

# Melbourne On-street Parking Dataset

As part of City of Melbourne's smart city initiatives, City of Melbourne is making data from 4300 in-ground sensors in their on-street parking bays available through their Open Data Platform.

Useful links:

- <https://www.melbourne.vic.gov.au/parking-and-transport/parking/parking-locations-fees/Pages/parking-locations-and-fees.aspx>
- <https://www.melbourne.vic.gov.au/about-council/governance-transparency/open-data/Pages/on-street-parking-data.aspx#map>

## Data Source

---

Historical Parking Bay dataset is available on the City of Melbourne Web site.

1. Historical On-street Car Parking Sensor Data - 2017 (File size: 5.31GB):

<https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i>

Please get the CSV file via Export -> CSV

2. On-street Car Park Bay Restrictions (File size: 700K)

[https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Car-Park-Bay-Restrictions/ntht-5rk7/data?\\_ga=2.170293031.649825504.1547467842-374571340.1547467842](https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Car-Park-Bay-Restrictions/ntht-5rk7/data?_ga=2.170293031.649825504.1547467842-374571340.1547467842)

Please get the CSV via Export->Download->CSV

3. On-street Parking Bays (File size: 12.8M)

[https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-Bays/crvt-b4kt/data?\\_ga=2.1497270.649825504.1547467842-374571340.1547467842](https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-Bays/crvt-b4kt/data?_ga=2.1497270.649825504.1547467842-374571340.1547467842)

Please get the file in GeoJSON via Export -> GeoJSON

# Content

---

## 1. Historical On-street Car Parking Sensor Data - 2017

The City of Melbourne has installed in-ground parking bay sensors in most CBD parking bays. These sensors record when a vehicle arrives and when it departs. Each record also includes the parking restriction for the bay and whether the vehicle has overstayed that restriction. This is the data for the 2017 calendar year.

## 2. On-street Car Park Bay Restrictions Data

Each row contains information about the restrictions that apply to one parking bay. Each restriction consists of a set of columns.

### Field information

- FromDay: The first day in the range on which this restriction applies (0=Sunday, 6=Saturday)
- ToDay: The final day in the range, inclusive. (1-5=Monday to Friday)
- StartTime: The time each day when the restriction applies.
- EndTime: The time each day when the restriction ends.
- TypeDesc: The type of restriction. Eg: "2P Meter" (two hour parking, paid for using a meter), "Disabled Only" (only vehicles with disabled permits can park).
- Description: A compact, human-readable description of the overall restrictions. TKT A stands for Ticket Area covers multiple bays within that area. DIS ONLY and DIS are both disabled parking spaces
- Duration: The time that a vehicle can park in the spot (assuming they can legally park there).
- DisabilityExt: For bays that aren't limited to disabled permits, how much time a vehicle with disabled permit can spend in the spot. Usually twice the regular amount of time.
- EffectiveOnPH: Does this restriction apply on public holidays.
- Exemption: Which vehicles are exempt.

There can be multiple restrictions, occurring at non-overlapping times. Each restriction is a set of columns with a number: FromDay1, ToDay1, StartTime1, EndTime1, etc.

## 3. On-street Parking Bays Data

This data contains spatial polygons which represent parking bays across the city. Each bay can also link to its parking meter, and parking sensor information.

# Challenge

---

1. Analyse the historical sensor data and report interesting findings such as:
  - "TopN most popular parking bays",
  - "TopN most violated parking bays", and
  - possible reasons (near shopping centres? KFC?).
2. Building Machine Learning classifier to predict "In Violation" info (True/False).
3. Create Simple Melbourne Parking Bay simulators "based on the Historical dataset".
  - Simple version: For each Parking Bay, compute the (hypothetical) current Parking Bay status (occupied or not) given the current time (such as 15/01/2019 11:30:14.000).

The following Python script computes a parking bay status "randomly". Can we enhance this script base on the historical dataset?

```
import random

num = random.randint(0,10)

if num < 3:
    print "Bay is available."
else:
    print "Bay is occupied."
```

- State-full version: For each Parking Bay, compute a hypothetical "last X-hours" history of Parking Bay status (a sequence of occupied at T1, empty at T2, occupied at T3, empty at T4, ...) given the current time.
4. Compare your simulated results and the historical data.

---- END ----