The joy of Clean Data!

San Francisco Big Data Science Meetup

15 Dec 2015

Matt Dowle

H_2O.ai
Machine Intelligence

Overview

- For beginners
- Examples from my background
- Tools along the way
- Live demo of “80% munging”
- How H2O fits in
- Q & A

1996 – Lehman Brothers

- Just graduated – Applied Maths & Computing
- Dividend claims
- Cleaning at source; e.g. data entry typos
- Estimate cash flows, alerts etc
- Nothing fancy
- Tools: VB & Sybase
- How I accidentally created messy data

1999 - Salomon Brothers

- Equity risk model
  - Multiple time series regression (10 year)
  - DEM proxy for EUR prior to 1 Jan 1999
  - IPOs get their sector's median; e.g. France Telecom
  - Abbey National X000445**5**
  - 90% of the lines of code was not the regression

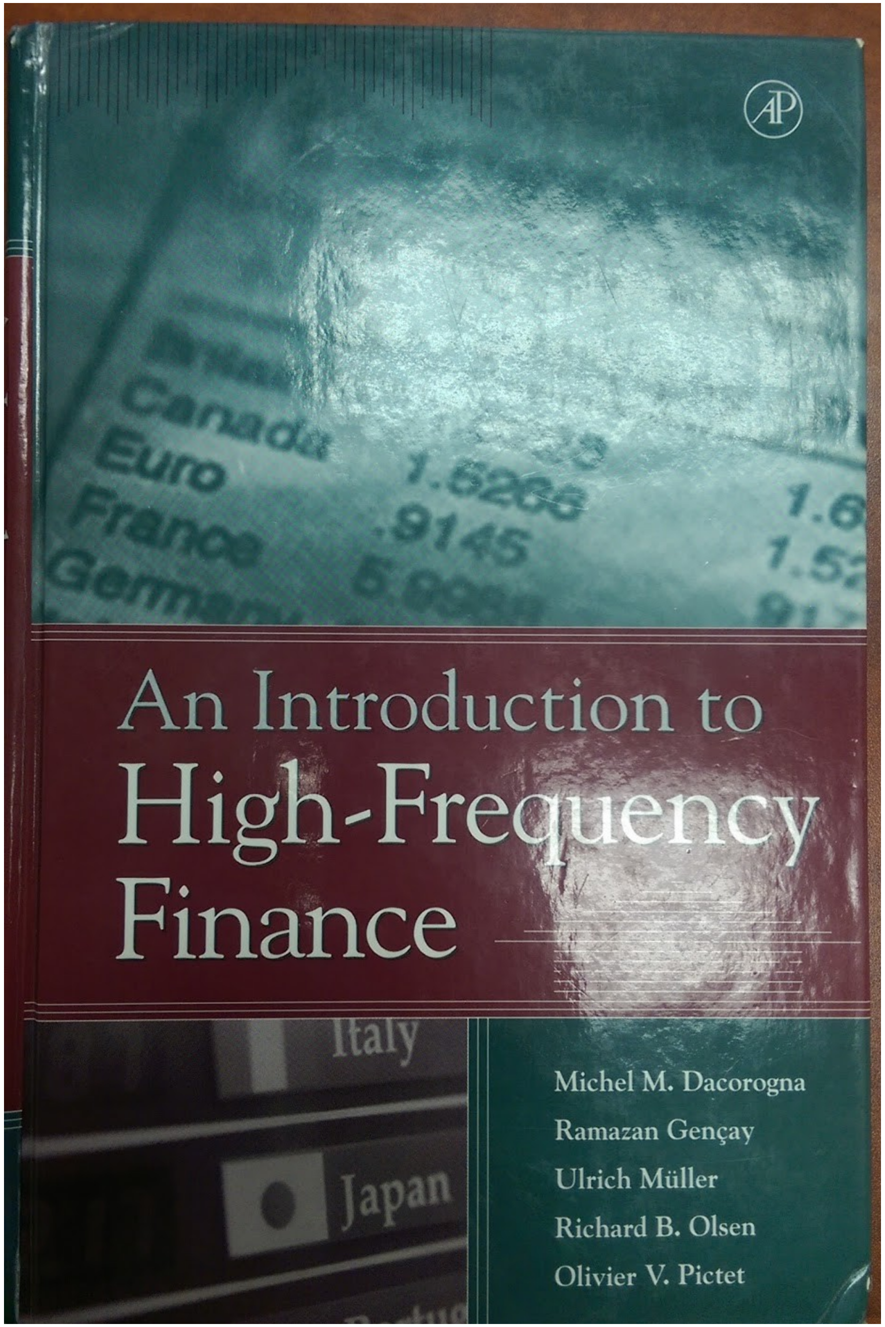
2002 - Citigroup

- Pairs Trading
- 200 most liquid stocks
- 200 x 199 / 2 = 19,900 pairs
- Stock splits, id changes
- Dickey Fuller test for stationarity
- Bollinger bands => buy/sell signal
- Excel spreadsheet to clients with embedded S-PLUS plot, daily, 50 custom variants
- Rebalance => orphan & surrogate pairs

2004
moved to fund management

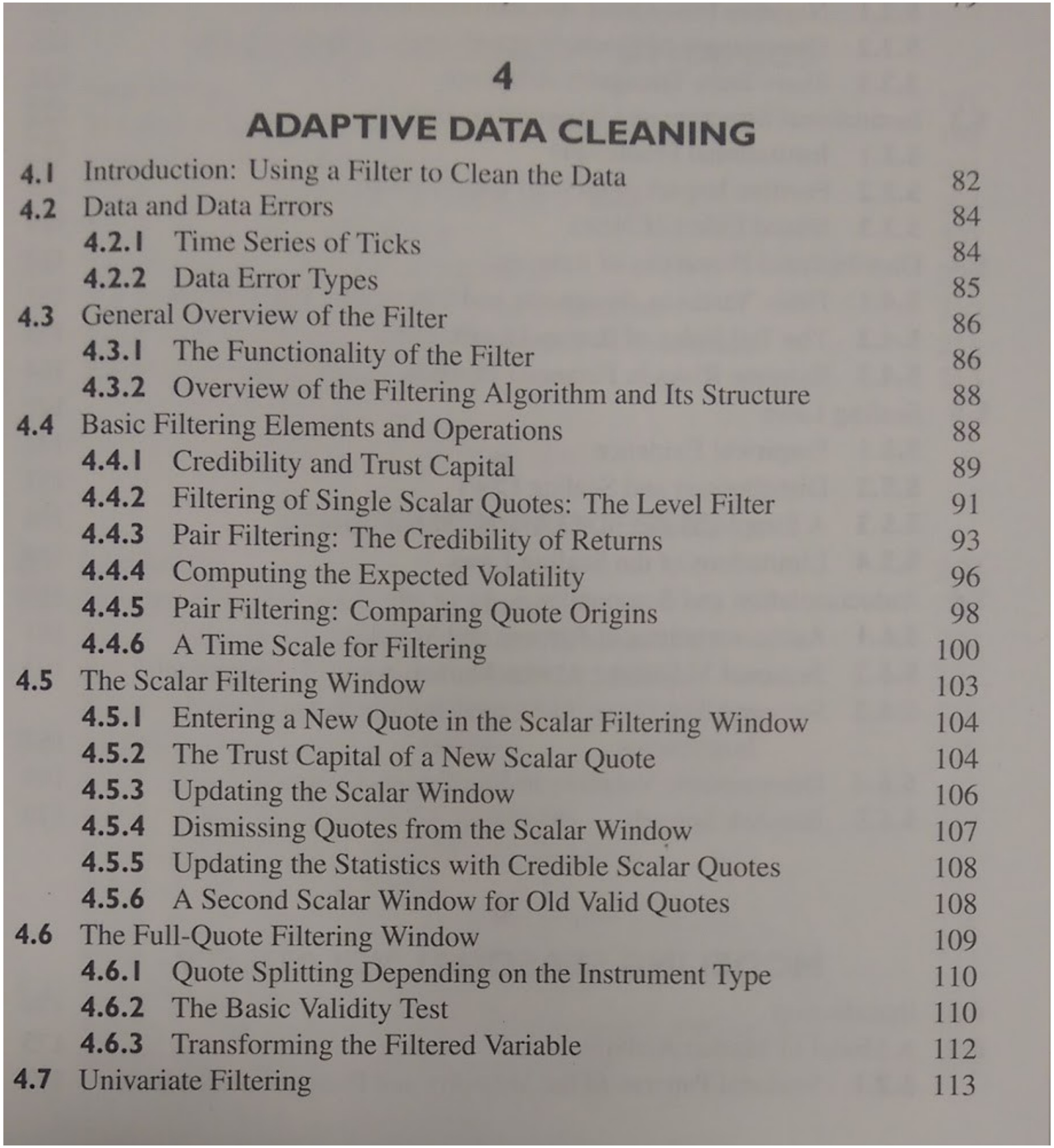
Bigger data

e.g. 25TB



An Introduction to High-Frequency Finance

Michel M. Dacorogna
Ramazan Gençay
Ulrich Müller
Richard B. Olsen
Olivier V. Pictet



4

ADAPTIVE DATA CLEANING

Over cleaning

1. I queried for intra-day auctions

   ```
   select from quote where bid>ask
   ```

   No results; i.e. all bid<ask.

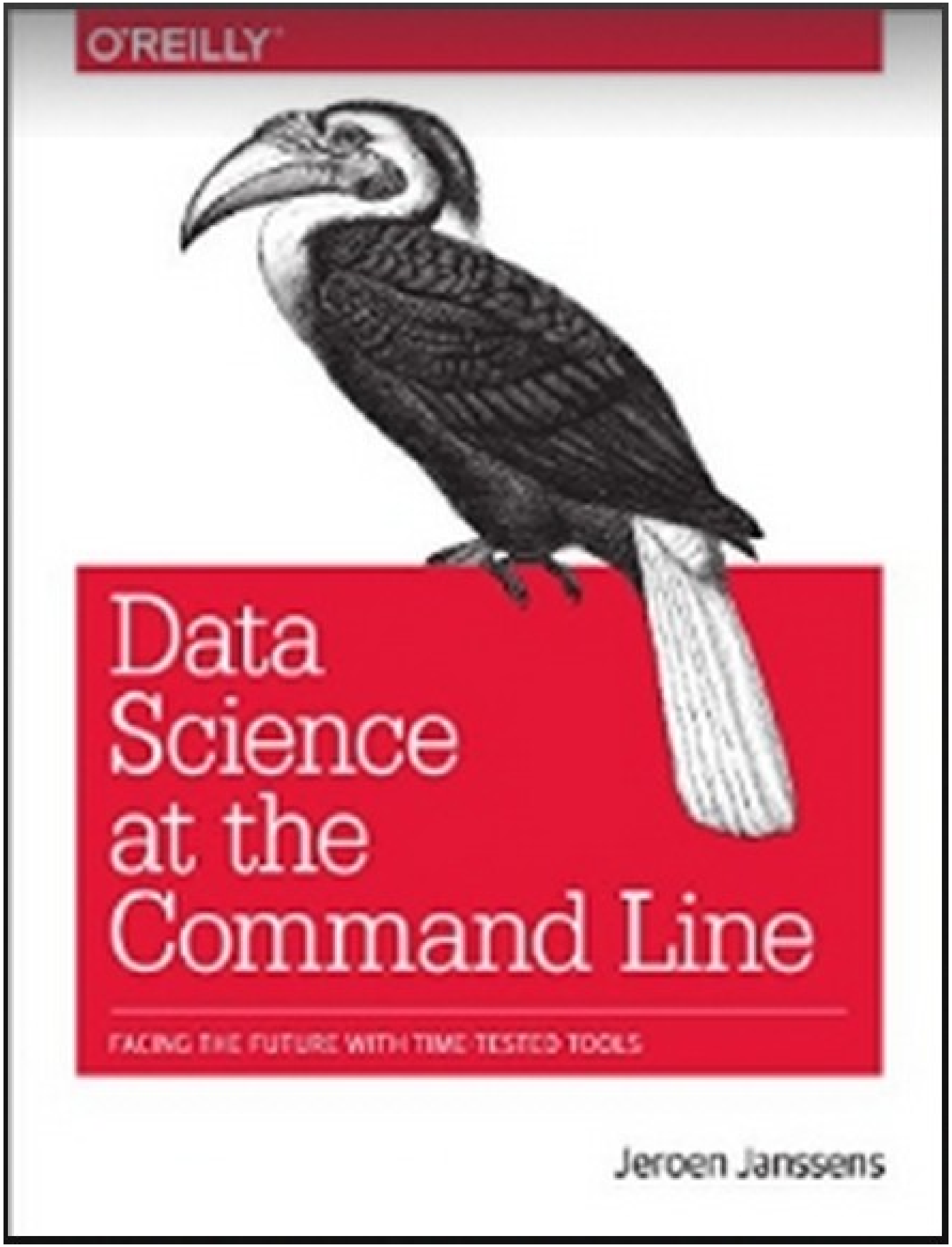
   Asked data provider

   Grrrrr

2. Negative prices can be correct

Tools

KDB	http://kx.com/ @kxsystems
OneTick	https://www.onetick.com/ @OneMarketData



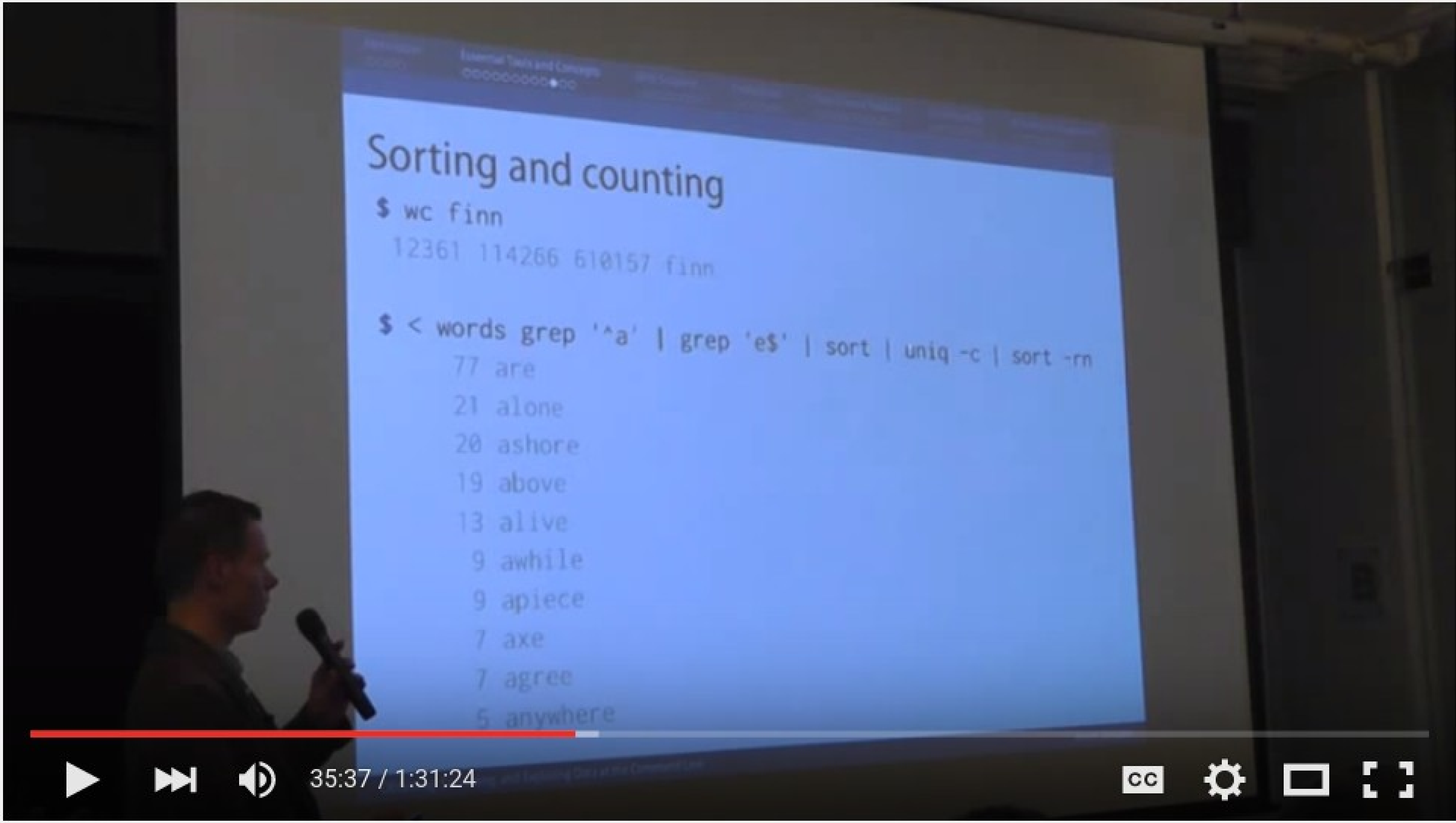
O'REILLY

Data Science at the Command Line

FACING THE FUTURE WITH TIME-TESTED TOOLS

Jeroen Janssens

https://www.youtube.com/watch?v=QxpOKbv-KQU



Sorting and counting

```
$ wc finn
 12361 114266 610157 finn

$ < words grep '^a' | grep 'e$' | sort | uniq -c | sort -rn
     77 are
     21 alone
     20 ashore
     19 above
     13 alive
      9 awhile
      9 apiece
      7 axe
      7 agree
      5 anywhere
```

alias

awk

aws

bc

bigmler

body

cat

cols

csvcut

csvgrep

csvjoin

csvlook

csvsort

csvsql

csvstack

csvstat

curl

cut

dseq

find

for

grep

head

header

in2csv

jq

json2csv

less

parallel

paste

pbc

python, R and r

Rio

Rio-scatter

run_experiment

sample

scrape

sed

seq

shuf

sort

split

sql2csv

tail

tapkee

tee

tr

tree

uniq

wc

weka

xml2json

- Can be faster than loading the whole file into R or Python
- Can be faster workflow
- Pre-processing before loading into R or Python

tidyr by Hadley Wickham

https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html

- Untidy data defined as :
  - Column headers are values, not variable names.
  - Multiple variables are stored in one column.
  - Variables are stored in both rows and columns.
  - Multiple types of observational units stored in the same table.
  - A single observational unit is stored in multiple tables.
- Solves by: gathering, separating and spreading
- That's the *shape* of the data. Yes, good, but not the kind of messy data I'm talking about in this presentation.

To illustrate

- In June 2013, RStudio made available download logs from their CRAN mirror http://blog.rstudio.org/2013/06/10/rstudio-cran-mirror/
- R-Bloggers search “CRAN download stats” 154 results; e.g.

  http://www.r-bloggers.com/finally-tracking-cran-packages-downloads/

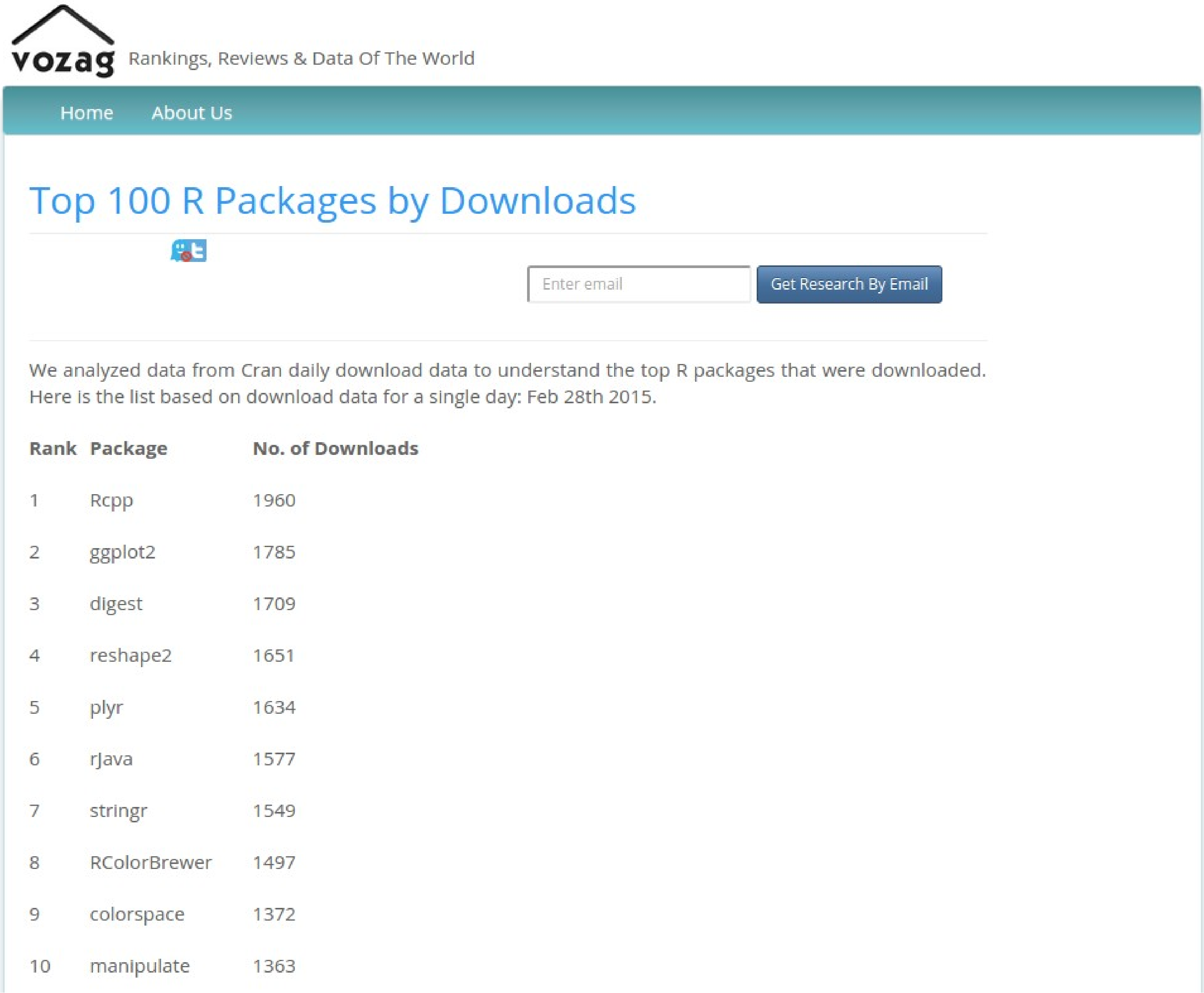
  https://github.com/metacran/cranlogs

  http://www.r-bloggers.com/working-with-the-rstudio-cran-logs/

  http://www.r-bloggers.com/cran-download-statistics-of-any-packages-rstats/

  http://www.r-bloggers.com/my-r-packages-worldmap-of-downloads/

March 2015

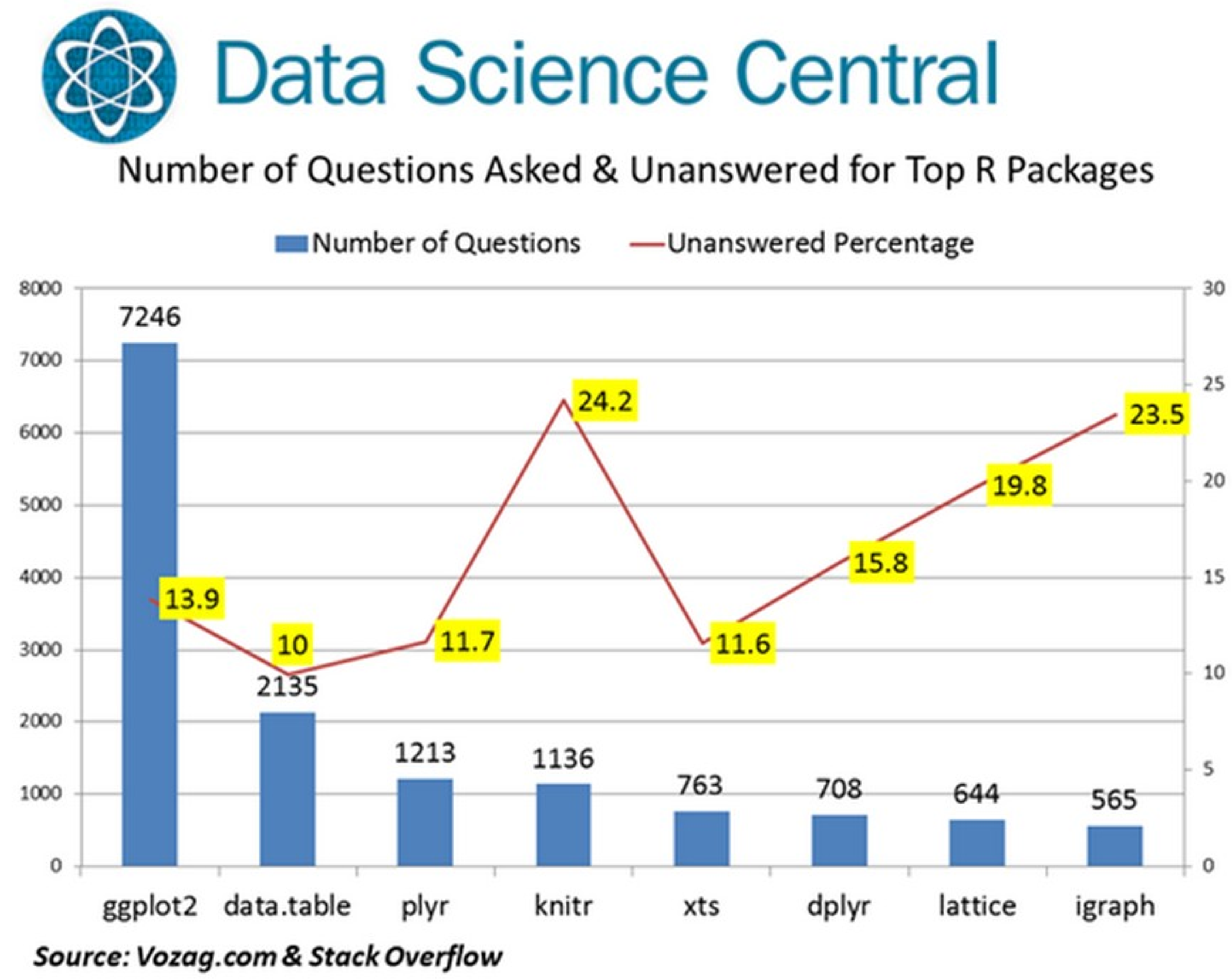


Top 100 R Packages by Downloads

We analyzed data from Cran daily download data to understand the top R packages that were downloaded. Here is the list based on download data for a single day: Feb 28th 2015.

Rank	Package	No. of Downloads
1	Rcpp	1960
2	ggplot2	1785
3	digest	1709
4	reshape2	1651
5	plyr	1634
6	rJava	1577
7	stringr	1549
8	RColorBrewer	1497
9	colorspace	1372
10	manipulate	1363

April
2015



Data Science Central

Number of Questions Asked & Unanswered for Top R Packages

Source: Vozag.com & Stack Overflow

Data Science Central

- Comparing the top downloaded packages with the most discussed packages shows little correlations between them. For Instance, ggplot2 has the most questions asked & is the second highest downloaded package **but data.table package (the second highest ranked R package for questions asked) is not even in the top 100 packages downloaded.** Knitr is another example which is in the top 5 questions asked, but is 27th ranked in downloaded packages.
- **So- does the R community need to focus on packages that have the highest questions to resolve their issues rather than the ones with the most downloads?**

Let's look at the data!

Live demo of munging

Observations and comments on meetup video recording

https://youtu.be/4VWQEvYIfV8

(~ 22 mins in)

“Big data”

1. **Data > 240GB**

   needle-in-haystack e.g. fraud

2. **Data < 240GB**

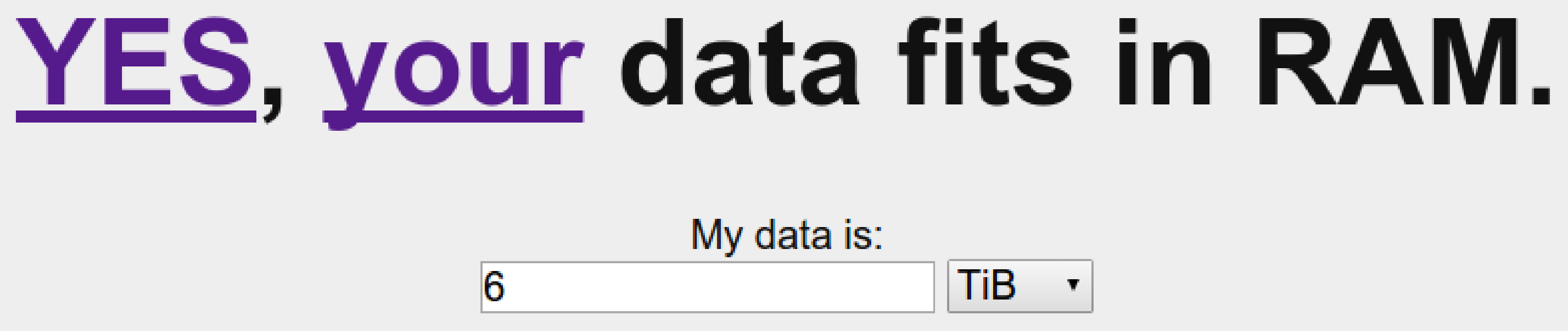
   compute intensive, parallel 100's cores

3. **Data < 240GB**

   feature engineering > 240GB

Speed for i) production and ii) interaction

NB: 240GB is currently largest available on EC2 http://yourdatafitsinram.com/



YES, your data fits in RAM.

My data is:

6 TiB ▾

Dell PowerEdge R920 60 core

(4 * Intel® Xeon® E7-8880L 2.2GHz, 37.5M Cache, 15 Core)

with 1.5TB RAM $60k (96 * 16GB)

with 6TB $150k-$200k? (96 * 64GB)

But, still “only” 60 cores

In the office here we already have 2.5TB RAM and 320 cores on 10 machines.

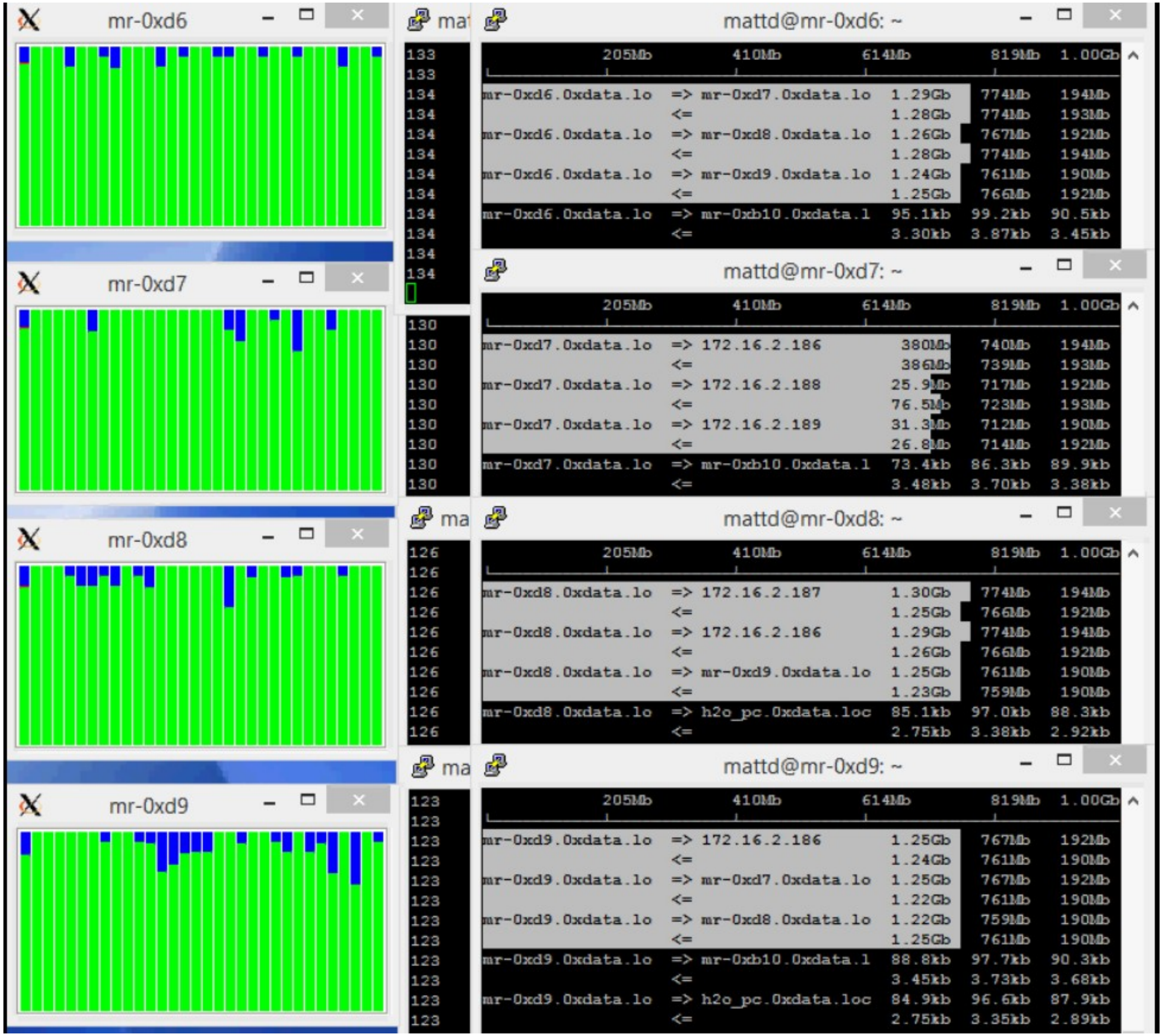
So do many businesses.

- data.table's radix join
- Now parallel and distributed
- e.g. high cardinality 1bn/1bn/1bn row join

data.table	10 min	
H2O 1 node 32 core	3.5 min	
H2O 4 node 128 core	1.5 min	**=> demo**
H2O 10 node 320 core	2.0 min	

- Known improvements to be made

https://www.youtube.com/watch?v=8VpzNibOme0



```
133
133
134
134
134
134
134
134
134
134
134
134
```

```
              205Mb        410Mb        614Mb         819Mb   1.00Gb
mr-0xd6.0xdata.lo  => mr-0xd7.0xdata.lo  1.29Gb   774Mb   194Mb
                   <=                    1.28Gb   774Mb   193Mb
mr-0xd6.0xdata.lo  => mr-0xd8.0xdata.lo  1.26Gb   767Mb   192Mb
                   <=                    1.28Gb   774Mb   194Mb
mr-0xd6.0xdata.lo  => mr-0xd9.0xdata.lo  1.24Gb   761Mb   190Mb
                   <=                    1.25Gb   766Mb   192Mb
mr-0xd6.0xdata.lo  => mr-0xb10.0xdata.l  95.1kb   99.2kb  90.5kb
                   <=                    3.30kb   3.87kb  3.45kb
```

```
130
130
130
130
130
130
130
130
130
```

```
              205Mb        410Mb        614Mb         819Mb   1.00Gb
mr-0xd7.0xdata.lo  => 172.16.2.186        380Mb   740Mb   194Mb
                   <=                     386Mb   739Mb   193Mb
mr-0xd7.0xdata.lo  => 172.16.2.188       25.9Mb   717Mb   192Mb
                   <=                    76.5Mb   723Mb   193Mb
mr-0xd7.0xdata.lo  => 172.16.2.189       31.3Mb   712Mb   190Mb
                   <=                    26.8Mb   714Mb   192Mb
mr-0xd7.0xdata.lo  => mr-0xb10.0xdata.l  73.4kb   86.3kb  89.9kb
                   <=                    3.48kb   3.70kb  3.38kb
```

```
126
126
126
126
126
126
126
126
126
126
```

```
              205Mb        410Mb        614Mb         819Mb   1.00Gb
mr-0xd8.0xdata.lo  => 172.16.2.187       1.30Gb   774Mb   194Mb
                   <=                    1.25Gb   766Mb   192Mb
mr-0xd8.0xdata.lo  => 172.16.2.186       1.29Gb   774Mb   194Mb
                   <=                    1.26Gb   766Mb   192Mb
mr-0xd8.0xdata.lo  => mr-0xd9.0xdata.lo  1.25Gb   761Mb   190Mb
                   <=                    1.23Gb   759Mb   190Mb
mr-0xd8.0xdata.lo  => h2o_pc.0xdata.loc  85.1kb   97.0kb  88.3kb
                   <=                    2.75kb   3.38kb  2.92kb
```

```
123
123
123
123
123
123
123
123
123
123
123
123
```

```
              205Mb        410Mb        614Mb         819Mb   1.00Gb
mr-0xd9.0xdata.lo  => 172.16.2.186       1.25Gb   767Mb   192Mb
                   <=                    1.24Gb   761Mb   190Mb
mr-0xd9.0xdata.lo  => mr-0xd7.0xdata.lo  1.25Gb   767Mb   192Mb
                   <=                    1.22Gb   761Mb   190Mb
mr-0xd9.0xdata.lo  => mr-0xd8.0xdata.lo  1.22Gb   759Mb   190Mb
                   <=                    1.25Gb   761Mb   190Mb
mr-0xd9.0xdata.lo  => mr-0xb10.0xdata.l  88.8kb   97.7kb  90.3kb
                   <=                    3.45kb   3.73kb  3.68kb
mr-0xd9.0xdata.lo  => h2o_pc.0xdata.loc  84.9kb   96.6kb  87.9kb
                   <=                    2.75kb   3.35kb  2.89kb
```

Thank you.

Q & A