Accessing bigger datasets in R using SQLite and dplyr

Nicholas J. Horton

Amherst College, Amherst, MA, USA

March 24, 2015

nhorton@amherst.edu

Thanks

- to Revolution Analytics for their financial support
- to the Five College Statistics Program and the Biostatistics Program at UMass/Amherst
- to Nick Reich, Emily Ramos, Kostis Gourgoulias, and Andrew Bray (co-organizers of the Meetup)
- suggestions of meeting topics, speakers and locations always welcomed

Upcoming meetups

working with geographic data in R 3/26 at 7:00pm

DataFest 3/27-3/29

learn about plotly 4/2 at 4:00pm

updates from RStudio: J.J. Allaire 7/15 (save the date!)

statistical analysis of big genetics and genomics data Xihong Lin, 11/16 (save the date!)

using docker for fun and profit anyone willing?

Acknowledgements

- joint work with Ben Baumer (Smith College) and Hadley Wickham (Rice/RStudio)
- supported by NSF grant 0920350 (building a community around modeling, statistics, computation and calculus)
- more information at `http://www.amherst.edu/~nhorton/precursors`

Prelude

- "New Frontier in statistical thinking" (Chamandy, Google @ JSM 2013)
- teaching precursors to big data/foundations of data science as an intellectually coherent theme (Pruim, Calvin College)
- growing importance of computing and ability to "think with data" (Lambert, Google)
- key capacities in statistical computing (Nolan and Temple Lang, TAS 2010)
- "Statistics and the modern student" (Gould, ISR 2010)
- "Building precursors for data science in the first and second courses in statistics" (Horton, Baumer, and Wickham, 2015)

Prelude (cont.)

How to accomplish this?

- start in the first course
- build on capacities in the second course
- develop more opportunities for students to apply their knowledge in practice (internships, collaborative research, teaching assistants)
- new courses focused on "Data Science"
- today's goal: helping DataFesters grapple with larger datasets

Data Expo 2009

Ask students: have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted it if you'd had more data? (Wickham, JCGS, 2011)

Data Expo 2009

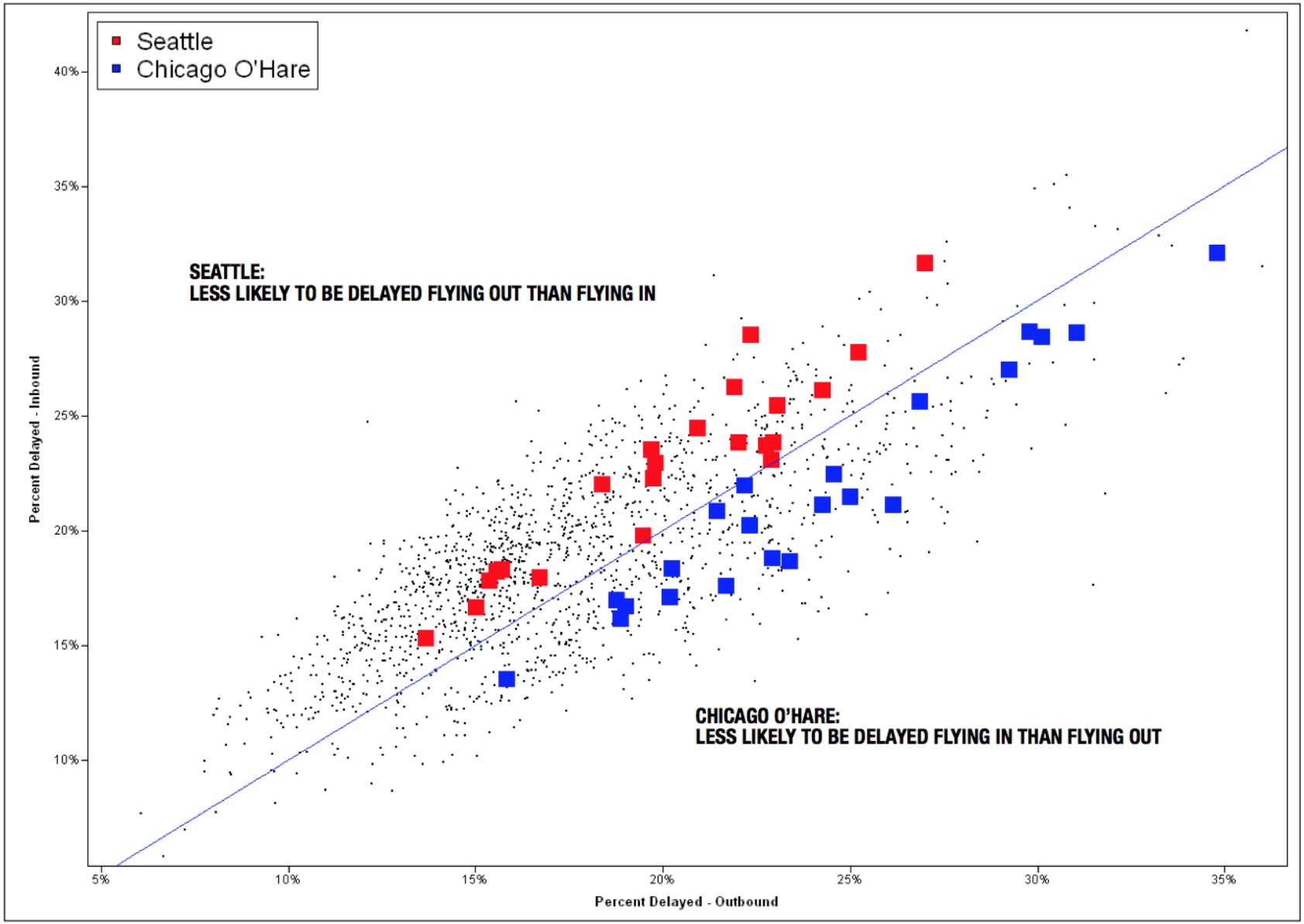
- dataset of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008 (but we now have through the begininning of 2015)
- large dataset: more than 180 million records
- aim: provide a graphical summary of important features of the data set
- winners presented at the JSM in 2009; details at `http://stat-computing.org/dataexpo/2009`

Airline Delays Codebook (abridged)

Year	1987-2015
Month	
DayOfMonth	
DayOfWeek	1=Monday
DepTime	departure time
UniqueCarrier	OH = Comair, DL = Delta, etc.
TailNum	plane tail number
ArrDelay	arrival delay, in minutes
Origin	BDL, BOS, SFO, etc.
Dest	

Full details at
`http://www.transtats.bts.gov/Fields.asp?Table_ID=236`

Data Expo 2009 winners



Data Expo 2009 winners

Ghosts of Flights

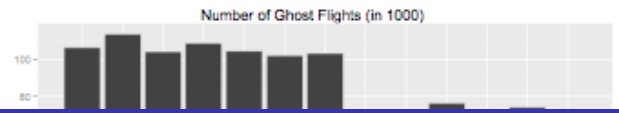
CAN WE SEE WHAT IS NOT THERE?

Planes have, for reasons such as maintenance, weather, or schedule fly empty between airports as so-called *Ghosts*. By tracking individual planes, we reveal their paths, including situations, where a plane lands in a different airport than where it takes off later, i.e. a ghost:

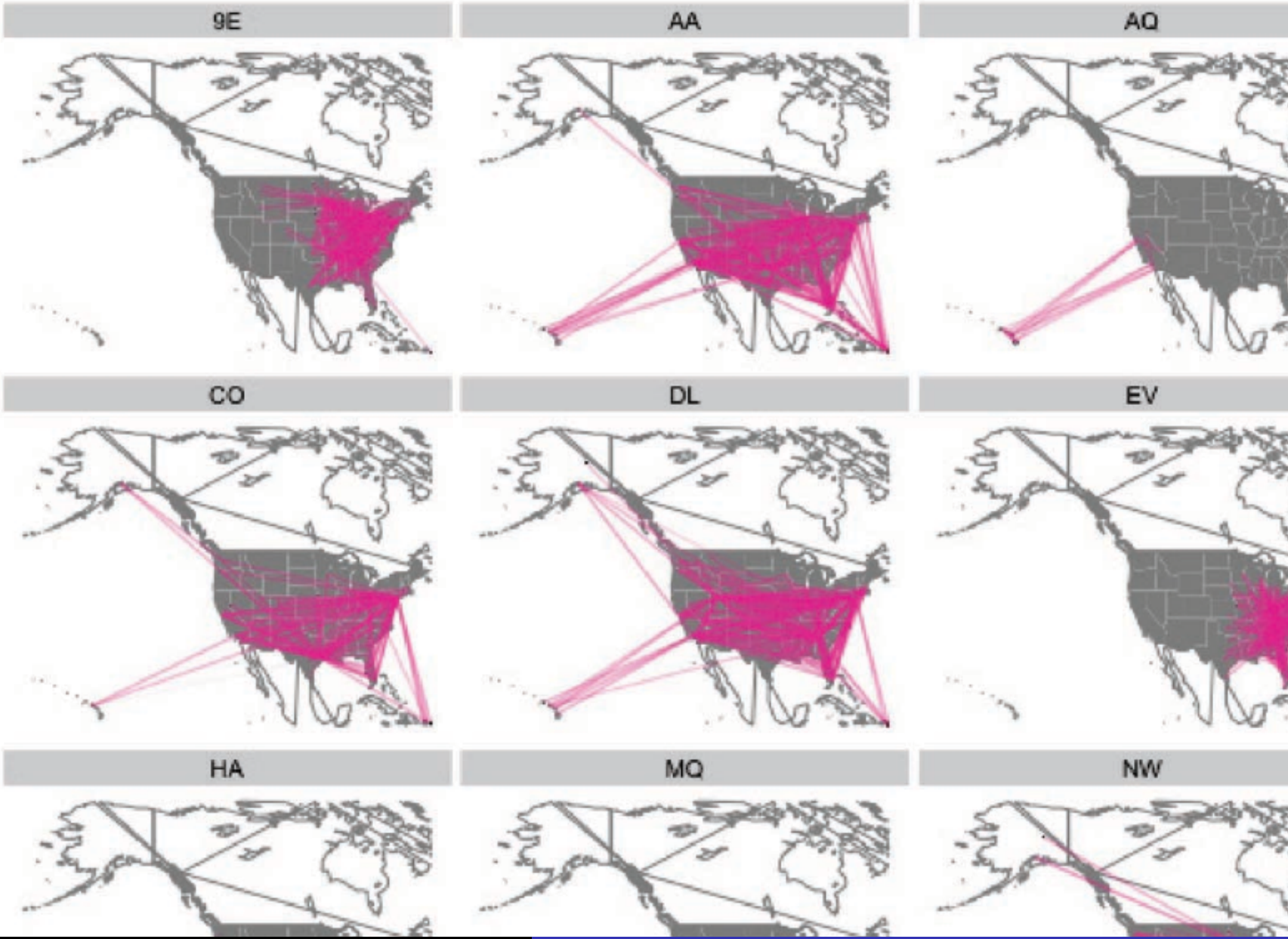
Example: US Airways Aircraft N-881 - Ghostflight from PIT to RIC (222 miles)

Year	Month	Day	DepTime	ArrTime	Origin	Dest	Diverted
1995	3	8	1102	1256	PIT	CVG	0
1995	3	8	1311	NA	CVG	PIT	1
1995	3	8	1913	2050	RIC	PIT	0
1995	3	8	2134	2300	PIT	MSY	0

Ghost Flight Totals: over 1 million flights since 1995, with an average distance between airports of 1000 miles, corresponding to about 1.5 million gallons of fuel. Since 2001 the number of ghost flights is cut, with major airlines also decreasing the fuel consumption. Smaller carriers are facing increasing costs

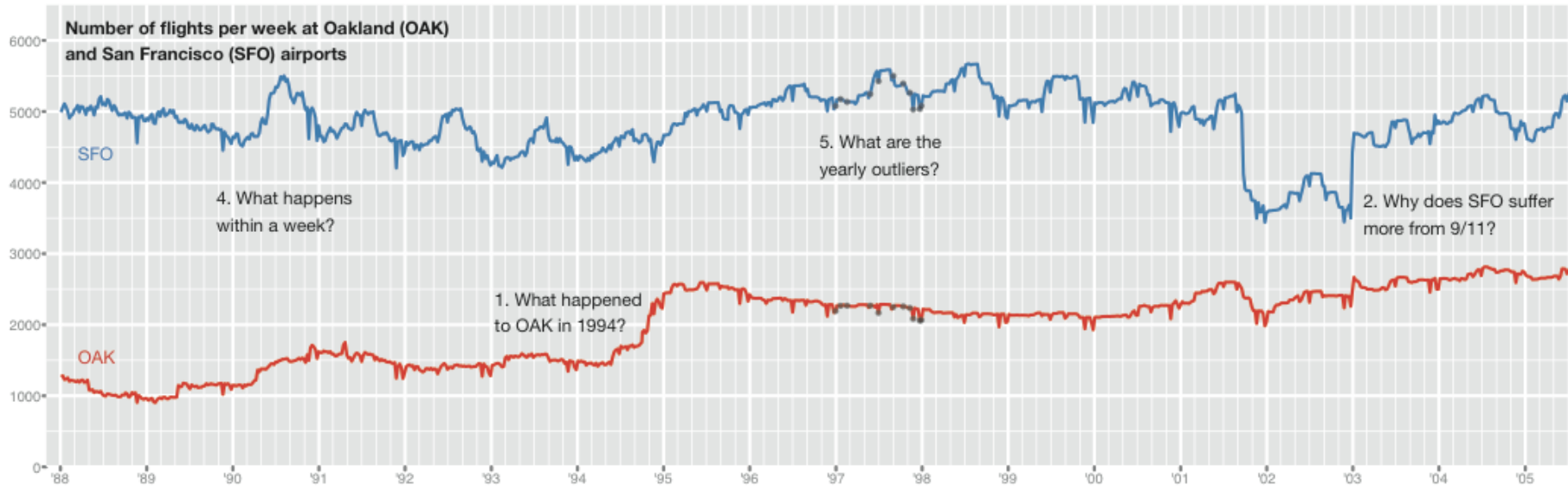


Data Expo 2009 winners

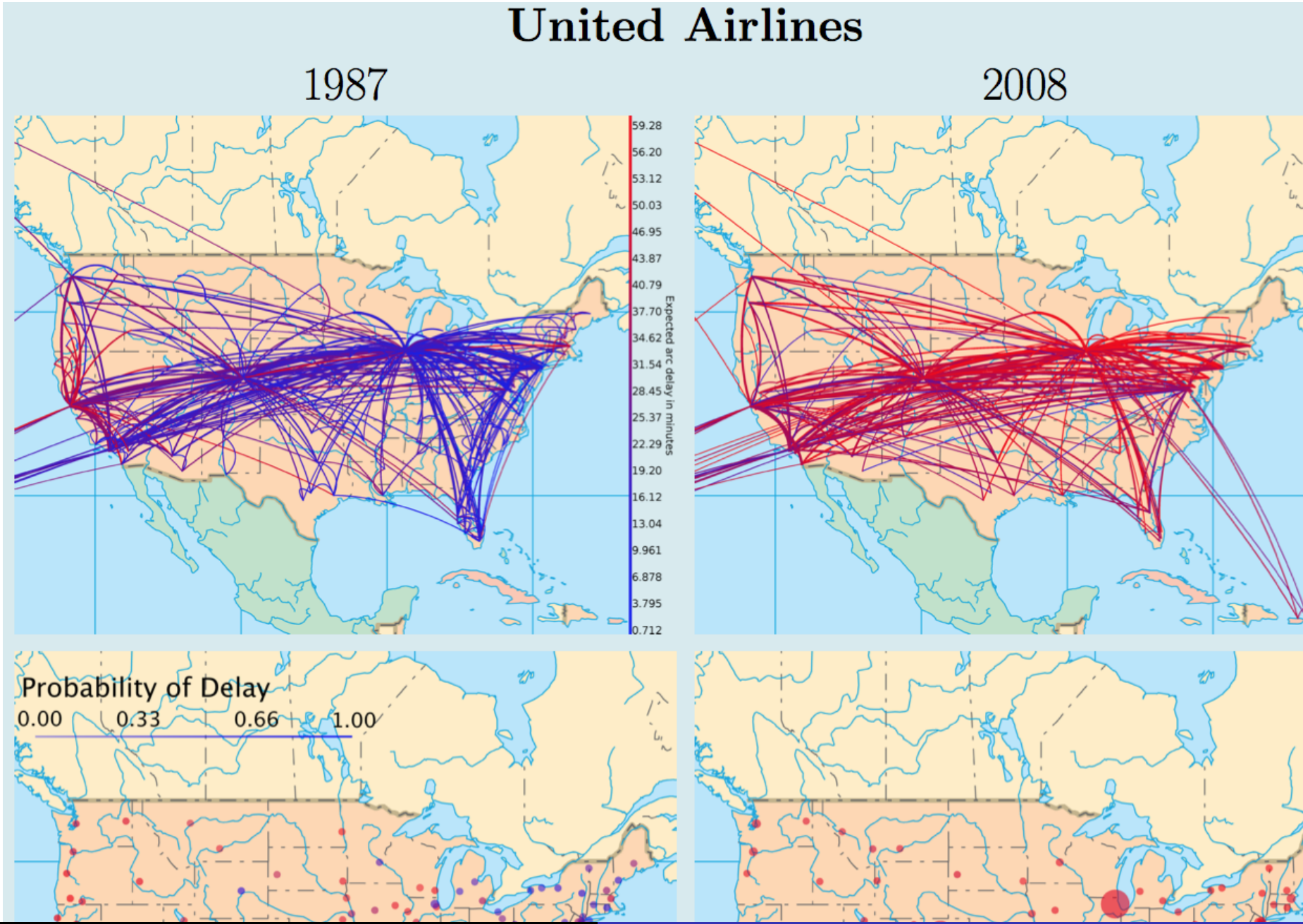


A Tale of Two Airports

AN EXPLORATION OF FLIGHT TRAFFIC AT OAK AND SFO



Data Expo 2009 winners



Goal

- facilitate use of database technology for instructors and students without training in this area
- demonstrate how to use SQLite in conjunction with `dplyr` in R to achieve this goal
- help enable the next generation of data scientists

Background on databases and SQL

- relational databases (invented in 1970)
- like electronic filing cabinets to organize masses of data (terabytes)
- fast and efficient
- useful reference: *Learning MySQL*, O'Reilly 2007
- Horton, Baumer, and Wickham (2015, in press), `http://www.amherst.edu/~nhorton/precursors`

Client and server model

- server: manages data
- client: ask server to do things
- use R as both the client and the server

SQL

- Structured Query Language
- special purpose programming language for managing data
- developed in early 1970's
- standardized (multiple times)
- most common operation is query (using `SELECT`)

Why SQLite?

advantage: free, quick, dirty, simple (runs locally), database can be copied from one system (Mac OS X) to Windows or Linux

disadvantage: not as robust, fast, or flexible than other free alternatives such as MySQL (which run remotely)

For personal use, SQLite is ideal (and is what I will demonstrate today).
For a class, I'd recommend MySQL.

Creating the airline delays database

1. download the data (30gb uncompressed)
2. load the data
3. add indices (to speed up access to the data, takes some time)
4. establish a connection (using `src_sqlite()`)
5. start to make selections (which will be returned as R objects) using `dplyr` package
6. features lazy evaluation (data only accessed when needed)

Key idioms for dealing with big(ger) data

select:	subset variables
filter:	subset rows
mutate:	add new columns
summarise:	reduce to a single row
arrange:	re-order the rows

Hadley Wickham, `bit.ly/bigrdata4`

Accessing the database (see test-sqlite.Rmd)

```
my_db <- src_sqlite(path="ontime.sqlite3")

# link to some useful tables
flights <- tbl(my_db, "flights")
airports <- tbl(my_db, "airports")
airlines <- tbl(my_db, "airlines")
airplanes <- tbl(my_db, "airplanes")
```

Accessing the database (see test-sqlite.Rmd)

```
> airports
Source: sqlite 3.8.6 [ontime.sqlite3]
From: airports [3,376 x 7]

   IATA                  Airport             City State
1   00M                 Thigpen      Bay Springs    MS
2   00R Livingston Municipal        Livingston    TX
3   00V             Meadow Lake Colorado Springs    CO
4   01G           Perry-Warsaw             Perry    NY
5   01J      Hilliard Airpark          Hilliard    FL
6   01M    Tishomingo County           Belmont    MS
..  ...                   ...               ...   ...
Variables not shown: Country (chr), Latitude (dbl),
  Longitude (dbl)
```

Accessing the database (see test-sqlite.Rmd)

```
airportcounts <- flights %>%
   filter(Dest %in% c('ALB', 'BDL', 'BTV')) %>%
   group_by(Year, DayOfMonth, DayOfWeek, Month, Dest) %>%
   summarise(count = n()) %>%
   collect()  # this forces the evaluation
airportcounts <- airportcounts %>%
  mutate(Date =
    ymd(paste(Year, "-", Month, "-", DayOfMonth, sep="")))
head(airportcounts)
```

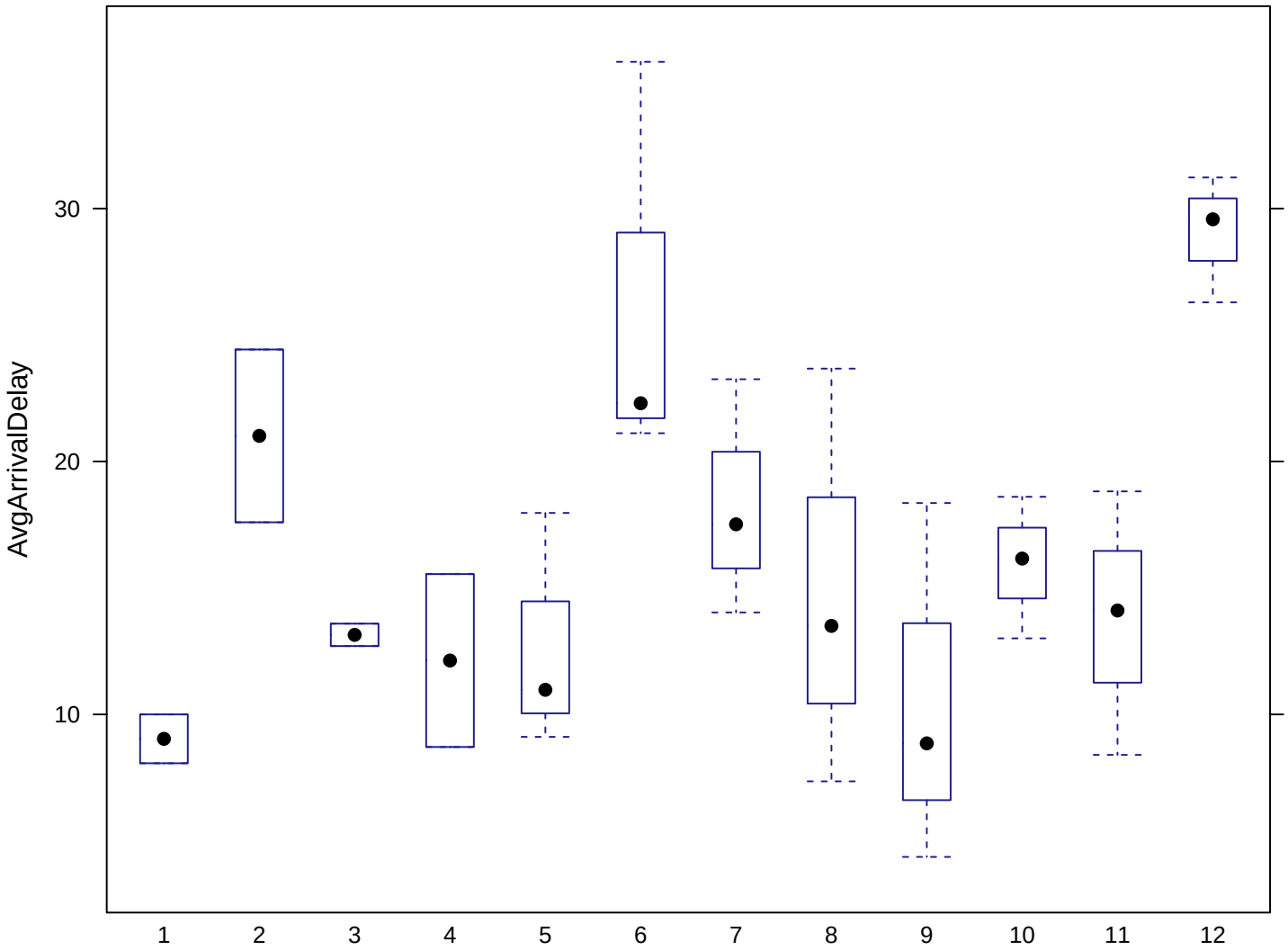
GROUP BY

```
## Source: local data frame [6 x 7]
## Groups: Year, DayOfMonth, DayOfWeek, Month
##
##   Year DayOfMonth Day Month Dest count Date
## 1 2014 1          3   1     ALB   17 2014-01-01
## 2 2014 1          3   1     BDL   52 2014-01-01
## 3 2014 1          3   1     BTV    6  2014-01-01
## 4 2014 2          4   1     ALB   24 2014-01-02
## 5 2014 2          4   1     BDL   60 2014-01-02
## 6 2014 2          4   1     BTV   11 2014-01-02
```

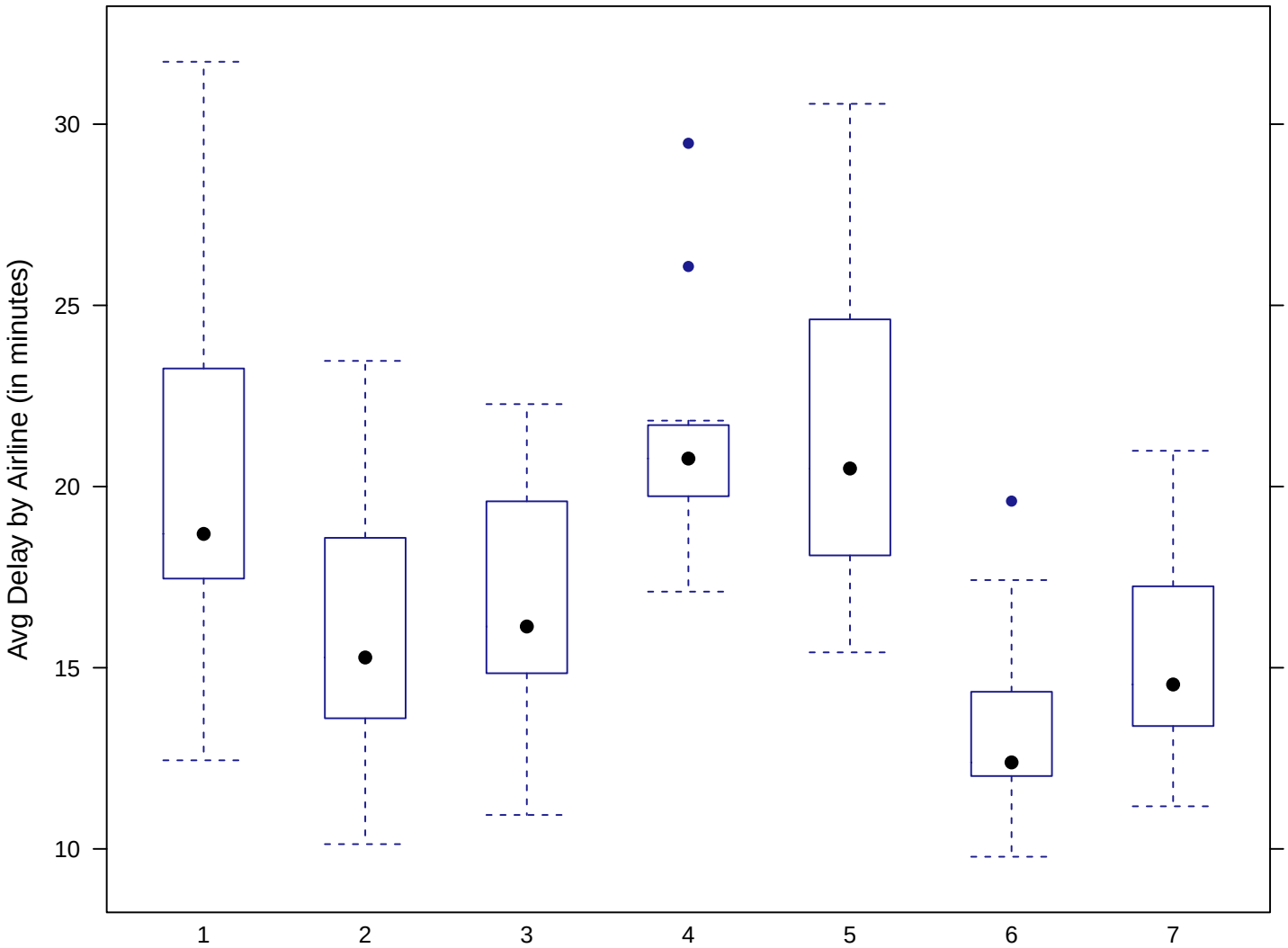
SQL commands

```
tbl(my_db, sql("SELECT * FROM flights LIMIT 4"))
# or tbl(my_db, "flights") %>% head()
## Source: sqlite 3.8.6 [ontime.sqlite3]
## From: <derived table> [?? x 29]
##
## Year Month DayOfMonth Day DepTime CRSDepTime ArrTime CRS
## 1 2014 1   1          3    914    900         1238 1225
## 2 2014 1   2          4    857    900         1226 1225
## 3 2014 1   3          5      0    900            0 1225
## 4 2014 1   4          6   1005    900         1324 1225
## Variables not shown: UniqueCarrier (chr), FlightNum (int
## ActualElapsedTime (int), CRSElapsedTime (int), AirTime
## (int), DepDelay (int), Origin (chr), Dest (chr), Distanc
## (int), TaxiOut (int), Cancelled (int), CancellationCode
## (chr), CarrierDelay (chr), WeatherDelay (chr), NASDelay
## SecurityDelay (chr), LateAircraftDelay (chr)
```

Which month is it best to travel (airline averages/BDL)?



Which day is it best to travel (airline averages from BDL)?



Setup on your own machine (data from 2014)

Update existing packages: `update.packages()`

Install necessary packages: RSQLite, dplyr, tidyr, mosaic, knitr, nycflights13, lubridate, igraph, markdown

Create a new project in RStudio and directory to store the data and data

Download files: see list at `http://www.amherst.edu/~nhorton/precursors`

Set up the database: run `load-sqlite.R`

Take it for a spin: run `test-sqlite.Rmd`

Next steps

Want more data? See the additional instructions to access more than just the data from 2014

See also: the dplyr package vignette on databases

See also: the precursors paper (in press) at `http://www.amherst.edu/~nhorton/precursors`

Accessing bigger datasets in R using SQLite and dplyr

Nicholas J. Horton

Amherst College, Amherst, MA, USA

March 24, 2015

nhorton@amherst.edu