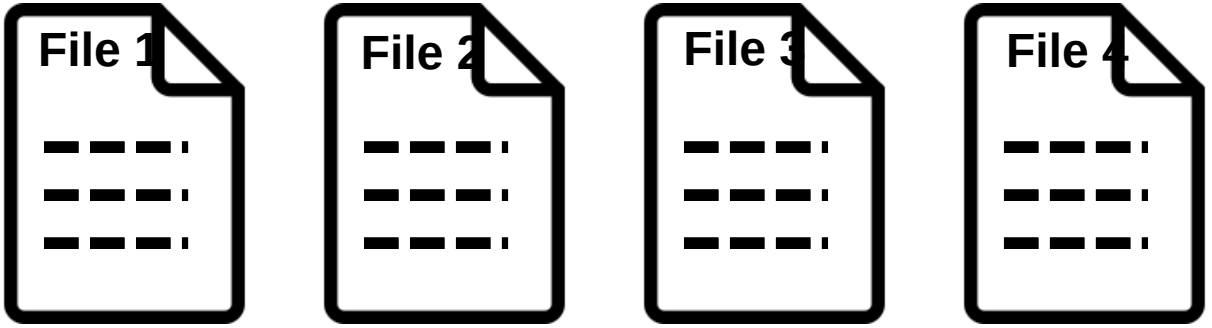
How is my data stored?

- On the mcu the files are equal in size

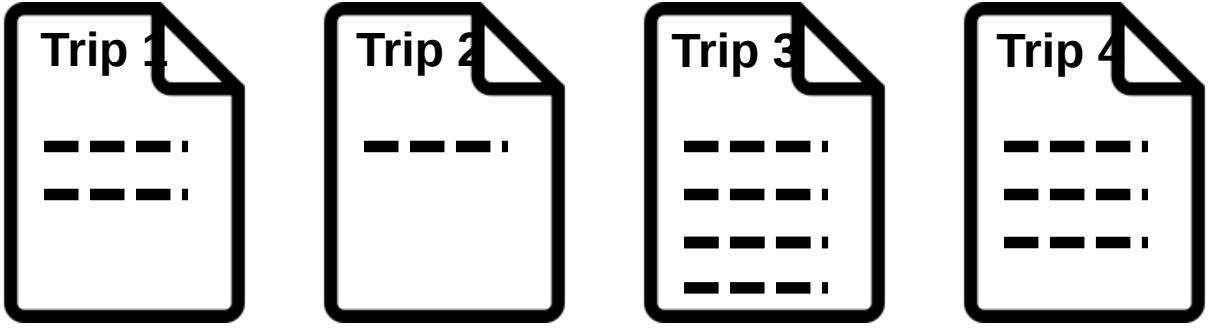


How is my data stored?

- On the mcu the files are equal in size



- On the PC, they are organised in trips



What does that function look like?

- The file naming convention is: 1_file.csv

What does that function look like?

- The file naming convention is: 1_file.csv
- I sort those files by the index in the beginning

What does that function look like?

- The file naming convention is: 1_file.csv
- I sort those files by the index in the beginning
- Look for csv line start to split in trips

What does that function look like?

- The file naming convention is: 1_file.csv
- I sort those files by the index in the beginning
- Look for csv line start to split in trips
- Something like: files.sort(key=sortMethod)

Where are the 250 KM lost?

- In the sorting function…

```
1 def sortMethod(fileName):
2     number = fileName.split('_')[0]
3     return number
```

Where are the 250 KM lost?

- In the sorting function…

```
1 def sortMethod(fileName):
2     number = fileName.split('_')[0]
3     return number
```

- The file naming convention is: “1_file.csv”

Where are the 250 KM lost?

- In the sorting function…

```
def sortMethod(fileName):
    number = fileName.split('_')[0]
    return number
```

```
def sortMethod(fileName):
    number = int(fileName.split('_')[0])
    return number
```

Questions you may have

READ THE TITLE BUCKO

- It’s how to LOSE 200.000 data points
- Not how to protect 200.000 data points