This keynote is based on:

- M. Jørgensen, T. Halkjelsvik, and B. Kitchenham. **Is there a magnitude bias in project cost estimation?** , International Journal of Project Management 30(7):751-862, 2012.
- T. Halkjelsvik and M. Jørgensen. **From origami to software development: A review of studies on judgment-based predictions of performance time**, Psychological Bulletin 138(2):238-271, 2012.
- M. Jørgensen and B. Kitchenham. **Interpretation problems related to the use of regression models to decide on economy of scale in software development**, Accepted for publication in Journal of Systems and Software, 2012.
- M. Jørgensen. **The Influence of Selection Bias on Effort Overruns in Software Development Projects**, Submitted to a journal, 2012.
- M. Jørgensen. **Myths and Over-simplifications in Software Engineering**, Submitted to a conference, 2012.
- M. Jørgensen and K. J. Moløkken-Østvold. **How Large Are Software Cost Overruns? Critical Comments on the Standish Group's CHAOS Reports**, Information and Software Technology 48(4):297-301, 2006.

(download these papers from: simula.no/people/magnej/bibliography)

[simula . research laboratory]

It is likely (in many contexts) that …

- if you do unusually well on one test, you will do worse on the following.
- any treatment of the worst performers will have a positive effect.
- your children will be worse than you on things you are exceptionally good at.
- the "rookie of the year" will disappoint the following year.

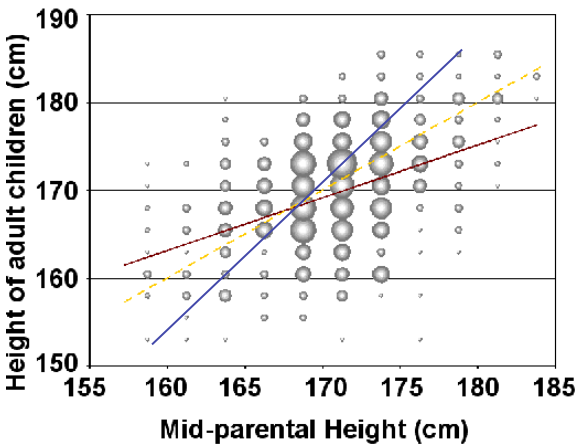
The reason is:
Regression towards the mean (RTM)

- RTM is a real, statistical effect easy to misinterpret.
- Milton Friedman once wrote that "*I suspect that the regression fallacy is the most common fallacy in the statistical analysis of economic data*".
- First formulated by Sir Francis Galton more than 100 years ago.
- The relevance of RTM increases with decreasing correlation between the variables of interest.

The beginning: Galton's study of inheritance

- Galton observed that children of tall (short) parents were typically shorter (taller) than their parents.
- Galton first described this finding as a biological force - "*filial regression to mediocrity*".
- BUT, then the human race would soon consists of people with the same, average, height!
- In addition, a time-reversal gave the opposite result. Parents of tall (short) children were typically shorter (taller) than their above average tall parents.

Galton's data
(probably the first regression analysis in history)



Yellow line: Same average height of parents and their children

Red line: Regressing children's height on parent's mid-height

Blue line: Regressing parent's mid-height on children's height.

Comment to the interpretation of RTM

- Observed height = “true” height + “noise”
  - “true” height is the inherited height
  - “noise” is the rest, i.e., luck, measurement error, etc.
- The “true” height of parents and their children is the same - given no increase in average height in a population over time.
- The observed height regresses towards the mean
  - The “luck” (☺) that made me tall, will on average not be repeated on my children.

Mutual funds

- Easy to show that there is close to zero correlation between a mutual fund’s ranked performance one year and the next.
- The relative performance of a mutual fund can be simulated through a model where the the true performance is the same, and the observed performance is fully determined by noise (luck, bad luck).

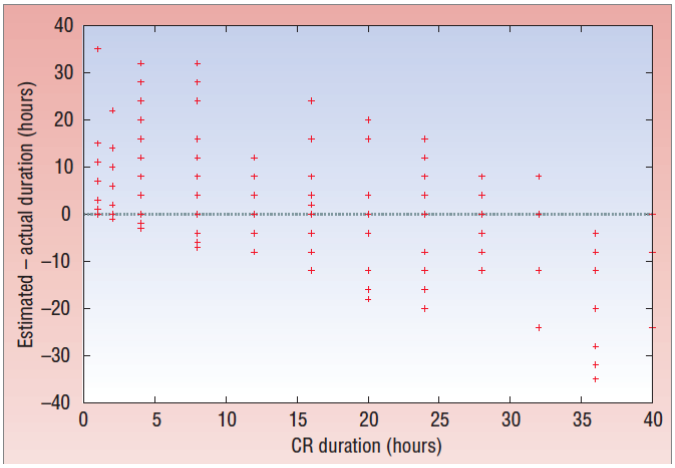
Mutual funds - simulation

No correlation between Rank(t) and Rank(t+1), i.e., no “true”, only RTM-induced relationship.

These results should NOT be interpreted to mean that good performers “get lazy” poor ones “pull them selves together”.

simula . research laboratory

How would you interpret this data?

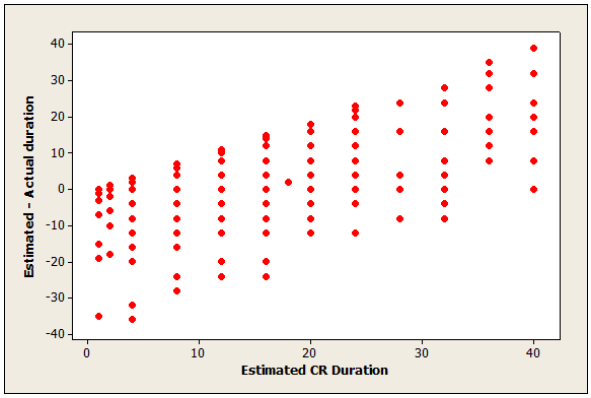


CR duration = Actual duration (effort) to complete a change request

Interpretation by author: Larger tasks are more under-estimated.

simula . research laboratory

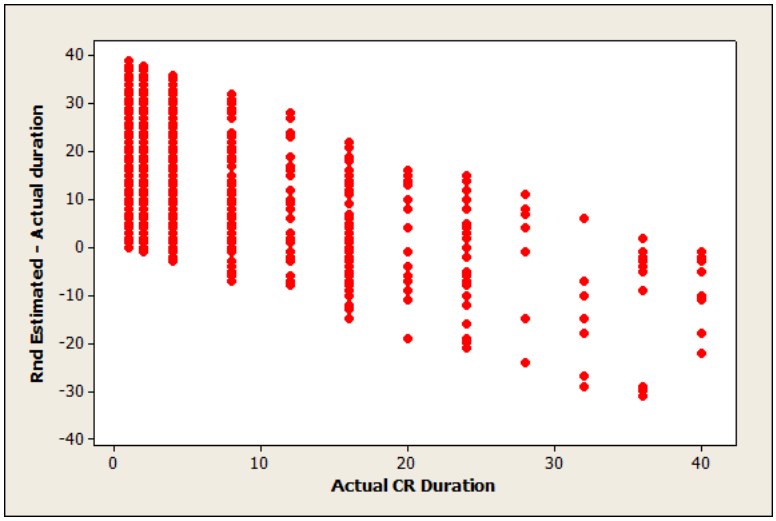
What about these data?



They are from the exact same data set! The only difference in the use of estimated and actual duration as the task size variable.

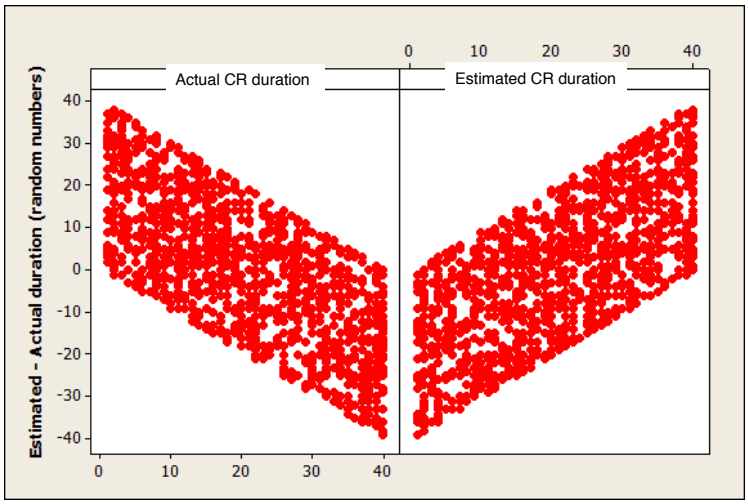
simula research laboratory

The data we get with actual effort being a random number between 1 and 40



simula research laboratory

The data we get when there is no correlation between estimated and actual duration



simula . research laboratory

What is going on?

Let us start with the relation
between
effort (work-hours)
and
project size (lines of code)

simula . research laboratory

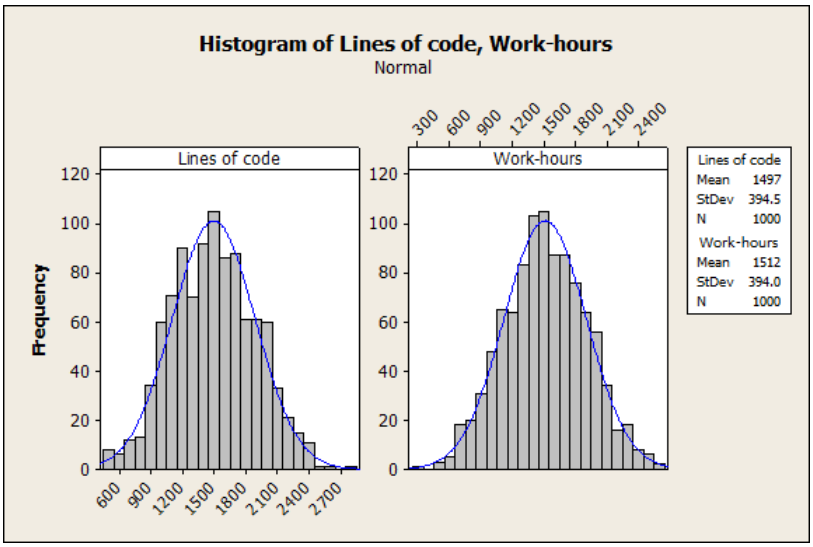
We generate a data set with a linear relationship between effort and task size

- The “true” relationship is that it takes one work-hour (WH) to develop one line of code (LOC), i.e., **WH = LOC**.
- There is substantial “noise” in the measurement of WH and LOC, e.g., :
  - Inconsistency in how lines of codes and work-hours are measured from project to project
  - “Unsystematic” variance in WH or LOC, e.g., differences in programming style that produces more LOC for the same amount of WH for one person than another.
- The noise leads to a correlation of 0.5 between the observed task size and effort.

Generation process (1000 tasks)

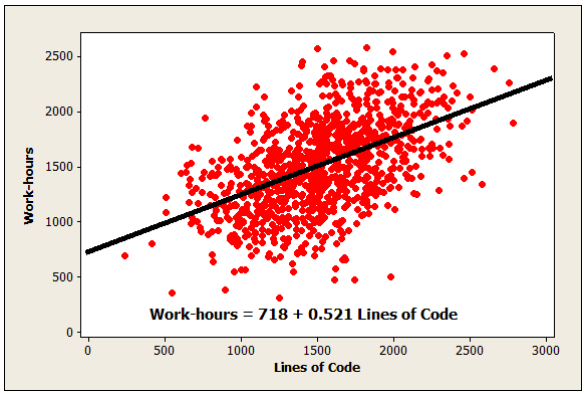
- Randomly draw a number, representing the **true number of LOC** of a system, from a uniform distribution with numbers between 1000 and 2000.
- Calculate **true number of WH** as:
  - WH_true =LOC_true.
- Randomly draw two numbers representing the “noise” of LOC and WH from a normal distribution with mean 0 and standard deviation 280.
- Calculate **observed number of LOC** as:
  - LOC_obs = LOC_true + LOC_noise.
- Calculate the **observed number of WH** as:
  - WH_obs = WH_true + WH_noise.

The generated observed LOC and WH have about the same normal distribution (same mean and same std)



simula research laboratory

Regressing LOC on WH

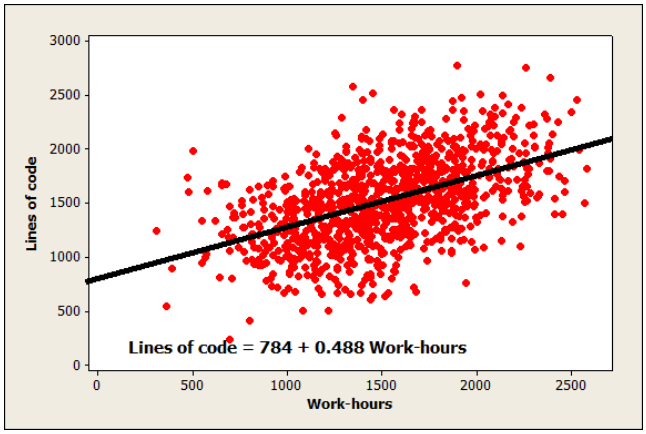


Does it suggest that it takes about 0.5 work-hour to develop one additional line of code?

(The true relationship is 1 WH per LOC.)

simula research laboratory

Regressing WH on LOC



Does it suggest that you get 0.5 lines of code by investing work-hour? This corresponds to 2 work-hours per line of code.

simula research laboratory

Statistical fact: Regression analyses tend to have too low b-values to reflect the true relationship

- The more noise in the dependent variable, the more deflated the b-value.
- The more noise in the measurement, the larger the difference between the TWO regression lines (regressing X on Y and Y on X).
- We may use the difference between the two regression lines as an indicator of noise and interpretation problems.

simula research laboratory

Is there an economy or diseconomy of scale in software development?

(Do we get more or less productive with increasing project size?)

Analyses of (dis)economy of scale

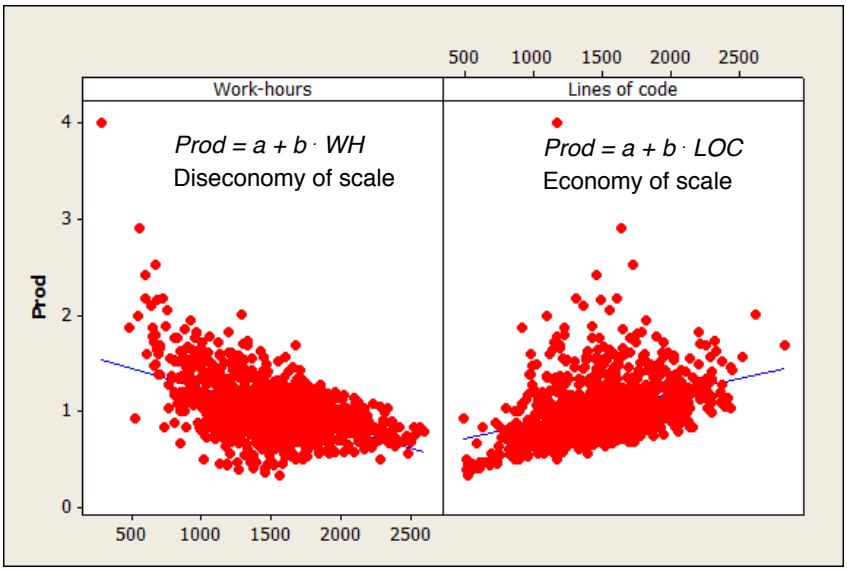
- Relevance:
  - Should we split or join tasks?
  - Are larger project less productive? If yes, this supports, for example, incremental development.
- Research results:
  - Most studies show an economy of scale, but some find linearity. Few reports diseconomy of scale.
- Practitioners' experience:
  - Diseconomy of scale, except for very small task.

Generated data where the true relationship is linearity

- Calculate productivity (PROD) = LOC/WH
- Regression model: PROD = a + b · TS.
- TS is a measure of the task size, e.g., WH, LOC, function points, number of requirements, etc.
  - The above model corresponds to Effort = $aSize^b$, when the tasks size is measured as LOC, function points etc..
- True relationship of generated data:
  - Constant productivity (a=1, b=0)
- Observed relationship: ?

simula research laboratory

The choice of task size measure decides whether we find economy or diseconomy of scale!



simula research laboratory

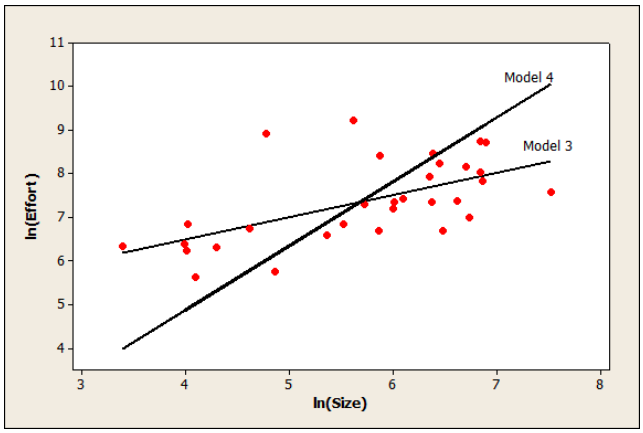
We find the same pattern in reported data set (based on paper to appear in JSS)

Data set	$Effort = a_1 Size^{b_1}$		$Size = a_2 Effort^{b_2}$		
	b_1	"Return on scale"	b_2	"Return on scale"	r
(Jørgensen 1997)	0.52*	EOS	0.68	Linear	0.54
(Desharnais 1988) All projects	0.94	Linear	0.53*	DOS	0.70
(Desharnais 1988) - Cobol projects	0.98	Linear	0.66*	DOS	0.81
(Desharnais 1988) - Advanced Cobol projects	0.99	Linear	0.85	Linear	0.92
(Desharnais 1988) - 4 GL projects	1.05	Linear	0.72	Linear	0.87
(Kitchenham, Pfleeger et al. 2002)	0.67*	EOS	0.78*	DOS	0.73
(Jørgensen 1995)	0.56*	EOS	0.96	Linear	0.74
"Finnish" data set[2]	0.99	Linear	0.71*	DOS	0.75
(Kemerer 1987)	0.81	Linear	0.76	Linear	0.79
(Kemerer 1987)	0.90	Linear	0.74	Linear	0.82
(Hill, Thomas et al. 2000)	1.02	Linear	0.68*	DOS	0.83
(Boehm 1981)	1.02	Linear	0.76*	DOS	0.86
(Jeffery and Stathis 1996)	0.80	Linear	0.97	Linear	0.88
(Miyazaki, Terakado et al. 1994)	O.99	Linear	0.78*	Strong DOS	0.89

EOS = Economy of scale, DOS = Diseconomy of scale

simula research laboratory

Visualization of data from one of the studies (Jørgensen 1997)



Model 3: Effort = a·Size^b
Economy of scale (b < 1)

Model 4: Size = a·Effort^b
Diseconomy of scale (b > 1)

simula research laboratory

We **cannot** trust the previous statistical analyses on economies of scale in SE

- The main reason for the dominance of reporting economy of scale or constant return on scale in SE research is the use of:
  - $Effort = a\ Size^b$ (factor input model), instead of
  - $Size = a\ Effort^b$ (production function)
- If SE, as most other disciplines, had used the production function instead of the factor input model, we would have, almost without exceptions, reported linearity or diseconomy of scale.

A comment on prediction vs. explaining

- In a prediction (e.g., effort estimation) context the regression towards the mean is not really a problem. It is, for example, rational to predict an effort closer to the mean effort with increasing noise (decreased correlation between size and effort).
- If, however, we want to understand or explain the true relationship, e.g., to decide whether we on average will benefit from splitting up or joining tasks, we have a problem with regression towards the mean effects!
- Not easy to see how to conduct trustworthy explanatory analyses of observational data.

Is there an increase in cost overrun with increased project size?

simula research laboratory

Previous studies on project size vs cost overrun

- Most software studies report an increase, while most infrastructure projects report a decrease in cost overrun with increased project size
- Software studies typically measure project size as the actual cost, while infrastructure projects measure it as the estimated cost.
- Can this difference in project size measure explain the difference in reported results?

simula research laboratory

Field data (published in IJPM): Cost overrun vs. size

Study	Size Measure	Analysis	Sample size and domain	Original analysis	Alternative analysis
(Heemstra and Kusters 1991)	ACT	TAB	388 software projects	ICO	-
(Gray, MacDonell et al. 1999)	ACT	TAB	77 software projects	ICO	-
(Hatton 2007)	ACT	REG	957 software projects	ICO	DCO
(Moløkken-Østvold, Jorgensen et al. 2004)	ACT	CAT	42 software projects	ICO	CCO
(Sauer, Gemino et al. 2007)[1]	ACT	TAB	519 software projects	ICO	-
(Yang, Wang et al. 2008)	ACT	CAT	112 software projects	ICO	-
(Dantata, Touran et al. 2006)	ACT	REG	37 rail projects	ICO	CCO
(van Oorschot 2005)[2]	EST	REG	108 software projects	CCO	ICO
(Flyvbjerg, Skamris Holm et al. 2004)[3]	EST	REG	131 infrastructure projects	CCO	-
(Odeck 2004)	EST	CAT	620 road projects	DCO	-
(Hill, Thomas et al. 2000)[4]	EST	TAB	506 software projects	DCO	ICO
(Bertisen and Davis 2008)	EST	REG	63 mining and smelting projects	DCO	-
(Creedy 2006)[5]	EST	REG	231 road projects	DCO	CCO

ACT = Actual cost, EST=Estimated cost, ICO=Increasing cost overrun, DCO=Decreasing cost overrun with increased project size, CCO=Constant cost overrun, TAB=Cross-tabulation, REG=Regression, CAT=Category-based analysis

- We **cannot** trust the previous statistical analyses on increasing cost overrun with increasing project size in software engineering

- If software engineering researchers had used estimated size (budget, cost estimate, effort estimate) as project size variable, we would typically find no difference or decreasing cost overrun with increased project size.

Non-random sampling

Non-random sampling and regression towards the mean

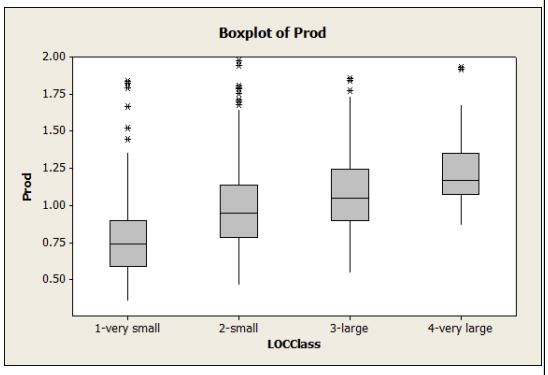
- Regression towards the mean implies that values higher than the mean are likely to have a positive random error (positive noise-value), while values lower than mean are likely to have negative random error.
- If we generate a data set through a non-random selection of observations, we may get a biased sample of random errors
- A biased sample of random errors may mislead the analysis.

Example: Project size vs Productivity

- We divide projects into size groups (very small, small, large, very large) based on the produced LOC.
- This gives four non-random groups and biased “noise”
- Assume, for example, that the “very small” group only include projects less than 1000 LOC.
- This will, amongst others lead to:
  - Inclusion of a project that has a true size of 1200 LOC, but due to forgetting to count some code, it is measured to be 800 LOC (negative noise value, too low measured productivity).
  - Exclusion of a project that has a true size of 800 LOC, but due to counting the same code twice, it is measured to be 1200 LOC (positive noise value, too high measured productivity).
- In short, we would expect the observation of below average productivity for “very small” projects.

An this is exactly what we observe …

- To illustrate the effect of non-random sampling we split the previously generated data set into projects based on their LOC:
  - very small: <1000 LOC,
  - small: 1000-1500 LOC,
  - large: 1500-2000 LOC
  - very large: >2000 LOC
- As before there is NO true increase or decrease in productivity with increasing project size.
- The results are, however, very convincing (p<0.0001).



Non-random sampling: The Winner's curse

- The client does not select randomly among the proposals (the bids), but is more likely to select among those with lowest price.
- Those with over-optimistic cost estimates are more likely to give lower bids than those with realistic or over-pessimistic cost estimates.
- It can be shown that, situations with no underlying bias towards over-optimism, results in observed cost overruns due to selection bias in accordance with:

$$rel = \frac{act - est}{\mu} = \frac{\mu\left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w)}{\mu} = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w).$$

- w = Percentage of average cost, μ = mean estimated cost, est = estimated cost, act = actual cost (more information in my paper on selection bias, see previous slide)

A controlled experiment on selection bias effect

(Strategy: Select the i-th highest cost estimate):

Other evidence in support of selection bias effects in explaining cost overruns

Fields studies report that:

- In-house software development, where there is no selection bias, have no systematic tendency towards cost overrun.
- The higher the focus on lowest price as selection criterion, the higher the cost overruns.
- The "average bid" selection format seems to remove the cost overruns.

Low awareness on selection bias effects

- Experimental evidence on the winner's curse
- The Standish Group (1994) claimed a 189% average cost overrun of software projects.
  - Selection process (page 13 of their report): "*We then called and mailed a number of confidential surveys to a random sample of top IT executives, asking them to share failure stories. During September and October of that year, we collected the majority of the 365 surveys we needed to publish the CHAOS research.*"
- Clearly, the average cost overrun of projects with large cost overruns can be very large, but how much does this say about the software industry?
- Still, this report is the one most frequently quoted on cost overruns in software projects.

Lessons learned - 1

- Future **explanatory** analyses of observational data should avoid go into the regression-towards-the mean traps.
  - Re-read Section 1 in the introduction to regression analysis, e.g., the assumption of fixed variables.
- Results from ordinary regression and related analyses can only be interpreted properly if the level of random error in the independent variable is reasonably low or we know how to adjust for it.

Lessons learned - 2

- If the level of random error in the independent variable is likely to be high (e.g., indicated by a low correlation) we may have to:
  - Conduct controlled experiments (fixed variables)
  - Change into other variables (with less random error)
  - Assess the level of random error (not easy)
  - Use other statistical methods
- There is a high number of alternative statistical methods made for adjustments or avoidance of the interpretations problems. As far as I can see, they are hard to use and/or solve one problem by introducing new ones in typical software engineering contexts.

Lessons learned - 3

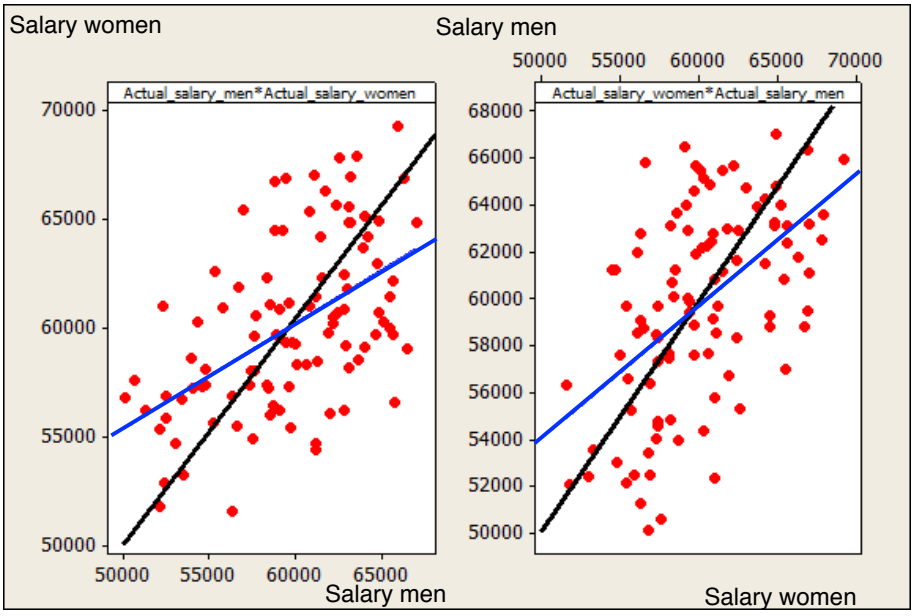
Know what you do, when conducting statistical analyses.

Extra material

Salary discrimination?

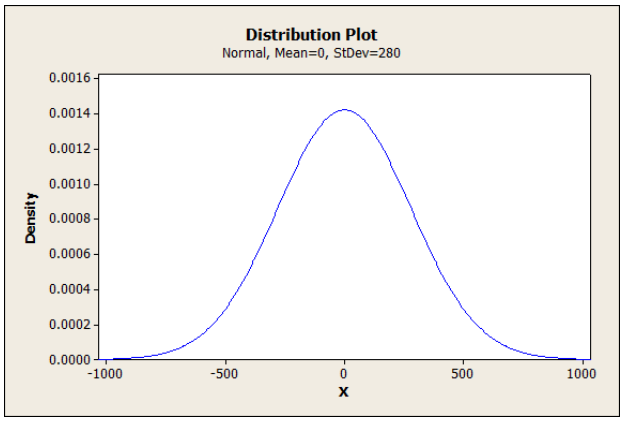
- Assume that there is an IT-company that:
  - Has 100 different tasks they want to complete.
  - For each task they hire one man and one woman (200 workers)
  - The “base salary” of a task varies from 50.000 to 60.000 USD and is the same for the man and the woman completing it.
  - The actual salary is the “base salary” added a random, gender independent, bonus. This is done through use of a “lucky wheel” with numbers (bonuses) between 0 and 10.000.
- This should lead to: Salary of women = Salary of men
- A regression analysis, however, gives that the women are discriminated (paid less) for the above average high salaries!
  - Salary of women = 26100 + 0.56 * Salary of men
- On the other hand:
  - Salary of men = 26900 + 0.55 * Salary of women

simula research laboratory



simula research laboratory

The distribution used to select “noise” numbers



Observed LOC = true_LOC + noise_LOC
Observed WH = true_WH + noise_WH

Productivity (LOC/work-hours)
Skewed, but close to normally distributed