A/B microtesting - exploring the incremental value created by testing more rapidly with a lower threshold of acceptance

A/B microtesting - a testing strategy that involves increasing the volume of testing in a given period while lowering the acceptance threshold.

This investigation comes out of a conversation about A/B Testing with Richard Fergie. In the course of this conversation, he said “But I think, in general, people are better off increasing their test velocity”.

Instinctively, this seemed right so I wanted to see if I could create a model to see if this is backed up or not.

How most A/B testing works

This is a bit of a straw man, but even if you do things differently I think you’ll recognise these patterns within the industry. At the moment, A/B tests happen one at a time, normally with a single change being tested against a control group (tests for media click through rates may be different). A test is run, generally for about two weeks, and then an analysis is run on the results.

The way almost all tools do this is to apply a frequentist analysis - there is a null hypothesis (“there is no difference in performance between the two cohorts”), and the data either proves or disproves this null hypothesis. In the ‘disprove’ case, this means that there is a statistical likelihood that any difference in performance is due to differences in customer interaction with the two different tests. The generally recommended standard applied to this is 95% certainty - anything less than this is deemed by the calculation tool to be ‘not statistically significant’.

This is where things start to break down. What happens if one of the test variants shows a performance increase, but only attains a significance of 85%? In some businesses, someone just decides that that’s enough and they accept the changes, and in others, there is a more strict adherence to what the tool says and the changes are rejected (there is a third option, which involves testing until you get the results you want, which needless to say is Not Good in a frequentist approach).

Neither of these outcomes is particularly satisfactory - on one hand, why bother calculating significance if you’re just going to ignore it? On the other, it seems wrong to turn down what seems to be an improvement just because you cannot be 95% certain that you will create more value.

It’s always struck me as slightly curious that most tool creators default to such a strict level of confidence. As an (ex) scientist, it makes sense - if I am to distribute my results and conclusions to my peers, I want to be as sure as I can that they are meaningful and will not waste people’s time in replicating something, or end up poisoning a bunch of people with a looser interpretation of a medical trial. However, in *most* cases, business is not science. In the case of something like a website, businesses can be more less averse in making decisions that drive value, knowing that in the aggregate the upside of the decisions they make will outweigh the downside of being mislead by data. Of course, this breaks down if you have a very small sample size, which is perhaps why e-commerce teams act in a more cautious way.

In other words, by seeking to eliminate Type I errors (ie seeking to minimise the risk of thinking something works when it doesn't), teams are ignoring Type II errors (ie, allowing real improvements to be ignored because they have been unable to prove them to their satisfaction).

(By the way, if you have read through the above and are thinking that this is why a Bayesian approach makes more sense for A/B testing, I AGREE - the tool on the Cufflinks Consulting website performs a Bayes calculation, and I've written more about that on the blog as well.)

How A/B microtesting should be beneficial

If A/B microtesting does indeed produce more value, we should see benefit from two aspects

1. You can perform more tests and hence integrate more improvements.
2. By lowering the significance acceptance threshold you will miss out on fewer real improvements.

Parts one and two go by together necessity - running more tests involves running shorter or more compartmentalised tests (ie tests with fewer customer impressions), which means that for the same uplift rate the results will have a lower significance than a longer test. At this point, it should also be recognised that there is also a point 3:

3. There is an increased risk that accepted tests will be a false positive (ie, an accepted test does *not* actually create more value from customers).

Creating a model for A/B microtesting

I want to create a model to compare A/B microtesting with a slower, more conventional strategy. I am not going to try to put a monetary value on this - average order value varies far too much to try that - but rather, I'm going to end up with a ratio so we can say a/ which of these approaches is more valuable and b/ by how much, given my initial set of parameters.

These are some initial assumptions:

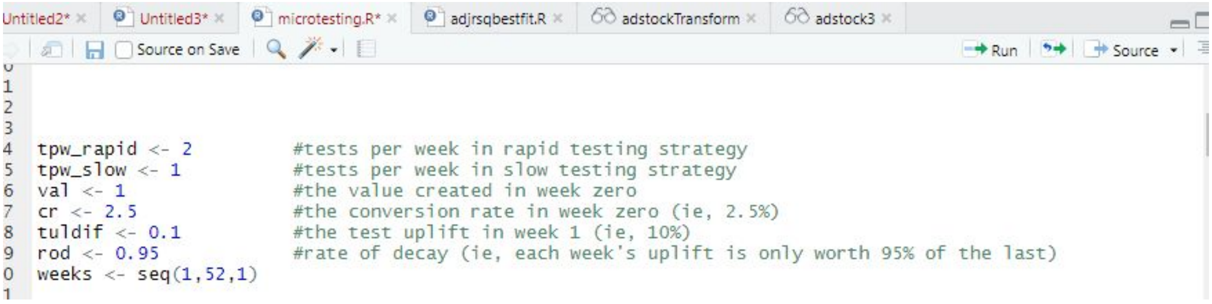
1. The team is able to order tests such that the most valuable ones happen first. This has the effect of each test being less valuable than the previous - ie, a decay over time.

2. The team is good enough that *on average* the test results will be positive (ie, they are identifying customer problems and fixing them, rather than just randomly trying stuff). This means we can use an uplift term for each strategy and account for Type I (incorrectly accepting a losing test) errors by reducing the uplift value by the probability of a Type I error (ie, Type one errors cause zero uplift, positive or negative).
3. There exists the technical capacity to implement tests and changes in both models.
4. The effects don't change by time of year ie I can apply the same calculations over 52 weeks.

You may disagree with any / all of these (for instance, it could be argued that conducting more tests raises the ceiling for improvement as it identifies behaviours that resonate well with customers) - I have detailed my calculations below so I invite alternative models for discussion.

Calculations I - create a per test uplift multiplier that diminishes over time.

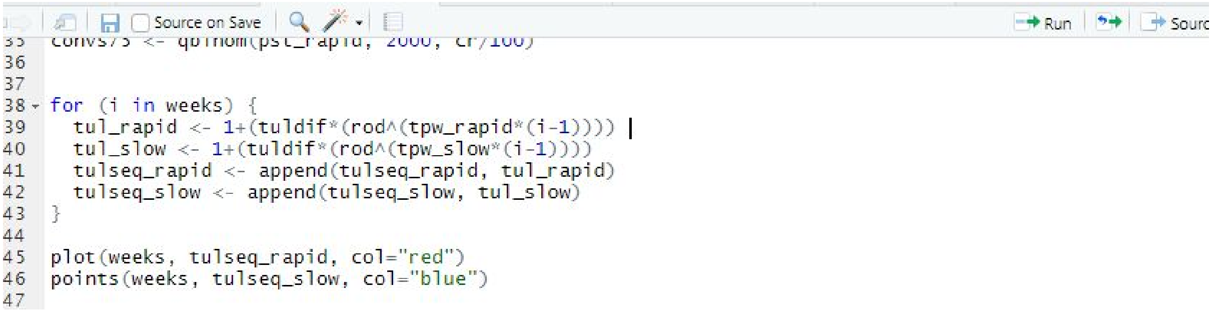
Creating some initial parameters:



```
tpw_rapid <- 2          #tests per week in rapid testing strategy
tpw_slow <- 1           #tests per week in slow testing strategy
val <- 1                #the value created in week zero
cr <- 2.5               #the conversion rate in week zero (ie, 2.5%)
tuldif <- 0.1           #the test uplift in week 1 (ie, 10%)
rod <- 0.95             #rate of decay (ie, each week's uplift is only worth 95% of the last)
weeks <- seq(1,52,1)
```

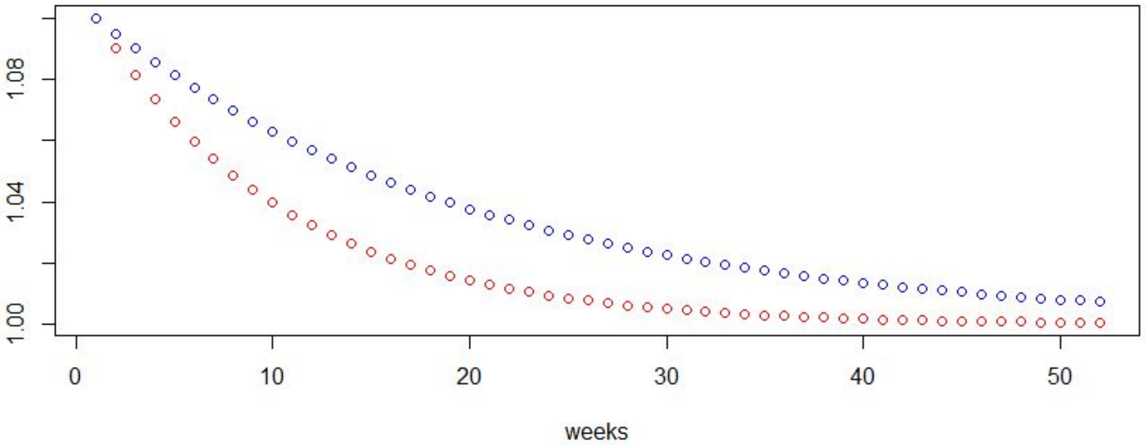
The important parameters to note here are 2 tests a week for our rapid testing strategy and one for our slower testing strategy.

I then calculate two sequences, tulseq_rapid and tulseq_slow, which are vectors for the uplift value for each test for each week, 1-52.



```
convs75 <- qbinom(pst_rapid, 2000, cr/100)


for (i in weeks) {
  tul_rapid <- 1+(tuldif*(rod^(tpw_rapid*(i-1))))
  tul_slow <- 1+(tuldif*(rod^(tpw_slow*(i-1))))
  tulseq_rapid <- append(tulseq_rapid, tul_rapid)
  tulseq_slow <- append(tulseq_slow, tul_slow)
}

plot(weeks, tulseq_rapid, col="red")
points(weeks, tulseq_slow, col="blue")
```

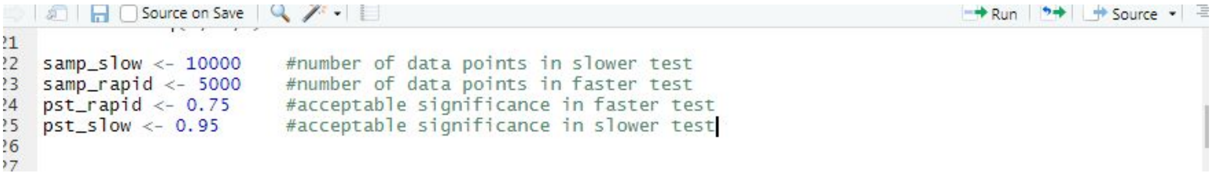
The plot of this looks like



Which, as expected, shows the uplift value of each test diminishes faster for the rapid strategy (red) than it does the slower one (blue). Note that with the rapid strategy, both tests in the week are assumed to have the same per test uplift value.

Calculations II - creating a ratio of Type II error probability

First we establish the parameters of a representative test

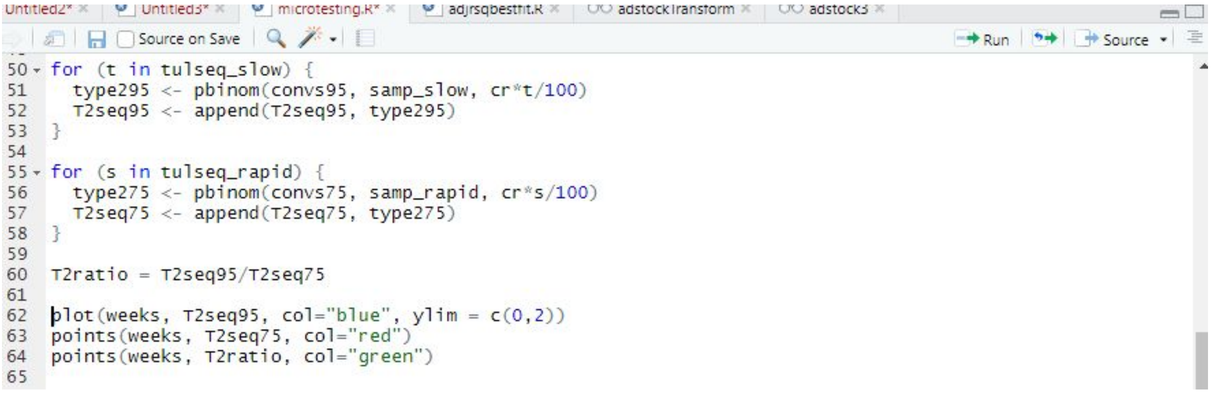


```
samp_slow <- 10000     #number of data points in slower test
samp_rapid <- 5000     #number of data points in faster test
pst_rapid <- 0.75      #acceptable significance in faster test
pst_slow <- 0.95       #acceptable significance in slower test
```

It’s worth noting that the probability of a Type II (incorrect rejection of a value creating test) goes down the more data points and the larger the difference between the two test cohorts is. In other words, the advantage for really big sites is lessened (more data), as is the advantage for sites that show large improvements in each test run. I’d suggest that 10% is already a pretty large number and if you are consistently seeing more than this for each test run either you are calculating incorrectly or your site is in bad shape.

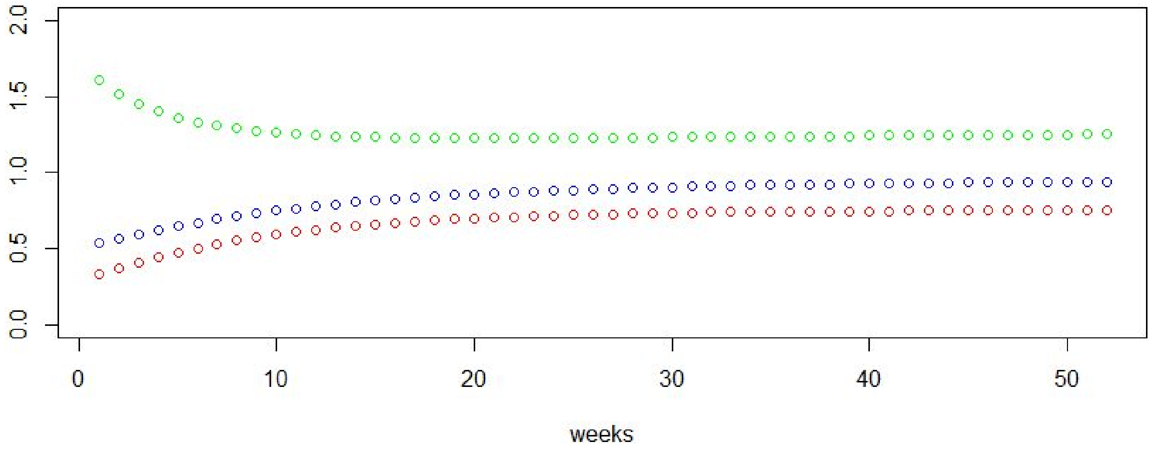
The other implication is that as time passes as the cohort differential drops (ie, there’s less of an uplift), the probability of a Type II error increases. This intuitively makes sense - the less obvious an improvement is, the harder it is to prove it.

Using the sequences we calculated in part one and the conversion significance values for each testing environment, we can calculate vectors for the probability and ratio of Type II errors:



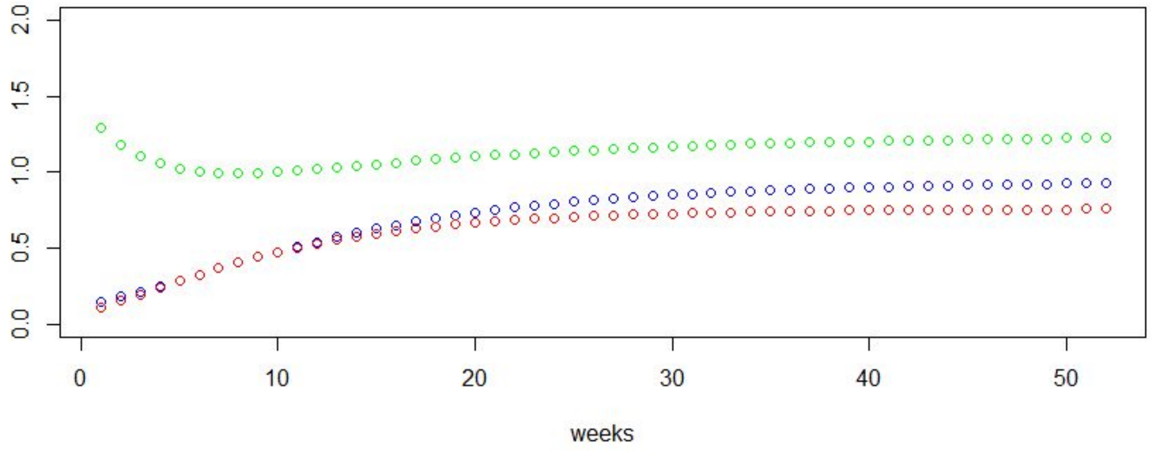
```
for (t in tulseq_slow) {
  type295 <- pbinom(convs95, samp_slow, cr*t/100)
  T2seq95 <- append(T2seq95, type295)
}

for (s in tulseq_rapid) {
  type275 <- pbinom(convs75, samp_rapid, cr*s/100)
  T2seq75 <- append(T2seq75, type275)
}

T2ratio = T2seq95/T2seq75

plot(weeks, T2seq95, col="blue", ylim = c(0,2))
points(weeks, T2seq75, col="red")
points(weeks, T2ratio, col="green")

```

When we plot this out, we see



The blue line, the probability of Type II errors when testing at a 95% significance acceptance threshold (the ‘slow’ strategy), is greater than the probability of Type II errors at 75%. The green point shows the ratio.

Increasing the sample size so that we are using 30k vs 15k for the two cohorts



We see rough parity between the two approaches for the first 13 weeks or so, which makes sense given the point about sample and signal size. Eventually, the diminishing signal will make a lower significance acceptance threshold less likely to suffer from Type II errors.

I suggest you use the code and investigate this effect with the numbers from your own website (or get in touch and I can do an audit).

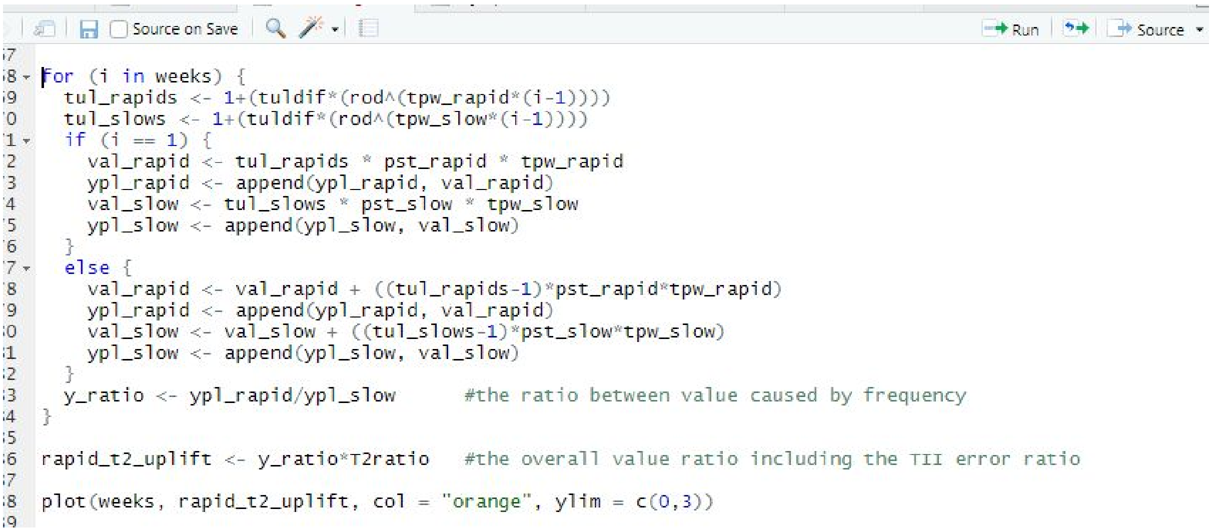
Calculations III - Integrating Type I error value and the value of running more tests

This final bit of code brings the whole model together.

It's here that we see the value of running more tests - double the number of tests creates double the number of improvements it's possible to find.

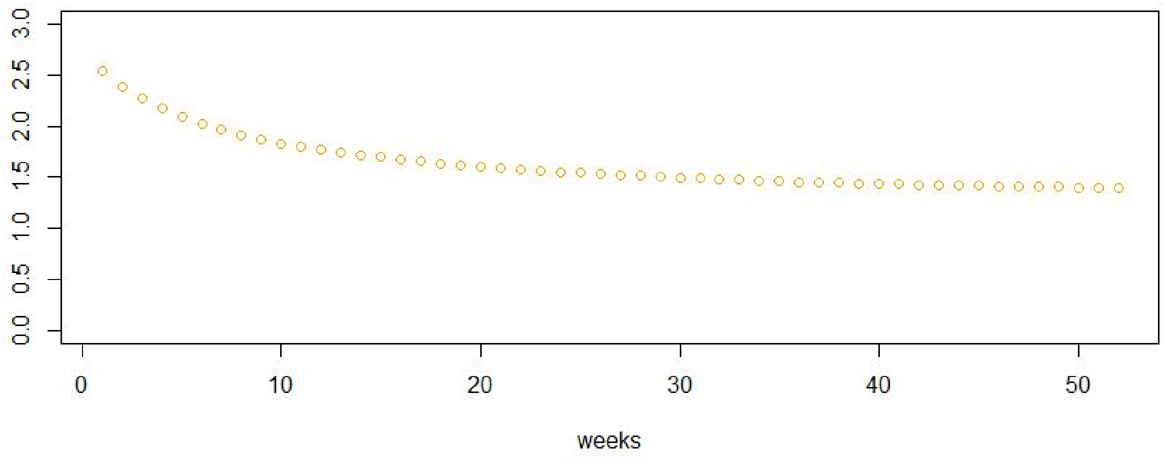
Remember - in the assumptions, I state that the uplift is the average uplift per test run (so accounts for successful and not successful tests).

The Type I error is accounted for by multiplying through the threshold acceptance, reducing the value of the test by (1 - threshold). In other words, before we account for the Type II errors, on average each test at a higher threshold increases the acceptance by more (less chance of a false positive).



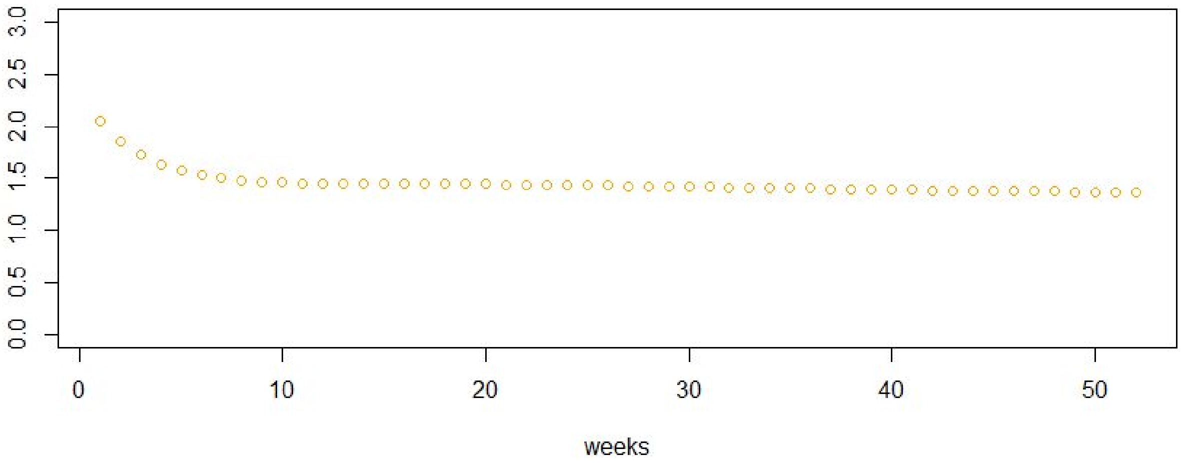
```
for (i in weeks) {
  tul_rapids <- 1+(tuldif*(rod^(tpw_rapid*(i-1))))
  tul_slows <- 1+(tuldif*(rod^(tpw_slow*(i-1))))
  if (i == 1) {
    val_rapid <- tul_rapids * pst_rapid * tpw_rapid
    ypl_rapid <- append(ypl_rapid, val_rapid)
    val_slow <- tul_slows * pst_slow * tpw_slow
    ypl_slow <- append(ypl_slow, val_slow)
  }
  else {
    val_rapid <- val_rapid + ((tul_rapids-1)*pst_rapid*tpw_rapid)
    ypl_rapid <- append(ypl_rapid, val_rapid)
    val_slow <- val_slow + ((tul_slows-1)*pst_slow*tpw_slow)
    ypl_slow <- append(ypl_slow, val_slow)
  }
  y_ratio <- ypl_rapid/ypl_slow      #the ratio between value caused by frequency
}

rapid_t2_uplift <- y_ratio*T2ratio   #the overall value ratio including the TII error ratio

plot(weeks, rapid_t2_uplift, col = "orange", ylim = c(0,3))
```

This produces a final curve that looks like this, for our original sample sizes of 10,000 and 5,000 per test:



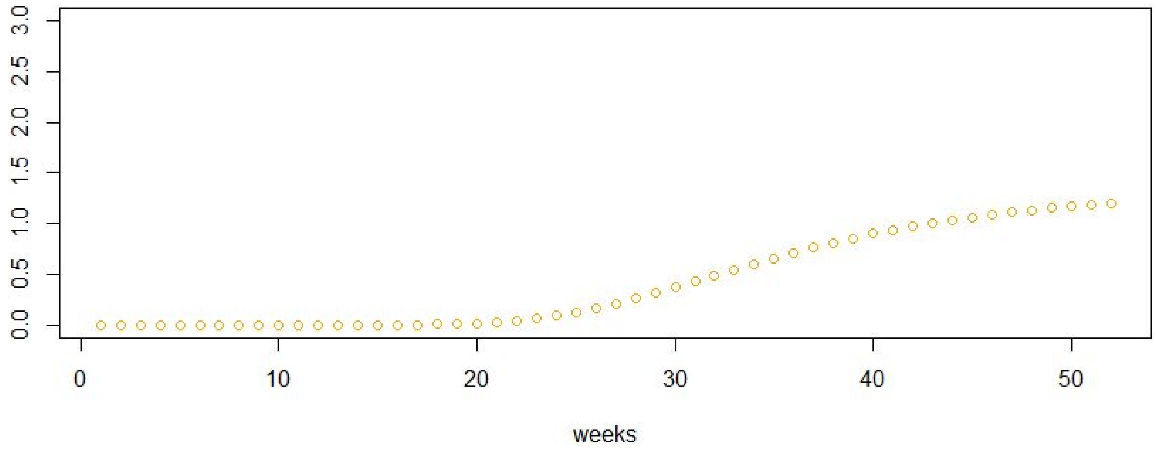
The points representing the ratio in value of the rapid strategy / the slower strategy. This trends over time to about a 50% increase as the Type II error influence decreases.

If we go again to our increased test size (30k vs 15k)



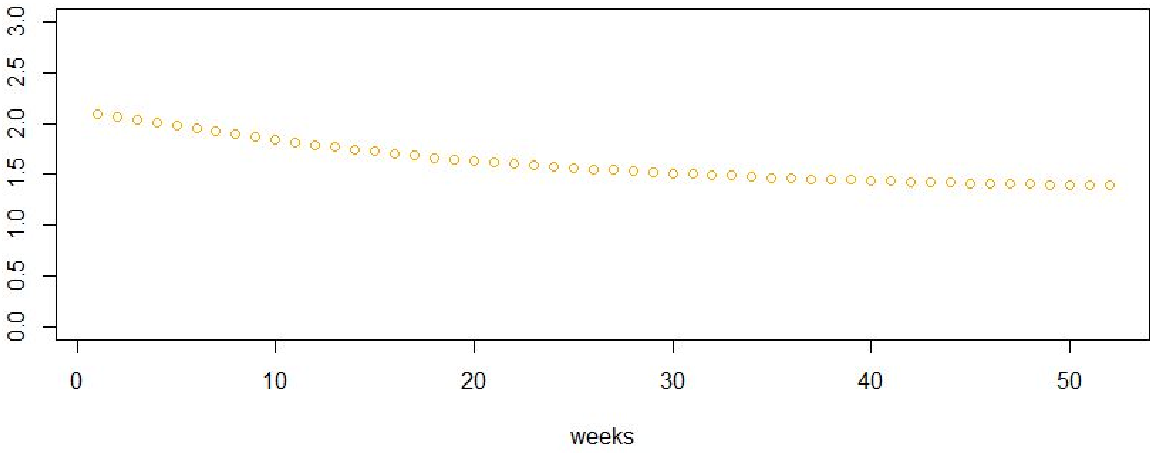
We still see an uplift, though less pronounced.

Looking at two extreme views, firstly with a very large test size (500k vs 250k per test)



We see that it takes a while for the more rapid testing strategy to be more valuable - the Type II errors are virtually non existent and we are unnecessarily increasing Type I errors. Simply running a more rapid strategy with an equivalent acceptance threshold (95%) would create more value.

On the other end, a very small sample size (400 vs 200) produces similar results to our original view, but with a slower rate of decline:



Conclusions and suggestions

1. Even if you don't want to work through the maths to calculate optimal testing size and frequency, you will almost certainly benefit from lowering the threshold at which you accept a test as showing greater value and from testing more rapidly.

2. All of the above really comes down to that - just be self consistent and make the decisions about certainty before the test starts or you will run into trouble.

3. This is slightly less important for websites that have a lot of traffic - these have the advantage of being able to increase test volume without sacrificing certainty (most tests will reach the required level of certainty).

4. This approach puts pressure on your team to produce quality customer research and solutions - this is a **good** thing as it will get more value out of them.

5. It's likely you can improve your output by following a Bayesian approach to formulation and calculation; using priors based on your team expertise to calculate the posterior probabilities means you can iterate more rapidly.

Want more?

If you want to discuss this further or if you'd like a more bespoke data analysis done of your website and media performance, please email me at damian.koblintz@gmail.com.

email: damian.koblintz@gmail.com