COLLECTING, PROCESSING AND ANALYSING LARGE GPS AND SMARTPHONE DATASETS

Dr Adrian B. Ellison and Dr Richard B. Ellison
Institute of Transport and Logistics Studies
The University of Sydney

THE UNIVERSITY OF SYDNEY

GIS for Transport Applications, Leeds, UK
14th April 2015

THE UNIVERSITY OF SYDNEY

AGENDA

- Background
- Data collection
- Software, databases and SQL
- Introductory tutorial

BACKGROUND

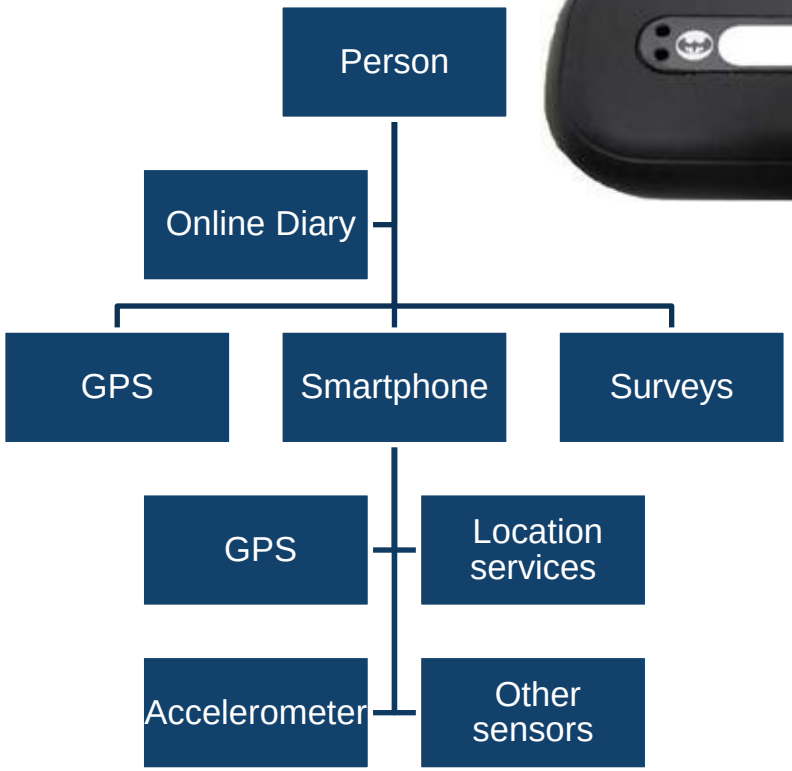
- Advances in technology have improved our ability to collect transport data
- Transport researchers are increasingly collecting datasets containing millions of observations from sources such as GPS, smartphones, smartcards and bluetooth sensors
- These provide the opportunity to improve our understanding of travel behaviour in a way that acknowledges the variability inherent within human behaviour
- Many of the common methodologies were not designed for datasets of this size
  - Managing and analysing these datasets requires a different ‘toolkit’ than are used for traditional surveys

TRAVEL SURVEYS - LIMITATIONS

- Reliant on participant recall
  - Particularly problematic for short and incidental trips
- Expensive, time consuming and labour intensive during and after data collection
  - Significant marginal cost for each additional participant
- Prone to human error by the participant and/or interviewer
- Limited ability to record information on routes
  - Where are people walking?
  - Why are they walking there and not elsewhere?
  - Do they use bicycle paths? If so, where?
  - Do people travel on local roads or main roads and how does this vary across purpose and time?
- Limited ability to observe the same people across time (and space)
  - Do they walk the same way every day?
  - Do they sometimes ride a bicycle instead?
  - Do they sometimes take the bus and the train at other times?

INTEGRATED DATA COLLECTION

- Technology allows for more data to be collected from each person
- Automated processing results in (potential) lower marginal cost
- Lower burden on participants
- Better quality of self-reported data



DATA COLLECTION

- GPS tracking devices
- Smartphone tracking using:
  - Wifi
  - Network position
  - Accelerometer
  - Gyroscope
  - GPS
  - Compass
  - Light sensor (photodiode)

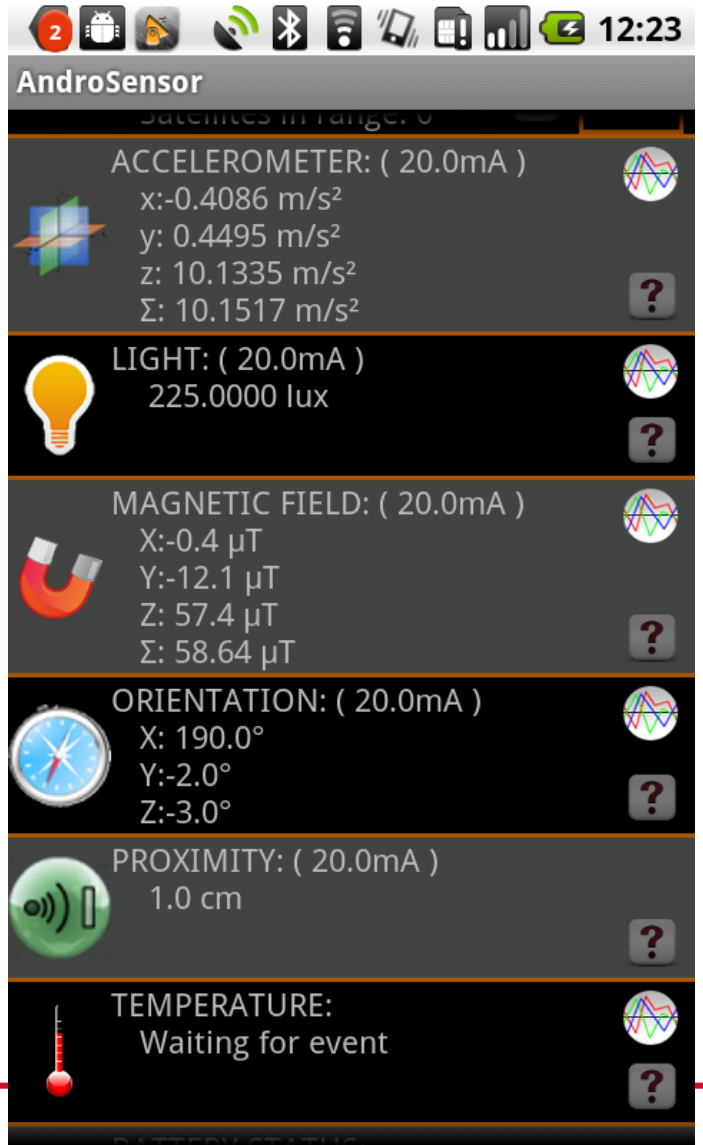


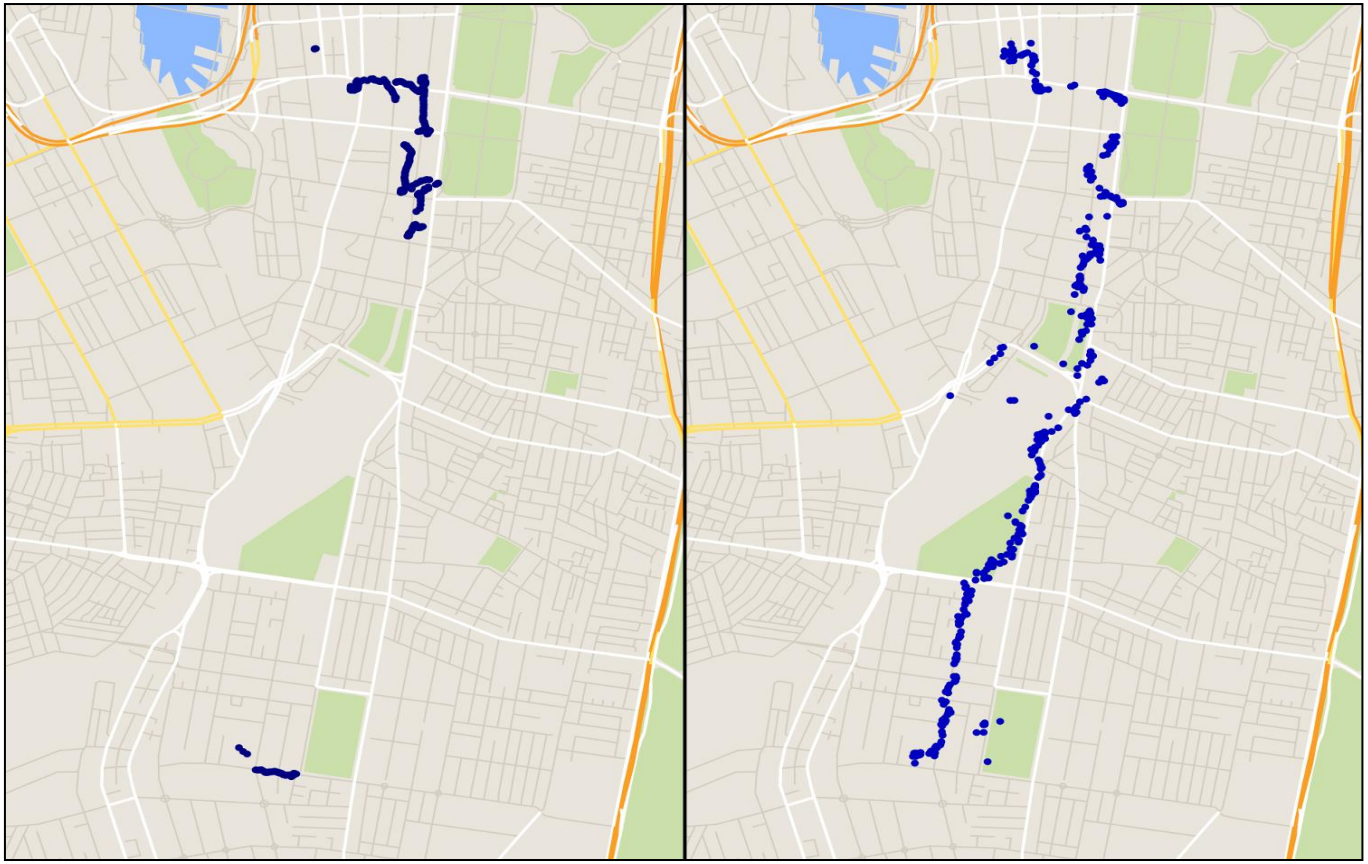
Image: Ellison (2012)

LIMITATIONS AND ISSUES

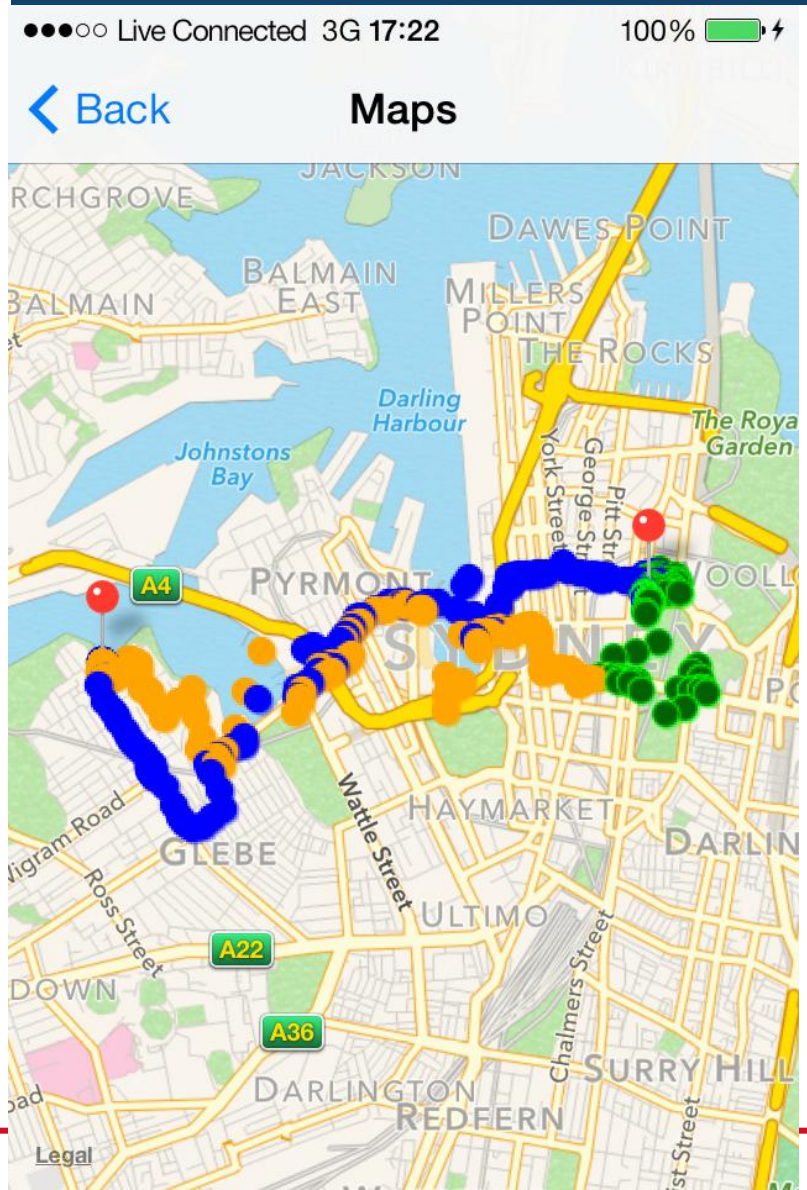
- Participants require internet access
  - Is an advantage for including younger participants
  - Older participants sometimes prefer older methods
  - Privacy concerns for some people
- Produces very large databases
  - Potentially millions of records/observations per day
  - Requires relational database skills to manage
  - Analysis more complex than for traditional sources (although there are solutions)
- GPS problematic in dense urban areas
  - Urban canyon
  - Signal acquisition
  - GPS also battery intensive

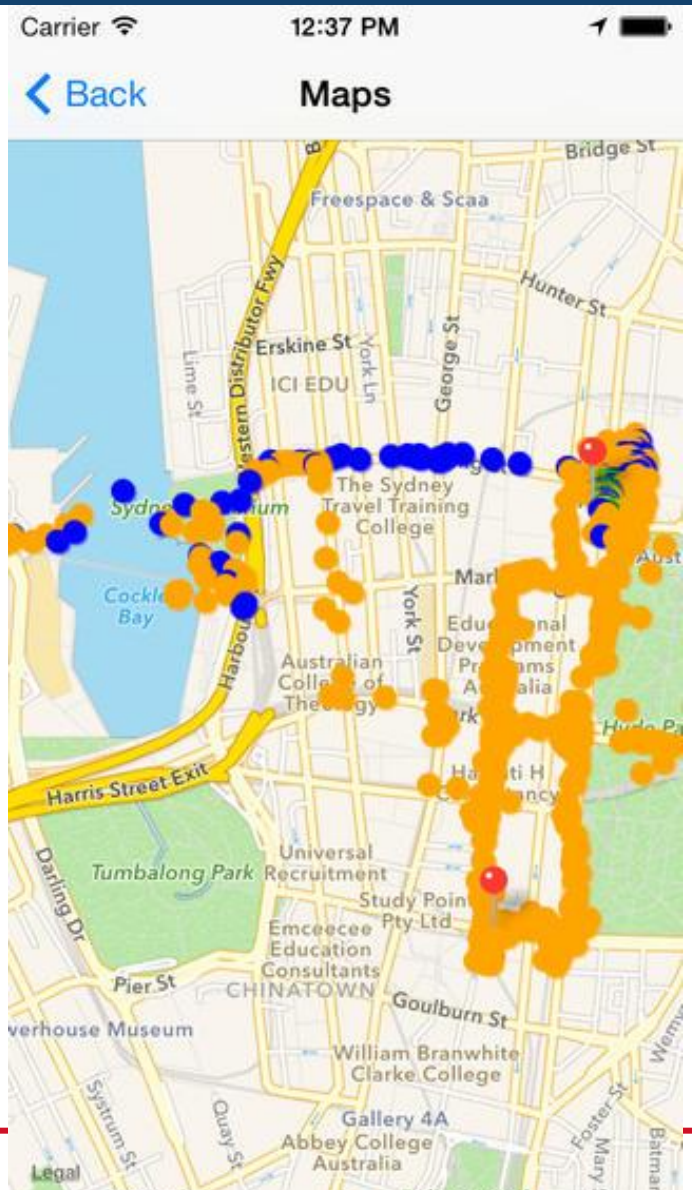
WI-FI AND NETWORK LOCATION DATA

GPS **Smartphone**



SMARTPHONE APP





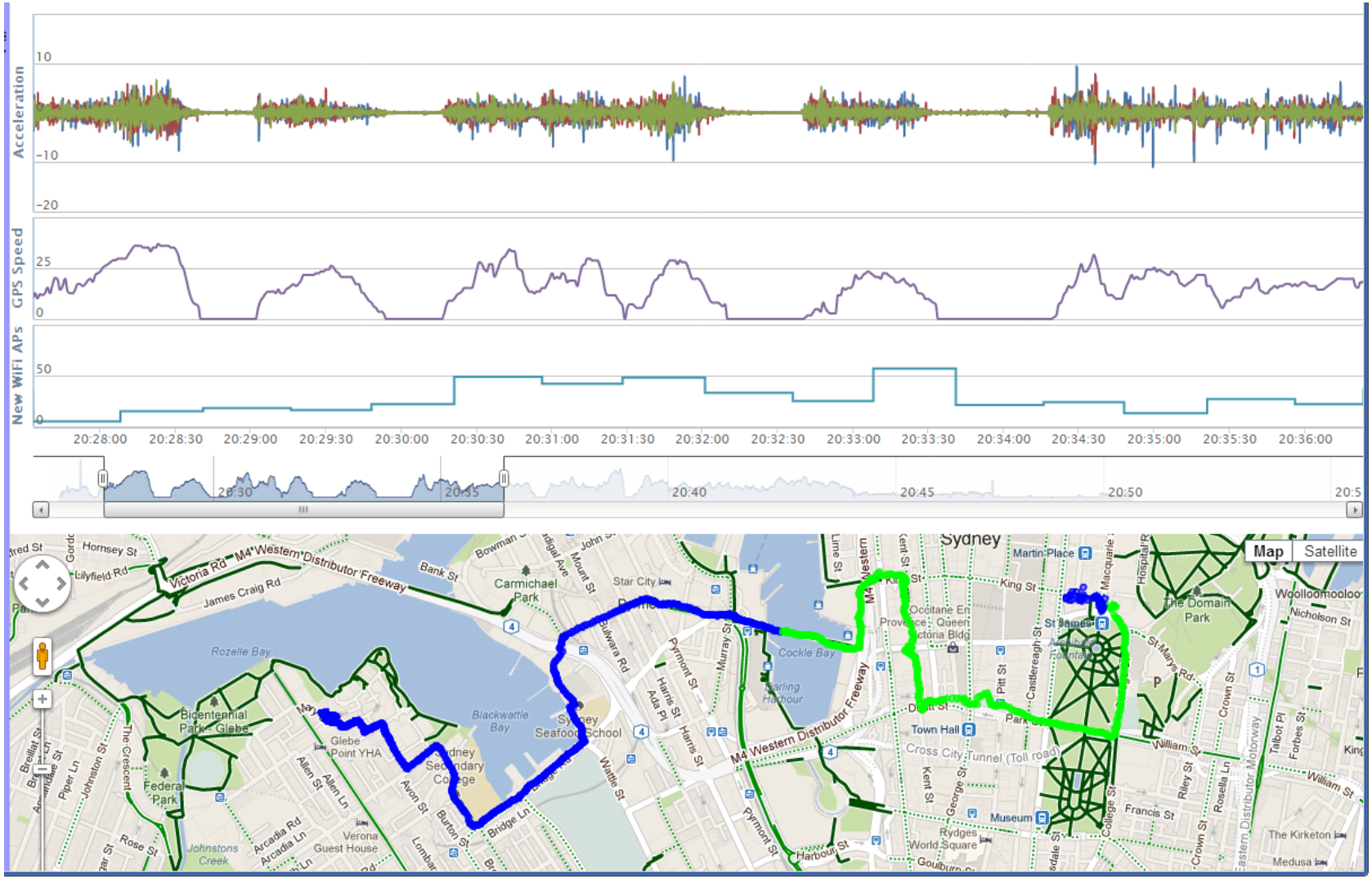
SMARTPHONE DATA AND TRAVEL DIARY

TripID	Date	Activity	Origin	Suburb	Destination	Suburb	Intermediate Stop	Legs	Completed
8596	**Friday 8th November** 10 15-10 35	Commuted to Work	Home (Home)	Glebe	ITLS (Work)	Sydney (CBD)	Shopping/Personal Business	1	☑
9033	**Friday 8th November** 20 00-21 00	Returned Home	ITLS (Work)	Sydney (CBD)	Home (Home)	Glebe	Shopping/Personal Business	1	☑

Leg	Mode	Duration	Origin	Suburb	Destination	Suburb	Bus Route
1	Bicycle	20 mins	ITLS	Sydney (CBD)	Home	Glebe	

› View multiple data sources on one screen

ACCELEROMETER AND GPS

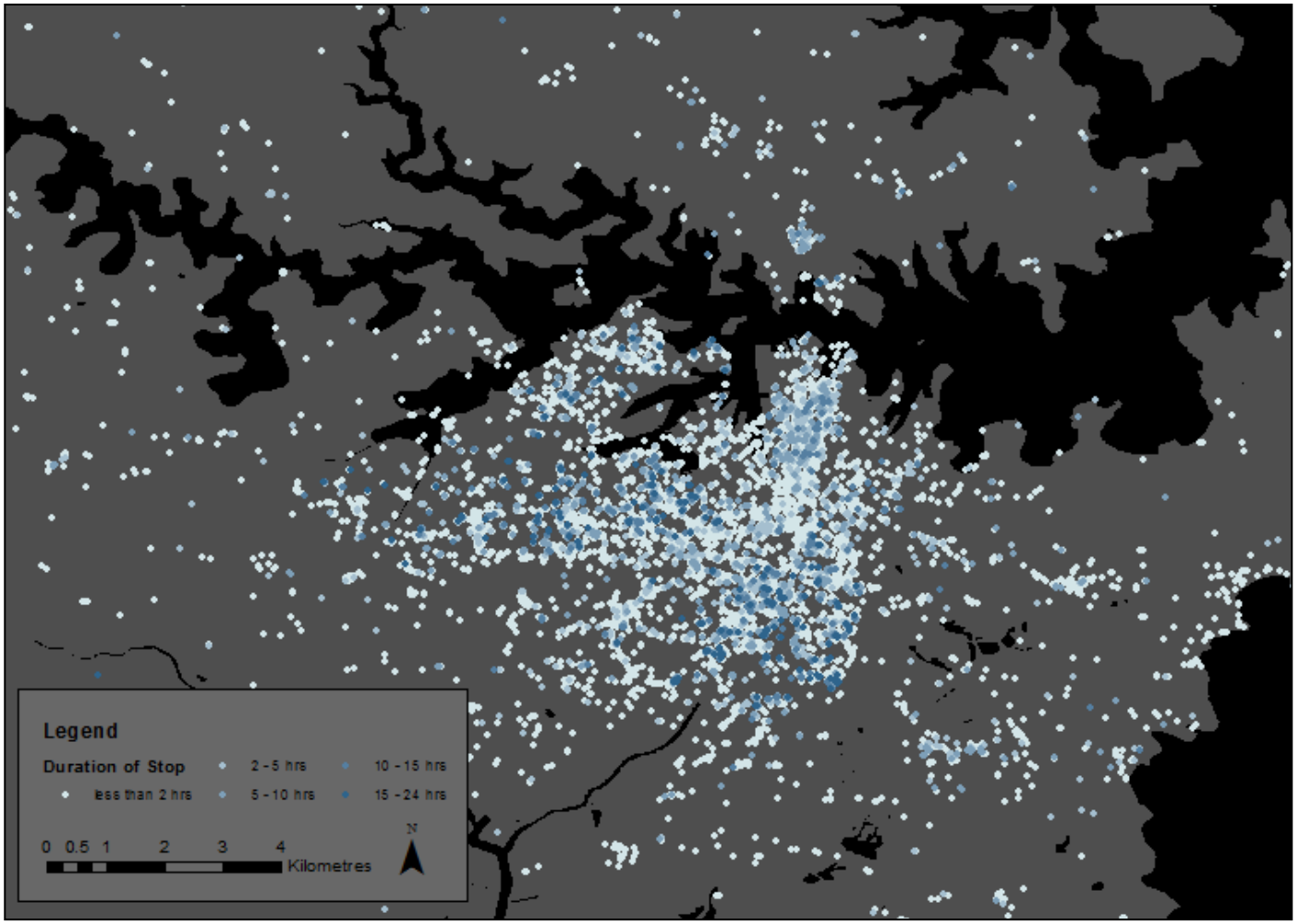


ROUTE CHOICE

› Route detection analysis from smartphone data

ANALYSIS OF STOPS

› Stop detection analysis from smartphone data



SYDNEY DRIVING STUDY

Import GPS data into database

- Create line “layers” using Google Maps encoding algorithm from GPS observations.
- Different colours for not speeding/speeding!

Display trips on map

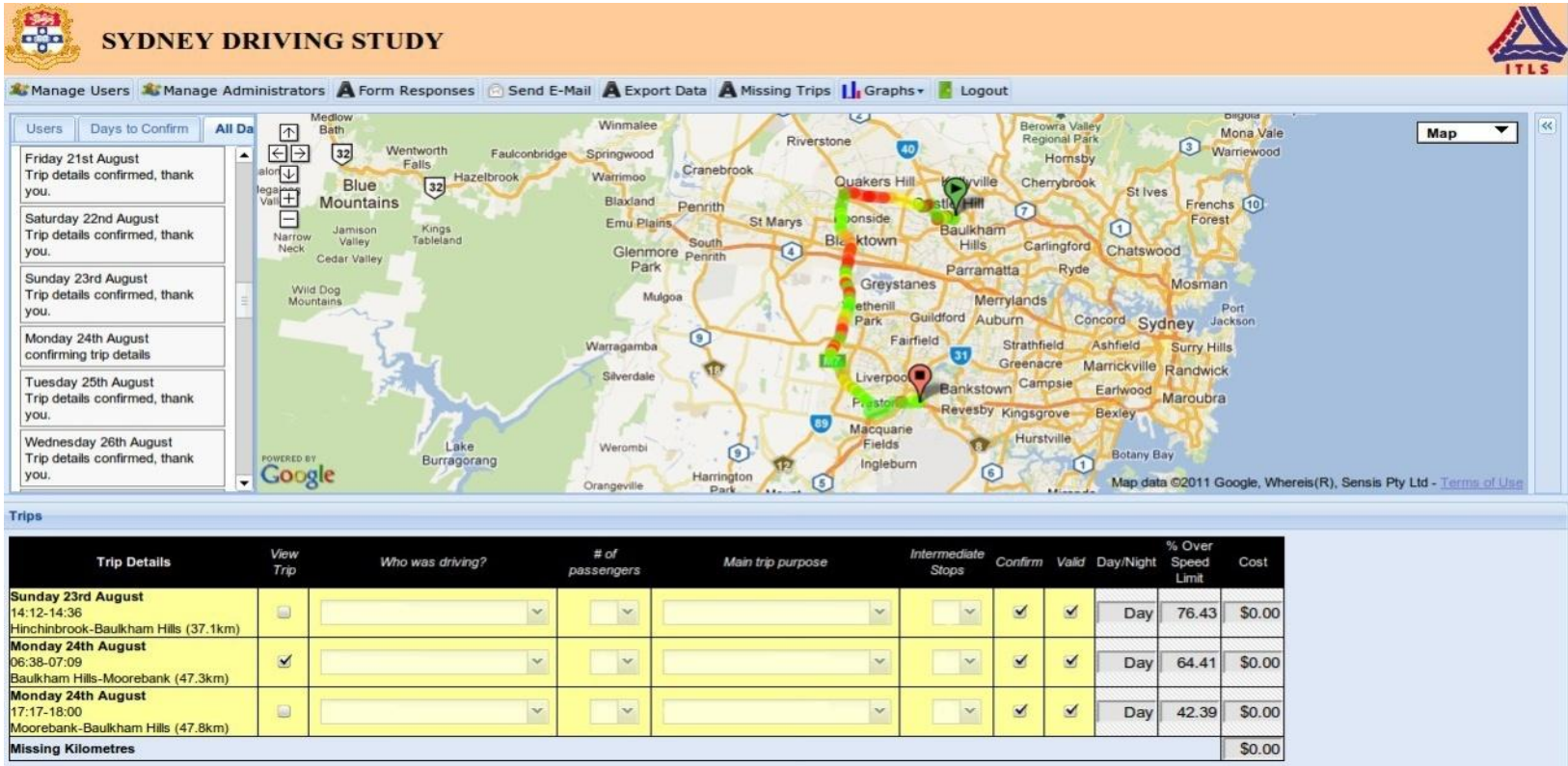
- Retrieve encoded lines from database.
- Use Google Maps API to plot GPS traces on map.

Add supplementary information to maps

- Google Maps API ‘tooltips’ – more information about trip shown when mouse hovers over trip.
- Clicking on origin or destination shows streetview if available.

SYDNEY DRIVING STUDY

- GPS Driving Study Interface:
  - Developed using open source software (including Linux/Apache webserver, MySQL database, PHP programming language and Google Maps API).



Trip Details	View Trip	Who was driving?	# of passengers	Main trip purpose	Intermediate Stops	Confirm	Valid	Day/Night	% Over Speed Limit	Cost
Sunday 23rd August 14:12-14:36 Hinchinbrook-Baulkham Hills (37.1km)	☐					☑	☑	Day	76.43	$0.00
Monday 24th August 06:38-07:09 Baulkham Hills-Moorebank (47.3km)	☑					☑	☑	Day	64.41	$0.00
Monday 24th August 17:17-18:00 Moorebank-Baulkham Hills (47.8km)	☐					☑	☑	Day	42.39	$0.00
Missing Kilometres										$0.00

SYDNEY DRIVING STUDY

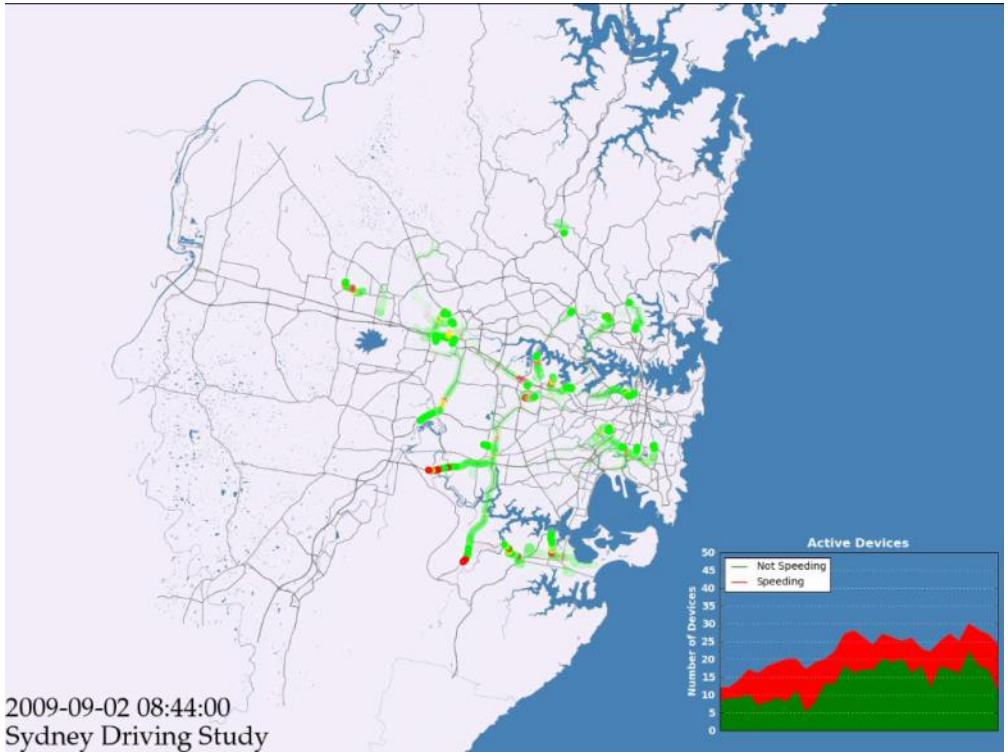
› Uses a combination of PostgreSQL/PostGIS database functions and geospatial data abstraction library.

Import GPS data into database
- Set correct projection and datum for coordinate fields.

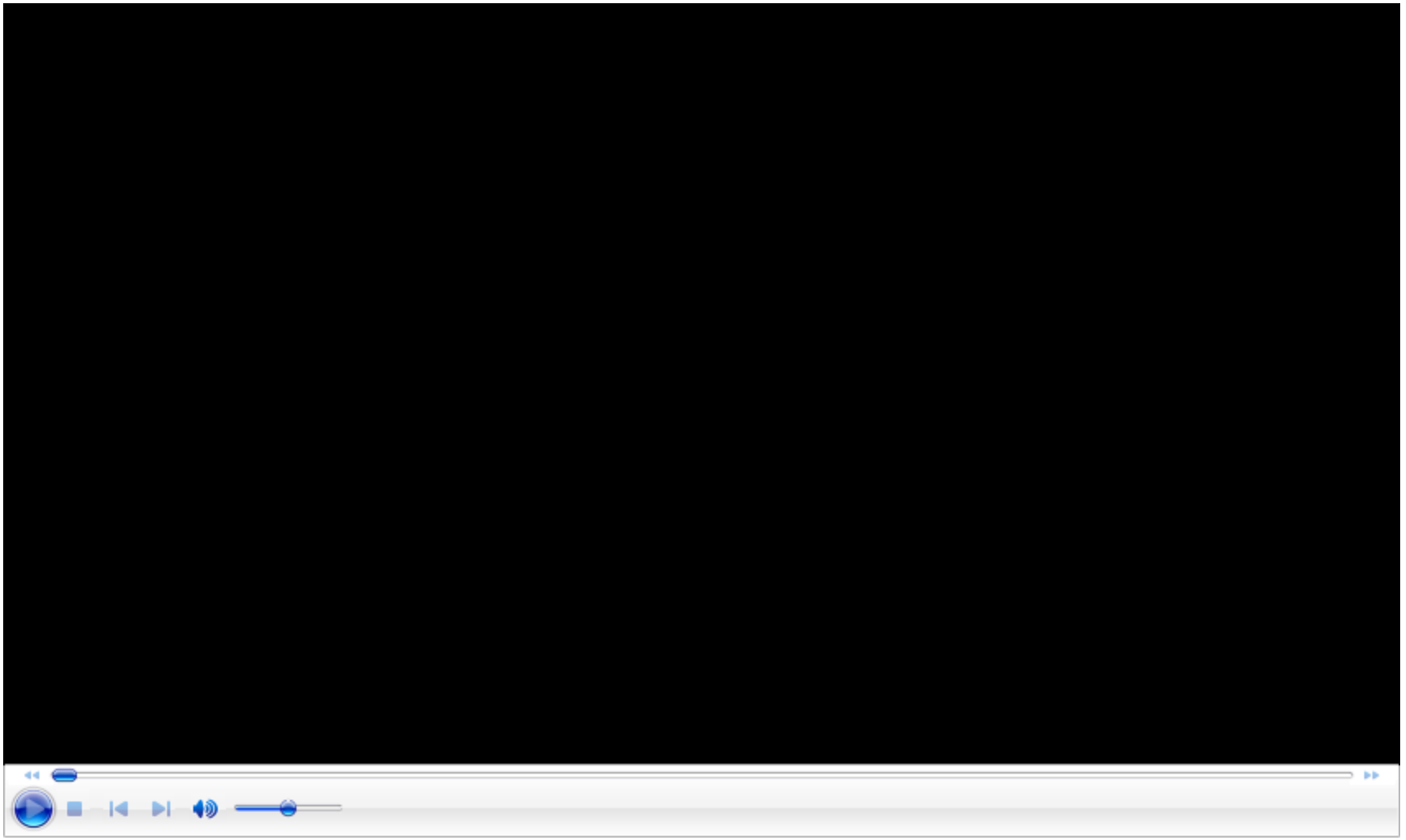
Plot base map layers onto a map using GDAL
- Ocean, main roads, waterways

Plot GPS points showing speeding from green (no speeding) to red (5km/h+).
- Generate speeding graph

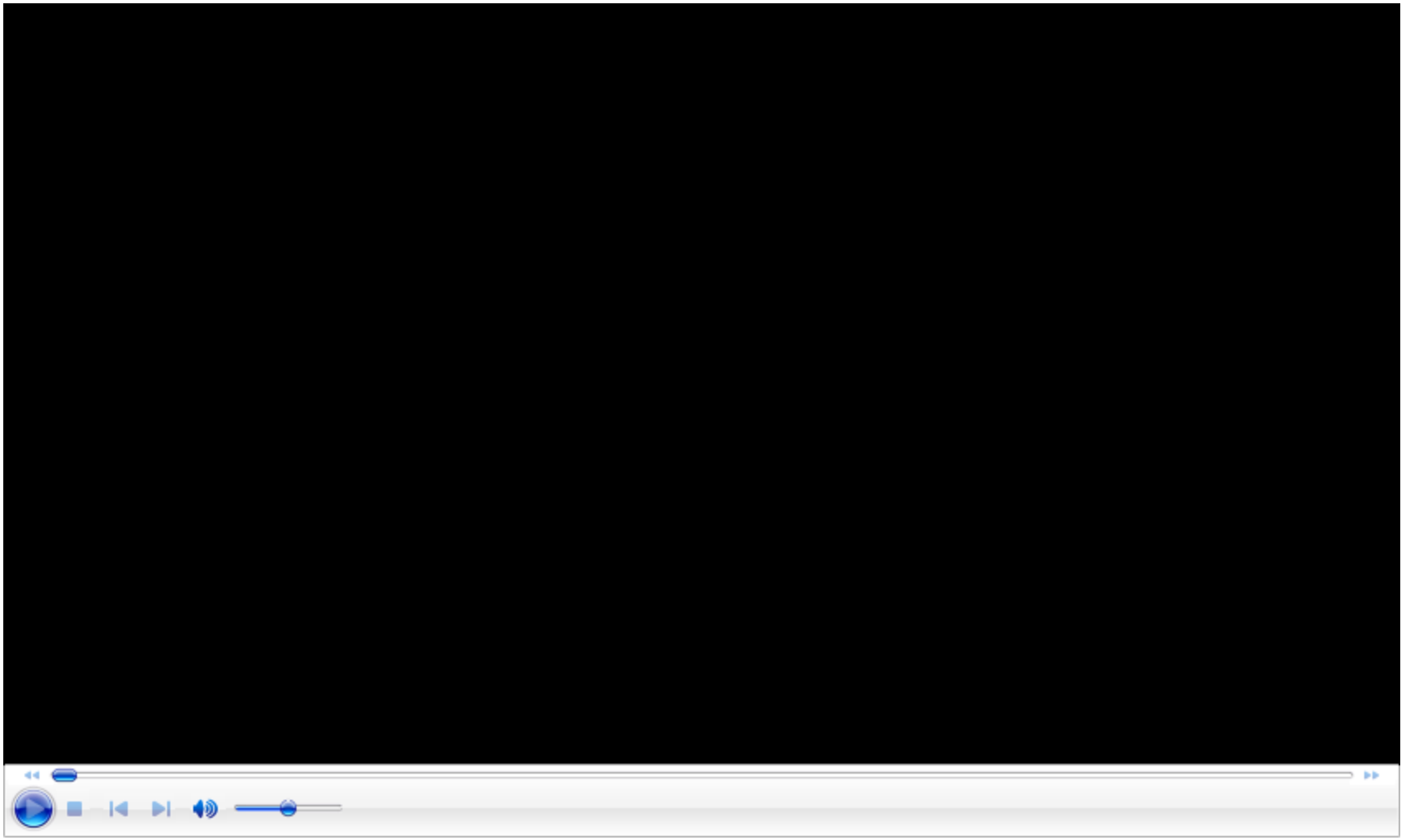
Repeat for 2 minute time periods reducing the opacity of previous time periods.



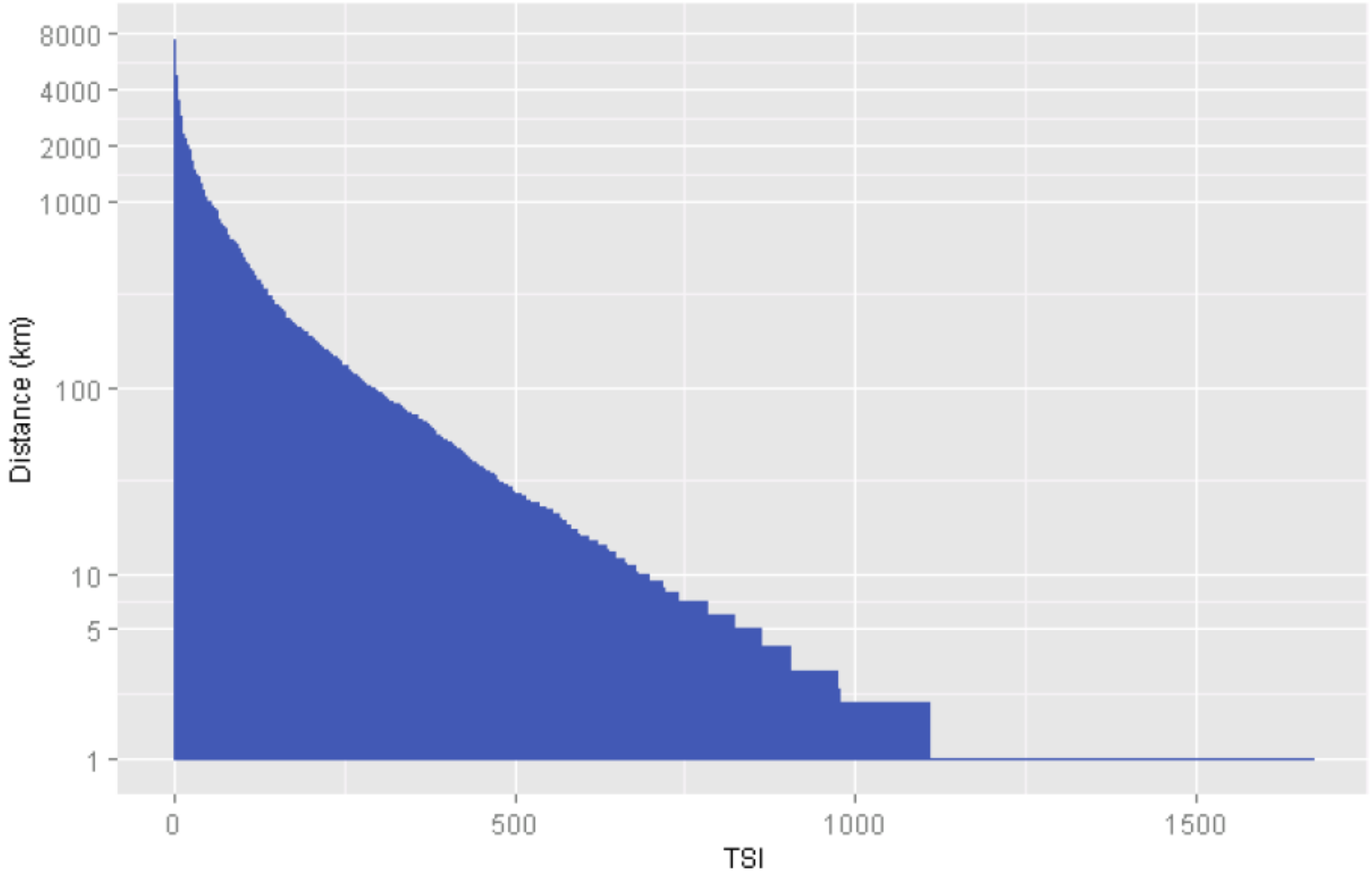
SYDNEY DRIVING STUDY



SYDNEY DRIVING STUDY

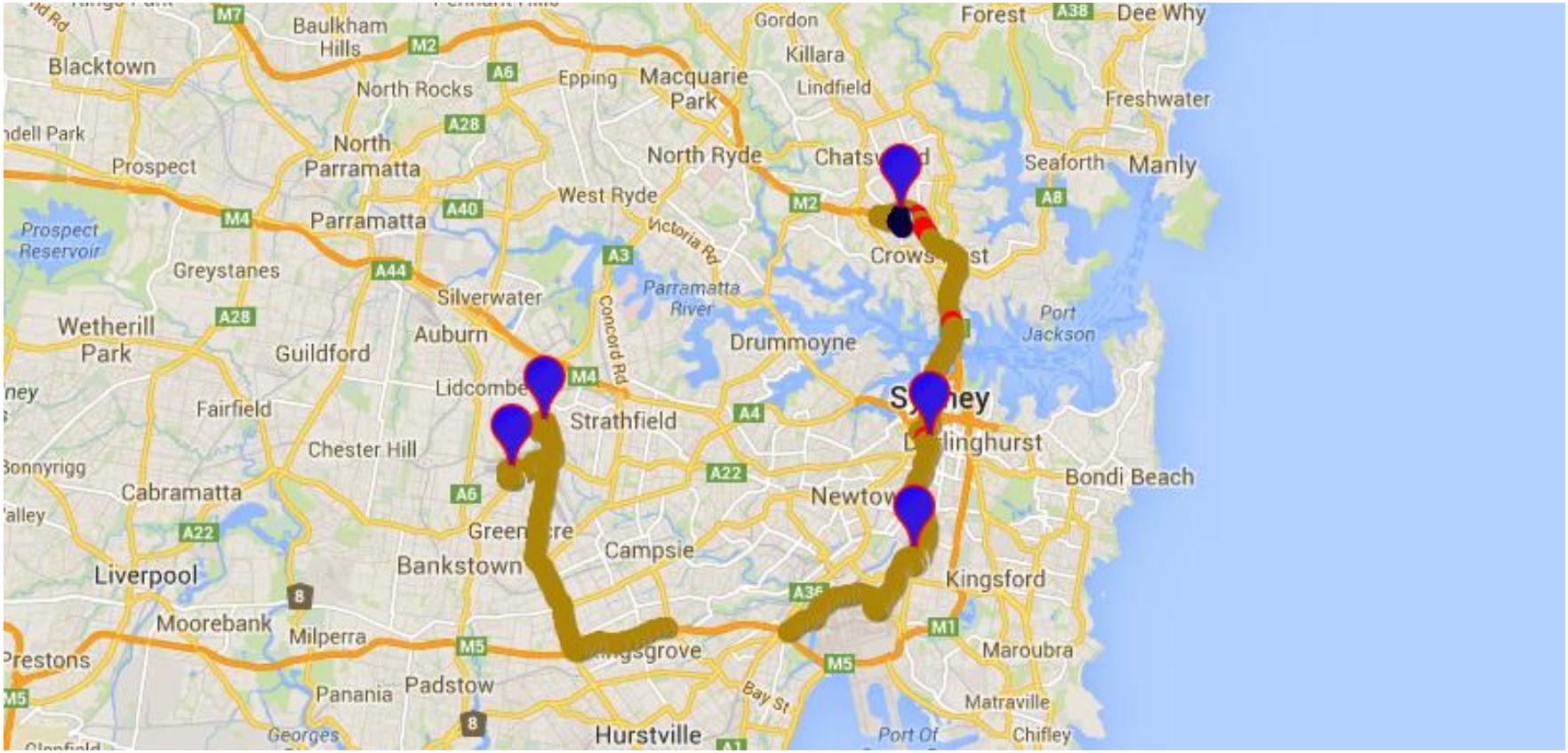


SYDNEY DRIVING STUDY



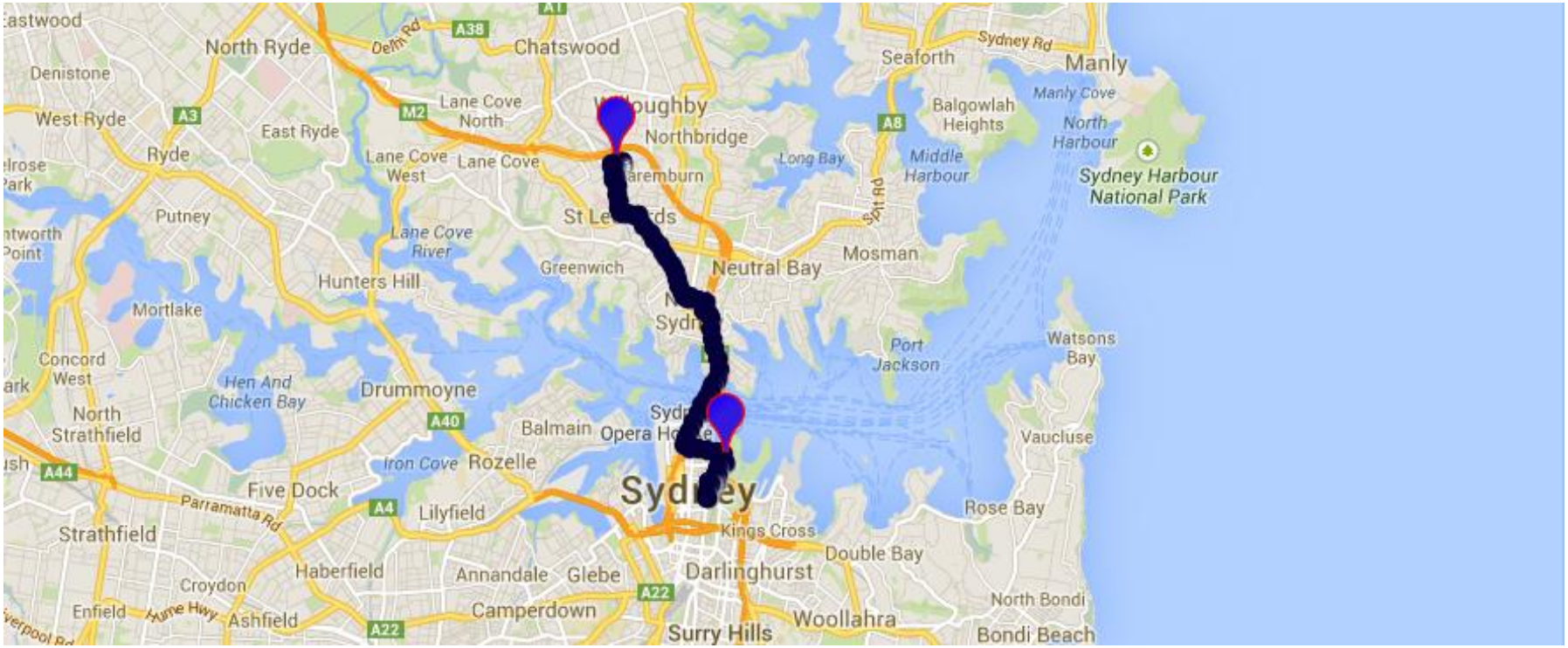
DATA COLLECTION FOR FREIGHT

GPS Example



DATA COLLECTION FOR FREIGHT

GPS Example



PROCESSING

- Sometimes the raw data requires further processing before it can be used
- For example, what if you wanted to calculate distances and durations?
  - Need to determine route and elapsed time between points
- Also sometimes necessary to merge data from several sources

- Processing the raw data allows for an even greater number of possible analyses
- The processed output can also be stored in a database and queried in the same way as the raw dataset

PROCESSING

- Two primary methods:
  - Write an R script to retrieve data from the database, process the data and then output the result back to the database
  - Write a PHP script to do the same

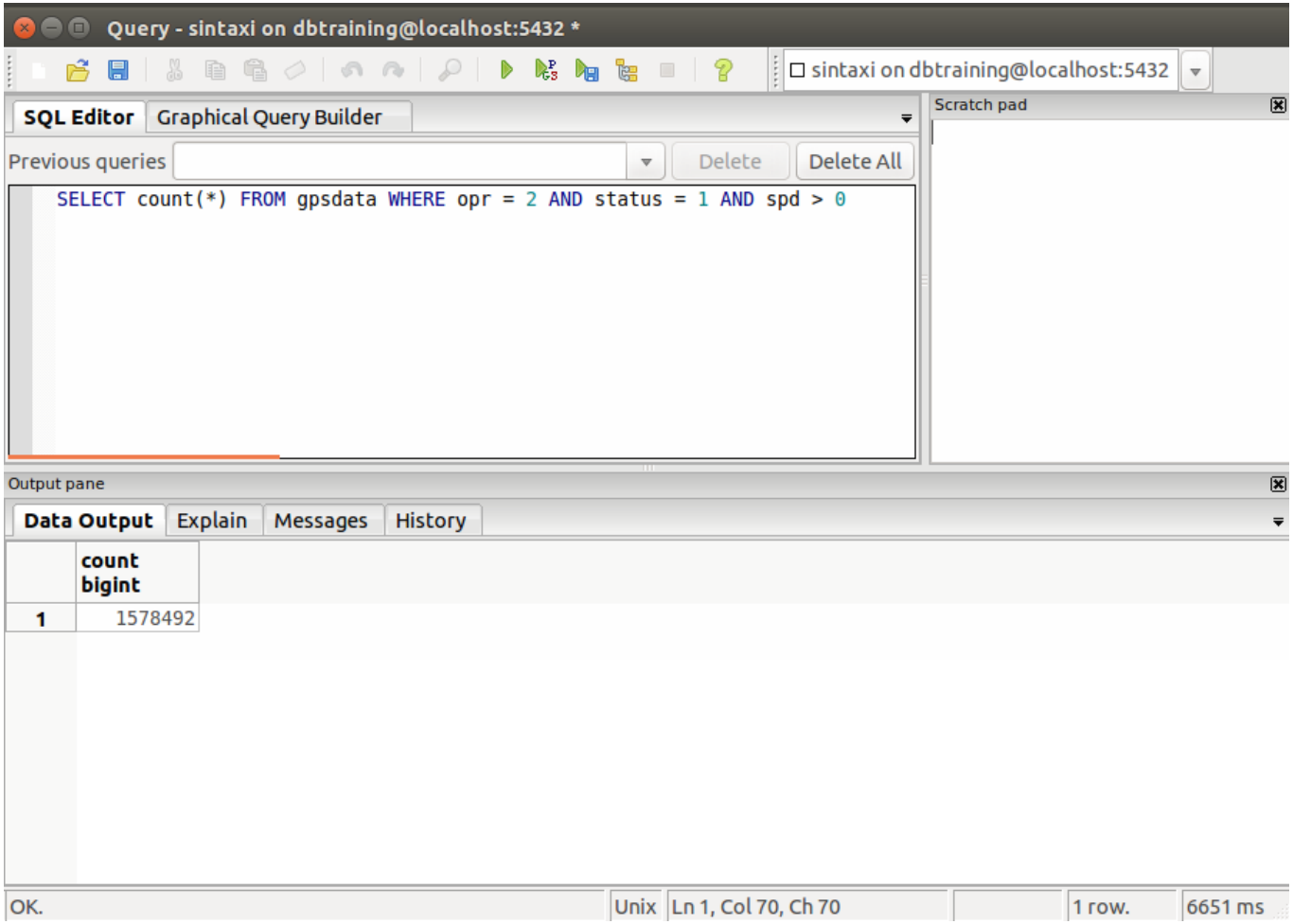
- There are advantages and disadvantages of each method:
  - If you are already using R it means only learning one scripting language if you also use it for processing
  - R is slower than PHP and less powerful in some ways
  - PHP is easier to use in an automated (batch) system

SOFTWARE

- This tutorial provides an interactive introduction to managing and analysing these datasets using a number of open source tools
- Database:
  - PostgreSQL with PostGIS
- Statistical package:
  - R with RStudio
- GIS:
  - PostGIS
  - Quantum GIS (QGIS)
- Programming: PHP and R

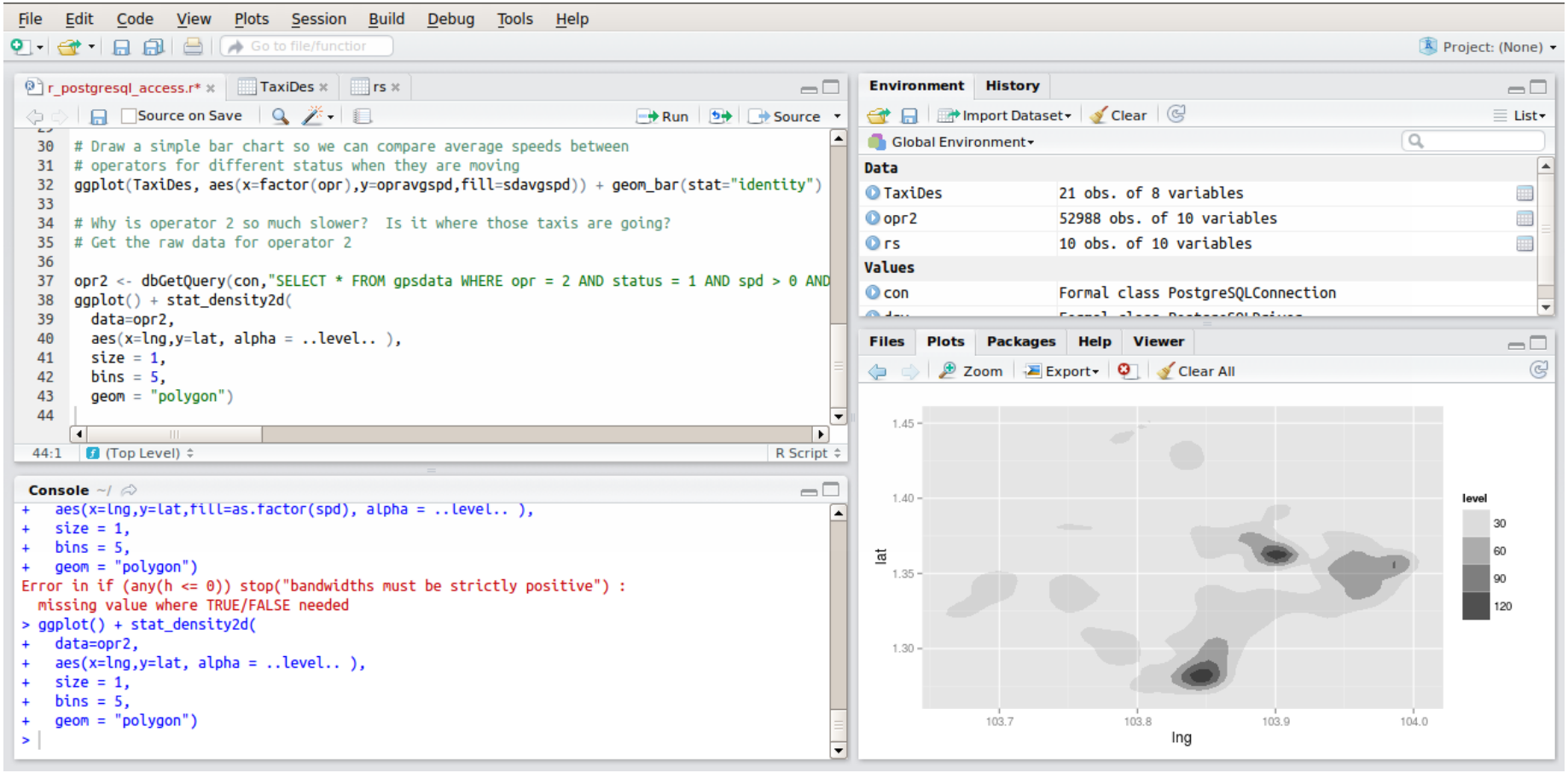
SOFTWARE

- Postgresql with PostGIS extension



SOFTWARE

› R using RStudio as a frontend



SOFTWARE

- PHP for data processing

```
_SERVER["XDG_RUNTIME_DIR"] => /run/user/1001
_SERVER["DISPLAY"] => :0
_SERVER["XDG_CURRENT_DESKTOP"] => Unity
_SERVER["GTK_IM_MODULE"] => ibus
_SERVER["LESSCLOSE"] => /usr/bin/lesspipe %s %s
_SERVER["TEXTDOMAINDIR"] => /usr/share/locale/
_SERVER["COLORTERM"] => gnome-terminal
_SERVER["XAUTHORITY"] => /home/dbtraining/.Xauthority
_SERVER["_"] => /usr/bin/php
_SERVER["PHP_SELF"] => -
_SERVER["SCRIPT_NAME"] => -
_SERVER["SCRIPT_FILENAME"] =>
_SERVER["PATH_TRANSLATED"] =>
_SERVER["DOCUMENT_ROOT"] =>
_SERVER["REQUEST_TIME_FLOAT"] => 1428418582.4849
_SERVER["REQUEST_TIME"] => 1428418582
_SERVER["argv"] => Array
(
    [0] => -
)

_SERVER["argc"] => 1

PHP License
This program is free software; you can redistribute it and/or modify
it under the terms of the PHP License as published by the PHP Group
and included in the distribution in the file:  LICENSE

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any
questions about PHP licensing, please contact license@php.net.
dbtraining@SINdata:~$
```

WHAT IS A RELATIONAL DATABASE?

- A data model based on multiple flat files, designed for records with dissimilar attribute structures connected by a common key identifier.
- Provides a much more efficient method of storing and querying data.
- Allows for powerful and complex queries where the relationship between entities is important.
  - "What is the average speed of drivers in Leeds who drive at least three days per week and have been to the University of Leeds at least once in the past month?"
- The relational database is integral to the collection, management and use of large datasets

EXAMPLES OF RELATIONAL DATABASES

- MySQL
- Postgresql
- IBM DB2
- Microsoft SQL Server
- Oracle Database

EXAMPLE OF A RELATIONAL DATABASE

Patient Admission Record				
ID	**Key**	**Admis-sion**	**Check Out**	**Room Number**
352	42	2/5/04	4/5/04	102
353	78	3/6/04	3/7/04	104

Accident Report				
ID	**Key**	**Date**	**Type**	**Address**
236	42	2/5/04	Car	91 Pitcairn Street
237	126	4/5/04	Pedestrian	12 Niue Ave

Person Record			
ID	**Birth Date**	**Name**	**Address**
42	1/1/80	John Smith	75 Elm Street
43	8/4/92	Robert Jones	112 Brooklyn Ave

Key is the key!

KEYS IN RELATIONAL DATABASES

- Two types of keys:
  - "Primary Keys"
    - A unique identifier for each record in the table (often a sequential ID field).
  - "Foreign Keys"
    - Are the unique identifiers for a record in a related table.
  - Primary keys must be unique within the table, foreign keys can be repeated.
- A table may have no foreign keys, one foreign key or several foreign keys.
  - Depends on the requirements and relationships between the tables.

CARDINALITY

- Cardinality is the type of relationship between related tables.
- Several types of cardinality:
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-Many

ONE-TO-ONE RELATIONSHIP

› Each record in Table A matches only one record in Table B

Table A: Products				
ID	**Name**	**Brand**	**Cost**	**SalePrice**
256	X200	575	$600	$800
257	X201	575	$650	$900
258	Pi Model A	576	$25	$25
259	Pi Model B+	576	$35	$35

Table B: Brand			
ID	**Name**	**Contact**	**ContactNumber**
575	Lenovo	Charles Babbage	02 9114 1234
576	Raspberry Pi Foundation	Alan Turing	02 9351 8765

ONE-TO-MANY RELATIONSHIP

› Each record in Table A matches more than one record in Table B but records in Table B match only one record in Table A.

Table A: Students

ID	Name	DateOfBirth	ContactNum	Degree
331	Adele Goldberg	1945-07-07	+1 242-111-9876	Computer Sci.
332	Linus Torvalds	1969-12-28	+1 353-987-6543	Software Eng.

Table B: Unit Marks

ID	Unit	StudentID	Result
575	COMP1000	331	HD
576	COMP1001	331	HD
577	SENG1000	332	HD
578	ARTS1000	332	D

MANY-TO-ONE RELATIONSHIP

› Each record in Table A matches only one record in Table B but each record in Table B can match more than one record in Table A

Table A: Students				
ID	**Name**	**DateOfBirth**	**ContactNum**	**Degree**
256	Charles Bachman	1924-12-11	+1 212-898-8888	991
257	Don Estridge	1937-06-23	+1 878-222-1322	991
258	Adele Goldberg	1945-07-07	+1 242-111-9876	991
259	Linus Torvalds	1969-12-28	+1 353-987-6543	992

Table B: Degrees			
ID	**Name**	**CourseworkCoordinator**	**ContactNumber**
991	Computer Science	Charles Babbage	02 9114 1234
992	Software Engineering	Alan Turing	02 9351 8765

MANY-TO-MANY RELATIONSHIP

› Each record in Table A can match more than one record in Table B and each record in Table B can match more than one record in Table A.

Table A: Students

ID	Name	DateOfBirth
256	Charles Bachman	1924-12-11
257	Don Estridge	1937-06-23
258	Adele Goldberg	1945-07-07
259	Linus Torvalds	1969-12-28

Table B: Units

UnitCode	Name	LectureRoom
COMP1000	Introductory Comp. Sci.	LT5
COMP1001	Intermediate Comp. Sci.	LT2
HIST1000	Introductory History	LT13

Table C: StudentEnrolments

StudentID	Unit
256	COMP1000
257	COMP1000
258	COMP1000
258	COMP1001
259	COMP1001
256	HIST1000
257	HIST1000
259	HIST1000

BASIC INTRODUCTION TO SQL

- SQL standards for “Structured Query Language”.
- SQL is the language used to interact with databases.
  - Different databases use different variants of SQL but all follow the same general structure.

BASIC INTRODUCTION TO SQL

SELECT * FROM students;

"SELECT" statement selects and retrieves the data from the database.

"*" selects all columns in the table(s).

"FROM" delineates the end of the list of fields and tells the database that the following word is a table name.

The (first) table used for the query follows the "FROM".

BASIC INTRODUCTION TO SQL

SELECT “Name”, “Email” FROM students;

“SELECT” statement selects and retrieves the data from the database.

Specifies that the “Name” and “Email” fields are selected.

“FROM” delineates the end of the list of fields and tells the database that the following word is a table name.

The (first) table used for the query follows the “FROM”.

BASIC INTRODUCTION TO SQL

```
SELECT * FROM students WHERE "DOB" < '2015-01-01';
```

Basic statement same as previous slide

"WHERE" delineates between select statement and conditions. Is followed by one or more conditional statements.

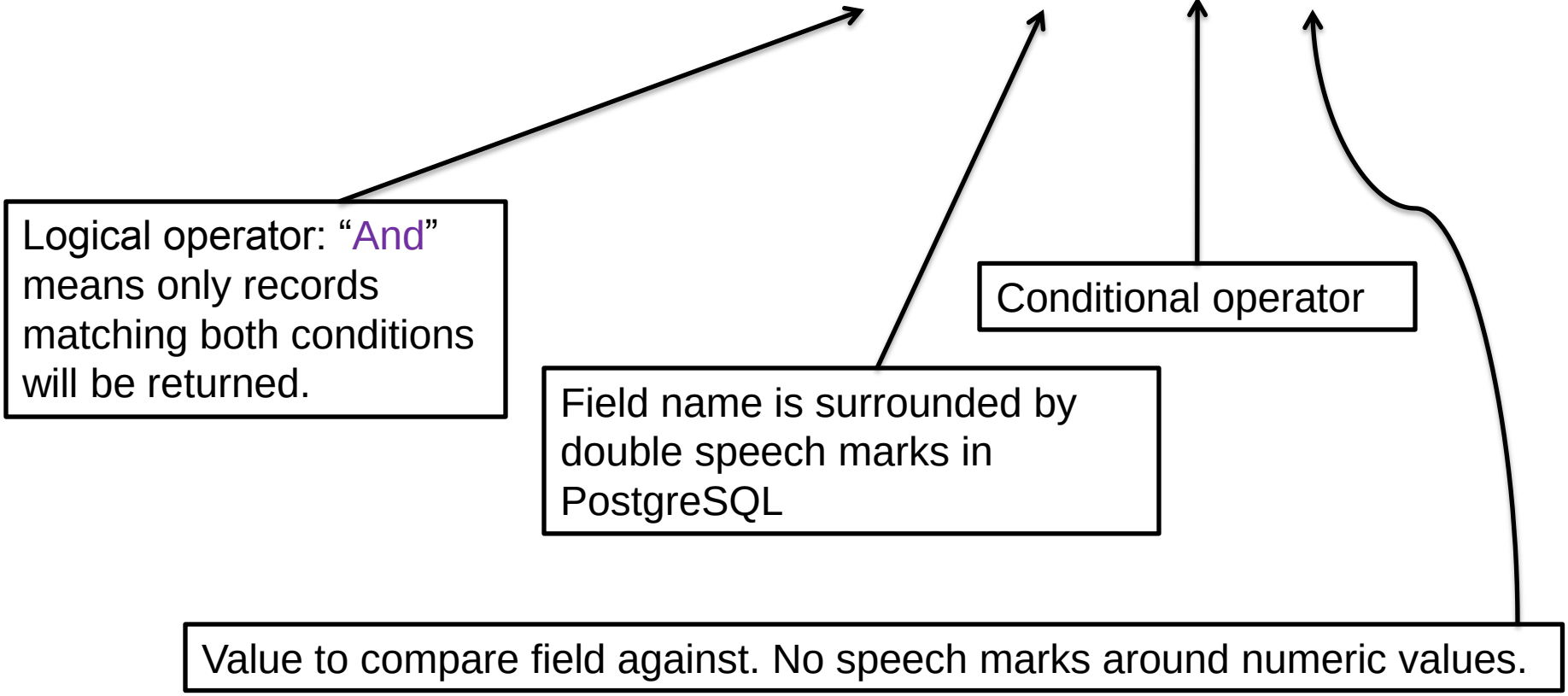
Conditional operator ("less than")

Field name is surrounded by double speech marks in PostgreSQL

Value to compare field against. Strings and dates are surrounded by single speech marks.

BASIC INTRODUCTION TO SQL

```
SELECT * FROM students WHERE
"DOB" < '2015-01-01' And "Degree" = 812;
```



BASIC INTRODUCTION TO SQL

- Main conditional operators:
  - = Equality operator (are two fields/values equal?)
  - < and > Less than/greater than operators
  - <= and >= Less than or equal to/Greater than or equal to
  - <> Not equal to
  - 'Like' – Comparison operator for strings with wildcards
  - 'Not' – Used before 'Like' to give opposite to result to 'Like'
- Main logical operators available:
  - 'And' – Records must match both conditions
  - 'Or' – Records must match at least one condition

SHORT COURSE

- Introduction to ‘Big Data’ Databases and Programming for Transport Researchers
  - Being held in Sydney from 10th to 12th June
  - http://sydney.edu.au/business/itls/courses/databases_and_programming

QUESTIONS?

› Any questions or comments?

Contact e-mail address:

adrian.ellison@sydney.edu.au

richard.ellison@sydney.edu.au