B15 DB2 UDB Advanced Analytics for Business Intelligence

Peter Haas, Research Staff Member, IBM

DB2 UDB supports a variety of statistical and analytical functions that can be used to extract valuable business information from data using SQL queries. Via a series of examples, we illustrate the power of DB2 for BI analytics. DB2's aggregation and OLAP functions can be used to understand the "shape" of the data by providing summary statistics, histograms, and smoothed time series. The correlation functions can be used to detect dependencies in the data, for example, between customers and over time. The RANK and linear regression functions can be used to develop statistical models of customer behavior for purposes of prediction and outlier detection. Performing BI analytics "in the engine" avoids the need to transfer and reformat large amounts of data, and allows the user to exploit DB2's powerful features, including automatic parallelization of analytic computations.

B15

DB2 UDB Advanced Analytics for Business Intelligence

Peter J. Haas

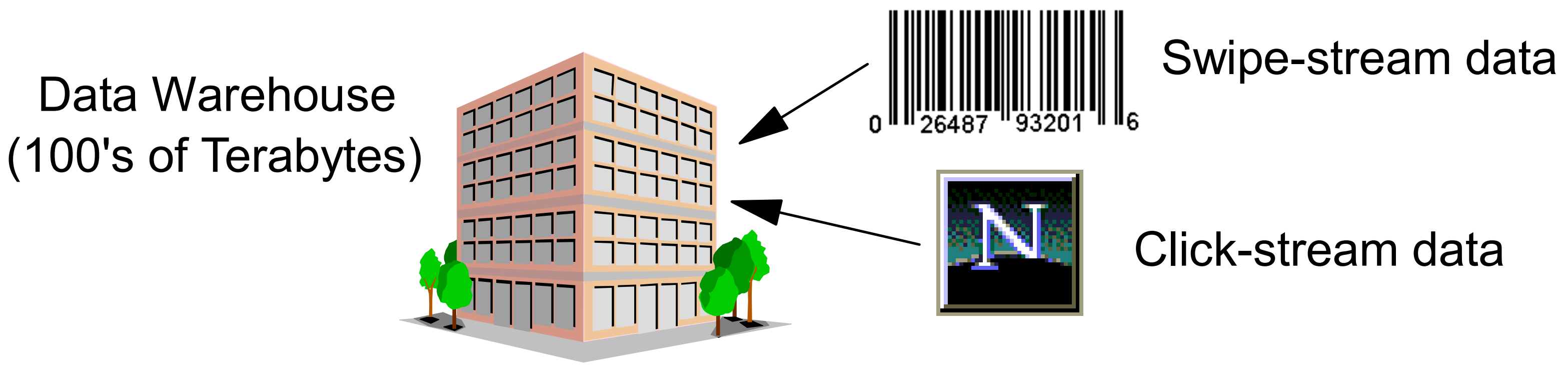
IBM Data Management Technical Conference

Anaheim, CA **Sept 9 - 13, 2002**

RETURN TO INDEX

From Data to Knowledge

- The challenge: extracting useful business information from (massive) data (automatically)



- Data analysis via SQL queries
  - processing occurs close to data
  - automatically exploits parallelism
  - can exploit other DB features: incremental maintenance, etc.

Some Different Types of Analyses

- Understanding the overall "shape" of the data
  - summaries
  - pictures

- Detecting outliers

- Detecting dependencies
  - between customers
  - over time

- Statistical modelling
  - for prediction and decision-making
  - functional relationships
  - inference (answering questions)

Pertinent Features of DB2 UWO

- Classical aggregation functions
  - SUM, COUNT, AVERAGE, ...
- Statistical functions
  - STDDEV, CORR, REGR_*
- OLAP functions
  - ROWNUMBER, RANK, window aggregates, ...
- Other V5-V7 enhancements
  - common table expressions
  - CASE
  - triggers
- Can use SQL to combine tools in new and powerful ways

Classical Summary Statistics

```
VIEW transvw1(country, year, amount)

select country, year,
   count(*) as count, sum(amount) as sum,
   avg(amount) as avg, max(amount) as max,
   stddev(amount) as stddev
from transvw1
group by country, year;
```

COUNTRY	YEAR	COUNT	SUM	AVG	MAX	STDDEV
GERMANY	1998	31	3126.04	100.84	109.75	6.18
GERMANY	1999	24	3549.06	147.87	160.87	7.49
USA	1998	20	4031.20	201.56	249.34	28.91
USA	1999	25	7820.09	312.80	607.98	281.36

Outliers: Credit Card Fraud Detection

- Create a customer card-usage profile table:

```
CREATE VIEW profile(cust_id, avg_amt, sd_amt) AS
   select cust_id, avg(charge_amt), stddev(charge_amt)
   FROM trans
   WHERE date BETWEEN '2002-01-01' and '2002-03-31'
   GROUP BY cust_id;
```

- Detect and flag unusually large charges

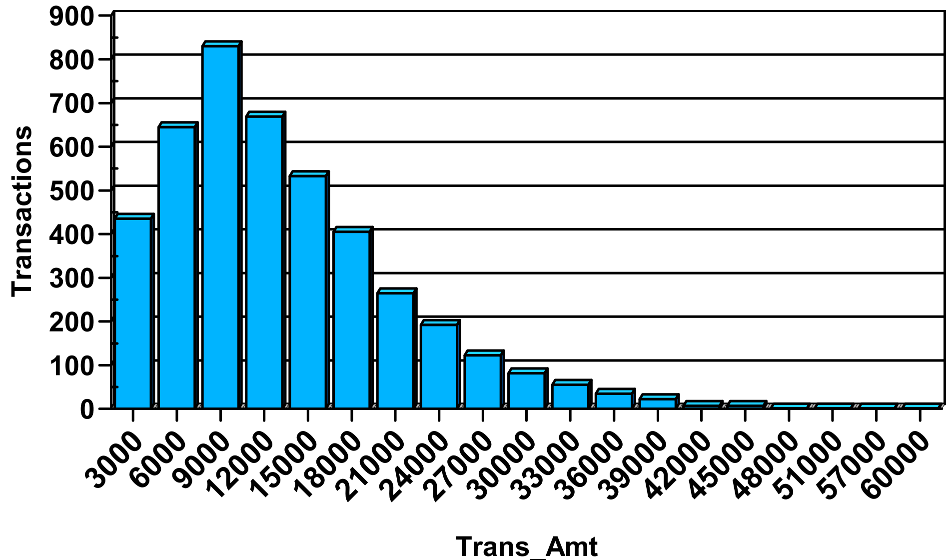
```
CREATE TRIGGER big_chrg
AFTER INSERT ON trans
REFERENCING NEW AS newrow FOR EACH ROW MODE DB2SQL
WHEN (newrow.charge_amt > (SELECT avg_amt + 2e0 * sd_amt
                           FROM profile
                           WHERE profile.cust_id =
                             newrow.cust_id))
INSERT INTO big_charges(cust_id,charge_amt)
   VALUES(newrow.cust_id, newrow.charge_amt);
```

(Equi-Width) Histograms

equi-width histogram for transaction amounts (20 buckets):

```
WITH dt as (SELECT t.transid, sum(amount) as trans_amt,
      case
         when     (sum(amount)-0)/((60000-0)/20) < 0 then 0
         when     (sum(amount)-0)/((60000-0)/20) > 19 then 19
         else int((sum(amount)-0)/((60000-0)/20))
       end as bucket
       FROM trans t, transitem ti WHERE t.transid=ti.transid
      GROUP BY t.transid)
SELECT bucket, count(bucket) as height, (bucket+1) *
(60000-0)/20 as max_amt FROM dt GROUP BY bucket;
```

BUCKET	HEIGHT	MAX_AMT
0	435	3000
1	645	6000
2	830	9000
3	669	12000
4	533	15000
5	405	18000
6	265	21000
7	192	24000
8	123	27000
9	82	30000
10	55	33000
11	35	36000
12	22	39000
13	7	42000
14	7	45000
15	1	48000

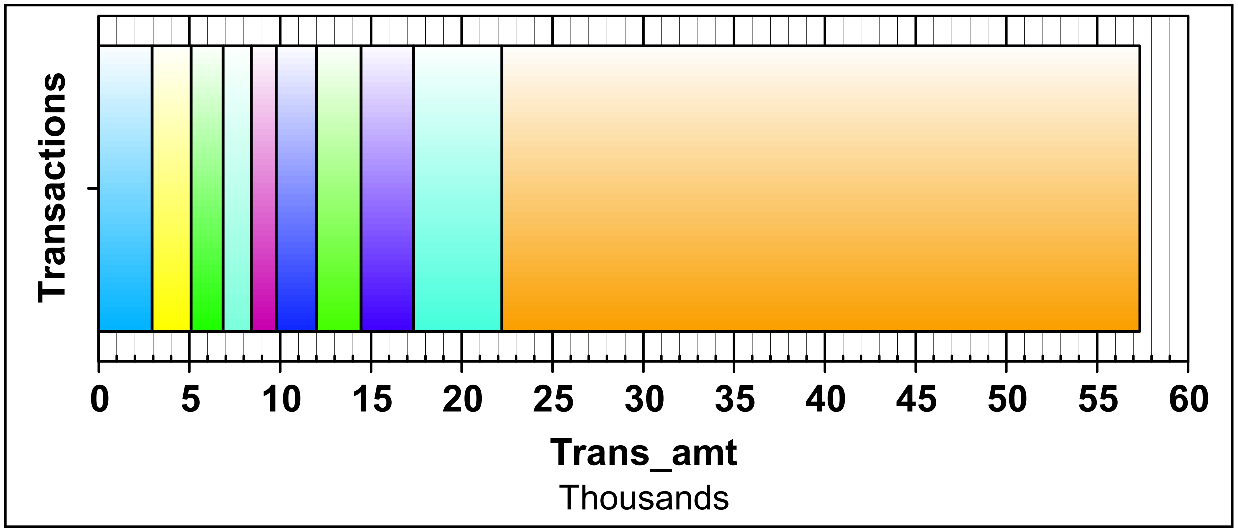


Quantiles (Equi-Height Histograms)

equi-height histogram for transaction amounts (10 buckets):

```
WITH dt as
  (SELECT t.transid, sum(amount) as trans_amt,
          rownumber() over (order by sum(amount)) * 10 /
           (select count(distinct transid)+1
            from stars.transitem) as bucket
   FROM stars.trans t, stars.transitem ti
   WHERE t.transid=ti.transid GROUP BY t.transid
  )
SELECT bucket, count(bucket) as b_count, max(trans_amt) as
  part_value
FROM dt GROUP BY bucket;
```

BUCKET	B_COUNT	PART_VALUE
0	430	2957.54
1	431	5094.14
2	431	6873.05
3	431	8429.81
4	431	9793.69
5	431	12019.40
6	431	14468.20
7	431	17355.26
8	431	22215.92
9	431	57360.41

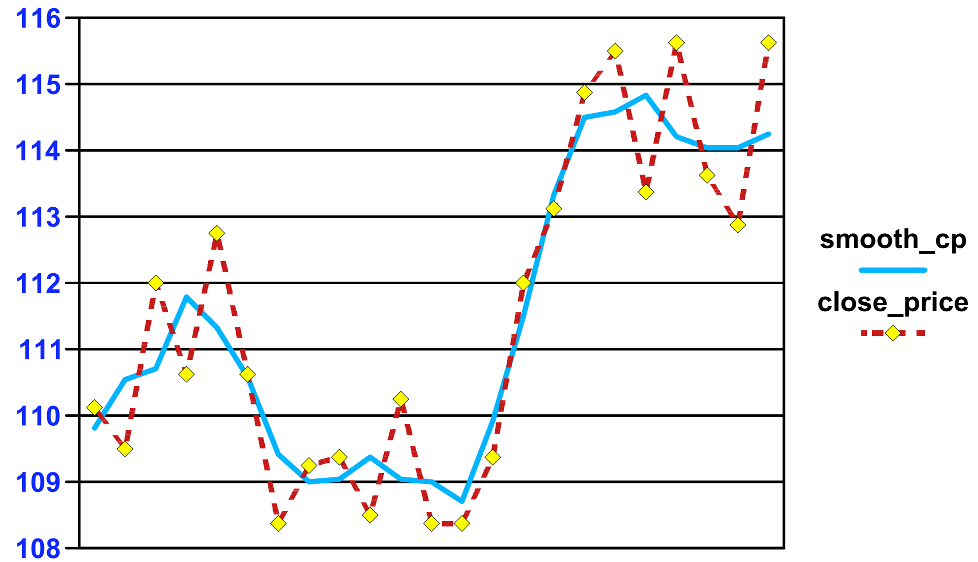


Smoothed Time Series

Three-day running-mean smoothed average of IBM stock prices:

```
SELECT date, symbol, close_price,
  avg(close_price) OVER (order by date rows between 1
  preceding and 1 following) AS smooth_cp FROM stocktab
WHERE symbol = 'IBM' and date between '1999-08-01' and
  '1999-09-01' ;
```

```
DATE       SYMBOL CLOSE_PRICE  SMOOTH_CP
---------- ------ ------------ -------------
08/02/1999 IBM         110.125      109.8125
08/03/1999 IBM         109.500      110.5416
08/04/1999 IBM         112.000      110.7083
08/05/1999 IBM         110.625      111.7916
08/06/1999 IBM         112.750      111.3333
08/09/1999 IBM         110.625      110.5833
08/10/1999 IBM         108.375      109.4166
08/11/1999 IBM         109.250      109.0000
08/12/1999 IBM         109.375      109.0416
08/13/1999 IBM         108.500      109.3750
08/16/1999 IBM         110.250      109.0416
08/17/1999 IBM         108.375      109.0000
08/18/1999 IBM         108.375      108.7083
08/19/1999 IBM         109.375      109.9166
08/20/1999 IBM         112.000      111.5000
08/23/1999 IBM         113.125      113.3333
08/24/1999 IBM         114.875      114.5000
08/25/1999 IBM         115.500      114.5833
08/26/1999 IBM         113.375      114.8333
08/27/1999 IBM         115.625      114.2083
08/30/1999 IBM         113.625      114.0416
08/31/1999 IBM         112.875      114.0416
09/01/1999 IBM         115.625      114.2500
```



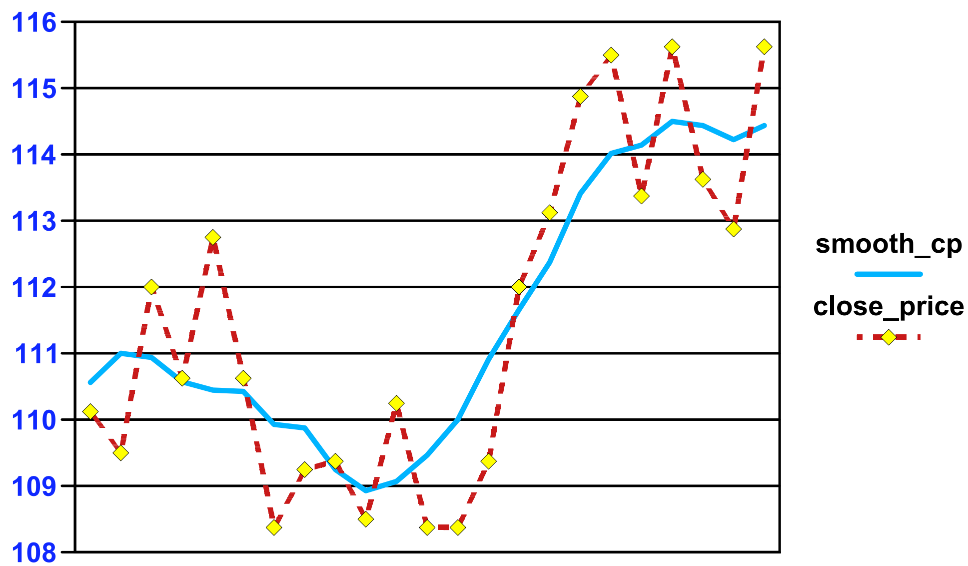
Three-day running-mean smooth

Smoothed Time Series

Seven-day running-mean smoothed average of IBM stock prices:

```
SELECT date, symbol, close_price,
  avg(close_price) over (order by date rows between 3
  preceding and 3 following) as smooth_cp
FROM stocktab
WHERE symbol = 'IBM' and date between '1999-08-01' and
  '1999-09-01' ;
```

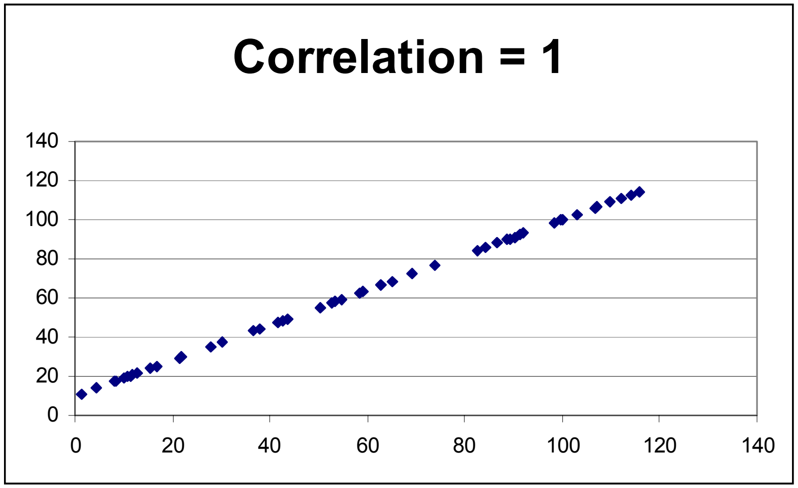
DATE	SYMBOL	CLOSE_PRICE	SMOOTH_CP
08/02/1999	IBM	110.125	109.8125
08/03/1999	IBM	109.500	110.5416
08/04/1999	IBM	112.000	110.7083
08/05/1999	IBM	110.625	111.7916
08/06/1999	IBM	112.750	111.3333
08/09/1999	IBM	110.625	110.5833
08/10/1999	IBM	108.375	109.4166
08/11/1999	IBM	109.250	109.0000
08/12/1999	IBM	109.375	109.0416
08/13/1999	IBM	108.500	109.3750
08/16/1999	IBM	110.250	109.0416
08/17/1999	IBM	108.375	109.0000
08/18/1999	IBM	108.375	108.7083
08/19/1999	IBM	109.375	109.9166
08/20/1999	IBM	112.000	111.5000
08/23/1999	IBM	113.125	113.3333
08/24/1999	IBM	114.875	114.5000
08/25/1999	IBM	115.500	114.5833
08/26/1999	IBM	113.375	114.8333
08/27/1999	IBM	115.625	114.2083
08/30/1999	IBM	113.625	114.0416
08/31/1999	IBM	112.875	114.0416
09/01/1999	IBM	115.625	114.2500

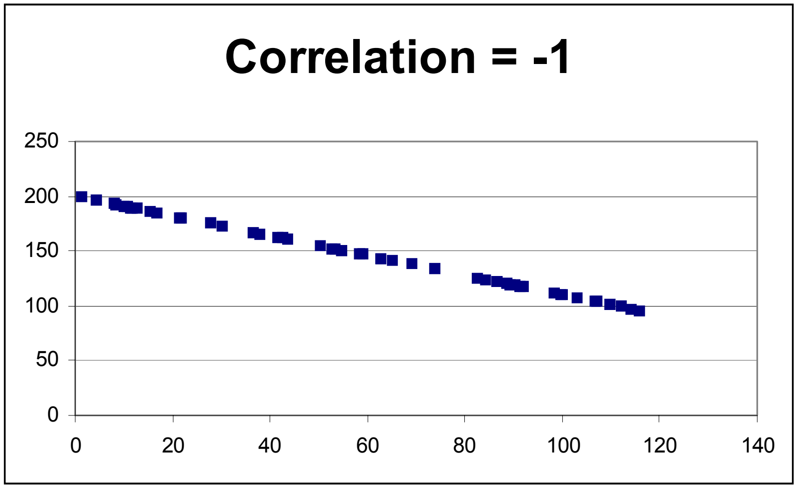


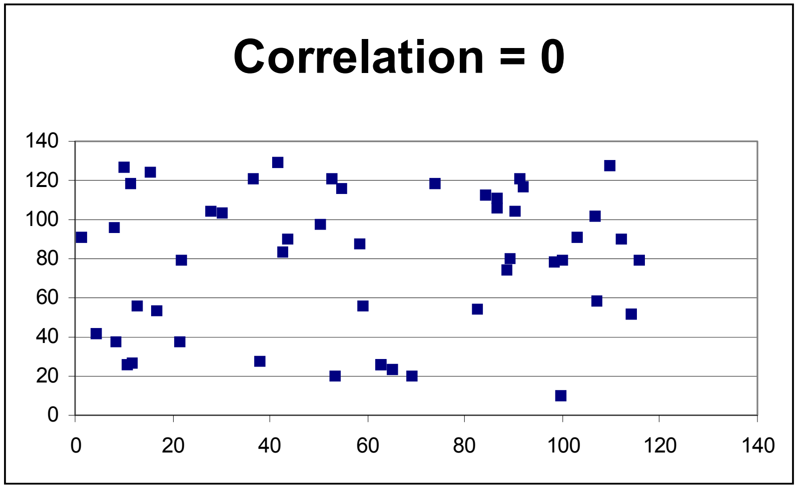
Seven-day running-mean smooth

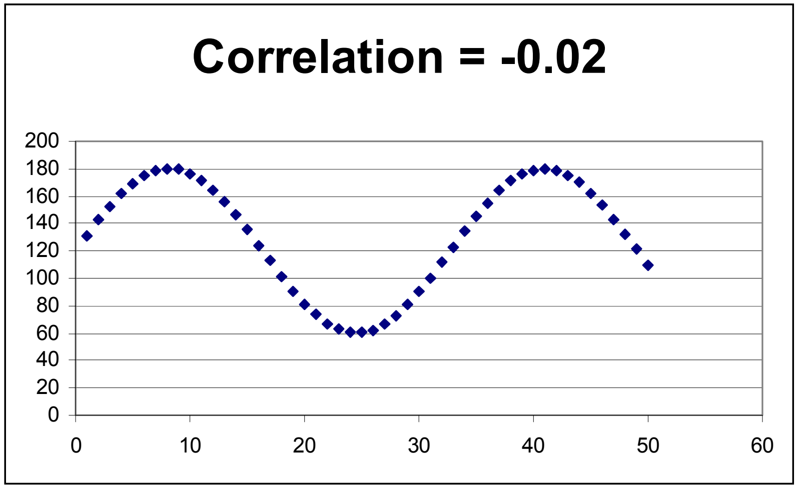
Detecting Dependencies: Correlation

- Correlation coefficient: measures strength of linear relationship









Correlation in DB2

Sales regions where income and purchases are not aligned:

```
VIEW transvw2(country, state, annual_purchases, income)

SELECT country, state,
  correlation(annual_purchases, income) AS correlation
FROM transvw2
GROUP BY country, state
having abs(correlation(annual_purchases, income)) > 0.10;
```

```
COUNTRY    STATE      CORRELATION
---------- ---------- -----------
USA        AK                0.78
USA        AL                0.68
USA        DE               -0.30 <=
USA        GA                0.14
USA        KS                0.69
USA        LA                0.48
```

Can also display covariance (= "unnormalized" correlation)

Another Use for Correlation

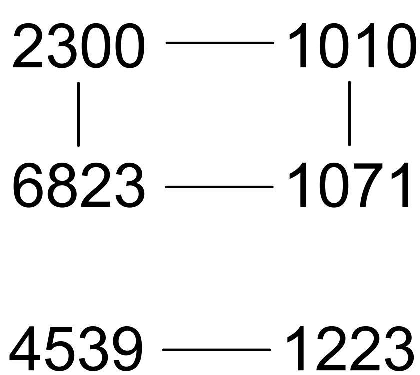
Customers with similar buying habits:

total amount puchased over all transactions

```
VIEW transvw3(custid, prodid, amount)

SELECT a.custid as custid1, b.custid as custid2,
       corr(a.amount, b.amount)as corr
FROM transvw3 a, transvw3 b
WHERE a.prodid = b.prodid and a.custid < b.custid
GROUP BY a.custid, b.custid
HAVING corr(a.amount, b.amount) >= 0.5
  and count(*) > 100
ORDER BY corr desc;
```

CUSTID1	CUSTID2	CORR
2300	6823	0.99
1071	2300	0.85
1223	4539	0.83
1010	1071	0.78
1010	2300	0.72
1071	6823	0.65



Autocorrelation

Correlate yearly sales with sales from previous years

```
VIEW transvw4(pgname, year, total_sales)

WITH dt (pgname, year, sales_0, sales_1, sales_2) AS
  (SELECT pgname, year, total_sales,
    max(total_sales) over
     (partition by pgname order by year rows between 1 preceding
      and 1 preceding),
    max(total_sales) over
     (partition by pgname order by year rows between 2 preceding
      and 2 preceding)
   FROM transvw4
  )
SELECT pgname, correlation(sales_0,sales_1)*100 as "correlation1(%)",
  correlation(sales_0,sales_2)*100 as "correlation2(%)",
FROM dt GROUP BY pgname;
```

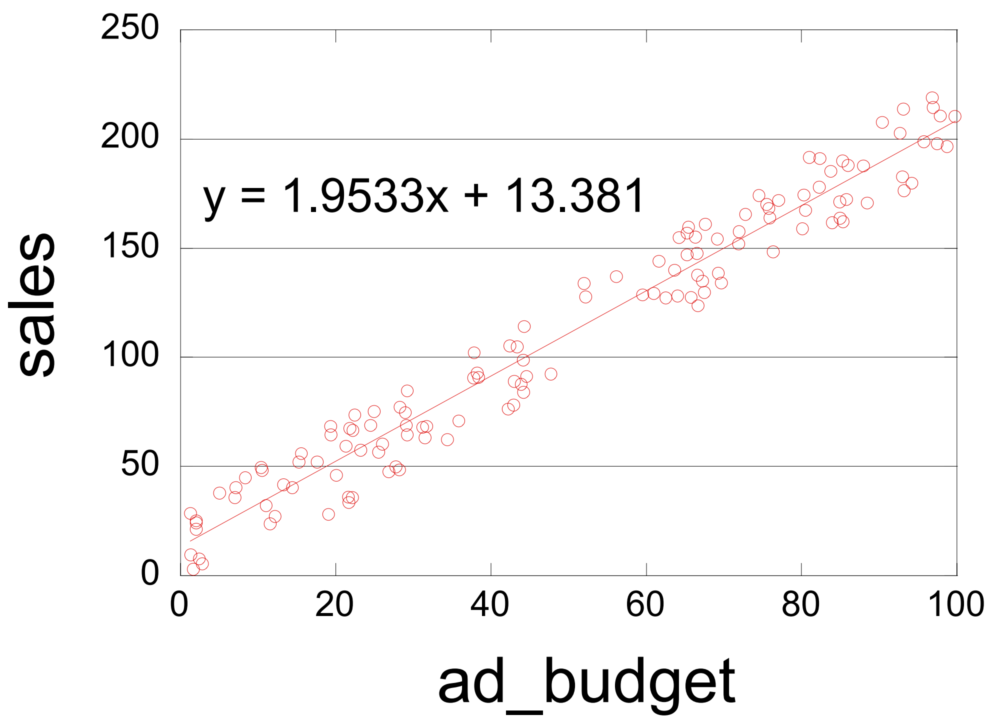
```
PGNAME       correlation1(%) correlation2(%)
------------ --------------- ---------------
antibiotics            -2.83          -29.07
camcorder             -54.78           20.75
coats                 -25.40           -1.68
vcr                    59.39           33.17
```

Least-Squares Fit (Linear Regression)

- Fits a line of the form y = ax + b   from (x,y) pairs
- Ex: effect of advertising budget on 1999 sales

```
SELECT           y         x
  regr_count(sales, ad_budget) AS num_cities,
  regr_slope(sales, ad_budget) AS a,
  regr_icpt(sales, ad_budget) AS b
FROM ad_camp;
```

```
num_cities          a      b
----------     ------ ------
126            1.9533 13.381
```



Fitting Other Types of Curves

- $y = ax^2 + b$

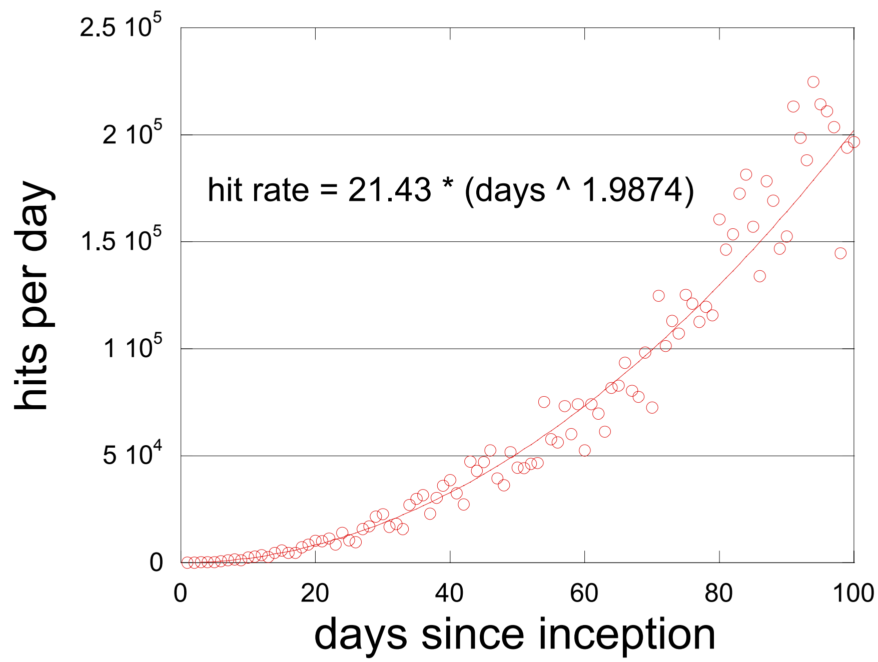
```
SELECT regr_slope(y, x*x) AS a,
       regr_icpt(y, x*x) AS b
```

- $y = bx^a$
  - $\log y = a \log x + \log b$

```
SELECT regr_slope(log(y), log(x)) AS a,
       exp(regr_icpt(log(y), log(x))) AS b   ...
```

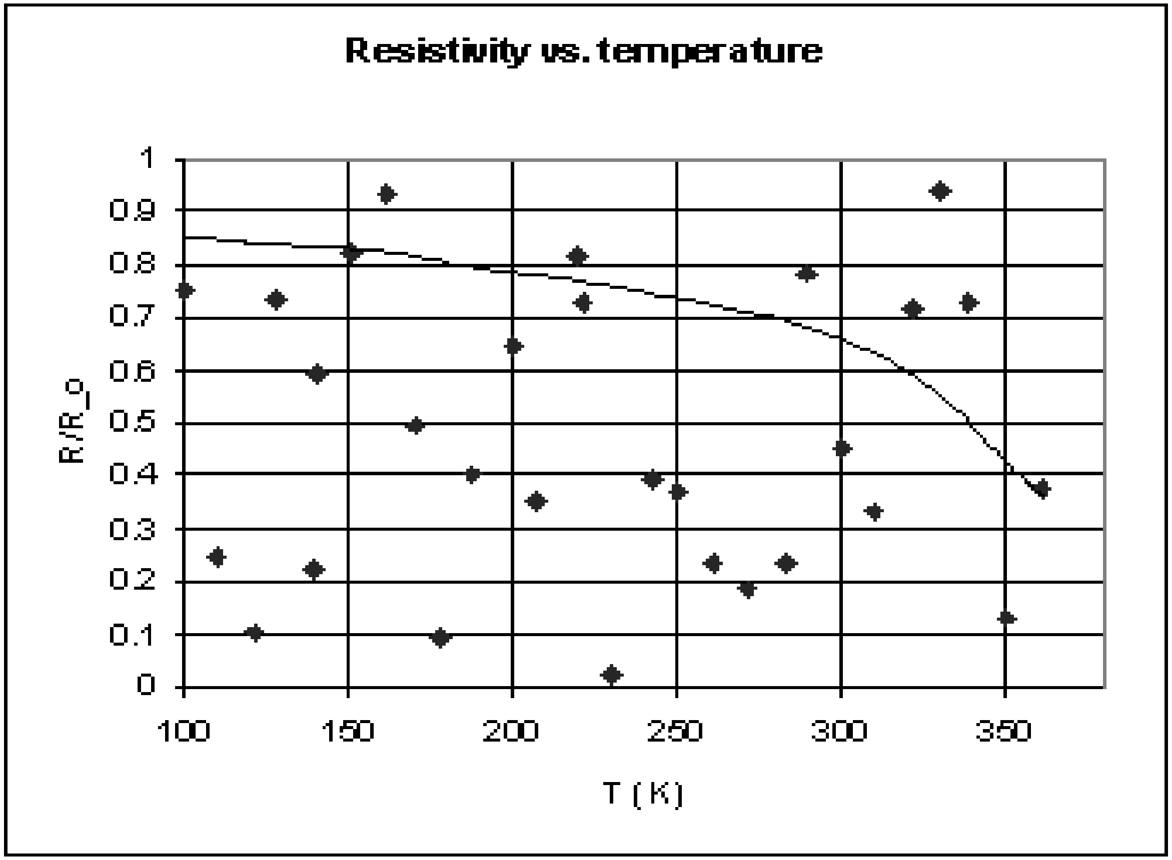
```
SELECT
 regr_count(hits,days) as num_days
 regr_slope(log(hits),log(days)) AS a,
 exp(regr_icpt(log(hits),log(days))) AS b
FROM traffic_data;

   NUM_DAYS       A       B
----------- ------ -------
        100 1.9874 21.4302
```



Quality of Fit

- Want diagnostics!
  - especially in automated environment



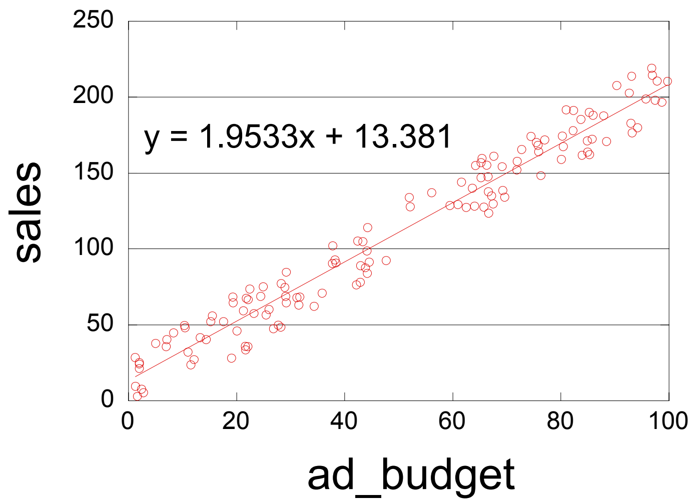
From L. Kovar, "Band Structure in Germanium, My #$%$"
Ann. Improbable Research, 7(3), 2001
http://www.improb.com/airchives/paperair/volume7/v7i3/germanium-7-3.html

Quality of Fit - Continued

- R-Squared
  - roughly, the square of the correlation of x and y
  - proportion of y-variation explained by the model

```
SELECT
  regr_count(sales, ad_budget) AS num_cities,
  regr_slope(sales, ad_budget) AS a,
  regr_icpt(sales, ad_budget) AS b,
  regr_r2(sales, ad_budget) as r-squared
FROM ad_camp;
```

```
num_cities         a      b   r-squared
----------    ------ ------   ---------
128           1.9533 13.381     0.95917
```

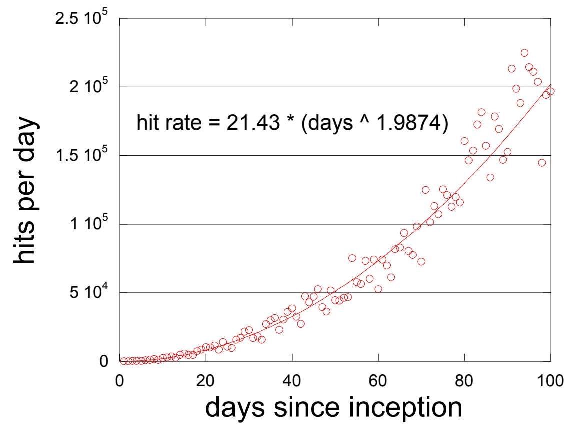


Quality of Fit for Nonlinear Curves

- **Incorrect:** Compute R-Squared for transformed data

```
select regr_r2(log(hits),log(days))as r2 from traffic_data;
```

```
r2: 0.9912
```



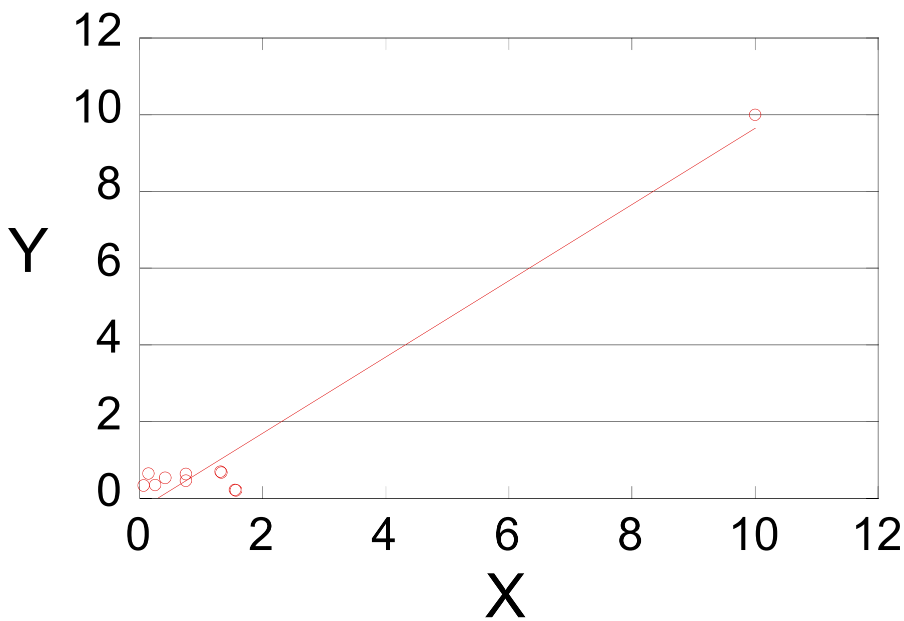
- **Correct:** Compute R-Squared for original data

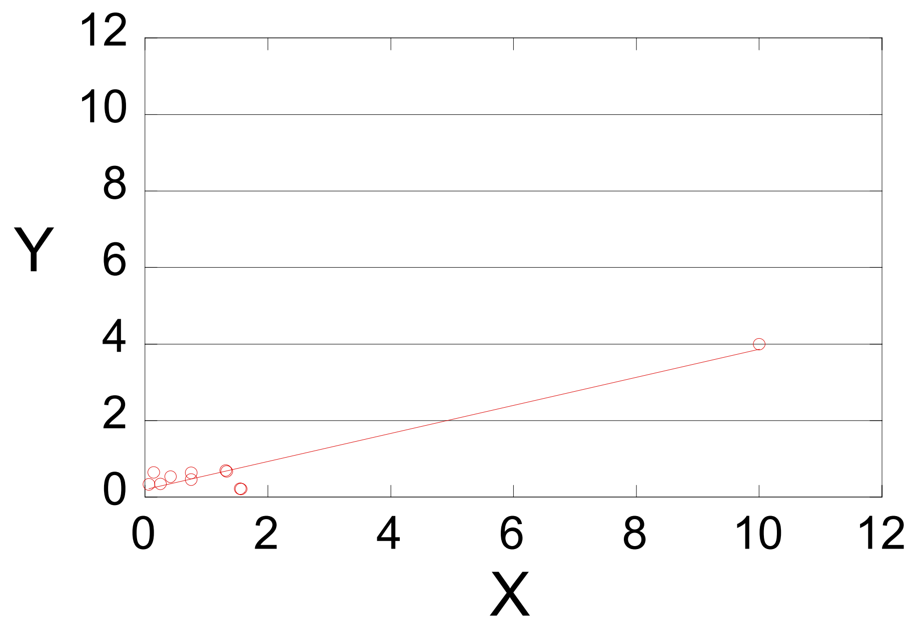
```
with coeffs(a,b) as
(select regr_slope(log(hits),log(days)) as a,
           exp(regr_icpt(log(hits),log(days))) as b
  from traffic_data),
residuals(days,hits,error) as
(select t.days, t.hits, t.hits - c.b * power(t.days,c.a)
  from traffic_data t, coeffs c)
select 1e0 - (sum(error*error)/regr_syy(hits,days)) as r2
from residuals;
```

```
r2: 0.9554
```

Influence of Individual Data Points

- Some data are more important than others



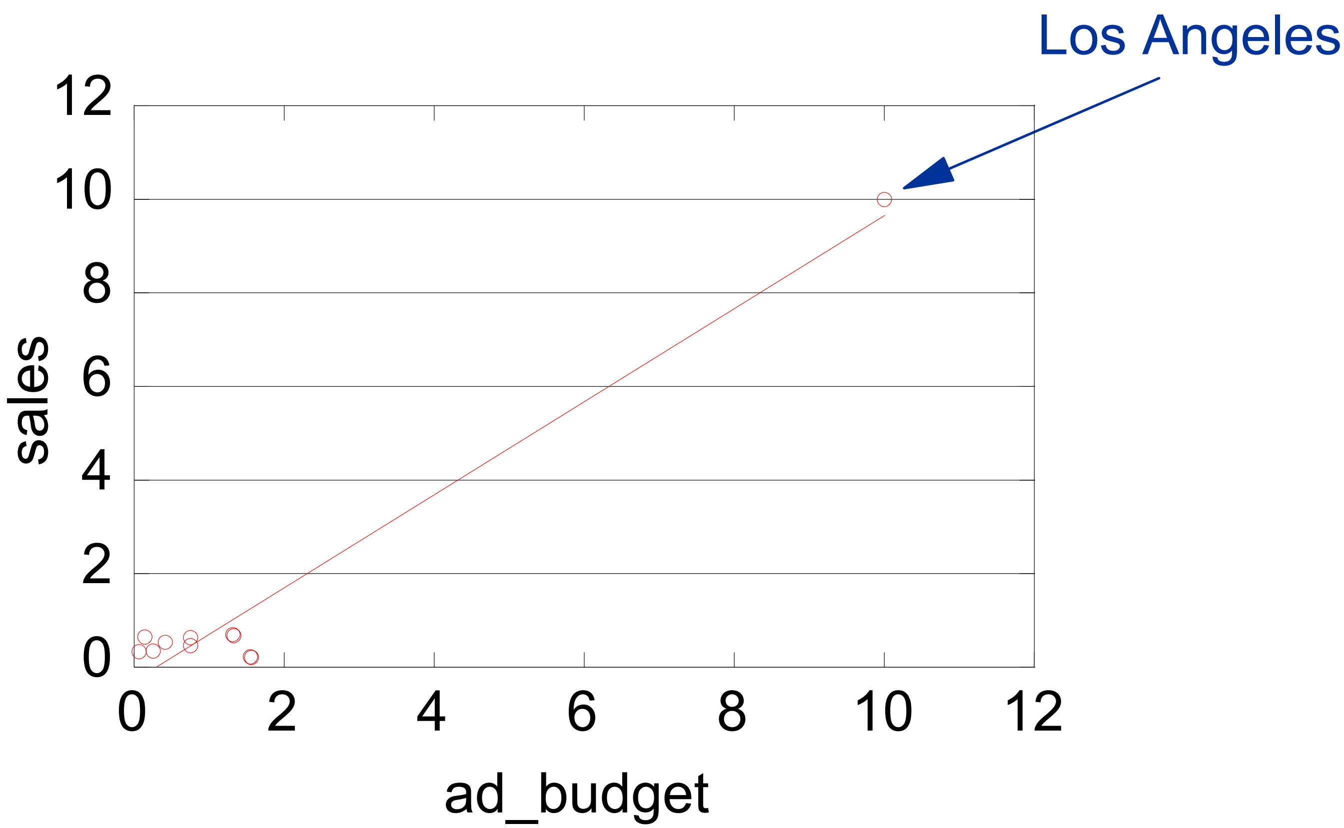


- Measure of influence: HAT diagonal
  - $h_i = (m_{x2} - 2m_x x_i + x_i^2) / s_{XX}$
    - $m_x = avg(x_1, \ldots ,x_n)$
    - $m_{x2} = avg(x_1^2, \ldots ,x_n^2)$
    - $s_{XX} = (x_1 - m_x)^2 + \ldots + (x_n - m_x)^2$

HAT Diagonal Computation

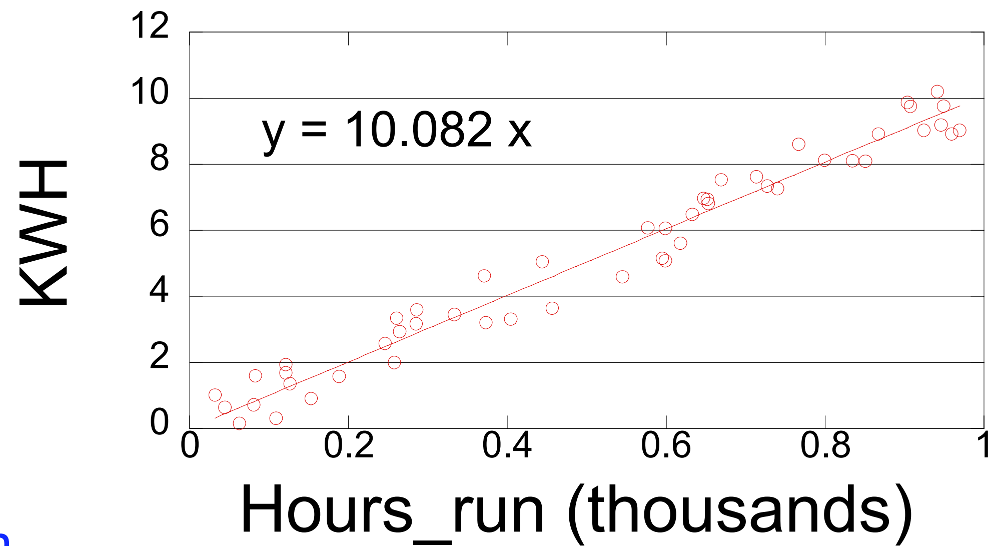
```
WITH stats(mx, mx2, sxx) AS
(SELECT regr_avgx(sales,ad_budget),
        regr_avgx(sales,ad_budget*ad_budget), regr_sxx(sales,ad_budget)
 FROM cal_ad_camp
)
SELECT d.label as city, (s.mx2 - 2 * s.mx * d.x + d.x * d.x) / s.sxx AS hat
FROM xy_data d, stats s
ORDER BY hat DESC;
```

```
       city       hat
-----------  ------
Los Angeles  0.9644
Boonville    0.1222
Grass Valley 0.1195
Yreka        0.1154
Gilroy       0.1099
Lemoore      0.1011
Hilmar       0.1011
Mendocino    0.0923
Turlock      0.0922
Morgan Hill  0.0910
Truckee      0.0910
```



Regression through the origin

- Fit a line of the form y = ax
  - $a = (x_1y_1 + \ldots + x_ny_n) / (x_1^2 + \ldots + x_n^2)$
  - recall regr_sxx, regr_sxy
    - regr_sxx: $(x_1 - m_x)^2 + \ldots + (x_n - m_x)^2$
    - here $m_x = \text{avg}(x_1, \ldots, x_n)$ as before
  - a trick:
    - $(x_1^2 + \ldots + x_n^2) = \text{regr\_sxx} + n\, m_x^2$
    - $(x_1y_1 + \ldots + x_ny_n) = \text{regr\_sxy} + n\, m_x\, m_y$




```
SELECT
  regr_count(kwh, hours_run) as num_machines,
  (regr_sxy(kwh, hours_run) + regr_count(kwh, hours_run) *
   regr_avgx(kwh, hours_run) * regr_avgy(kwh, hours_run))
  / (regr_sxx(kwh, hours_run) + regr_count(kwh, hours_run) *
     regr_avgx(kwh, hours_run) * regr_avgx(kwh, hours_run))
   as a;
FROM power_data;
```

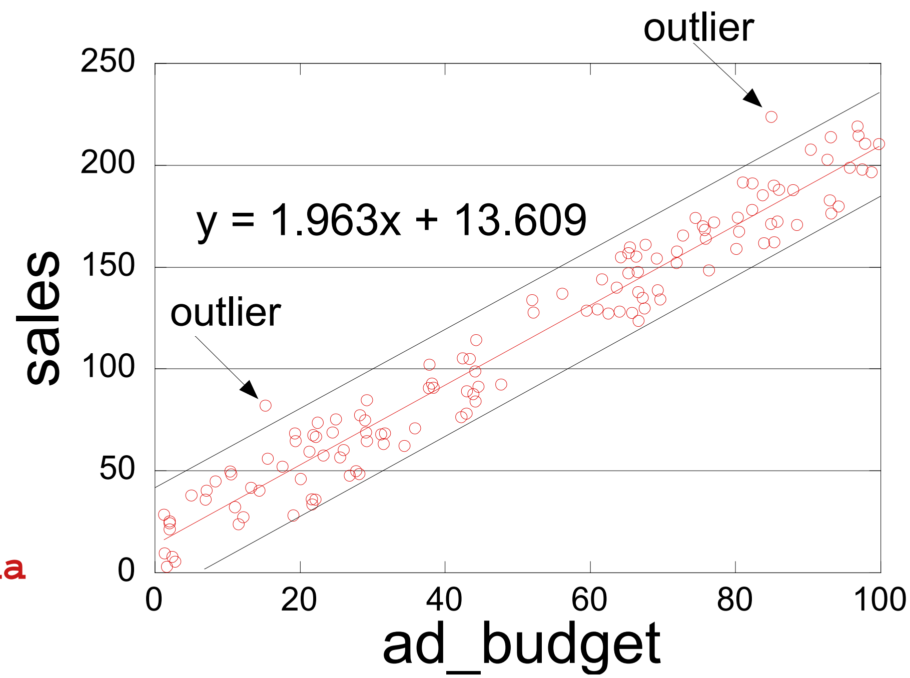
```
num_machines        a
------------   ------
50             10.082
```

Outliers II: Effective Ad Campaigns

- Identify unusually effective campaigns, controlling for ad budget
- Use sigma, the standard deviation about the regression line

```
WITH dt(a, b, sigma) AS
 (SELECT
   regr_slope(sales,ad_budget),
   regr_icpt(sales,ad_budget),
   sqrt((regr_syy(sales,ad_budget)
    - (regr_sxy(sales,ad_budget)
    *regr_sxy(sales,ad_budget)
    /regr_sxx(sales,ad_budget)))
    / (regr_count(sales,ad_budget) - 2))
  FROM ad_camp)
SELECT city, ad_budget, sales
FROM ad_campx ac, dt
WHERE sales > a*ad_budget + b + 2e0*sigma
ORDER BY ad_budget;
```

```
CITY        BUDGET    SALES
------      ------   ------
Fresno       15.26    82.00
San Diego    84.99   223.81
```



Multiple Linear Regression

- Fit a line of the form $y = b_0 + b_1x_1 + \ldots + b_nx_n$
  - ex: $y = a_1z + a_2 z^2 + b$
- To fit: write in matrix form and solve "normal equations"

$$y = Xb + e$$

$$\begin{vmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{vmatrix} = \begin{vmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \ldots & \ldots & \ldots & & \ldots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{vmatrix} \begin{vmatrix} b_0 \\ b_1 \\ \ldots \\ b_k \end{vmatrix} + \begin{vmatrix} e_1 \\ e_2 \\ \ldots \\ e_n \end{vmatrix}$$

Find b's that minimize $\mathrm{T}e_i^2$

given

choose

minimize

Multiple Linear Regression, Continued

- Compute b by solving "normal equations": $[X^TX]b = [X^Ty]$

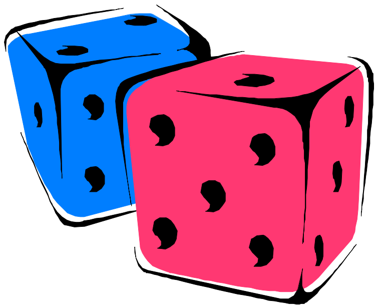
- Multiple regression in DB2 UWO --- a simple approach
  - compute entries of $[X^TX]$ and $[X^Ty]$ using SQL queries
    - `SELECT x1*x1, x1*x2, x2*x2, x1*y, x2*y ...`
    - matrices are incrementally maintainable
  - solve for b by feeding entries into a UDF that solves equations (e.g., by Gaussian elimination)

- Research prototype (two x variables) developed at Almaden
  - Good for fixed problems with changing data

- Issues for an industrial-strength solution
    - general equation-solving UDF
    - robustness to "difficult data"
    - diagnostic statistics (R^2, etc.)

From Description to Modeling

- Scenario 1: business decisions
  - need to make predictions/inferences
  - view data as sample from real world
  - need to distinguish real effects from luck-of-the-draw
- Scenario 2: quick analysis of massive data
  - data is small sample from large database
  - need to assess validity of sample-based computations



Significance of Regression Fit

- View data as a sample
  - $y_i = ax_i + b + error_i$

- Real effect of x on y, or luck-of-the-draw?

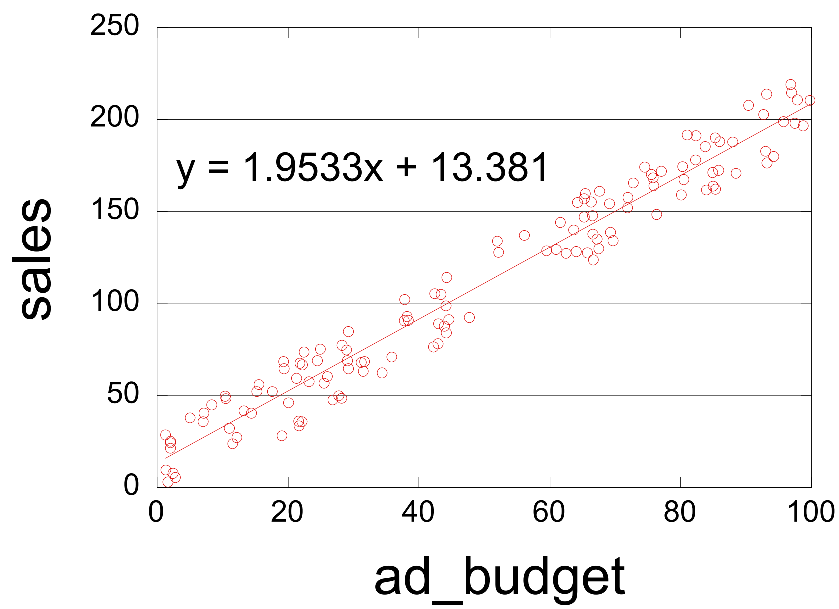
- Look at F statistic (with 1 and n-2 "degrees of freedom")
  - measures (y variability caused by x) / (unexplained y variability)
  - if $a = 0$, then F should take on "small" values

- A statistical test:
  - let f be observed value of F
  - compute Prob(F >= f) assuming that a = 0
  - suppose that f is so big that Prob(F >= f) is very small
    - unlikely to see this if a = 0
    - therefore, effect of x on y is statistically significant

Significance of Fit, Continued

```
WITH dt(num_cities, a, b, sxx, sigma2) AS
(SELECT
  regr_count(sales,ad_budget),
  regr_slope(sales,ad_budget),
  regr_icpt(sales,ad_budget),
  regr_sxx(sales,ad_budget),
  (regr_syy(sales,ad_budget)
   -(regr_sxy(sales,ad_budget)*regr_sxy(sales,ad_budget)
      /regr_sxx(sales,ad_budget)))
  / (regr_count(sales,ad_budget) - 2)
 FROM ad_camp
)
SELECT num_cities, a, b,
 ((a*a*sxx)/sigma2) AS F
FROM dt;
```

```
num_cities          a      b           F
----------  ------ ------  ---------
128         1.9533 13.381    2959.83
```



Prob($F_{1,126}$ > 2959.83) << 0.01

Caveat: errors need to be iid normal

Inference: Effectiveness of Ad Campaign

- An experiment
  - Campaign run in City B in January (8 stores)
  - No campaign in "control" City A (9 stores)
  - Monthly sales computed in stores in both cities for February
    - `VIEW feb_sales(city,store_id,sales)`
- Did campaign result in increased sales?
- Classical test: 2-sample t test
  - restrictive normality assumption
  - restrictive equal-variance assumption
- Wilcoxon Rank Test
  - a more modern "nonparametric" procedure
  - avoids restrictive assumptions

Wilcoxon Rank Test

- Test statistic: sum of ranks of City B in combined ranking

```
WITH ranked_sales(city, ranks) AS
(SELECT city, rank() over (order by sales)
 FROM feb_sales
)
SELECT sum(ranks) as W
FROM ranked_sales WHERE city = 'B'
```

- Test if W is significantly > than expected value (assuming no diff.)
  - expected value: $[(n_A n_B) + n_B(n_B+1)] / 2$
    (n_X = number of stores in City X)

- example: $n_A = 9$, $n_B = 8$, and W = 94
  - expected value = 72
  - Prob(W > 93) =2% (from tables) if no real difference in sales

Inference: Test for Independence

- Is there a relationship between operating system and DB product?
- Contingency-table analysis (number of users)

	Sybase	Oracle	
Linux	120	80	200
Unix	45	95	140
Windows	30	12	42
	195	187	382

- A lesser-known "maximum likelihood" chi^2 test for independence
  - test statistic (r rows and c columns):

$$X = 2n\log(n)$$
$$+ [2n_{11}\log(n_{11}) + \ldots + 2n_{rc}\log(n_{rc})]$$
$$- [2n_{1+}\log(n_{1+}) + \ldots + 2n_{r+}\log(n_{r+})]$$
$$- [2n_{+1} + \log(n_{+1}) + \ldots + 2n_{+c}\log(n_{+c})]$$

n_{ij}: # in cell (i,j)
n_{i+}: row i sum
n_{+j}: column j sum
n: total # users

Test for Independence, Continued

```
WITH c_table(os, db, n, g1, g2) AS
(SELECT os, db, count(*), 2e0*( 0.5e0-grouping(os)), 2e0*(0.5e0-grouping(db))
 FROM survey
 GROUP BY CUBE(os,db))
SELECT sum(g1*g2*2e0*n*log(n)) as X
FROM c_table
```

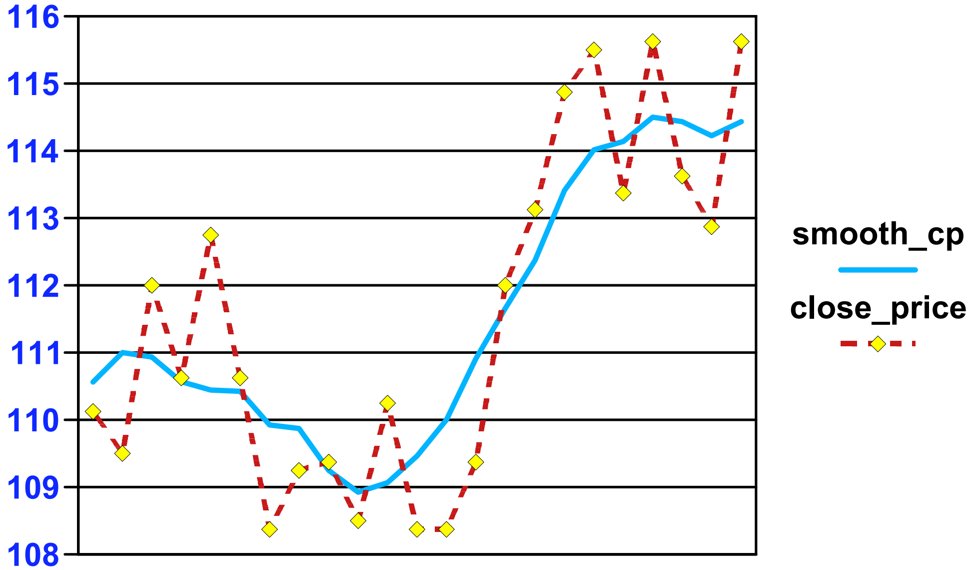
```
     X                  os      db       n      g1      g2
------              ------  ------  ------  ------  ------
34.114               Linux     SYB     120     1.0     1.0
                     Linux       -     200     1.0    -1.0
                         -     SYB     195    -1.0     1.0
                         -       -     382    -1.0    -1.0
                   ...
```

c_table

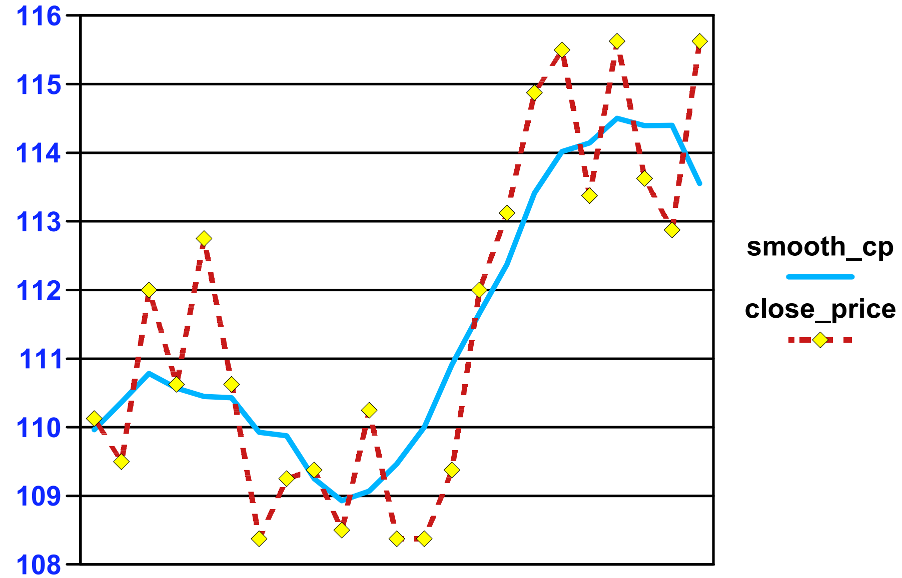
- If data is truly independent:
  - X should be close to 0
  - X has χ^2 distribution with (r-1)(c-1) degrees of freedom
- Computer example: r = 3, c = 2
  - if independent, Prob(X > 34.114) < 0.001%

Combining Regression and Windowing

- Running-line estimator (Hastie & Tibshirani, 1990)
  - fits a line $y = a_i x + b_i$ to local neighborhood of each (x_i, y_i) point
  - smoothed y value is given by $y_i^{smooth} = a_i x_i + b_i$
  - better behavior at endpoints, better statistical properties



Seven-day running-mean smooth



Seven-day running-line smooth

Regression and Windowing: Cont'd

- Would like to execute the following query:

```
WITH dt(day,date,symbol,close_price) AS
 (SELECT cast(row_number() over (order by date) as float),
   date, symbol, close_price
  FROM stocks WHERE symbol = 'XYZ' and
   date between
   '1999-08-01' and '1999-09-01'
 )
SELECT date, symbol, close_price,
  day * (regr_slope(close_price,day)
   over (order by day rows between
          3 preceding and 3 following))
  + regr_icpt(close_price,day) over (order by day
      rows between 3 preceding and 3 following)
  AS smooth_cp
FROM dt;
```

- Doesn't quite work yet
  - Work-around by expanding: regr_slope(y,x) = covar(y,x) / var(x) etc.

A Regression and Windowing Query

- Final query (assumes no NULLs):

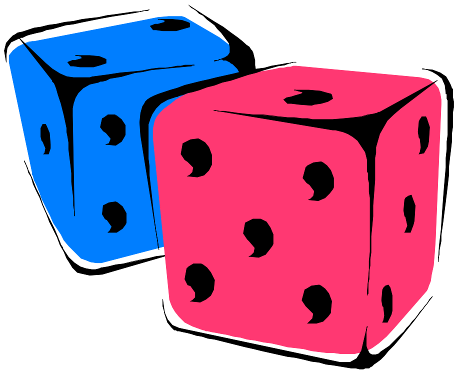
```
with
dt(day,date,symbol,close_price) as
  (select cast(row_number() over (order by date) as real),
   date, symbol, close_price from stocks
  ),
ddt(day,date,symbol,close_price,slope,avgx,avgy) as
  (select day, date, symbol, close_price,
    covar(close_price,day)
     over (order by day rows between 3 preceding and 3 following) /
    var(day)
     over (order by day rows between 3 preceding and 3 following),
    avg(day)
     over (order by day rows between 3 preceding and 3 following),
    avg(close_price)
     over (order by day rows between 3 preceding and 3 following)
   from dt
  )
select date, symbol, close_price,
 day * slope + (avgy-slope*avgx) as smooth_cp
from ddt;
```

Taming Massive Data: Sampling

- Sampling for
  - auditing or "fuzzy exploration"
  - quick approximate answers to aggregation queries
  - making analytics and datamining scalable

- Technology challenges
  - generating a representative sample efficiently
  - estimating an aggregate
  - assessing precision of estimate

- Sampling in DB2 --- Present and Future
  - Go to session B16



Selected References

- **Introductory Statistics**
  - Larry Gonick, et al : *The Cartoon Guide to Statistics*, HarperCollins, 1994
  - Jeffrey Clark and Douglas A. Downing: *Forgotten Statistics : A Self-Teaching Refresher Course*, Barrons, 1996
  - Lloyd R. Jaisingh: *Statistics for the Utterly Confused*, McGraw-Hill, 2000
- **Advanced Statistics**
  - T. J. Hastie and R. J. Tibshirani: *Generalized Additive Models,* Chapman & Hall/CRC, 1999
  - B. W. Lindgren: *Statistical Theory*, 3rd Ed., MacMillan, 1976
  - R. G. Miller: *Beyond ANOVA, Basics of Applied Statistics,* Wiley, 1986
  - R. H. Myers: *Classical and Modern Regression with Applications*, 2nd Ed., Duxbury, 1990
  - C.-E. Sarndal, B. Swenson, and J. Wretman: *Model Assisted Survey Sampling*, Springer-Verlag, 1992

- Also: Database + files are available for queries in this talk
- Also: Forthcoming Redbook

Many Thanks To...

- Kevin Beyer
- Guy Lohman
- Eric Louie
- Bob Lyle
- Hamid Pirahesh
- Ashutosh Singh
- Markos Zaharioudakis
- ...