2014

Moore's Law

Domaille, S.

Moore's Law

Susannah Domaille, Jacob Hughes, Samantha Metters and James Sharwin

Project Advisor: Tom Heinzl, School of Computing and Mathematics, Plymouth University, Drake Circus, Plymouth, PL4 8AA

Abstract

Moore's law originally was the observation that the number of transistors on integrated circuits doubles roughly every 18 months. However, many other areas of technology progress with a similar exponential growth. For instance, can one find an analogous law in the context of super-computing? The aim of this paper is to answer this question by showing how a variant of Moore's law emerges from an analysis of the "Top 500" lists of super computers from 1993 to 2013.

1 Introduction

"The number of transistors on integrated circuits doubles approximately every two years" [1].

This is an observation made by, and named after, Intel co-founder Gordon E. Moore in 1965 (see Fig. 1). Despite the name, it should be viewed as an observation and an informed prediction, rather than a law. There is no fundamental law that states how powerful a newly made integrated circuit will be at any given time. Moore wrote an internal paper in which he drew a straight line through five points representing the number of components per square inch on an integrated circuit. He noticed that the number of transistors had doubled every year since the invention in 1958, up until 1964. His paper, *"Cramming more components onto integrated circuits"*, was published in 1965 by Electronics magazine [2]. Since then, his law has been generalised to the claim that computer power is doubling every 18 months.
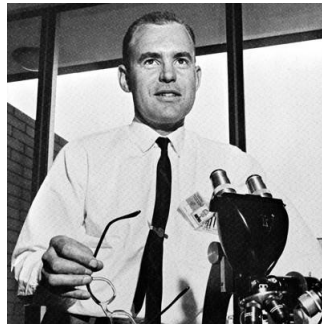
Figure 1: Gordon Moore at Fairchild R & D in 1962 [3].

Moore's law predicts that this trend will continue into the foreseeable future. So far, this prediction has proven to be quite accurate. Because Moore's law suggests exponential growth, it is however unlikely to continue forever. Some studies have shown that within the next few years physical limitations could be reached, meaning the law may become infeasible, unless it is modified.

Some theoretical physicists believe that the rate of increase is already starting to slow [4]. Silicon computer power can no longer maintain the exponential rise as we have reached such a high measure of computer performance that will be extremely hard to improve. Due to this many physicists argue that Moore's Law could flatten out completely by 2022. It is extremely unlikely that transistors could be made smaller and smaller forever, and eventually the limits of miniaturization at atomic levels would be reached. If we went beyond these limits processors would just overheat. We may have another 10 to 20 years before we reach this fundamental (quantum) limit.

There are two main problems we encounter if we make silicon chips too small, heat and leakage. Today a chip has a layer almost down to 20 atoms across, but when that layer gets down to about 5 atoms across major complications will occur.

The heat generated will be so intense that the chip will melt and begin to disintegrate.

Also we will have no idea where the electron is, meaning it could be outside the chip, if quantum theory takes over. To summarise: There is an ultimate limit set by the laws of quantum mechanics as to how much computing power you can achieve with silicon.

2 Regression analysis of Top 500 list of super computers

The information gathered for this regression has been collated from the TOP500 list of supercomputers [5], extending over a 20 year period (1993-2013). Over this period of time we are looking to assess if the performance of supercomputer behaves in a way as described by Moore addressing 'the rate of increase of the number of transistors on an integrated circuit'. We will fit a least squares model to the top 3 supercomputer over this time period, at two intervals per year.

For our statistical analysis of supercomputers, we shall be considering 3 sepa- rate factors: R_{max} being the observed power, R_{peak} the theoretical power of which the supercomputer can reach and the total number of cores within a computer. All the proceeding graphs have been plotted in Rstudio [6].
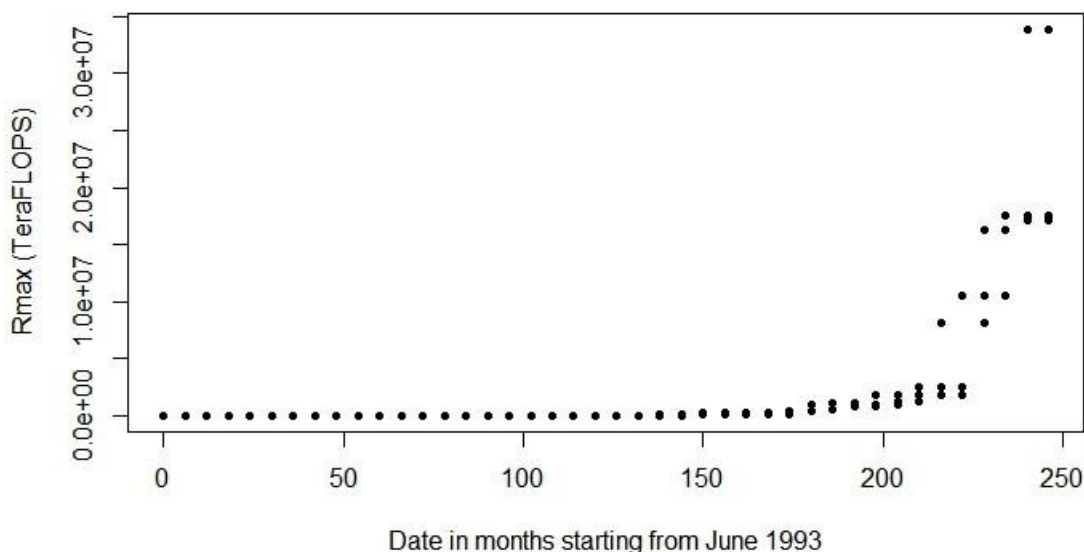

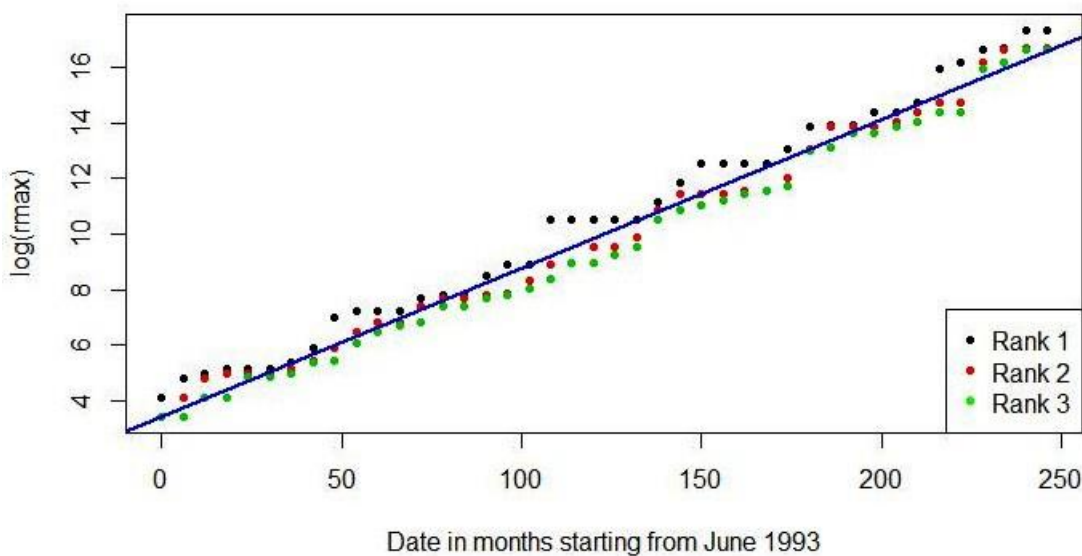**Figure 2:** Plot of Rmax values for top 3 supercomputers in Top500 list since 1993.

2.1 Regression analysis of R$_{max}$

We measure computer power in terms of flops, that is floating point operations per

second. A few years ago, computers had reached a performance of more than a Petaflops (Pflops), where the prefix 'Peta' denotes 10^{15}. In what follows we will choose Teraflops (Tflops) as our units (1 Tflop = 10^{12} flops).

Let us denote the maximum performance by R_{max}. Figure 2 shows a plot of R_{max} against time, starting from the first list of June 1993. R_{max} is the fundamental quantity on which the Top 500 ranking system is based on.

As expected one can see a relatively tiny growth between 1993 and 2009 com- pared to the last 4 year which, taken on their own, would have noticeable exponential growth. It should be noted that we are trying to find a trend and the most logical way of doing so is to linearise the variable in *y* i.e. take the natural logarithm of the R_{max} values. The result is displayed in Fig. 3.

Figure 3: Logarithmic plot of R_{max} values for top 3 supercomputers in Top500 list since 1993

By linearising in the *y* variable we can fit a least squared estimator, visualised as the blue line [12][13], the idea being to minimise the sum of the squared residuals. From observation, we can say there is a linear increase in $\ln(R_{max})$ over time. Com- paring actual data to the linear trend line for ($\ln(R_{max})$), patterns are easily noticed. In particular, we identify a step pattern within the data suggesting that there are sudden leaps in power followed by periods of stagnation. It is particularly obvious for the top computer (blue dots in Fig. 3) and may be rather straightforwardly be explained by new technology becoming available every few years. If one averages over each sam- ple of size 3 the pattern persists. We expect, though, that the steps will less distinct for larger samples (say upon averaging over the top 10 computers of each list).

One particular statistic to note is the R-squared value. The R-squared value is a means by which you can tell how well fit your least squares estimator is fit to the data points and it is calculated as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} ,$$
(1)

where SS_{res} is the sum of squared residuals and SS_{tot} is the total sum of squares. A value of R-squared close to 1 suggests that the model is a good fit and we can be confident in our result(s). Our R-squared value of 0.9796 suggests that our model is a very good fit. Using the summary command in R for this particular model, we obtain the linear equation

$$\ln R_{max} = y = 0.05349x + 3.439 ,$$
(2)

which shall be used later to discover the doubling time.

2.2 Regression analysis of R_{peak}

R_{peak} is the theoretical speed at which the computer can run (measured in Tflops). This is always going to be greater than the R_{max} value, but by not too big a margin. Therefore, it seems logical to expect that the graphs will be rather similar. This is indeed borne out by Fig. 4. Clearly, the graphs of Fig. 2 and Fig. 4 look rather similar
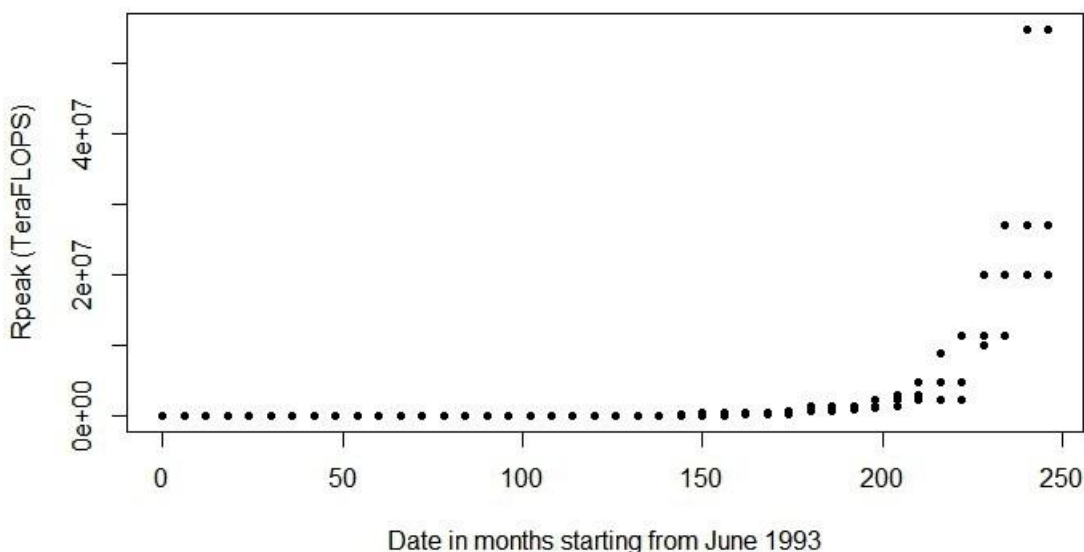
Figure 4: Plot of R_{peak} values for top 3 supercomputers in Top500 list since 1993. (as the observables on the y axes are closely related). Again we linearise the y variable by taking the logarithm of R_{peak} to identify any trends in the data. The result is shown in Fig. 5.
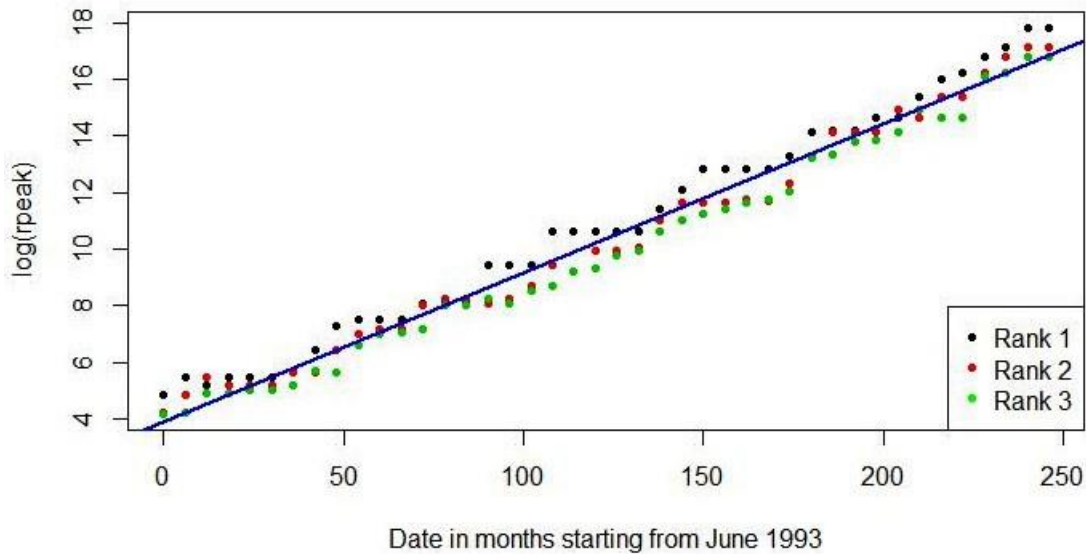
Figure 5: Logarithmic plot of Rpeak values for top 3 supercomputers in Top500 list since 1993.

As expected Figure 5 is very similar to Figure 3 reflecting the similarity of R_{max} and R_{peak}. We also see the step pattern again here. The similarity can be quantified by looking at the equation for the least squares model fit, which is

$$\ln R_{peak} = y = 0.05248x + 3.926 \ , \tag{3}$$

to be compared with (2). Both coefficients in the equation are rather similar, in particular the slopes which coincide to one significant figure. Here the R-squared value is 0.9812, slightly closer to 1 than for the R_{max} data. Again, we can be confident that the fit (3) accurately represents (the trend of) our data.

2.3 Regression analysis of the total amount of cores within a supercomputer

The total number of cores within a super computer may not seem like an obvious predictor for us to set a least squares model on our data, given that the computers are ranked by R_{max} in the top500 list. Some computers ranked 2 or 3 for R_{max} can be ranked higher than position 1 for the number of cores. But here we are not using the rankings as predictor variables, so are not interested in the interaction between rank, core and R_{max}. It thus makes sense to have a look at the time evolution of core number by plotting it against time, see Fig. 6. We still seem to find an exponential pattern emerging. To check, we again linearise in y by considering the log of total core number and plot the result in Fig. 7. Compared to R_{max} and R_{peak} raw data (Fig.s 3 and 5), the data points are more spread out, both horizontally and vertically. This

[190]
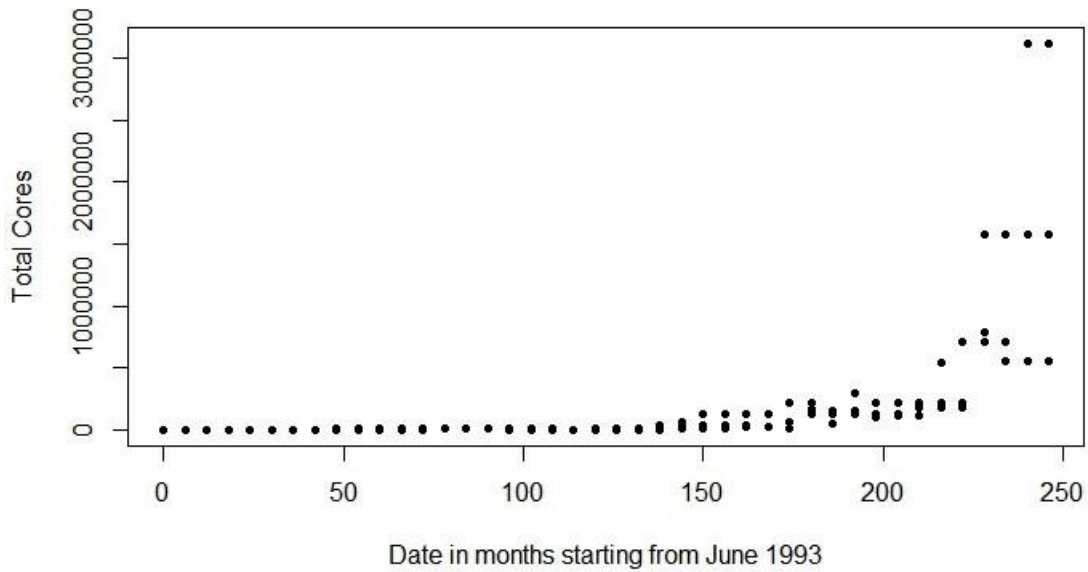
Figure 6: Plot of Total Cores for top 3 supercomputers in Top500 list since 1993

spread could be reduced by considering averages, but in this way we would of course coarse-grain the information. From the raw data, it seems fair to conclude that core number is not as strongly correlated with computing power as R_{max} or R_{peak}. This will be quantitatively corroborated in what follows. To measure the correlation we again
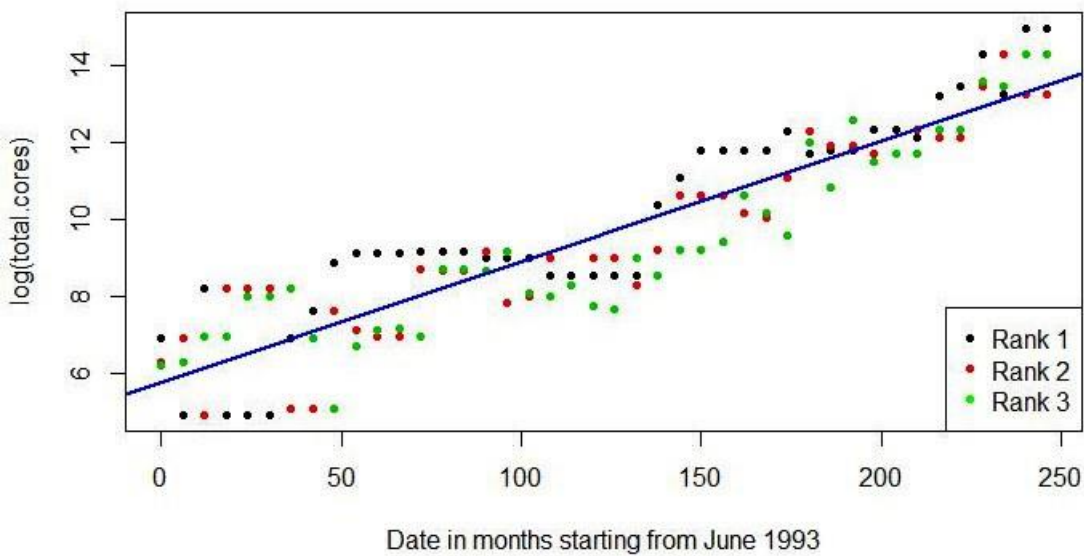
Figure 7: Logarithmic plot of Total Cores for top 3 supercomputers in Top500 list since 1993

perform a linear regression resulting in the line

$$\ln(\text{core number}) = y = 0.03129x + 5.767 \,, \tag{4}$$

with slope and intercept different from those in (2) and (3), but still in the same ball park. The R-squared value of this least squares model, however, is 0.845, hence significantly less than the previous values (0.9796 and 0.9812). Obviously, the linear correlation is somewhat weaker, but still reasonably strong in view of the amount of data points ($d = 63$). Taking this figure as the number of degrees of freedom, the critical value for R, below which one would reject the hypothesis of linear correlation, is only about 0.25. So we are on fairly safe ground to assume a linear relationship also for log core number against time, hence exponential increase of core number with time.

2.4 Doubling time

To calculate the doubling time T_d from the linear models we use the standard relation,

$$Td = \frac{\ln(2)}{m}, \tag{5}$$

where m is the slope in the straight-line fit equations, $y = mx + c$, associated with the logarithmic plots. Our findings can be summarised as follows:

• For R_{max} the value of m is 0.05349, therefore the doubling time is 12.96 months.

• For R_{peak} the value of m is 0.05248, therefore the doubling time is 13.21 months.

• For total core number the value of m is 0.03129, therefore the doubling time is 22.15 months.

3 Conclusion

Fitting linear regression models to our logarithmic plots of R_{max}, R_{peak} and the total core number unveils strong linear relationships with R-values close to unity. Hence, we can assume with confidence that the indicators in question indeed grow exponentially, consistent with Moore's law. It thus makes sense to calculate the respective doubling times.

Our findings are that R_{max} and R_{peak} (measured and theoretical performance) both double at a rate of once every 13 months. This is a large growth rate compared to others in technological development. For example, alternative forms of Moore's Law forecast the number of transistors on integrated circuits doubling every 18 months [7].

When the rate of increase begins to seriously slow, fitting a linear regression line to the logarithmic plot would produce a weak R squared value. This would lead us to conclude that exponential growth was no longer being realised. But as we have found large R squared values there is no indication (yet) that the growth has started to slow.
As said in the introduction, the exponential growth rate is bound to eventually decrease – unless there is an unforeseen major advancement in computing technology in the coming years. Gordon Moore himself stated: "It can't continue forever. The nature of exponentials is that you push them out and eventually disaster happens" [8]. Heat dissipation and power usage are currently the main obstacles for the advancement of supercomputing technology. Supercomputers are housed in massive facilities, with a large amount of specially designed cooling systems and power transformers to keep the behemoths in check. Regarding power consumption, we note that the current record holder for performance, the Tianhe 2 supercomputer, uses 17,808 kW of power [5]. This is a massive amount of power equivalent to the power usage of roughly 5000 homes [9].

One way in which supercomputers could maintain an exponential rate of increase is by forming a larger "grid" of supercomputers, in which multiple supercomputers work together. This is likely to be costly, however, as it requires a large amount of expensive fibre optic cables between remote supercomputer sites [10].

Nevertheless, despite these infrastructure related barriers, the future of supercomputing still looks positive. The next major milestone to reach is a supercomputer capable of operating at an exascale level. Exascale computing refers to a capability of operating at 1 Exaflop (1 Eflop) or more, an Eflop being 1000 Pflops, or 1 Eflop $= 10^{18}$ flops. Intel hopes to be able to deliver this technology by 2018. Many other governmental agencies across the world have similar plans for the development of exascale supercomputers before 2020 [11].

References

[1] http://en.wikipedia.org/wiki/Moore's law, accessed 26-1-2014.

[2] http://jp.barralis.com/articles/Gordon Moore 1965 Article.pdf, accessed 29-1-2014.

[3] http://www.computerhistory.org/semi-conductor/assets/images/400x400/1965 1 1.jpg accessed 4-2-2014.

[4] http://www.computerworld.com/s/article/9226758/Physicist says Moore s Law is collapsing, accessed 4-2-2014.

[5] http://www.top500.org/lists/, accessed 4-2-2014.

[6] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, 2013, http://www.R-project.org

[7] http://dujs.dartmouth.edu/spring-2013-15th-anniversary-edition/keeping-up-with-moores-lawU̇vzGPvI uBQ, accessed 4-2-2014.

[8] http://news.techworld.com/operating-systems/3477/moores-law-is-dead-says-gordon-moore/, accessed 4-2-2014.

[9] http://en.wikipedia.org/wiki/Tianhe-2, accessed 4-2-2014. [10] http://kb.iu.edu/data/angf.html, accessed 1-2-2014.
[11] http://fcw.com/articles/2013/07/17/exoflop-supercomputing.aspx, accessed 2-2-2014.

[12] http://stackoverflow.com/questions/15102254/how-do-i-add-different-trend-lines-in-r, accessed 11-2-2014.

[13] https://sites.google.com/site/manabusakamoto/home/r-tutorials/r-tutorial-3, accessed 11-2-2014.

[14] J.J. Faraway, *Linear models with R*, Chapman and Hall/CRC, 2005.

[15] P. Hewson and J. Eales, *Problems in regression modelling*, Lecture Notes 2010-2013, compiled October 9, 2012.