The Ballmer Peak: An Empirical Search

Clinical Study

Twm Stone
A Real Job
The Private Sector
Cambridge, UK
`twm.stone@cantab.net`

Jaz Stoddart
A Real Scientist
Royal Botanic Gardens, Kew
London, UK
`js2231@cantab.ac.uk`

Abstract

The concept of a ‘Ballmer Peak’ was first proposed in 2007, postulating that there exists a very specific blood alcohol content which confers superhuman programming ability. More generally, there is a commonly held belief among software engineers that coding is easier and more productive after a few drinks. Using the industry standard for assessment of coding ability, we conducted a search for such a peak and more generally investigated the effect of different amounts of alcohol on performance. We conclusively refute the existence of a specific peak with large magnitude, but with $p < 0.001$ find that there was a significant positive effect to a low amount of alcohol—slightly less than two drinks—on programming ability.

Keywords Alcohol · Problem solving · Cognition · Software engineering · Programming · Coding

ACM Reference format:
Twm Stone and Jaz Stoddart. 2024. The Ballmer Peak: An Empirical Search. In *Proceedings of SIGBOVIK, Pittsburgh, PA USA, April 2024 (SIGBOVIK’24)*, 7 pages.

1 Introduction

There has long been a belief among programmers that their coding ability is significantly improved after a couple of drinks. Although there has been significant previous work in this area—notably showing a minor beneficial impact of alcohol on creative problem solving (Benedek et al., 2017), and showing a detrimental impact of a particular (high) level of inebriation on novice coders (Brabrand et al., 2024)—there has not been any direct scientific investigation of the effect of differing levels of alcohol intoxication on coding ability.

It was posited in 2007 by renowned popular science author Randall Munroe, in the adjacent comic (Munroe, 2007), that there existed a peak of width approximately 0.01%, centred on 0.1337% blood alcohol content, which confers superhuman programming ability. More recently the intrepid comedic minds of Mitchell, Webb et al. explored the huge potential benefits to all human activities of having a very precise level of inebriation (slightly less than two drinks (Mitchell, Webb et al, 2010)) and an insightful Danish-language documentary by Vinterberg demonstrated the wide-ranging positive effects of maintaining a blood alcohol level of above 0.050% (Vinterberg, 2020) .

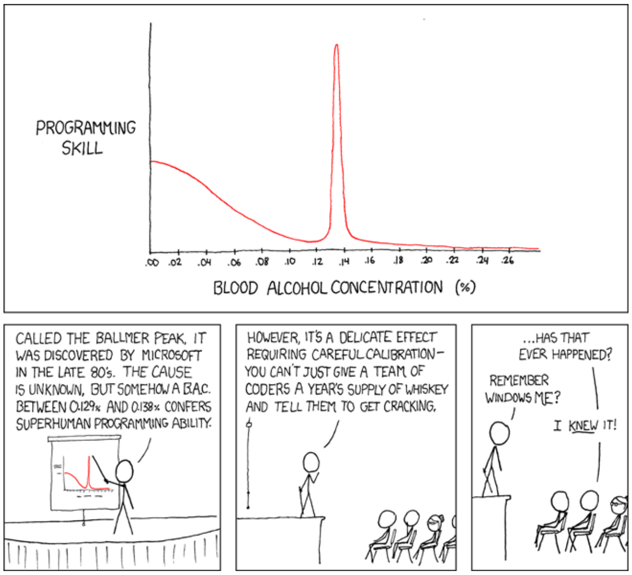

PROGRAMMING SKILL
.00 .02 .04 .06 .08 .10 .12 .14 .16 .18 .20 .22 .24 .26
BLOOD ALCOHOL CONCENTRATION (%)
CALLED THE BALLMER PEAK, IT WAS DISCOVERED BY MICROSOFT IN THE LATE 80's. THE CAUSE IS UNKNOWN, BUT SOMEHOW A B.A.C. BETWEEN 0.129% AND 0.138% CONFERS SUPERHUMAN PROGRAMMING ABILITY.
HOWEVER, IT'S A DELICATE EFFECT REQUIRING CAREFUL CALIBRATION– YOU CAN'T JUST GIVE A TEAM OF CODERS A YEAR'S SUPPLY OF WHISKEY AND TELL THEM TO GET CRACKING.
...HAS THAT EVER HAPPENED?
REMEMBER WINDOWS ME?
I KNEW IT!

Figure 1: *Apple uses automated schnapps IVs.*

In this study, we attempt to establish the existence of the Ballmer Peak and more generally the existence, magnitude and location of any benefit to programming conferred by alcohol intoxication. We additionally aim to gain a qualitative understanding of the different ways in which alcohol consumption impacts the different facets of programming, and to guide future research in this area.

2 Methodology

2.1 Test subject

The test subject was chosen so that they had substantial previous experience in both competitive programming and using the chosen programming language professionally, but had never used the LeetCode platform before. This minimized the impact of the practice effect on problem completion rate whilst also preserving the novelty of the problems.

The selected candidate was a male, 27 year old Caucasian of Northern European descent, currently employed as a Software Engineer. Weighing around 65 kilograms and being approximately 170cm tall, he is generally fit and well with no formally diagnosed cognitive or physical conditions which might impact problem solving ability or result in abnormal metabolism of alcohol.[1]

2.2 Means of intoxication

To achieve a given level of percentage blood alcohol content (%BAC) we needed to administer potentially large amounts of alcohol to the test subject but in a predictable manner. Mitchell, Webb et al. suggested several options for administration—Cointreau enemas, intravenous claret, enteral Special Brew—which were deemed infeasible by the Ethics Board due to lack of available medical expertise (Mitchell, Webb et al, 2010).

Frequent ingestion of liqueur chocolates was considered as the method of intoxication but the required quantities and rate of consumption was deemed financially impractical and liable to induce emesis. It was thus decided that the most cost-effective means of administering ethanol was to imbibe it. This was done with a combination of beer, cider, premixed cocktails, and something which could generously be called 'homemade punch', the proportions of which depended on the target %BAC. These dilute sources of ethanol were chosen as pure ethanol is quite unpalatable, far more expensive to purchase and infeasible for us to produce ourselves, owing to the difficulty associated with separation from the azeotrope it forms with water.

2.3 Assessment of coding ability

To assess coding ability in this investigation, we opted to use the industry standard—the speed of solving LeetCode problems.[2] The chosen language for solving the LeetCode problems was a relatively new one—akin to pseudo-pseudocode with duck typing—introduced by Van Rossum and Drake, since it was the language with which the test subject had the greatest familiarity (Van Rossum and Drake, 2009). The LeetCode problems were filtered using inbuilt functionality to select at random exclusively from the subset of problems ranked as "Easy"; problems requiring a subscription or the use of other languages were also discarded, as were problems previously completed.[3] The problems were solved by typing directly into the provided in-browser IDE; no code completion, debugging tools or AI assistance were used, although it did have syntax highlighting.[4]

Since our budget did not extend to a premium LeetCode subscription, we did not use the associated debugger but instead simulated the problem-solving workflow of a professional software engineer by

- Adding lots of `print()` statements, running all of the provided test cases, and looking at `stdout`.
- Getting annoyed and looking up how various language features actually work on Stack Exchange.

The time to solve was taken from the point the page loaded to the point the submission stats loaded for a correct solution, including all reading, coding, debugging, and run time. No attempt to measure the asymptotic behaviour of the solution—in particular, as long as the platform accepted a solution and did not return a Time Limit Exceeded error, it was considered solved even if the algorithmic complexity was not as good as required in the problem description—and no other measure of code 'quality' was quantified, although observations were made by the author contemporaneously.[5]

2.4 Recording of blood alcohol content

Blood alcohol content (%BAC) was recorded using a BACtrack C6 electronic breathalyser,[6] which was recently calibrated and claims an accuracy of ±0.001%BAC. The accuracy of the breathalyzer was verified post-calibration through multiple means.

First, the zero point reading was verified through testing on sober volunteers.[7] Repeated readings taken on them consistently returned values of 0.000%BAC.[8] Four sober volunteers were then each given several pints of approximately equal strength beer or cider (Fuller's London Pride, 4.7% or Aspall Cyder, 4.5%)[9] and periodically breathal-

[1] Due to requirements of the Ethics Board, the test subject was the first author.

[2] `https://leetcode.com/`

[3] Also any problem involving binary trees on account of the test subject not being bothered to learn how they work.

[4] This was mildly useful on a number of occasions, although it did not prevent the test subject spending 10+ minutes failing to realise he had misspelled `lambda` as `lamdba`.

[5] We originally used Microsoft Word for this but it kept putting the entire document into 8pt Times New Roman whether we wanted it to or not.

[6] BACtrack Breathalyzers / KHN Solutions Inc., San Francisco, USA

[7] Here sober is defined as no alcohol having been consumed within the preceding 24 hours.

[8] It was observed during later testing that it always rounded down readings below 0.007% which contradicts the manufacturer claimed precision.

[9] Due to availability at The Old Wheatsheaf, Enfield.

ysed until a peak value was reached.[10] The expected peak %BAC for each volunteer was calculated from known volume of consumed alcohol and expected blood volume using Equation 1, which roughly matched the observations taken using the breathalyser.

$$\%BAC = 100 \times \frac{V_{alcohol}}{V_{blood}} = 100 \times \frac{V_{alcohol}}{M_{patient} * B} \tag{1}$$

where $V_{alcohol}$ is the volume of alcohol consumed in ml, V_{blood} is the volume of blood in ml, $M_{patient}$ is the mass of the patient in kg, and B is the gender and age appropriate value for the average blood volume in $mL.kg^{-1}$ (Nadler et al., 1962).[11]

Repeated readings taken from the same subject had a spread of $\pm 0.002\%$, in line with the advertised accuracy of the breathalyzer.

2.5 Data collection

Data collection took place approximately every second night for forty days and forty nights. The test subject was inebriated to a given level and then, after a pause to minimise residual alcohol on the breath,[12] began solving problems. The %BAC of the test subject was monitored through measurements taken immediately before and after each problem was attempted. After the subject became bored with coding,[13] he resumed drinking and the process was repeated up to several times.

3 Results

A total of 100 problems were attempted[14] over the course of forty days and forty nights preceding the SIGBOVIK '24 conference, on 15 separate occasions. The results were distributed as follows. Note that there is a bias towards data collection in the lower ranges of %BAC; owing to the average time per problem being greatly reduced more were completed before more drinks were consumed.

Table 1: Sample distribution

Range	Sample size	Mean time /s
$x = 0.000\%$	22	394
$0.000\% < x \leqslant 0.025\%$	16	402
$0.025\% < x \leqslant 0.050\%$	20	249
$0.050\% < x \leqslant 0.075\%$	12	310
$0.075\% < x \leqslant 0.100\%$	11	610
$0.100\% < x \leqslant 0.125\%$	6	614
$0.125\% < x \leqslant 0.150\%$	8	1204
$0.150\% < x \leqslant 0.175\%$	3	2550
$0.175\% < x \leqslant 0.200\%$	2	3281

Our first figure, below, is a scatter plot of the raw data—showing completion time against %BAC.

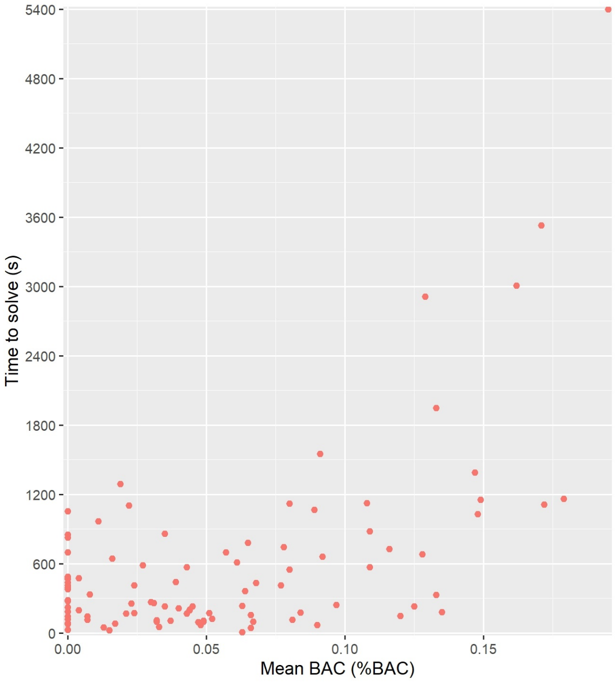


Figure 2: Time to solve in seconds against mean %BAC.

The next two figures presents the same data, but with a quadratic line of best fit and a 95% confidence ribbon. This model was chosen from many potential ones (linear, quadratic, exponential, cubic, power, etc.) since it matched most closely; it is fit with $p < 0.001$. Figure 3 presents the raw data and has a minimum at (0.043%, 222s) and y-intercept (0.000%, 466s), whereas Figure 4 presents "adjusted time" (see Limitations) against %BAC in the same way, with a minimum at (0.047%, 64.5s) and y-intercept (0.000%, 214s).

[10] They were all former university students and thus experienced drinkers, mitigating any risk factors that may have been grounds for ethical concerns.

[11] B for an adult male = $75mL.kg^{-1}$ and for an adult female = $65mL.kg^{-1}$

[12] Including efforts to wash the mouth out with water.

[13] Or was observed buying crisps, or trying to remember the Oscar-winning films of the 70s.

[14] Of the 100, 99 problems were completed and 1, attempted at $\sim$0.2%BAC, was not owing to the the test subject succumbing to unconsciousness after half an hour. The recorded completion time of 90 minutes was projected based on a sober assessment of the problem and the progress made.

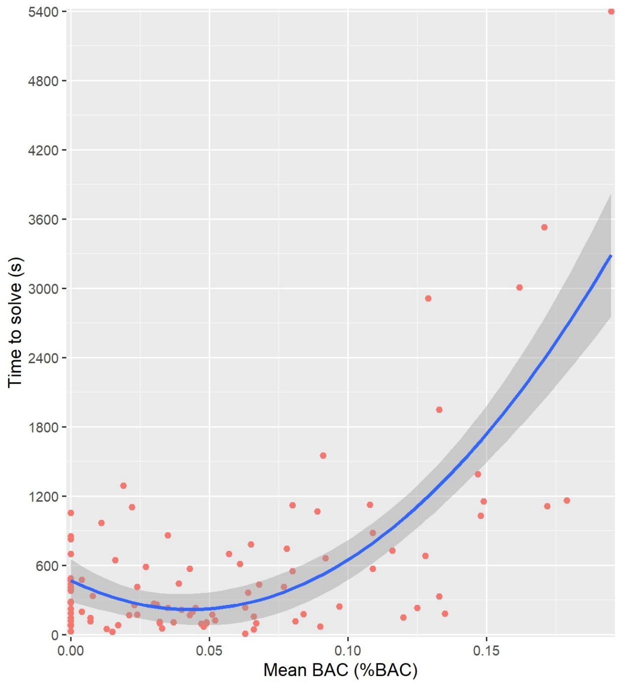


Figure 3: Time to solve against mean %BAC.

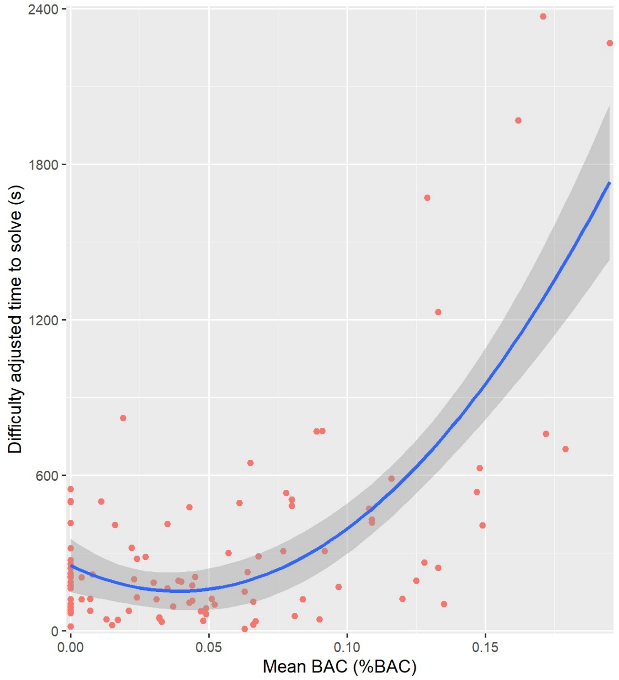


Figure 4: Difficulty-adjusted time against mean %BAC.

In order to visualise whether there was a "practice effect"—where performance improved over time across all problems irrespective of intoxication—Figure 5 was produced. The relative difference between observed and predicted time to solve was a metric calculated to attempt to account for %BAC and thus allow for trends in performance over the problem set to be seen. The relative difference between observed and predicted time was calculated as:

$$\mathit{Difference}_{relative} = \frac{T_{observed} - T_{predicted}}{T_{predicted}} \qquad (2)$$

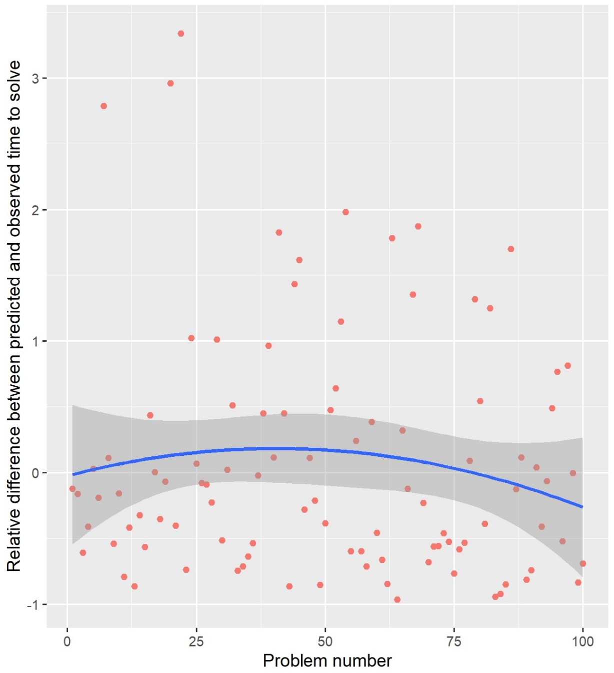


Figure 5: Prediction error throughout the study.

4 Discussion

4.1 Results and Implications

Over the course of this study the test subject descended through, at least once, every possible %BAC between 0.000% and 0.200%. Despite this, no spike in perceptual or actual competence was observed from problems suddenly becoming easier at any given level of inebriation. Thus, we believe we can conclusively rule out the existence of the Ballmer Peak as originally formulated.

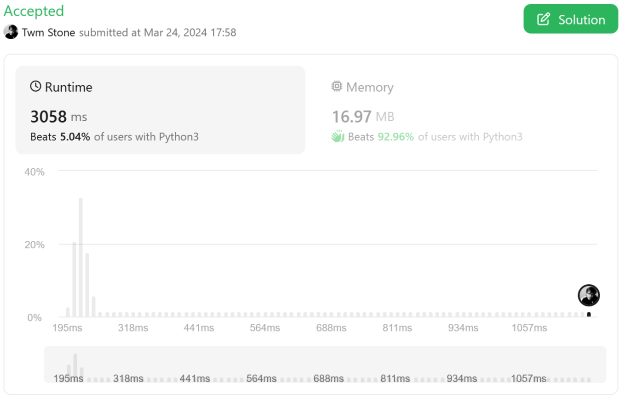
However, looking at Figure 3 it is clear that a moderate amount of alcohol has positive effects on problem-solving speed. The peak of this effect is at about 0.043%BAC, where the test subject was able to solve problems around 45% faster than while sober. To get to this level requires slightly less than two standard drinks—that is, around a pint and a half of 5% beer, a glass and a half of rosé, or three shots. Additional alcohol intake beyond that consistently worsened performance, with only slightly higher

%BAC having parity with sobriety and then higher levels having an increasingly negative effect. After the initial peak there was no positive impact at any point for a marginal drink.

A further observation is that alcohol intake increases spread but does not necessarily lower the ‘best case’ solving speed. One of the fastest problems solved during this study was at 0.090%BAC, the equivalent of having 6 straight shots of vodka.[15] This is in accordance with the notes taken contemporaneously; there was a perception that sometimes one can just "see it" and get the solution straight away; this ability did not actually seem to be affected by alcohol intake less than a very high level. However, as soon as the problem required debugging or trying a different approach the ability of the test subject regressed significantly. This was due to a combination of continual typographic and logical errors—each requiring some amount of effort to fix, the fix itself potentially adding more—but also the test subject remained convinced for longer that ‘this approach is basically perfect I just need to fiddle with the algorithm a bit’ rather than trying a different approach.

For the sober problems the test subject made more of an attempt to use so-called ‘idiomatic Python’ in the hope of getting a pithy one-line solution as fast as possible. This was sometimes successful but often wasted a lot of time fiddling with list comprehensions[16] although it did reliably place him in the top 95% of submissions (see below).

The possibility of a practice effect was considered. Once a line of best fit had been determined for the full data set, we measured the deviation from this ‘predicted time‘ for each of the problems solved. The results, Figure 5, of this analysis suggest there is no significant practice effect in our dataset. This conclusion is reached given the best fit for this data was a quadratic curve rather than a line with negative gradient. Additionally, the confidence ribbon entirely overlaps the line $y = 0$; indicating the observed values never significantly deviate from predicted values made with a model that assumes no practice effect.



Accepted

Twm Stone submitted at Mar 24, 2024 17:58

Runtime

3058 ms

Beats 5.04% of users with Python3

Memory

16.97 MB

Beats 92.96% of users with Python3

Figure 6: Somehow, 5% of users did worse than this...[17]

4.2 Limitations

Assessing how long a problem ‘should’ have taken was difficult; LeetCode does not provide average solution time for its problems, only ‘acceptance rate’, i.e. what proportion of submissions for the problem were correct. This was used as a proxy for difficulty in our ‘adjusted time’ but might reasonably also be considered to reflect how many traps or tricks there were in the question. We directly divided the real completion time by the acceptance% to adjust but one might be sceptical that is realistic—it is improbable that "add two numbers together" (#2235, 87.6%) is really half the difficulty of "find a substring length m which repeats k times" (#1566, 43.0%).[18] Figure 4 shows the impact of adjusting for difficulty in this manner. The location of the peak is almost unchanged and the overall data distribution is similar. Alternatives were considered for measuring difficulty but getting sober volunteers to solve enough of the problem set to have meaningful results on this was deemed infeasible within the time constraints of the study.

The approach to measuring %BAC used during this study may have impacted the accuracy of results. It should be noted that whilst it might have been possible to complete this work without the assistance of a breathalyser, using estimates based upon Equation 1, the use of one is likely to have increased accuracy and ease with which %BAC was measured. Estimates based solely upon blood volume are subject to introduced errors from assumptions made regarding rates of metabolism, efficacy of intake alcohol into blood, body fat distribution and proportion, and, of course, blood volume (Cederbaum, 2012) (Nadler et al., 1962). However, breathalyzers *are* known to over estimate %BAC immediately following consumption and so this

[15] This clearly had nothing to do with the fact that the problem in question—return true if there are three consecutive odd numbers in this integer array—was orders of magnitude simpler than some of the other problems in the so-called ‘Easy’ category.

[16] For example, the solution for 2404: Most Frequent Even Element was the following concise yet readable code:

```
def mostFrequentEven(self, nums: List[int]):
    evens = sorted([elem for elem in nums
        if elem % 2 == 0])
    counts = {thing: evens.count(thing) for
        thing in evens}
    return -1 if not counts else sorted([elem
        for elem in counts if counts[elem] ==
        max([counts[thing] for thing in
        counts])])[0]
```

[17] Astute readers will have noticed that the x-axis labels are not consistent with the achieved runtime—in other words, this result was off-the-charts bad.

[18] In particular, this second problem requires not only adding but also counting, and we know from (Linderholm, 1972) that, for the superstitious, the combination of the two requires dealing with both cardinals and ordinals.

may present a further source of inaccuracy—though measures were taken to mitigate this, such as waiting between initial consumption and starting the test and washing the mouth out with water.

Alternative more accurate approaches to %BAC measurement were not considered due to the associated prohibitive costs, impracticality, and ethical issues presented by repeated frequent sampling of blood and urine.

4.3 Further confounders

This study was not double-blinded or indeed blinded at all. The test subject knew[19] how much alcohol they had consumed. Even worse, given that they had taken the measurement of %BAC themselves and were aware of previous results, there may have been a placebo effect around how hard the test subject expected the problem to be, which could have informed their approach.

Increased tolerance of the subject to alcohol over the course of the study was not accounted for. Whilst initial data gathering was performed after pre-existing social events / pub trips, it was quite quickly established that this level of drinking was much too slow to realistically reach above $\sim$0.08%BAC. In order to get data for higher ranges (including, crucially, the originally postulated 0.1337% peak) the test subject had to intentionally drink large quantities rapidly on multiple occasions. This was observed to have a significant positive effect[20] on his ability to withstand high levels of inebriation during subsequent research.[21]

Tiredness and caffeine intake may have influenced the ability of the test subject to solve the problems. Whilst the test subject was never knowingly under-caffeinated some of the problems were solved quite late at night. The test subject's sleep cycle *did* suffer adverse consequences from this research—hence much of the data was gathered on Friday or Saturday nights to minimize the impact on their day job—and it is possible data gathered towards the end of the study was impacted negatively by this.

Prolonged alcohol consumption followed by periods of unconsciousness in which one does not drink water causes changes in composition of the blood. Of these changes, it is the reduced volume from dehydration and the presence of alcohol metabolites—primarily acetaldehyde, as well as additional congeners such as tannins and other phenolic compounds—which are believed to be the principal causes of a hangover (Mackus et al., 2020). Given the nature of the study and the concentrated periods of data collection throughout the forty days and forty nights of study, it is possible that the test subject achieved a state of both being drunk and hungover at the same time. Since this was not a study aimed at examining the effects of programming while hungover, we chose to ignore any impacts this might have had.

5 Conclusions

From this brief case study, we believe there is sufficient evidence to conclude that the Ballmer peak, sensu Munroe (2007), does not exist. The absence of a specific narrow peak for improved programming ability does not however discount the more widely held beliefs of general improvements in performance after very nearly two drinks.

To that end, our work supports the hypotheses of Mitchell, Webb et al. and we believe that the high importance of the subject matter means that further work to replicate this study on a larger scale is thus needed.[22]

Cheers.

5.1 Additional conclusions

In addition to our primary areas of research, we discovered various other items of scientific value:

- Recording data for research while becoming progressively more inebriated is challenging.[23]
- The UK drink-driving limit is dangerously high. The test subject could literally[24] five-and-drive—drink 5 beers over the course of an evening and reasonably expect to be under the legal limit while leaving the pub. This is not to say that the test subject could reasonably expect to be safe to drive in this condition, just that it would be legal.
- It was learned that for the dial in a fridge "2" means "2 power" and not "2 °C" and the fridge will happily chug along at 6+ °C. Furthermore milk stored at 6+ °C will spoil and spoiled milk actually isn't okay to drink. Unintentionally drinking several glasses of slightly off milk will make you rather ill.[25]

5.2 Suggestions for further research

We have identified several promising avenues for continuing research in this area. It seems likely that the optimal level of alcohol consumption is different for design, cod-

[19] At least, if they were paying attention.

[20] From a certain point of view...

[21] At least perceptually, their ability to focus without feeling ill was improved substantially. The fact that that the test subject ran a marathon mid-way through the study and then ceased training may have also impacted this.

[22] Additionally, we currently lack understanding as to the mechanistic, psychological and metabolic processes which have led to this result. There are large potential benefits to the field of software engineering from improving our knowledge of how to exploit this effect.

[23] See bit.ly/98K8eH for more evidence on this.

[24] Although did not!

[25] Spoiled milk typically contains some small amount of alcohol and thus there may be grounds to return to the study of spoiled milk as a low concentration source of alcohol for maintenance of %BAC at the optimal level. Products such as Kumis contain mild levels of alcohol and there are vodkas now distilled purely from milk based alcohol.

ing, testing, debugging, documentation, etc. All of these could be interesting areas to explore.

The immediate applicability of this research to software engineering is also potentially limited, since professional coding work is likely to require working with a larger codebase, not having the entire context for the problem in your head, and negotiating with stakeholders. Each of these could favourably or adversely be impacted by alcohol intake.

Auto-brewery syndrome is a rare condition in which stomach flora can produce a constant supply of alcohol keeping the afflicted individual in a permanent state of intoxication (Din et al., 2020). Further work may be required to induce auto-brewery syndrome to see if one can achieve a constant level of inebriation sufficient to exploit the findings of this work and improve programming performance.

Further research will commence as soon as we secure the funding to replenish Twm's drinks cabinet.

6 Acknowledgements

We would like to thank the following people, without whom this paper would have been slightly worse:

- Amanda Chua and Kevin Lim for procuring supplies, for helping with data collection and for advice on our experimental design.
- Our Ethics Board, Tom Flynn and Rachel Newhouse, for rubber-stamping all of our dubious decisions.
- Joonatan Laulainen and Dan Robertson, for steering us clear from even worse ideas for a study, for convivial ideation and for moral support.
- Nephele Penny, Jack Rickard and Oliver Shenton for assisting in the measurements of calibration accuracy of our equipment.
- Clara Ding, Tim Coulter, Lily Mills, and Dmitri Whitmore, for putting up with Twm drinking for this study while taking part in a video game tournament.
- Betty la Chatte for guidance on creative typography.
- Herbie Bradley, Patrick Kennedy-Hunt and Alexandra Souly for proof-reading and advisement.

References

Mathias Benedek, Lisa Panzierer, Emanuel Jauk, and Aljoscha C. Neubauer. Creativity on tap? Effects of alcohol intoxication on creative cognition. *Consciousness and Cognition*, 56:128–134, 2017. ISSN 1053-8100. doi:https://doi.org/10.1016/j.concog.2017.06.020. URL `https://www.sciencedirect.com/science/article/pii/S1053810016303713`.

Claus Brabrand, Nanna Inie, and Paolo Tell. Programming under the influence: On the effect of Heat, Noise, and Alcohol on novice programmers. *Journal of Systems and Software*, 210:111887, 2024. ISSN 0164-1212. doi:https://doi.org/10.1016/j.jss.2023.111887. URL `https://www.sciencedirect.com/science/article/pii/S0164121223002820`.

Randall Munroe. Ballmer Peak. *xkcd*, 2007. URL `https://xkcd.com/323/`.

David Mitchell, Robert Webb, James Bachman, Abigail Burdess, Mark Evans, Sarah Hadland, and Stuart Scudamore. The Inebriati. In *That Mitchell and Webb Look*. BBC, 2010.

Thomas Vinterberg. *Another Round*. Zentropa Entertainments, Film i Väst & Topkapi Films, 2020.

Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Samuel B Nadler, John H Hidalgo, and Ted Bloch. Prediction of blood volume in normal human adults. *Surgery*, 51(2)):224–232, 1962.

Carl E. Linderholm. *Mathematics Made Difficult*. World Publishing, 1972. ISBN 978-0-529-04552-2.

Arthur I Cederbaum. Alcohol metabolism. *Clinics in liver disease*, 16(4):667–685, 2012. doi:https://doi.org/10.1016/j.cld.2012.08.002.

Marlou Mackus, Aurora JAE van de Loo, Johan Garssen, Aletta D Kraneveld, Andrew Scholey, and Joris C Verster. The role of alcohol metabolism in the pathology of alcohol hangover. *Journal of Clinical Medicine*, 9(11): 3421, 2020.

Asim Tameez Ud Din, Faran Alam, Farooq Mohyud Din Chaudhary, et al. Auto-brewery syndrome: a clinical dilemma. *Cureus*, 12(10), 2020.