Centrum Wiskunde & Informatica

COMMIT/

Capturing the Data Laws of Nature

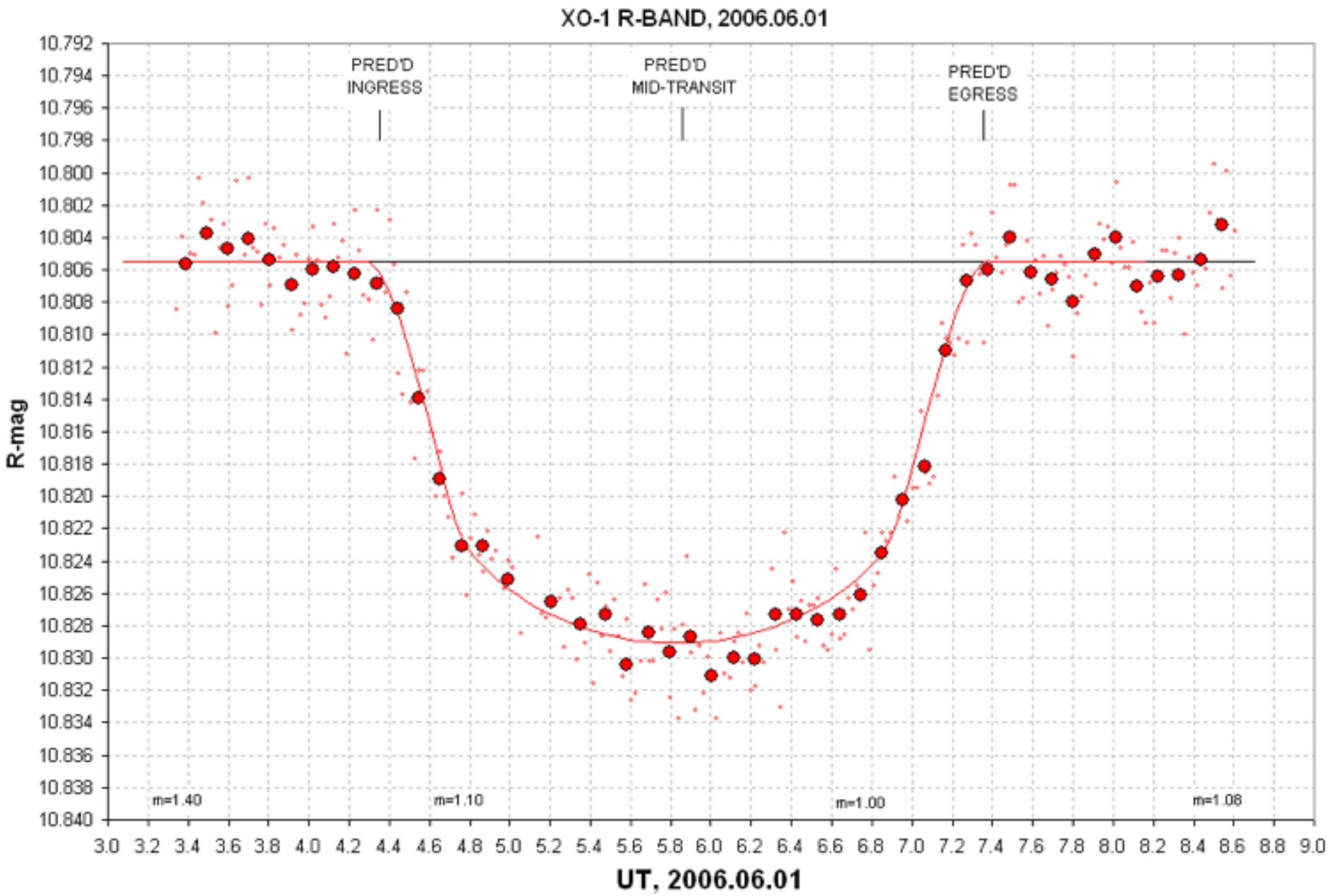
Hannes Mühleisen, Stefan Manegold & Martin Kersten [DA]

CWI Scientific Meeting, 2014-06-05

“Essentially, all models are wrong, but some are useful.”

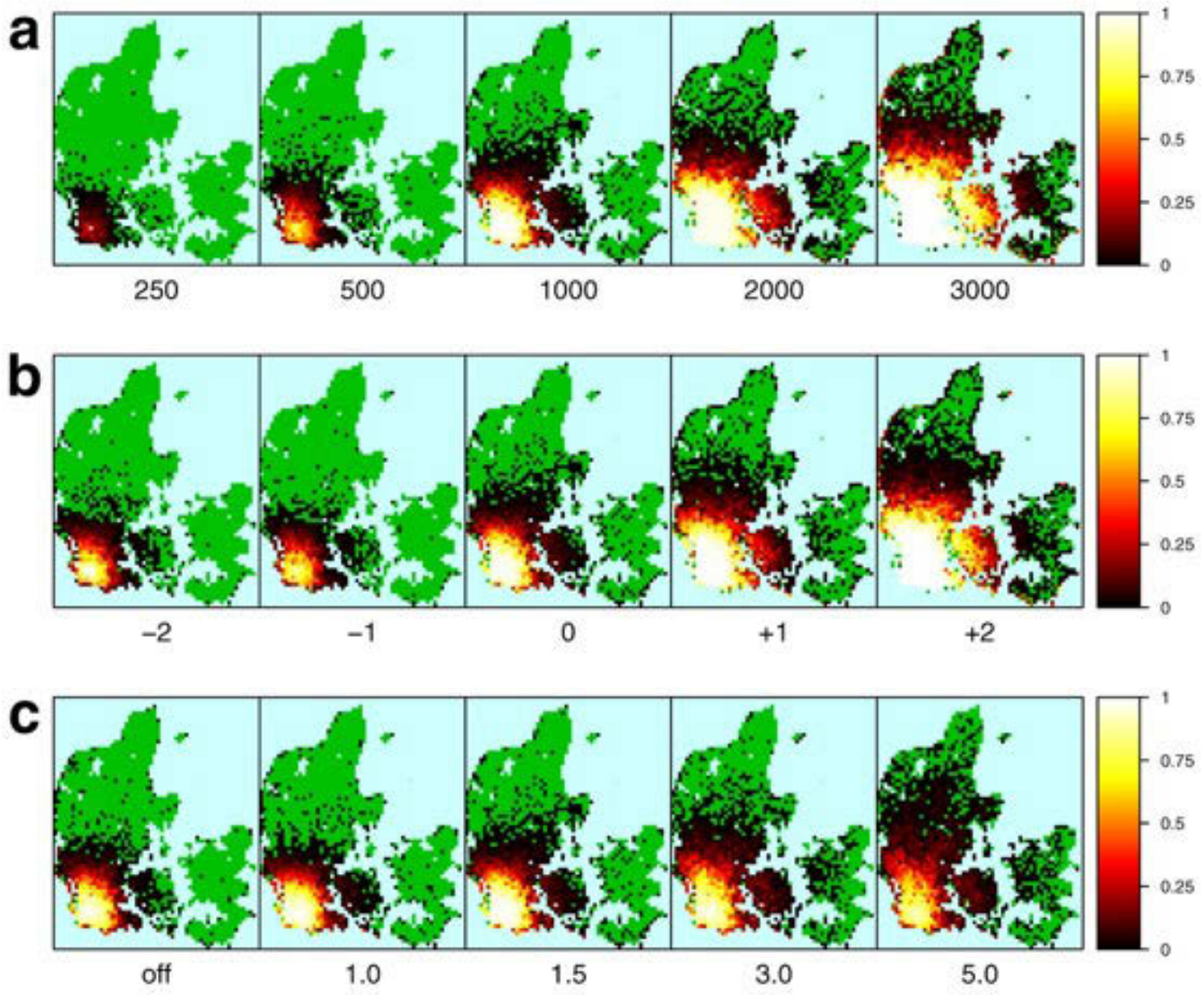
George E. P. Box

Astronomy…



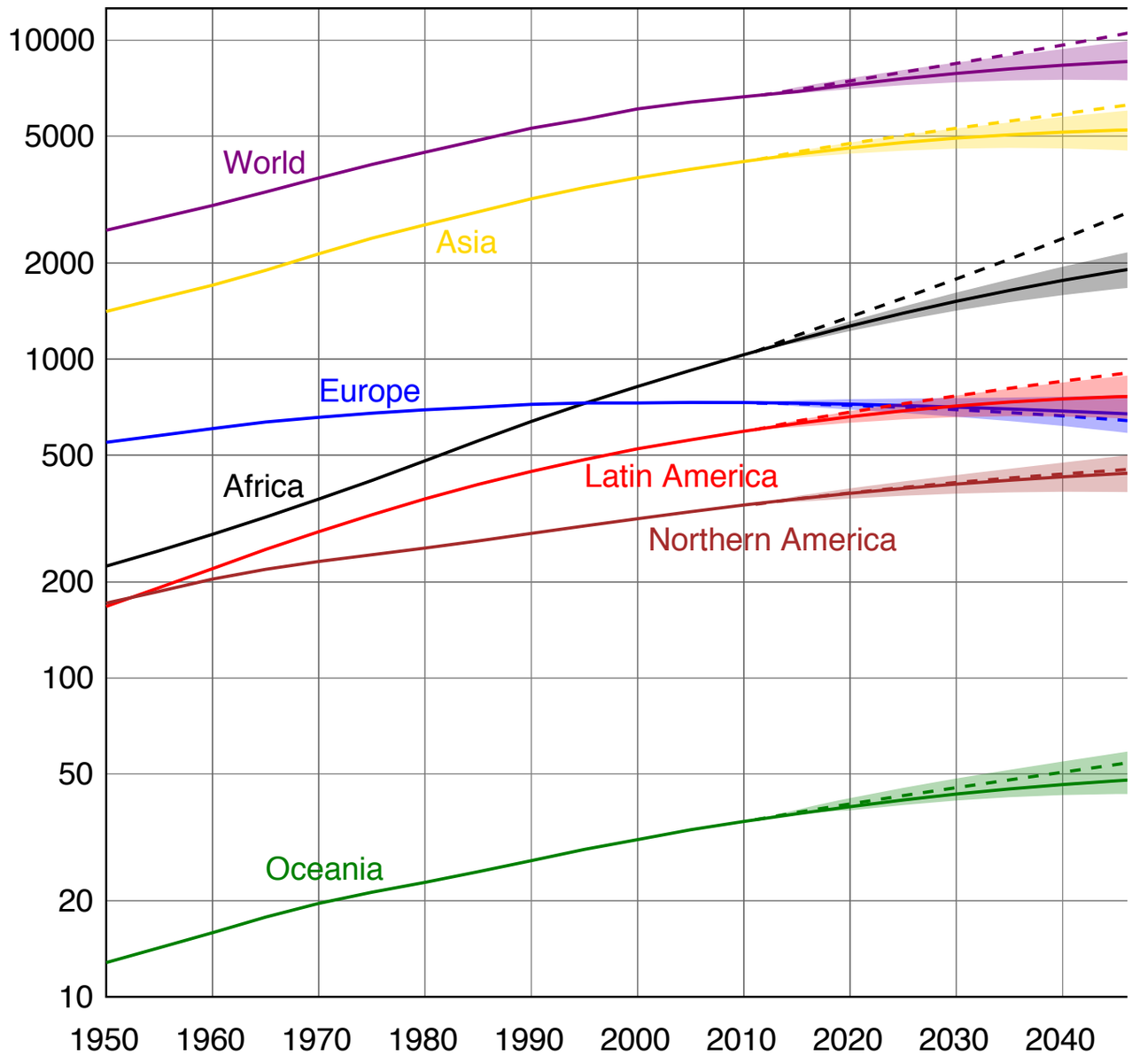
Exoplanet light curve

Epidemology…



Bluetonge epidemy in Denmark

Demographics…



Population development

Generic Process?

Collect observations

Training Set

Validation Set

Hypothesis/ Model

Trained Model

Validated Model

Publish paper/ Profit

The point?

- Everyone has models, **they encode our understanding of the world**
- Everyone has data to train and validate a model
- So far, data management community has **ignored** these models
    - But they hold precious domain knowledge!

User gave me a model, let's see.

I am storing some data.

I need some of the observations to fit the model.

This other guy is reading some of my data.

Cool, the model seems to fit the data well!

Let's get some more data to validate the fit…

This other guy is reading some more of my data.

Amazing, model fit is validated!

To Polder!

I am storing some data.

Integrate & Intercept

- Integrate model fitting infrastructure into data management system.
  - Also: **Huge** performance benefits!
- Intercept model fitting and validation operations by the user and store the model for later use.
  - Storage format: Model code + Parameters

Example: Ohm’s Law

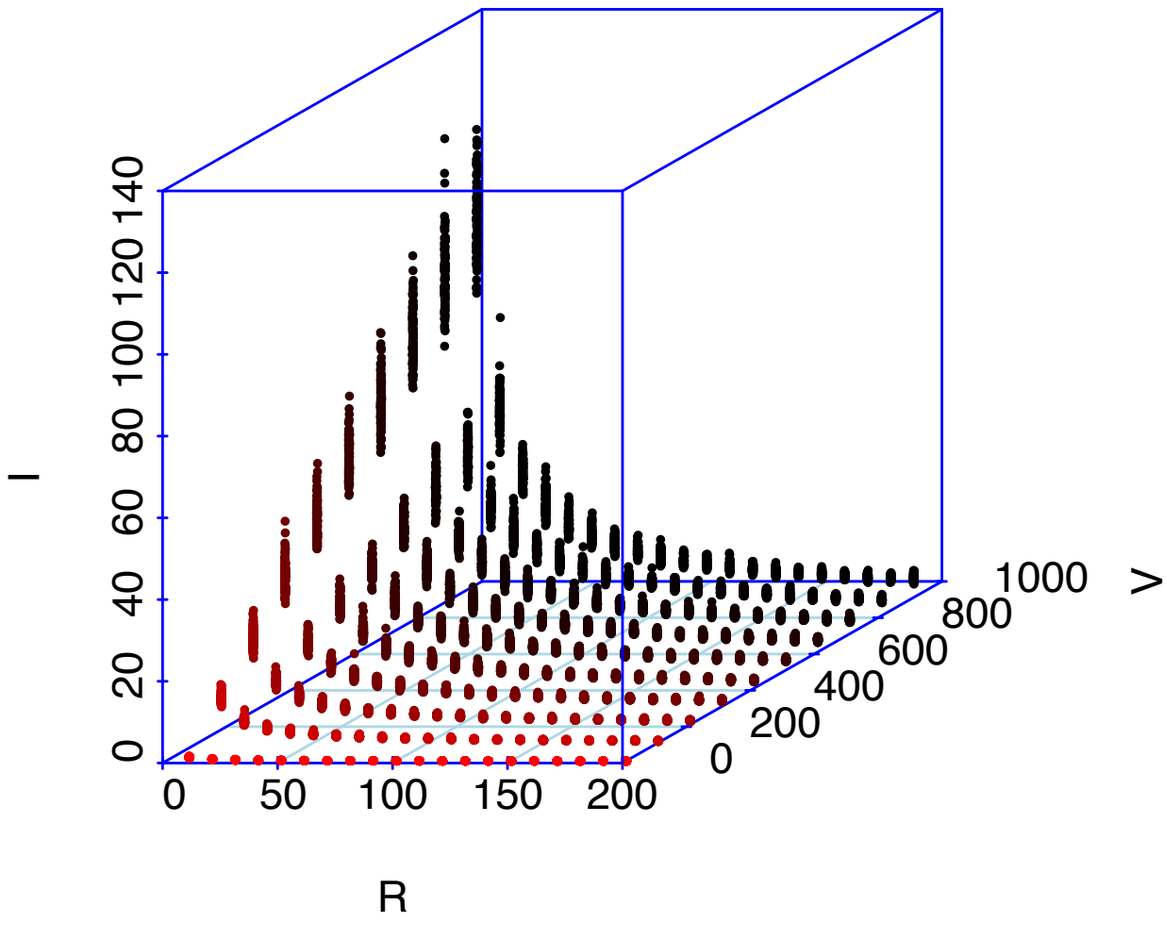
	ohm	volt	current
1	10	10	1.04569893
2	20	10	0.51646973
3	30	10	0.34002053
4	40	10	0.24729716
5	50	10	0.23197141
6	60	10	0.17356846
7	70	10	0.15761114
8	80	10	0.14021253
9	90	10	0.11309898
10	100	10	0.09291330
11	110	10	0.08632890
12	120	10	0.08286361
13	130	10	0.07358118
14	140	10	0.07054635
15	150	10	0.06108650
16	160	10	0.06746850
17	170	10	0.06582711
18	180	10	0.05333042
19	190	10	0.05301234
20	200	10	0.04816665
21	10	110	10.27267801
22	20	110	5.53502417
23	30	110	5.10863069
24	40	110	3.21255743
25	50	110	2.25928635
26	60	110	2.13666999
27	70	110	1.87122111
28	80	110	1.32818943
29	90	110	1.37449274
30	100	110	1.21708974
31	110	110	0.93508147
32	120	110	0.94363256
33	130	110	0.76718314
34	140	110	0.67562132
35	150	110	0.70524426
36	160	110	0.61914850
37	170	110	0.57320802
38	180	110	0.57169049
39	190	110	0.53375340
40	200	110	0.55153969

Parameters Observations

```
> fit <- lm(current ~ volt/ohm, data=measurements)
> summary(fit)

Residuals:
    Min      1Q  Median      3Q     Max
-15.401  -4.170  -0.644   1.653  79.410

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1.066e-02  1.144e-01    0.093    0.926
volt         4.647e-02  2.930e-04  158.610   <2e-16 ***
volt:ohm    -2.697e-04  1.937e-06 -139.267   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.566 on 19997 degrees of freedom
Multiple R-squared:  0.5727,   Adjusted R-squared:  0.5726
F-statistic: 1.34e+04 on 2 and 19997 DF,  p-value: < 2.2e-16
```

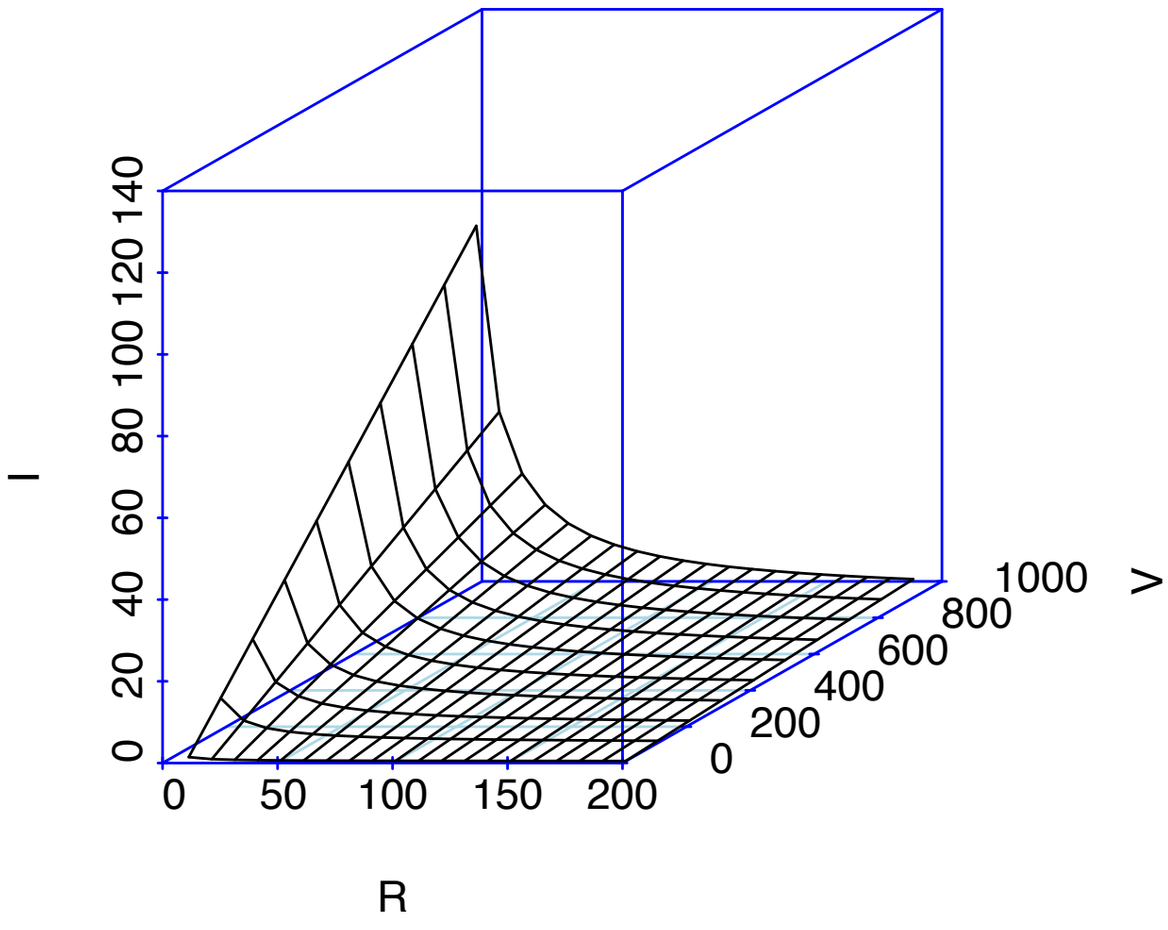
I=V/R...

Measurements



n=20000

Fitted Model



n=0

```
SELECT I FROM measurements WHERE R = 100 AND V = 400;
```

Storage Layer

Compression

- “True” semantic compression
  - General-purpose compression algorithms have to be very generic.
  - But we have a model, so we could just use that!
  - Store only model + deltas!
- Less storage space, faster access

Zero-IO Scans

- For approximate queries, we can ignore data and scan the model parameter space instead
    - Requires enumerable parameters
- Turns a IO-bound problem into a CPU-bound problem!
    - Reading from disk vs. recalculating values from model

Data & Model Changes

- What should we do if the user gives us a better model?
  - Recompressing could be very expensive
  - Threshold for improvement?
- Changes in the data affect the model quality, too
  - Switch models?
  - Constant Monitoring?

Multiple, partial or grouped

- There could be many models for a table with overlapping parameters
  - Which one to pick?
- Models do not have to cover the entire table/column
  - “Patching”?
- Models could be fitted on aggregation results
  - Can we still use them?

Approximate Queries

Analysis of Linear Models

- If we have a linar model, we can use analytical methods to calculate query results

  - min, max, sum, avg, …

Model Exploration

- Find interesting subsets of the data through gradient analysis etc.

Parameter Space

- What do we do if parameters are not specified in the query?
  - Even integer parameters have an infinite number of values
  - Ask user to restrict?
  - Scan the data to find boundaries?

Parameter Closure

- Given multiple parameters, it is far from certain that all combinations of values are allowed in the model.
  - Might crash, might be giving ridiculous results…
- How to find only legal values?
  - Scan data and generate legal parameter space shape?
    - Might be expensive to store…

Wrapup

- Make models a first-class citizen in data management
- Exploit models for storage optimization and approximate query answering
- Work in progress, many open questions!
- However
  - Watch out for MonetDB with embedded R. This you can use soon!

Thank You!

Questions?