ICT for Smart Mobility
ICT4SM

2024/2025

Luca Vassio – luca.vassio@polito.it
Kai Huang – kai.huang@polito.it

Lab organization

Lab organization

- Students work in groups of 3 people
  - 2 is too small, 4 is too big
- Better having a mixed background
  - As when you get your job
- Bring your own device: laptop, tablet, …
- We give you access to a MongoDB database
- **Python** programming language is strongly suggested
- Useful tools
  - Data extraction: A bit of MongoDB
  - Data postprocessing: **Python** | Java | awk | bash | yourchoice
  - Data plotting: **Matplotlib** | Matlab | Gnuplot | excel | yourchoice
  - Report editing: Word | Openoffice | LaTeX | yourchoice

Laboratory reports

- You have to write reports on laboratory assignments
- 3 reports per group in total
- Each report: 6 pages + appendix + source code
  - Describing what you have done
  - The results you got
  - Source code (Jupyter notebooks and/or Python scripts)
- The group report must be uploaded on Moodle

Report deadlines

- **WHEN**:
  - Deadline Report #1: Tuesday 19/11/2024 - 23:59 CET
  - Deadline Report #2: Tuesday 10/12/2024 - 23:59 CET
  - Deadline Report #3: Tuesday 14/01/2025 - 23:59 CET
- It is possible to submit all reports together
  - Winter session deadline: Tuesday 14/01/2025 - 23:59 CET
  - Summer session deadline: Tuesday 10/06/2025 - 23:59 CET
  - Autumn session deadline: Tuesday 02/09/2025 - 23:59 CET
  - Highly encouraged to submit reports during the course to avoid working on them only at the end of the semester → **2 bonus points** for each report if submitted on time and sufficient

Exam: Lab reports

- After each deadline during the semester each student will receive a report to correct/evaluate from one of the other groups (through Moodle)
  - The student will have a week to submit the corrections
  - The evaluation given to the report will **not** be considered for the evaluated group’s grade
  - According to how the report was corrected/evaluated, each student (individual evaluator) can receive up to **2 additional points**

Exam: Lab reports

- The teachers will evaluate each report (maximum 30/30 + 2 bonus points)
  - If the grade is insufficient (less than 18/30), the group **must update** the insufficient reports
- The reports will be valid for 2 academic years for all students
  - Students that did not pass the oral exam do not have to prepare other reports
  - For 2024/2025 -> up to September 2026

Formation of lab groups

- Choose your group on Moodle
  - Join one group in **Group 1 ~ Group 20**
  - Deadline: Friday 01/11/2024 - 23:59 CET

LAB 1
CAR-SHARING ANALYTICS

SCENARIO

- Car sharing platforms are becoming popular
  - Only in Torino: ~~Car2Go~~, ShareNow, Enjoy, ~~CarCityClub~~, ~~BlueTorino~~, ~~Leasys,~~ e+Share
- They offer a “click-and-rent” paradigm
  - Using an APP, or the web, users can check which cars are available, book one, then start the rental
- Platforms offer web API to interact with the system
  - Which cars are available, status of car, gas, etc.

Some API ~~are~~ used to be public -- https://github.com/sharenowTech/openAPI

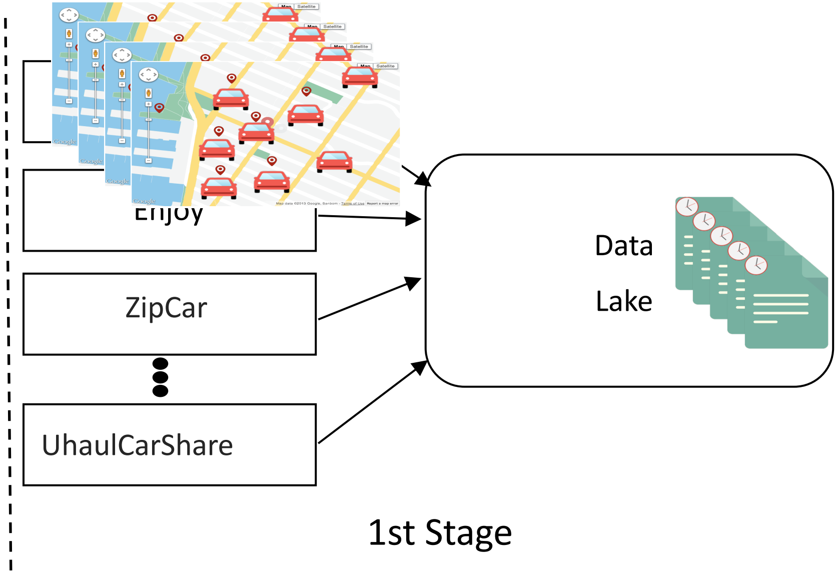
- **Idea: use these platforms as a source of information for mobility studies**

OVERALL PLATFORM

- Big Data approach
- **Data Collection**: Collect data from Open Access Repositories
  - All car position in all Italian cities
  - Integration with maps information, public transportation, traffic status
- **Data management and analysis**: Create a big data platform to harvest the raw data
  - Integrate data, and compute higher level metrics (car density, car flows, etc.)
- **Design a graphical interface to visualize data**
  - Time series
  - Coloured maps
  - Dashboards

IDEA

- **Collect this data and create a platform to monitor car sharing**

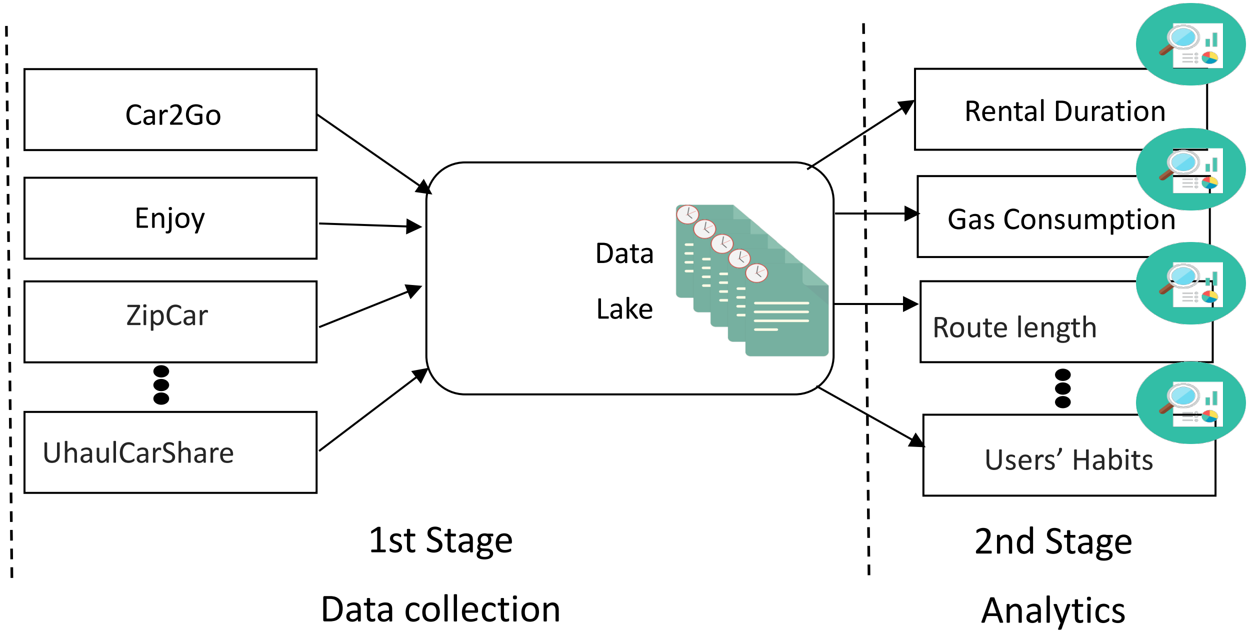


1st Stage

Data collection

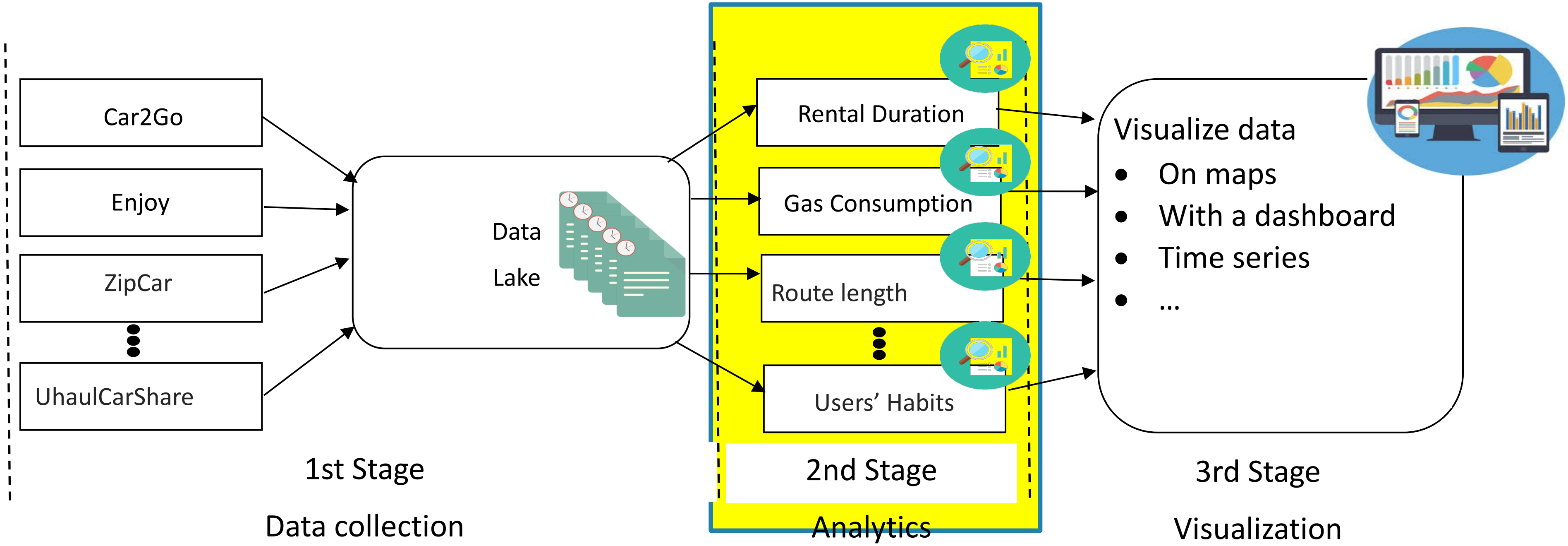
IDEA

- **Collect this data and create a platform to monitor car sharing**



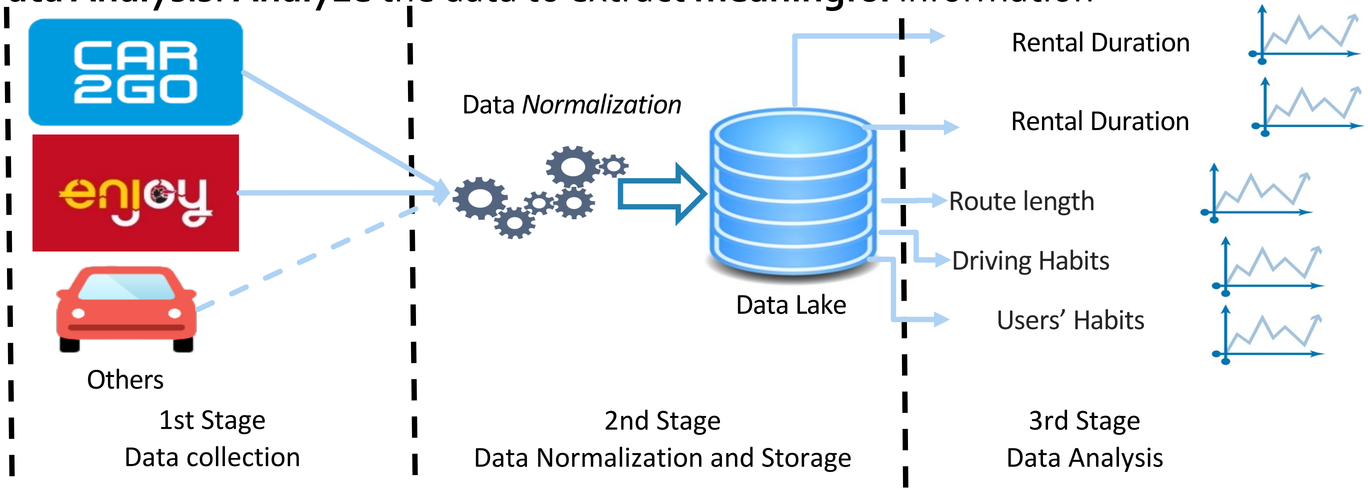
IDEA

- **Collect this data and create a platform to monitor car sharing**



The Big Data approach

- Multiple stage process
  1. **Data Collection**: **Collect** data from **different FFCS** providers
  2. **Data Normalization and Storage**: **Process** the collected data to **normalize** it in a **common format** and save it in a **Data Lake**
  3. **Data Analysis**: **Analyze** the data to extract **meaningful** information



Data Collection

Lab focus – Car2Go

- **30 cities**: 15 in Europe, 15 In North America

Amsterdam
Berlin
Firenze
Frankfurt
Hamburg
Kobenhavn
Madrid
Milano
München
Rheinland
Roma
Stockholm
Stuttgart
Torino
Wien

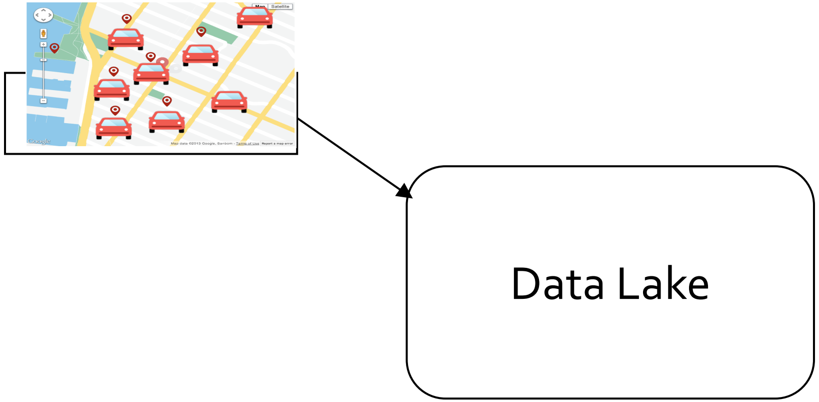
Arlington County
Austin
Calgary
Columbus
Denver
Miami
Montreal
New York City
Portland
San Diego
Seattle
Toronto
Twin Cities
Vancouver
Washington DC

Data collection

- Harvest the data offered by car sharing platforms, and store them in a repository
- Consider the car2go platform
  - It offered public API to collect the data [need a key]

http://www.car2go.com/api/v2.1/vehicles?loc=turin&oauth_consumer_key=getacar&format=json

- It returns a json file



Data Lake

```
{
    "placemarks": [
        {
            "address": "Via Argelati Filippo, 24, 20143 Milano",
            "coordinates": [
                9.1727,
                45.44989,
                0
            ],
            "engineType": "CE",
            "exterior": "GOOD",
            "fuel": 96,
            "interior": "GOOD",
            "name": "581/FF119NT",
            "smartPhoneRequired": true,
            "vin": "WME4533421K145153"
        },
        {
            "address": "Via Santander, 9, 20143 Milano",
            "coordinates": [
                9.16443,
                45.43981,
                0
            ],
```

Data processing– high level intuition

- Every **minute**, for every city, we have information about parked (available) cars.

- Every minute, we get a snapshot of available cars

Data processing– high level intuition

The car is booked

- When a car **’disappears**’ at time T1
  - It has been booked [!=rented]
  - We save the state “car booked at time T1”
    - We have temporary collection for activeBookings
  - [And we compute “**parking period = T1-To**”]

Data processing– high level intuition

- When the car ‘**reappears**’ at T=T2
  - The booking has ended
  - We save the state “car available at time T2”
    - We have temporary collection for activeParkings
  - [And we compute “**booking period= T2-T1**”]

Data processing– high level intuition

- For every city, we have information about parked (available) cars

- Every minute, we get a snapshot of available cars
- We compare this with the latest status for each car
    - We have temporary collections
        - For **activeBookings**
        - For **activeParkings**

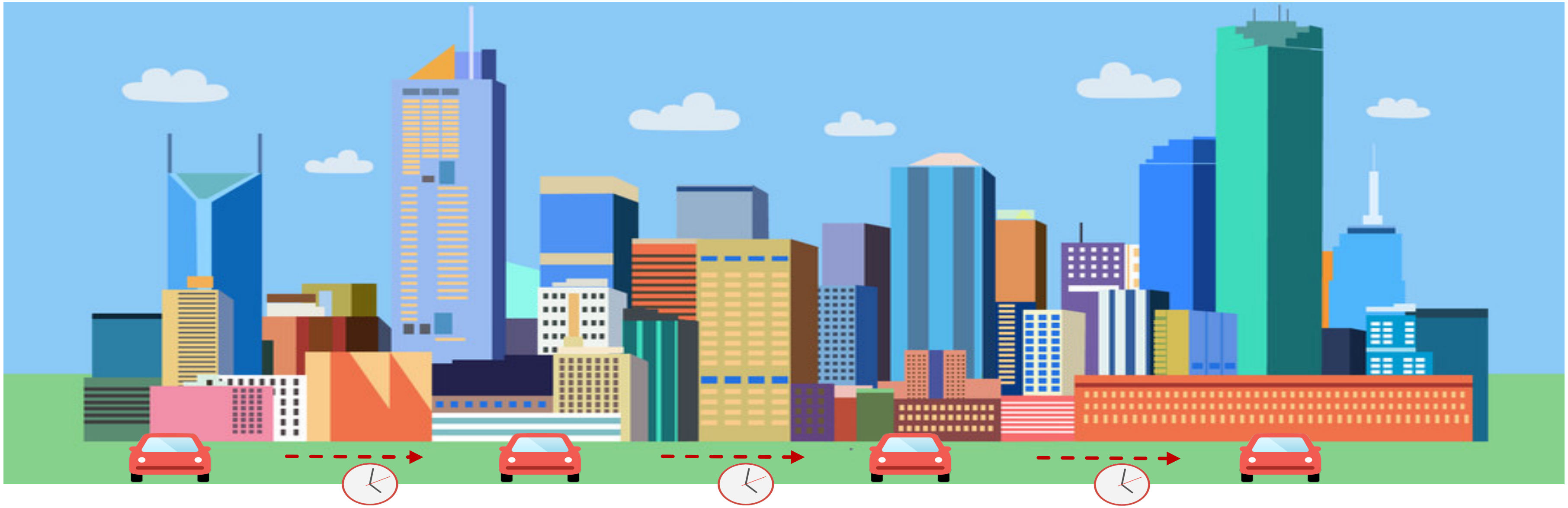
Dataset creation

- **Create a dataset**

The car is back

- By repeating this forever, we obtain a longitudinal dataset

DATASET



Dataset creation

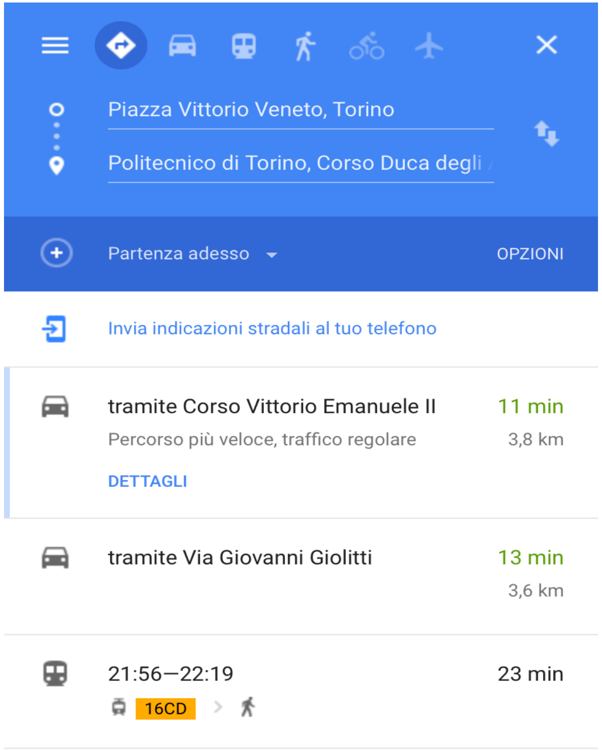
- **Create a dataset**

- By repeating this forever, we obtain a longitudinal dataset
  - A collection for **PermanentBookings**
    - With durations of car booking periods
  - A collection for **PermenentParkings**
    - With duration of car parking periods

Data Lake

Dataset integration

- Integrate the data with other sources
  - Given a rental from [x0,y0] to [x1,y1]
  - Which was the possible path the driver followed?
  - Which should be the minimum time to complete journey?
  - How long would it take to walk instead of driving?
  - How long would it take to use a bus/tram?
- Use google map to integrate the data
  - When a booking ends, query google map for data
    - Limitations due to the number of (free) queries



Piazza Vittorio Veneto, Torino

Politecnico di Torino, Corso Duca degli

Partenza adesso

OPZIONI

Invia indicazioni stradali al tuo telefono

tramite Corso Vittorio Emanuele II — 11 min

Percorso più veloce, traffico regolare — 3,8 km

DETTAGLI

tramite Via Giovanni Giolitti — 13 min

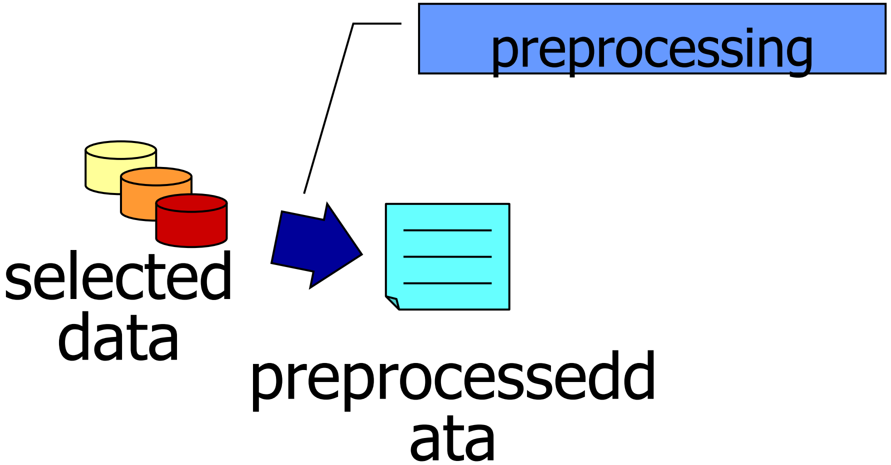
3,6 km

21:56—22:19 — 23 min

16CD

Caveats

- In reality, things are more complicated
- A car may disappear, and reappear in the same place
  - GPS fix position error
  - Booking that has been cancelled
  - System issues
- Car may disappear for (very) long time
  - Car brought to maintenance, stolen, destroyed, …
- A lot of car may disappear at the same time
  - System maintenance
  - System issues
- Google maps data may be missing
- …
- Always double check what you get!



Real world data is "dirty"

Without good quality data, no good quality pattern

Typically 60-80% of time spent here

Repository for dataset

Data collection

- Which technology for our project?
- We have
  - json files
  - Heterogeneous platforms (car2go, enjoy, bluetorino, …)
- NoSQL technology seems more adequate
  - Data can be not structured
  - Data may be missing
  - They are simpler to be used
  - They scale horizontally
- MongoDB is a free and open-source document-oriented DB
  - Well suited for our goals

Data Lake

MongoDB

- It is a scalable NoSQL DB
- Uses JSON-like documents with schemas
- It has drivers for a variety of popular programming languages
  - Python, C, C++, Java, …
- Simple query mechanisms

```
db.getCollection('ActiveBookings').find({})
  db.getCollection('ActiveBookings').find({city: “torino”})

db.getCollection('ActiveBookings').find({city: “torino”}).count()

db.getCollection('ActiveBookings').find({city: ”torino"}).sort({plate: 1})

db.getCollection('ActiveBookings').distinct("city")
```

- We will learn how to use mongoDB for queries

MongoDB - concepts

- Think of **documents** as database **records**
    - Documents are just JSON objects that MongoDB stores in binary
- Think of **collections** as database tables

RDBMS (mysql, postgres)	**MongoDB**
Tables	Collections
Records/rows	Documents/objects
Queries return a record	Queries return a **cursor**

MongoDB - concepts

- Queries return ”cursors” instead of a collections
  - A cursor allows you to iterate through the result set
  - A big reason for this is performance
  - Much more efficient to load results into memory
    - Especially if results are big as in big data

- The find() function returns a cursor object

```
var c = db.ActiveBookings.find( {city: "Torino"} )
var i = 0
while (c.hasNext() && i<10)
{
  var o = c.next() // this is the object
  print(o.init_time + " " + o.city)
  i++
}
```

Analytics

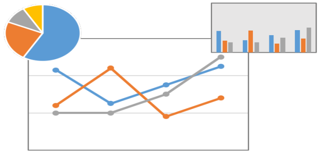
CARSHARING DASHBOARD

- The goal of the labs is to derive higher level information from a dataset
- Develop analytics which, starting from data and external information, show results and statistics in plots
  - Given a dataset
    - Car2Go dataset in different cities
  - Derive typical statistics related to transports
  - Get insight about the usage of free-floating car sharing
  - Estimate demand model

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

Number of booked cars



CarSharing Statistics

Booking duration

(Guessed) time of booking

(Guessed) time of driving

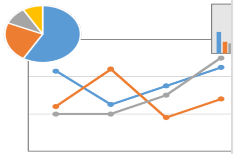
Parking Duration

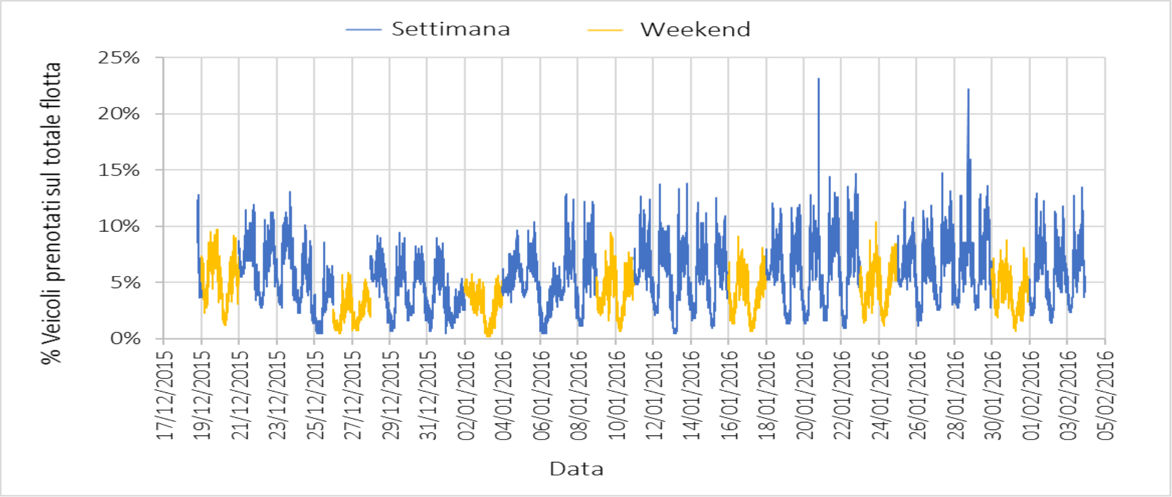
Locations

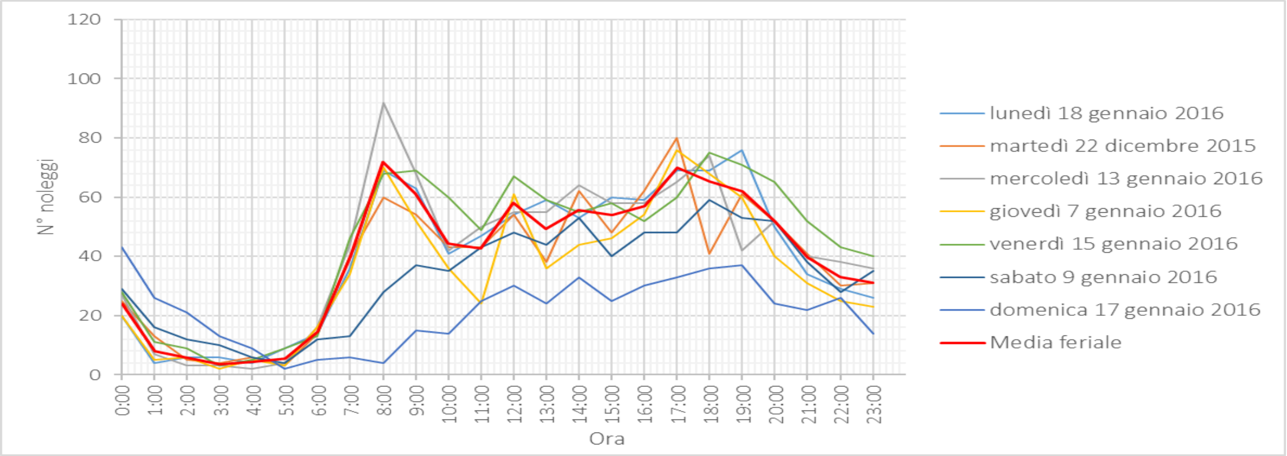
CARSHARING DASHBOARD

- Collect and visualize informatio

Number of booked cars



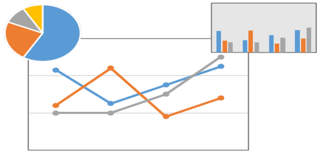


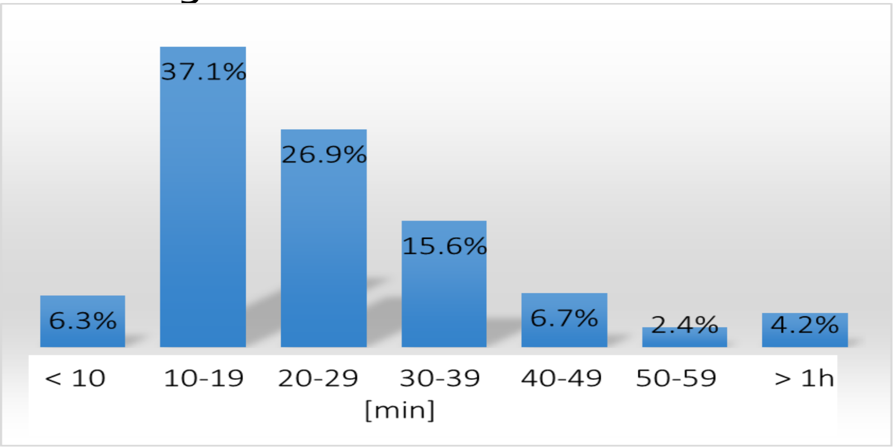


CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

CarSharing Statistics





Booking duration(*)

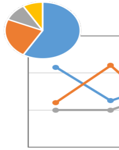
(*) for a proper definition of duration...

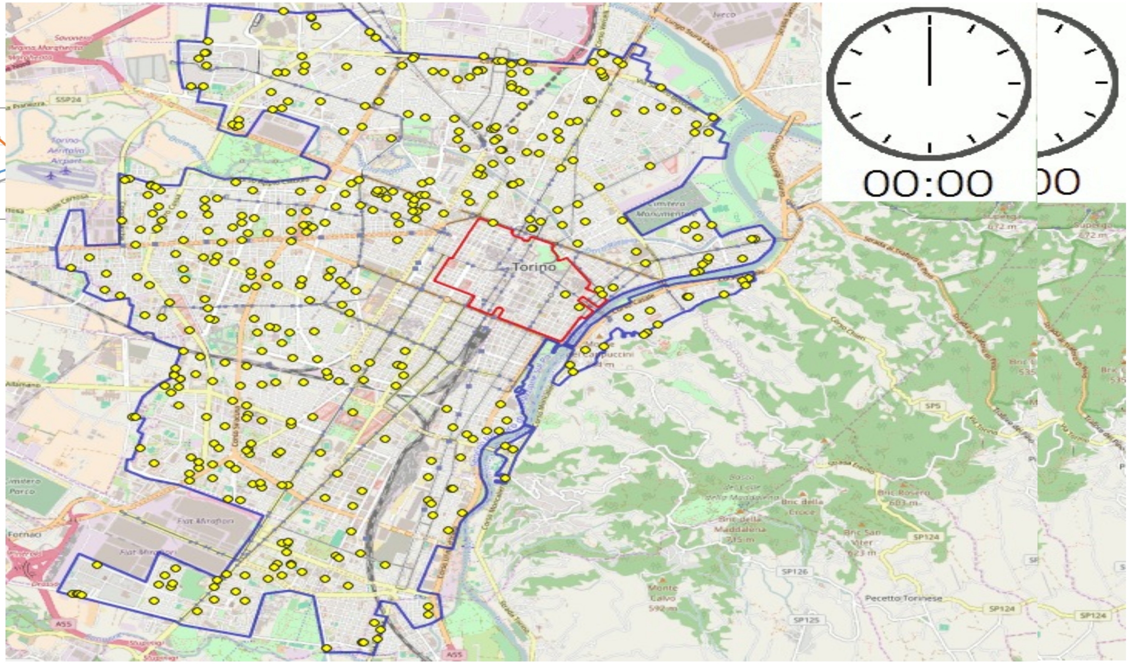
Rental Type(**)	Total	Fraction [%]
Chain	3297	8.8%
Single	31940	84.9%
Single*	1517	4.0%
tour	849	2.3%

(**) for a proper definition of type

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.

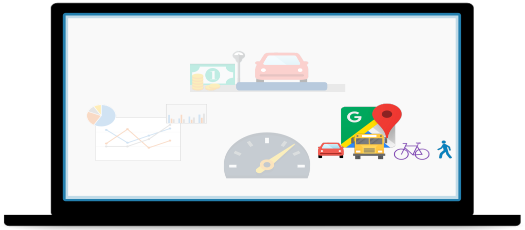




Locations

CARSHARING DASHBOARD

- Collect and visualize information about the car sharing service.



Google APIs

[Google Maps and Google Direction API]

- Develop a system which, starting from data, shows a dashboard offering statistics
  - Correlate rentals
  - with Public Transport availability
  - Walking distance
  - Time of day
  - …

Any questions?

References

- Marco Mellia – Politecnico di Torino