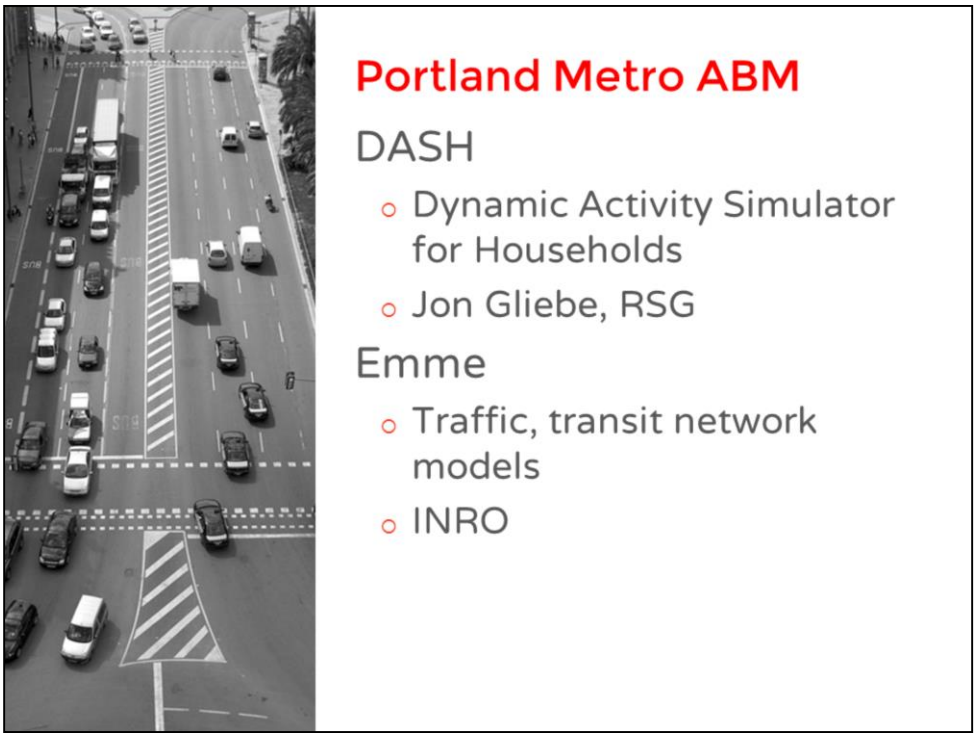
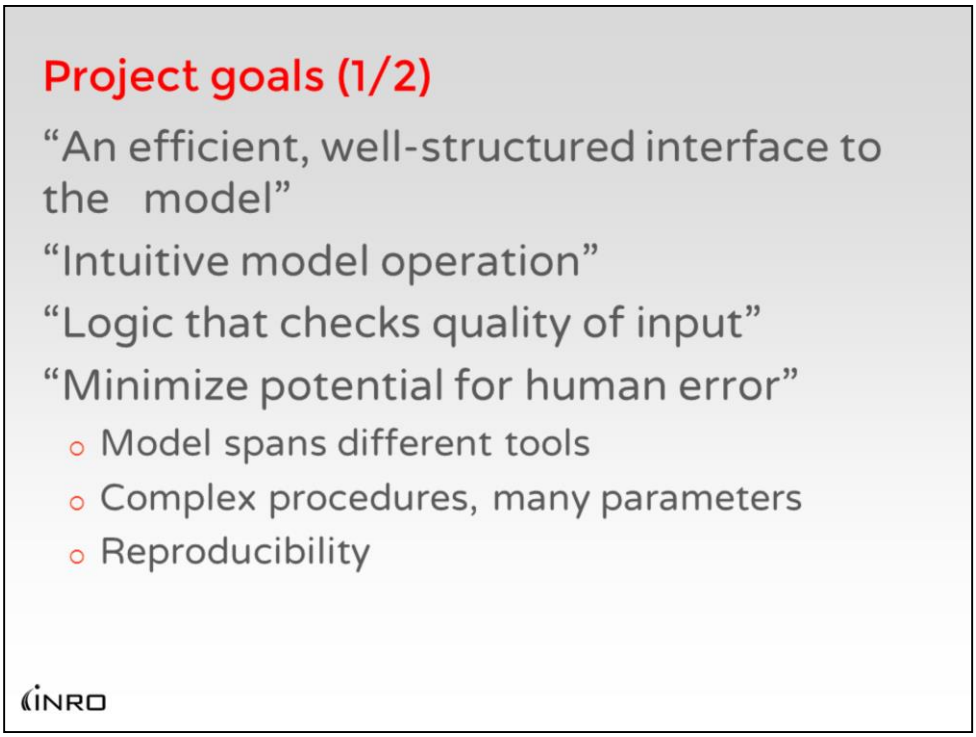
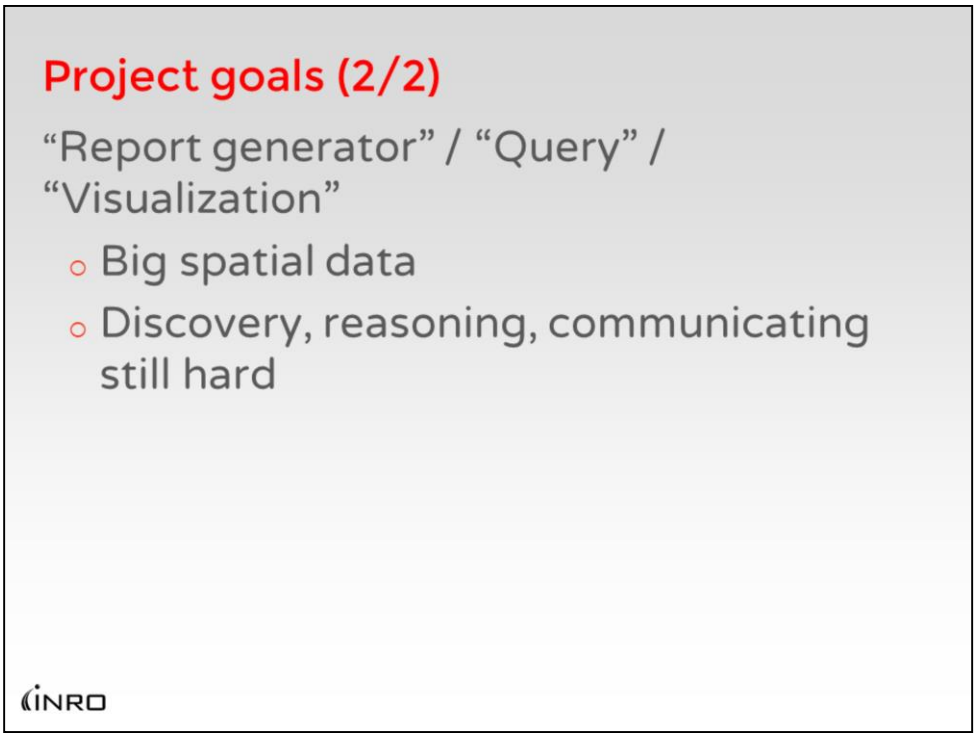

An interactive, visual computing environment for the Portland Metro activity-based travel forecasting model
Daniel Florian | dan@inrosoftware.com
Kevin Bragg | kevin@inrosoftware.com
INRO



INRO is collaborating with Portland Metro and RSG to develop a coherent user interface, operating environment and reporting/query interface for a new activity-based model. This has been a chance to introduce some exciting new technologies for running models as well as for visualization and analysis.

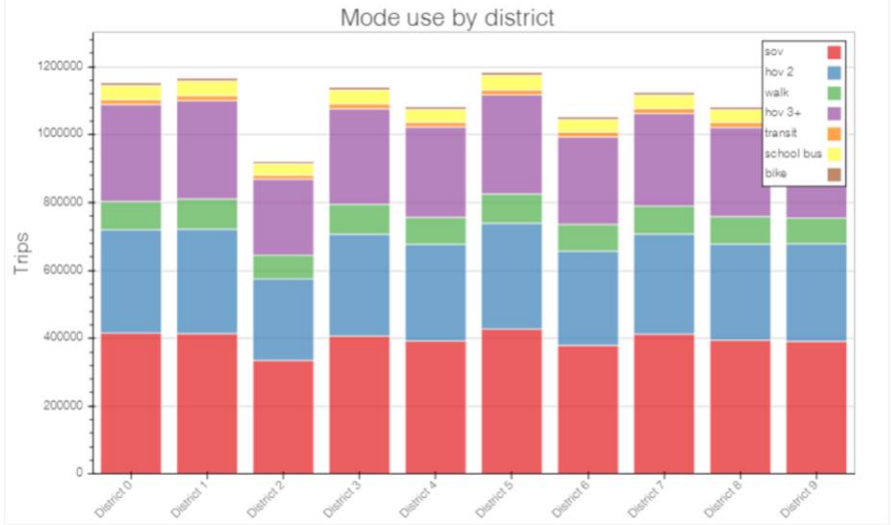


Project goals include a usable, reproducible system in which to perform model specification, model production and model audit.

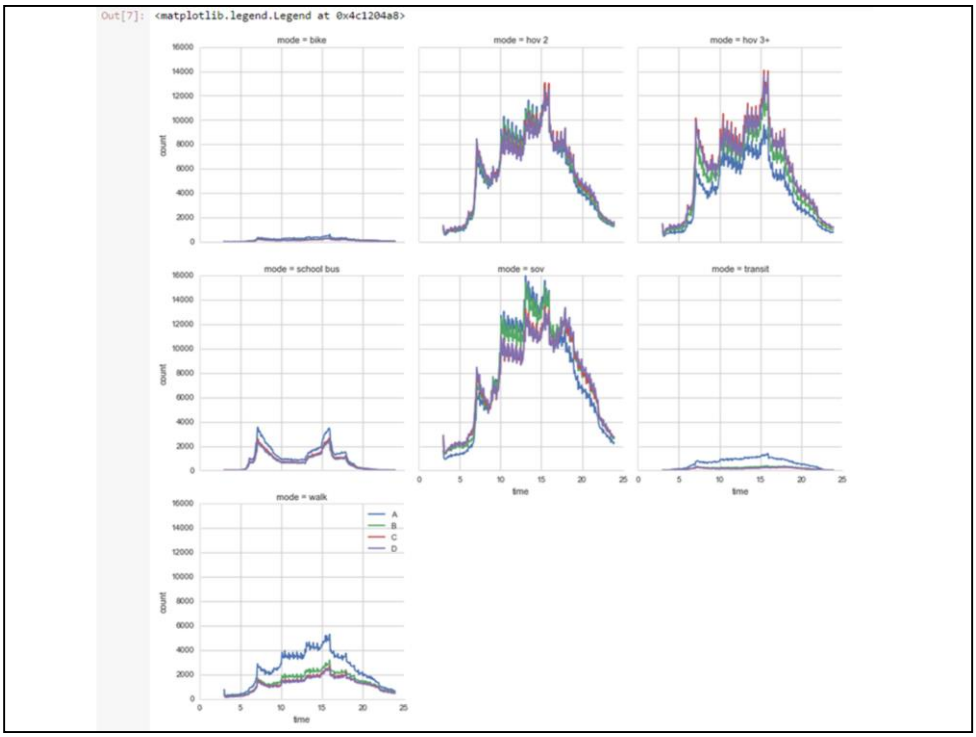


Reporting, query and visualization of ABM output is also a key project goal; one with fundamental challenges for which we propose a solution here.

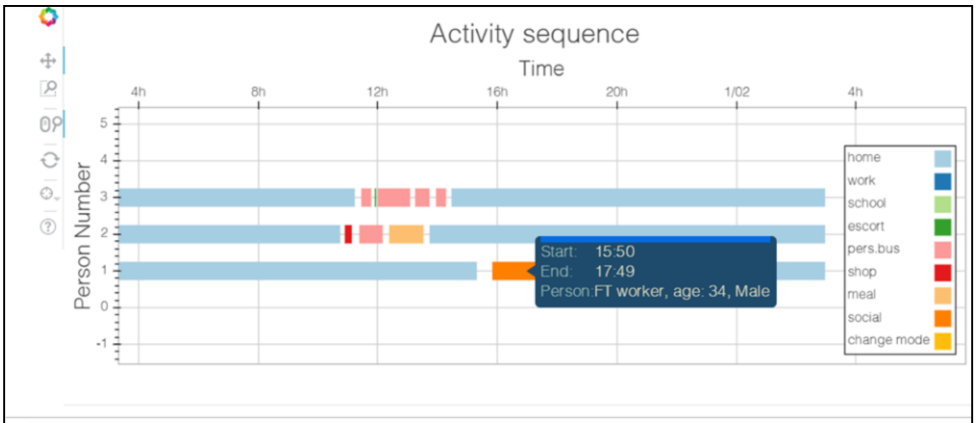
```
In [6]: bar = Bar(raw_data,
                  ["District %s" % x for x in range(10)],
                  title="Mode use by district", xlabel="", ylabel="Trips",
                  stacked=True,  legend="top_right", tools="save", yscale="linear",
                  width=1000, height=600, palette=colours)
        bar.left[0].formatter.use_scientific = False
        _plot.show(bar)
```



The IPython Notebook is an interactive computing system which mixes text, code execution, math, graphics and other media in literate, reproducible computational documents. The project is open-source and intended to facilitate computational workflows across individual exploration, collaboration, production-scale execution and communication, making it a natural technology choice. Using popular Python libraries associated under the SciPy and PyData banners provides scientific and plotting libraries for analytics, like this chart showing mode use by district…



…and this mode use by time of day and income panel…

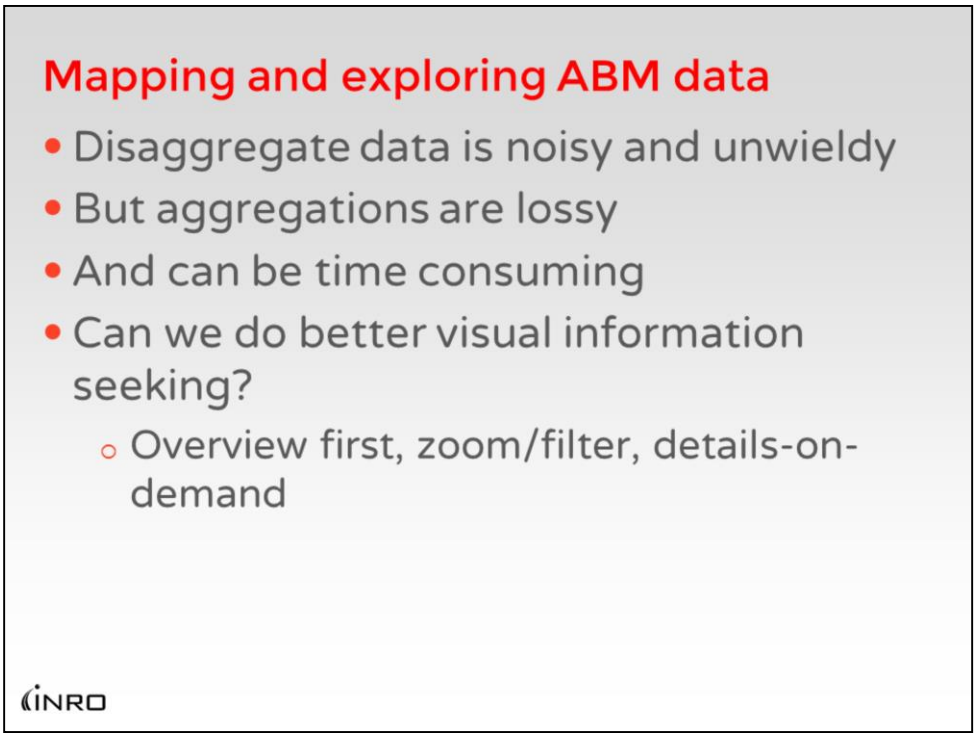



```
]: # Show activities / trips in a table
   trip_table.query("hhno==%s" % household_id)
```

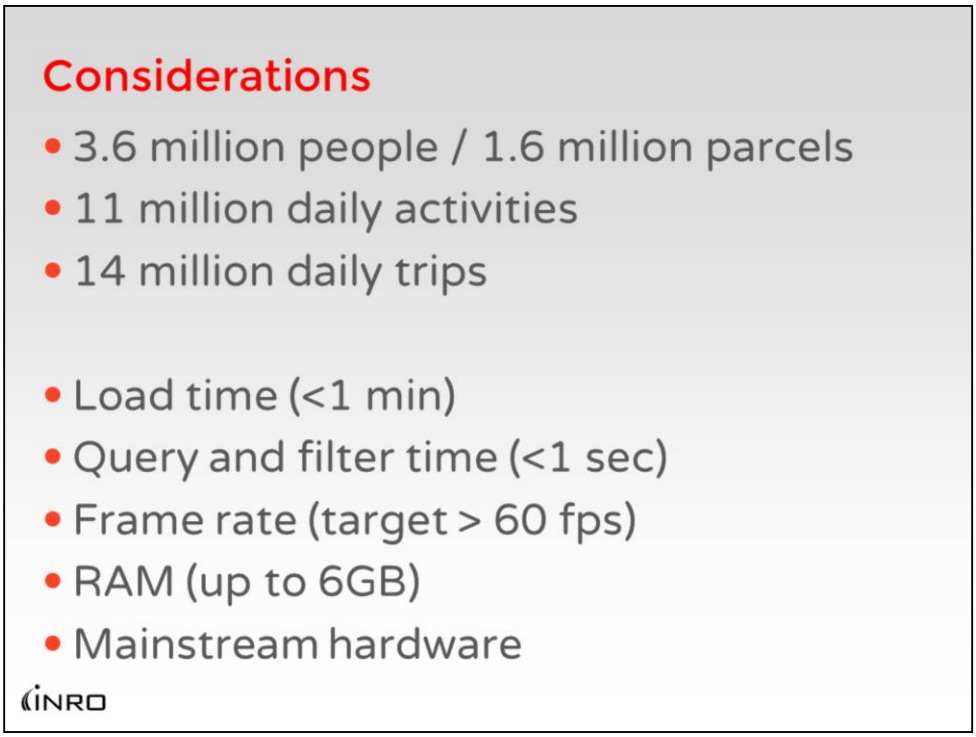
]:

	hhno	pno	tour	half	tseg	deptm	arrtm	endacttm	purpose	mode
3477674	774347	2	1	1	1	419	431	807	school	sov
3477675	774347	2	1	2	1	808	820	894	home	sov
3477676	774347	2	3	1	1	895	926	1045	social	walk
3477677	774347	2	3	2	1	1046	1077	1086	home	walk
3477678	774347	2	2	1	1	1087	1090	1140	meal	sov
3477679	774347	2	2	2	1	1141	1144	1283	home	sov
3477680	774347	2	4	1	1	1284	1292	1582	social	sov
3477681	774347	2	4	2	1	1583	1591	1618	home	sov

…and these detailed household activity sequence diagrams. Again, all the charts illustrated thus far are developed using open-source tools.

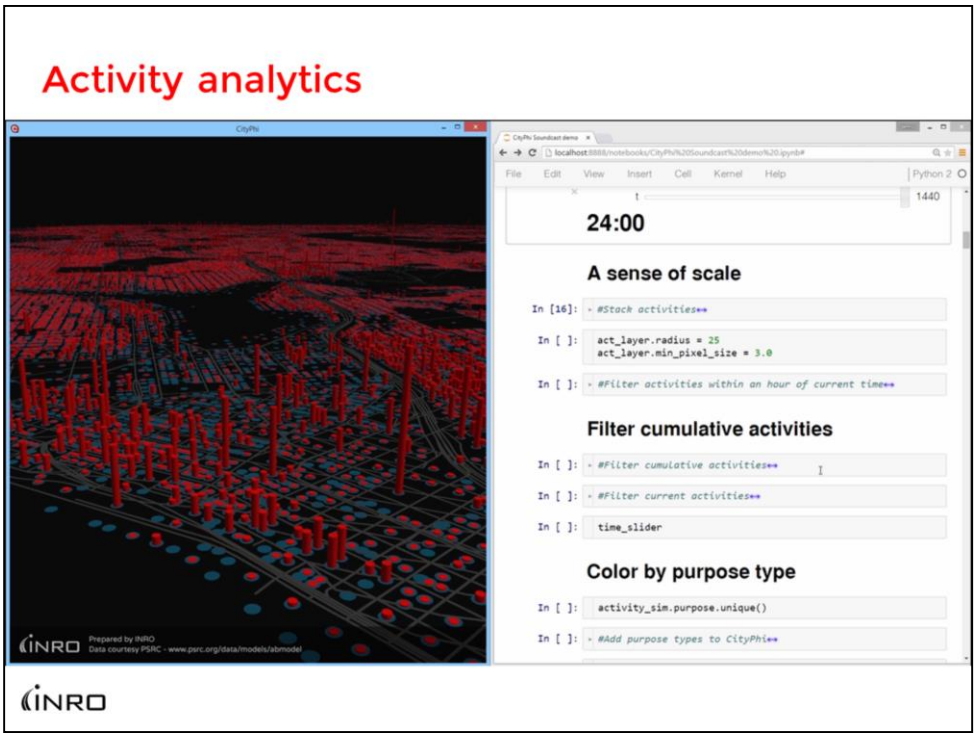


But mapping, animation and visual interaction with ABM output, which consists of tens of millions of trip trajectories and activities for even medium-sized cities, breaks down using traditional tools. We can do better.



In order to properly support visual analytics with ABM output data, we need to be able to query and filter disaggregate data quickly, and explore space and time interactively. We'd like this to happen on reasonable hardware, too.

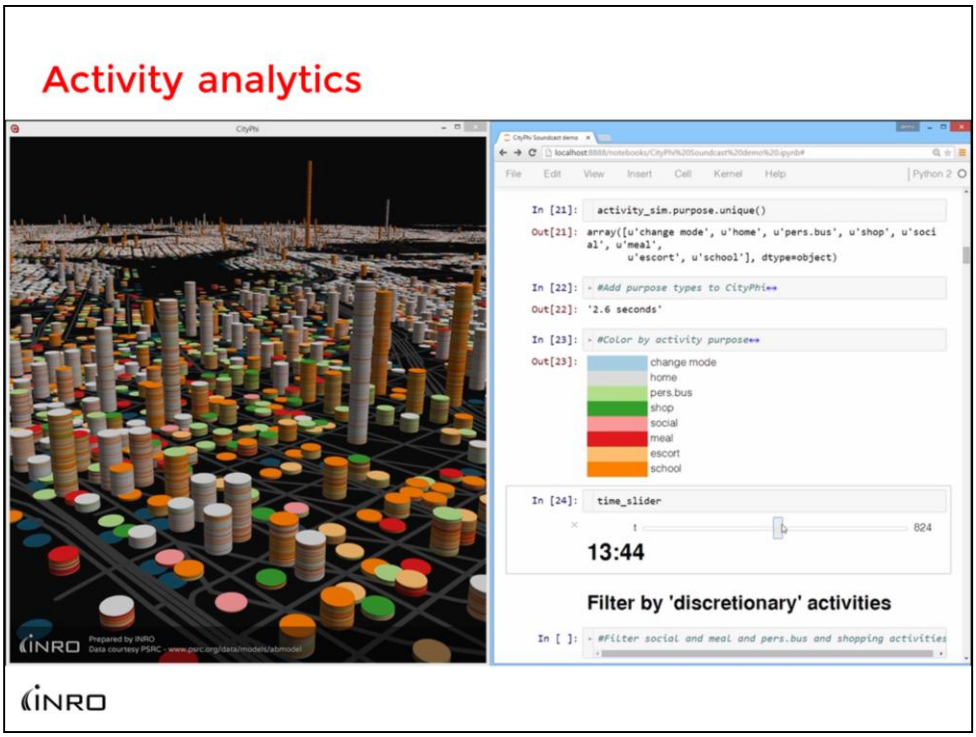
DASH ABM data for Portland is still being delivered, so instead we show examples from the metropolitan Seattle area, graciously provided by PSRC (http://www.psrc.org/data/models/abmodel/)



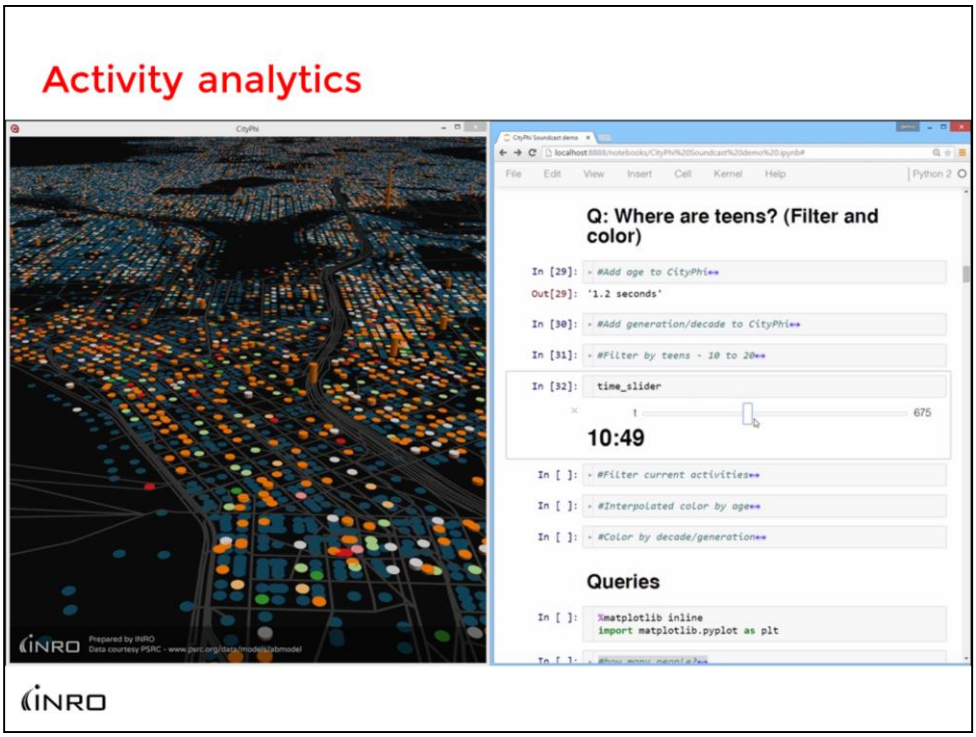
The right-hand side in the ensuing video demonstrations shows the IPython Notebook, discussed briefly earlier.

The left-hand side is a new INRO-developed framework for visual analytics of large-scale spatial and mobility data. This works together in core memory to meet the computational requirements laid out on the last slide, and to facilitate visual analytics on ABM data.

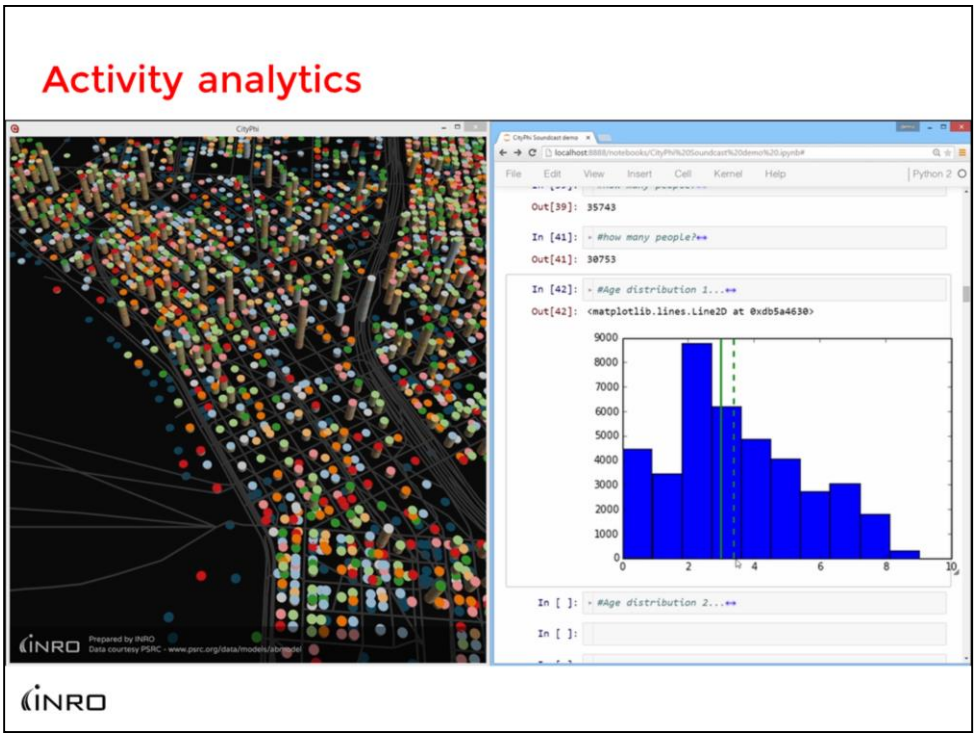
This image shows disaggregate activities displayed vertically at parcels and animated by time. The ensuing video stills provide additional examples of visual analytics.



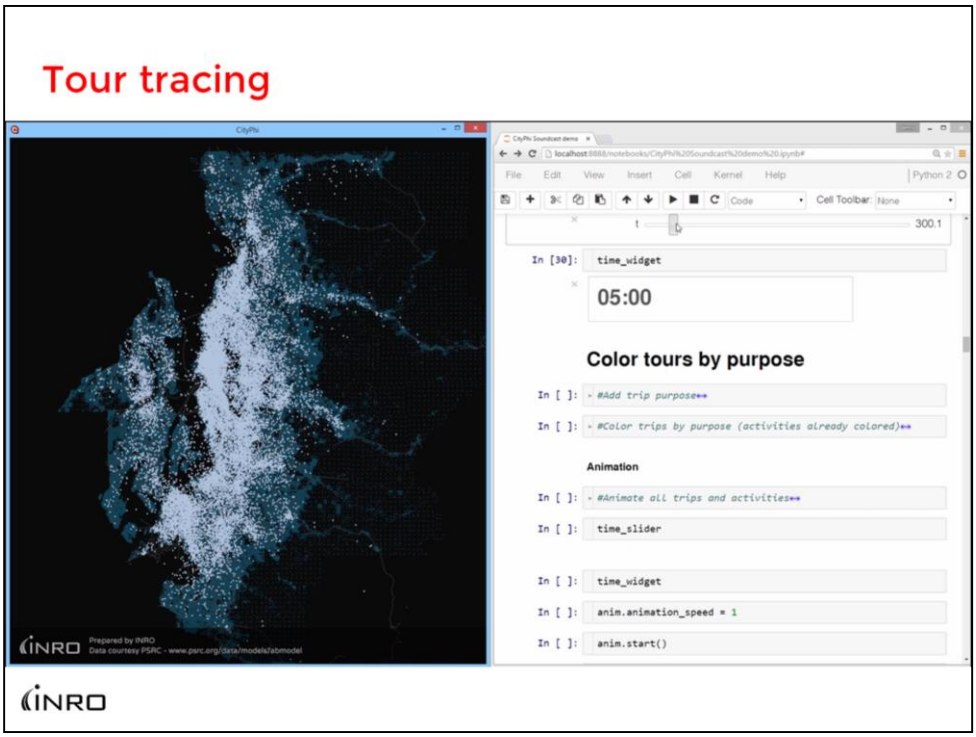
This image shows activities colored by purpose at 1:44pm. Working in core memory with the notebook, it takes 2.6 seconds to make activity purposes available for visual analytics, facilitating ‘discussion time’ visual analytics.



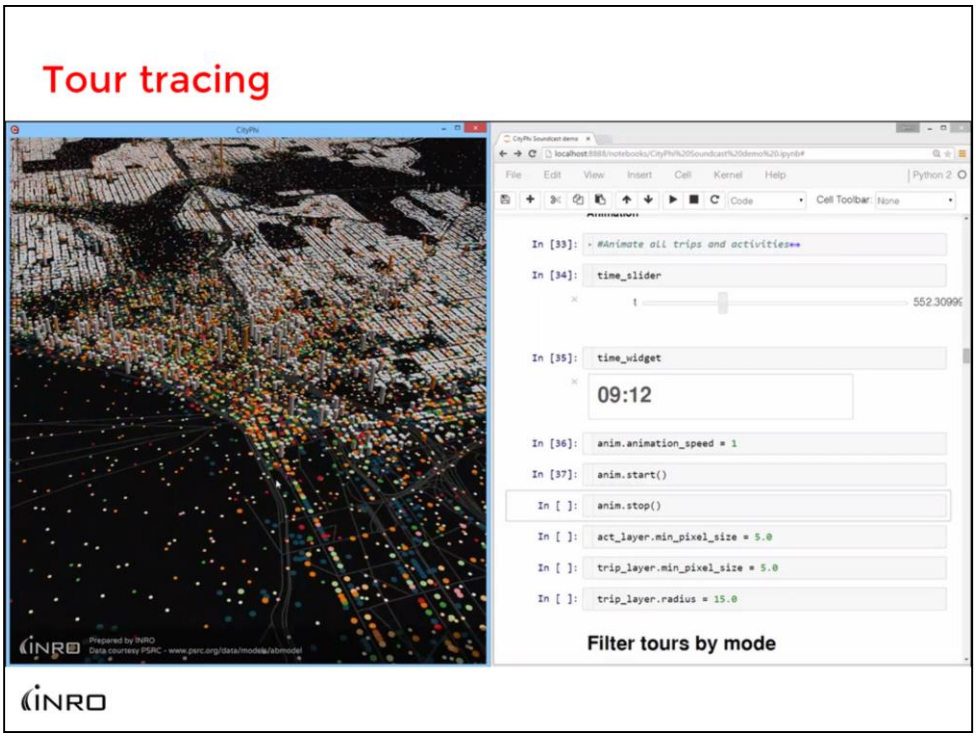
Uploading age information in 1.2 seconds and filtering by teens (10-20 year olds). At 10:49am many of them are in school (orange).



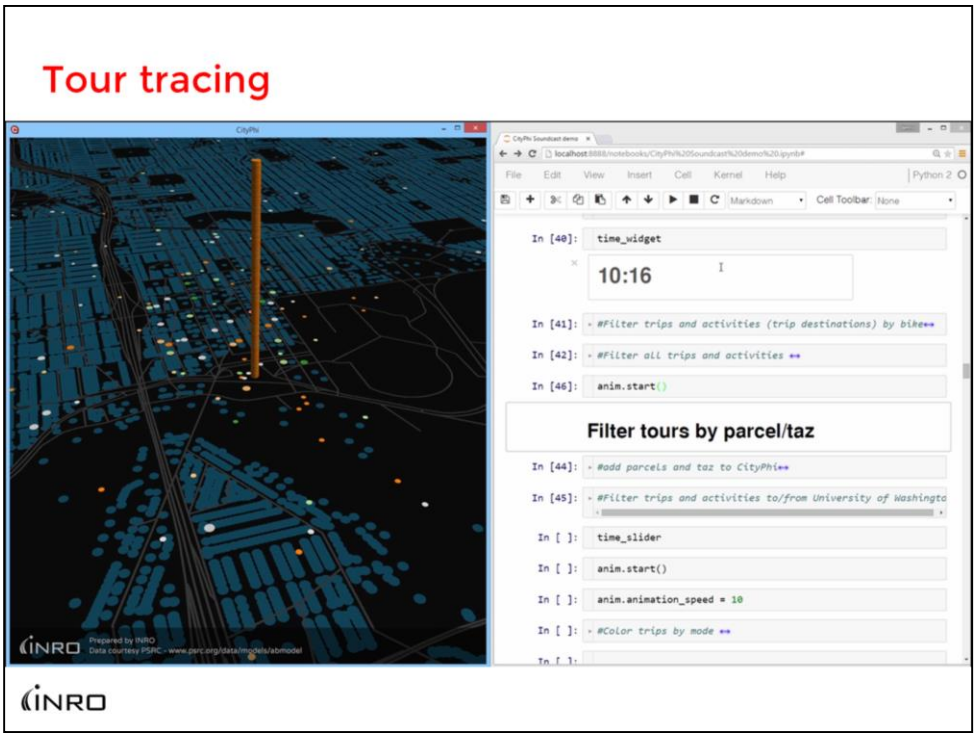
Keeping the data entirely disaggregate means that it can be queried. Here, we produce an age distribution of over 30000 people in downtown Seattle in the early afternoon, interactively. This can be easily compared with other regions or time periods of interest.



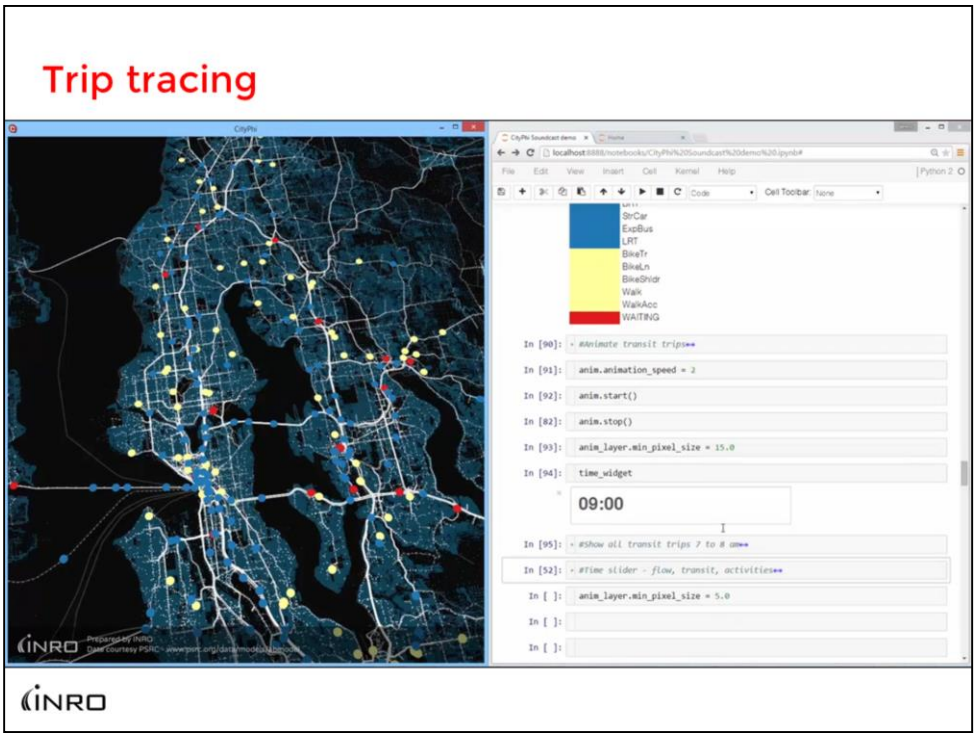
Tour tracing shows as-the-crow-files animations of mobility, from a fully synthesized set of >10,000,000 daily trips. (Note: work trips are excluded from this data set due to data privacy requirements)



Tour traces can be also colored, for example by activity purpose, or filtered, for example by mode (eg. transit, bike, etc…)

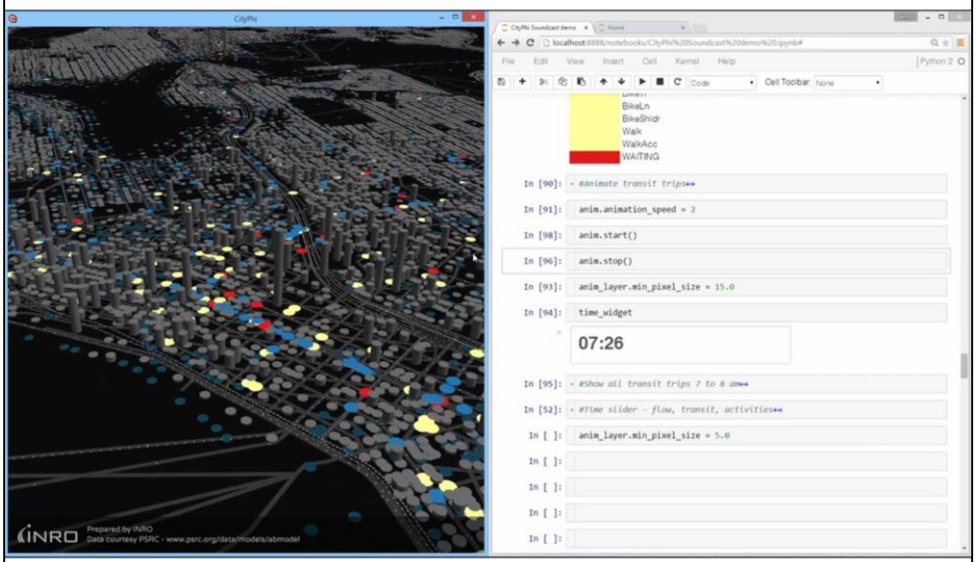


Activities and tours can also be filtered by parcel or TAZ, for instance, to show the rate of arrival at the university in the AM, and the corresponding parcels from which students and faculty arrive.

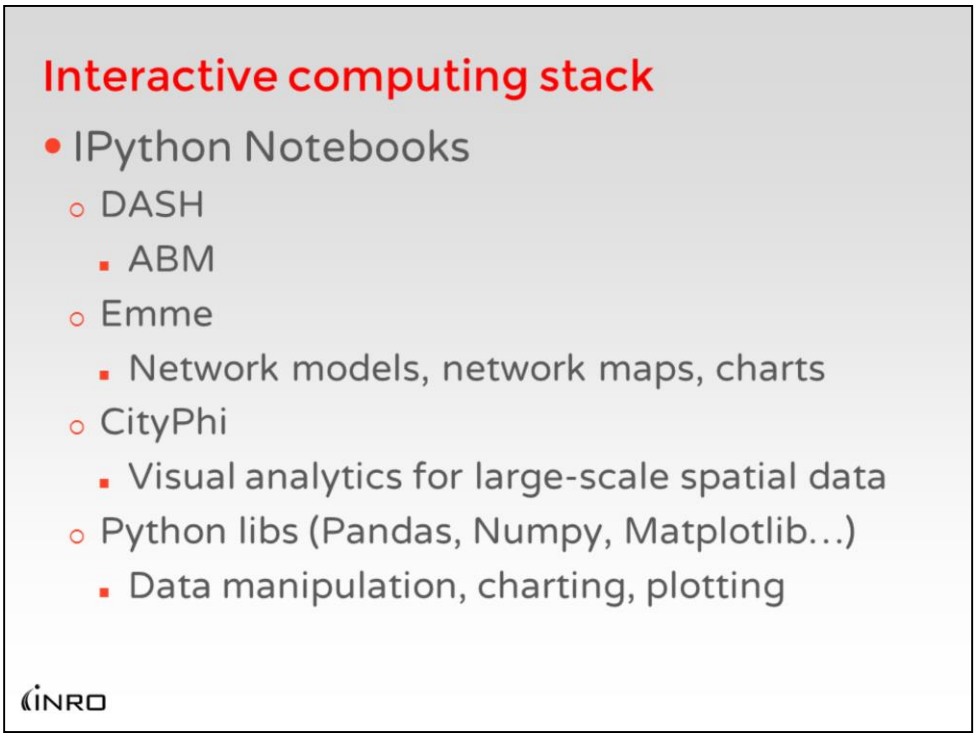


Traffic flow and transit trips can also be animated, filtered, colored and queried. Here paths from a morning transit assignment are used to visualize the in-vehicle, walk/access/egress and waiting portions of discrete transit trips.

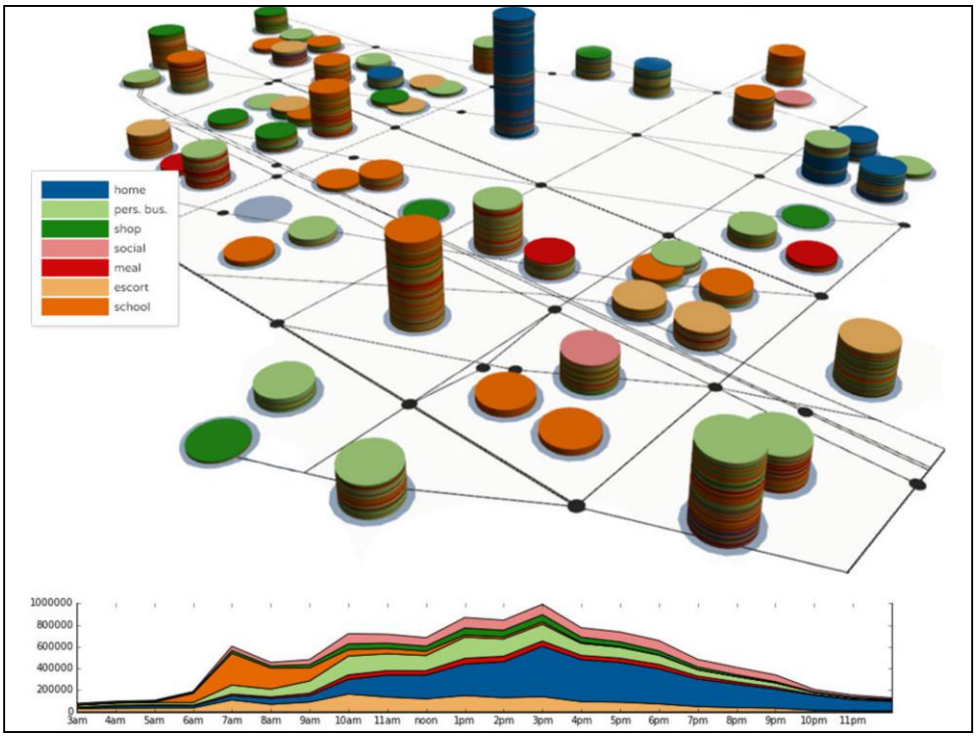
Trip tracing



INRO



The IPython Notebook aims to solve interactive, reproducible computation, and together with Python and many popular open-source libraries is quickly becoming a standard in scientific computing. We are excited about applications to travel demand and transportation forecasting, with both PSRC and Portland Metro investing in corresponding staff skills. DASH, Emme and CityPhi will participate in this framework at Portland Metro.


home
pers. bus.
shop
social
meal
escort
school
1000000
800000
600000
400000
200000
0
3am
4am
5am
6am
7am
8am
9am
10am
11am
noon
1pm
2pm
3pm
4pm
5pm
6pm
7pm
8pm
9pm
10pm
11pm


An interactive, visual computing environment for the Portland Metro activity-based travel forecasting model
Daniel Florian | dan@inrosoftware.com
Kevin Bragg | kevin@inrosoftware.com
INRO