# Background

The notes I have for this are:

projecting unemployment rates - how they've changed with covid and projecting them forward.

scenario exploration. Keep the model and framework simple. Get a simple and straightforward measure of unemployment rate, furloughing, jobs under threat, build timeseries assumption as to how those convert to job losses. A couple of days on it would be reasonable. Shannon has unemployment data. Covid>filesSC for unemployment data. What are the assumptions of the shape of rise and fall of jobs at risk etc. conversion rate of that to unemployment. Keep it simple.

That’s not a lot to go on. Also, as all of this depends on what happens with covid-19 and the government’s response, the answer is at least as complex as modelling both of those.

Chris said to keep it simple – scenario exploration. So we don’t need to know what is going to happen, just what might happen.

# What data do I have already?

C:\Users\WilliamLetton\Crystallise Limited\ModelDev - 2019nCoV\files\_SC\Unemployment\_Health

Has a series of bicswave files with questionnaire responses. This includes population furloughed

individualDataSheets has total number of jobs furloughed over time.

LabourMarketWeekly has employment and unemployment statistics up to week13 of q1 for a sample size of around 5k people.

It also has ‘temporarily away from paid work’

Also has ‘made redumant is last three months

NOMIS\_Unemployment has quarterly unemployment numbers for 1st quarter of 2020.

Unemployment by age and duration

Has quarterly statistics including 2020 q1

Ok.

# Initial model idea

So for a simple model have no geographical segmentation, age segmentation or industry segmentation.

Get a baseline employment rate from previous years (if cyclical) and/or trend

Get data for proportion furloughed over time so far. Use this to project into the future (with assumptions).

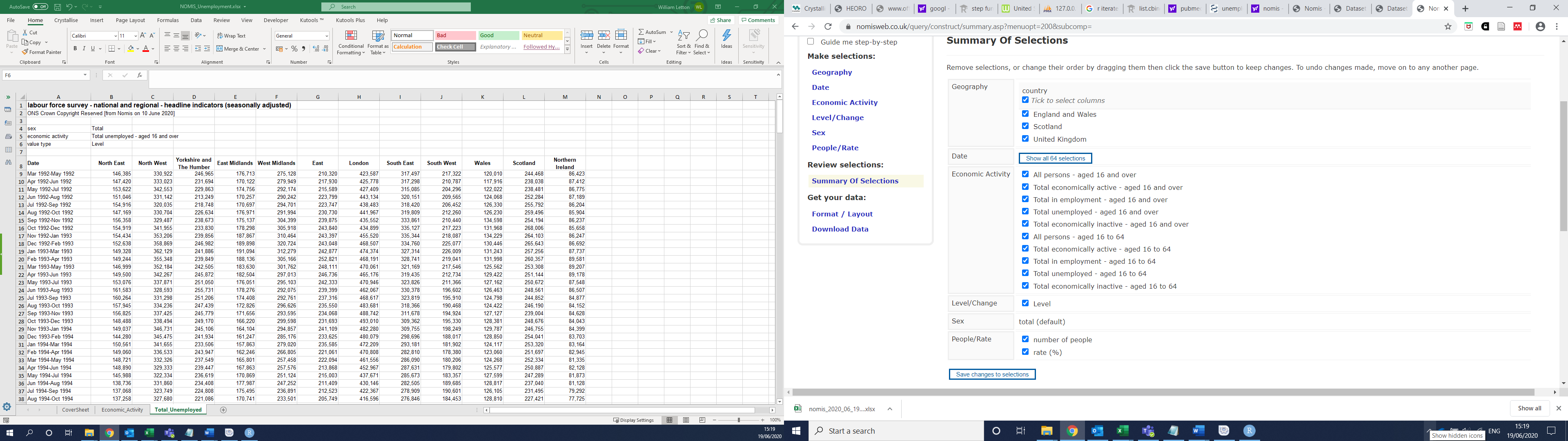
Estimate the proportion of furloughed that are made redundant per day.

Use all this with a range of assumptions and parameters to estimate excess unemployment over time under different scenarios.

Use R for this.

## Unemployment % baseline

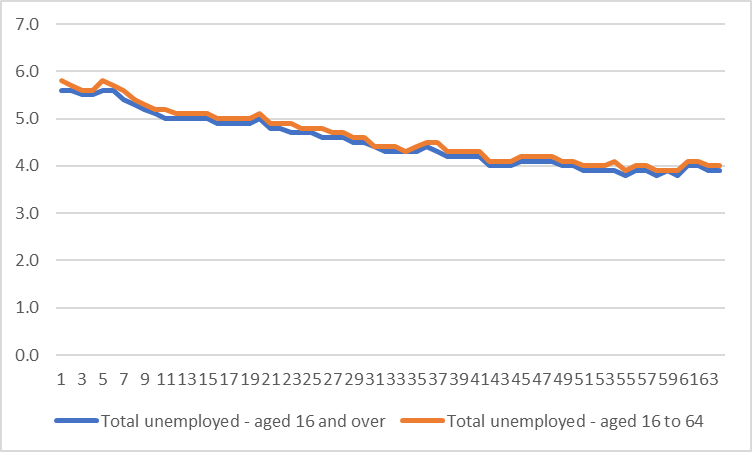
Going back to Nomis for employment data:



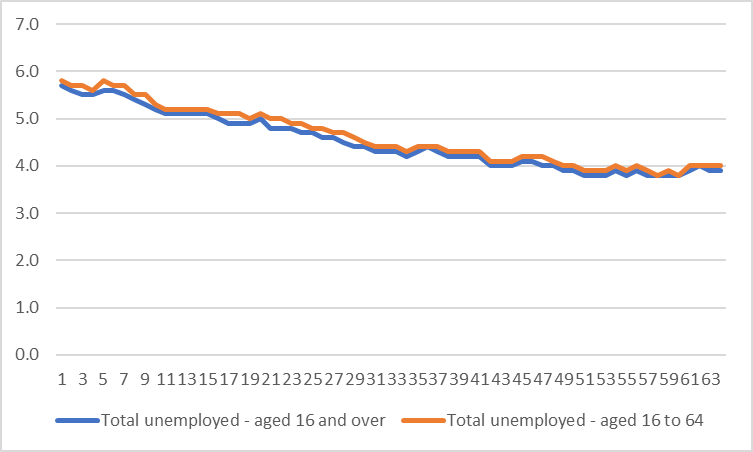
<https://www.nomisweb.co.uk/query/construct/summary.asp?menuopt=200&subcomp=>

called this file: 200619\_UNEMPLOYMENT\_nomis

England and Wales:



UK:



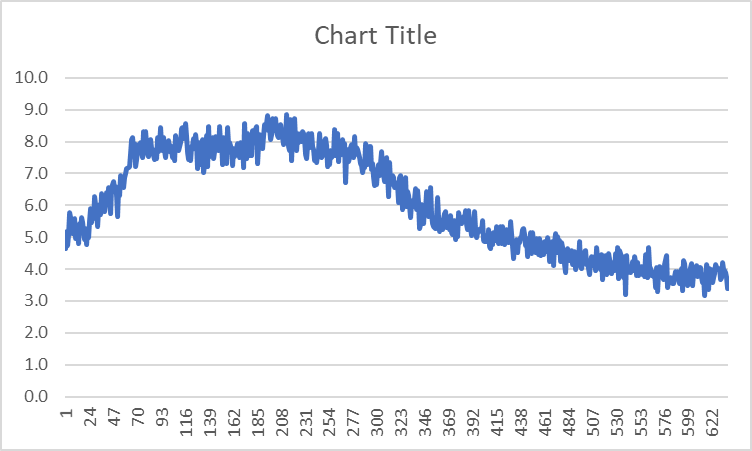
Each data point is a 3-month moving average I think. At any rate the baseline unemployment looks like it’s around 3.9-4.0%. this value is seasonally adjusted.

That’s a good baseline assumption.

## Unemployment during covid

LabourMarketWeekly has employment and unemployment statistics up to week13 of q1 for a sample size of around 5k people.

It only goes up to week 13, but presumably there is more recent version available by now. The data up to week 13 supports a baseline of 4%



## Furloughed numbers over time so far

Two possible sources for this.

### The bicswave data:

*The indicators and analysis presented in this bulletin are based on responses from the new voluntary fortnightly business survey, which captures businesses responses on how their turnover, workforce prices, trade and business resilience have been affected in the two week reference period. These data relate to the period 4 May 2020 to 17 May 2020.*

*The survey was sent to around 20,000 UK businesses, and results presented in this release are based on a limited number of responses, around 30.9% (6,364) of all businesses surveyed who responded.*

*Estimates from the Business Impact of Coronavirus (COVID-19) Survey (BICS) are currently unweighted\* and should be treated with caution when used to evaluate the impact of COVID-19 across the UK economy. Each business was assigned the same weight regardless of turnover, size or industry, and the data in the latest period are final.*

*\*For certain workforce tables the proportions are based on employment within responding businesses.*

Has data on **working remotely proportions and % on furlough leave and % made redundant**.

Only goes up to the 17th may, but presumably there’s more recent data available now.

# Model code ver1

For this basic model I’ll base it on Chris’ covid compartment model. This means I’ll be able to more fully understand how the timesteps and differential equations work.

I need to define compartments and rates of flow between them. To keep it simple the total fraction will be total economically active (i.e. employed or unemployed, none outside employment).

Within that three compartments?:

Employed, furloughed, unemployed

Then maybe later I can split employed into employedWorkplace and employedFromHome?

What I can then do is have the movement from employed to furloughed and furloughed to unemployed dependent on changeable parameters.

Or perhaps a compartment model isn’t the best way to go here? Certainly from scratch it would seem a bit much, but I think for the purposes of self-training, and for potential elaboration, it’s probably the best idea I have.

Employed = E

Furloughed = F

Unemployed = U

dEF = projection from current trend

dFE = estimate from available data

dEU = estimate from available data (including business closing down, which won’t be furloughing!)

dUE = ?

dFU = estimate from available data

so a lot of unknowns there. The basic model is that any movement is possible except U>F.

the model is driven by dEF, which will be determined by a projection from current trends. The rest will be various estimates.

Maybe it would also make sense to include

# Estimate of dEF

Looking for more recent bicswave data.

<https://www.ons.gov.uk/surveys/informationforbusinesses/businesssurveys/businessimpactofcoronaviruscovid19survey>

<https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandwellbeing/datasets/coronavirusandthesocialimpactsongreatbritaindata>

saved as 200622\_ONS\_referencetables190620, has survey reported furloughing.

<https://www.ons.gov.uk/businessindustryandtrade/business/businessservices/datasets/businessimpactofcoronavirusanalysisovertimeukwaves2to5panel>

200622\_BICS\_analysisovertimewave2to5dataset

Ahah, here we go: <https://www.ons.gov.uk/economy/economicoutputandproductivity/output/datasets/businessimpactofcovid19surveybicsresults>

Downloaded all 6 files to their own folder. Runs to the 31st may, which was released on 18th June.

Looking at the most recent of the 6:

There are furlough data and return from furlough data, but there are versions for ‘apportioned by workforce size’ or not. I don’t know what that means, and I can’t find an answer in the worksheets or online. Maybe if it’s not apportioned then if two companies have 10 and 30% furlough that will average to 20%, but if it is apportioned then if company 2 is twice the size of company 1 then the average is 25%?

>ask chris about this.

It makes a big difference, 37.8% for not apportioned and 29.8 for apportioned.

If that is the case then it makes sense to use the apportioned data. This is sheet ‘Proportion Furloughed (4)’. Note that this does not include business that have permanently stopped trading (which wouldn’t have furloughed workers anyway).

The questions changed over time. Wave 1/6 had no question on furloughing. Wave 2/6 had a question on furloughing, but it’s broken into still trading and paused/stopped trading rather than still/paused, and also the data is not apportioned.

Also part of this problem is that we have statistics for the furloughing in businesses that continue to trade and for those that have temporarily paused or ceased, but not how those numbers relate, or how the furloughing relates to unemployment.

Amount businesses who have not permanently stopped trading, the furlough % was around 30% from 20th April to 31st may.



This is in agreement with the ONS survey of individuals (200622\_ONS\_referencetables190620 table 13)



IndividualDataSheets.xlsx

Data from hmrc:



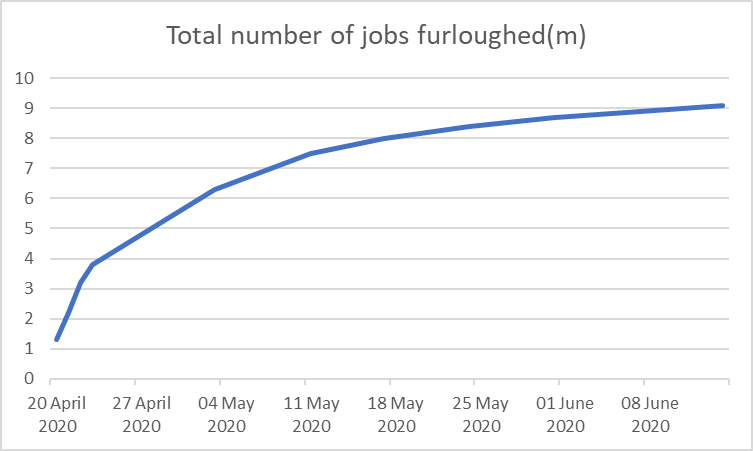
From links in the above file:

<https://www.gov.uk/government/collections/hmrc-coronavirus-covid-19-statistics>

has updated numbers:



Which looks like:



So if around 30% of workers are furloughed and 9m people are furloughed, that should mean that the uk labour force is around 30m. from the nomis data it looks like it’s actually around 33m, but that’s not so different as to make me massively worry.

In any case it looks like the furloughing is leveling off at around 9-10m. from the nomis report it seems that some of those are already returning to work. It’s a big unknown as to how many of the remaining furloughed will become unemployed. The furloughed staff also don’t include those made redundant, although the ONS impact on work survey suggests that this rate is less than 1%, which seems very low.

<https://www.statista.com/topics/1989/unemployment-in-the-united-kingdom/>

no idea of the reliability of their analysis, but they are forecasting 10% unemployment in the second quarter of 2020. This would represent an increase of around 6% on pre-corona levels, and around 5% on the levels as of the 14th June (very very very roughly).

# Meeting with chris

the point of the model is to then forecast health harms

can fix an arbitrary long term view and then fix the model to that.

long term view would be 2021

parameter for expected unemployment rate at the end of Q1 in 2021

starting at jan 2020

start with baseline unemployment by age

get an idea of the furloughing rate

how does furloughing convert to unemployment?

over time (i.e. from august)

we will also need to plug in a recovery function (arbitrarily set)

could use furloughing rate by industry. (i.e. bad in tourism and hospitaly).

look up 'apportioned by employment size' - ~~email jack.~~

# Creating basic model

Ok, so I think we have enough to start putting a model together. This will be based on the Covid-19 SIRD model.

I think that having the model outside the main code as was done for the covid model makes sense.

I need to be careful as this could get very complicated very quickly.

Most of the covid complexity can be left out. This includes all the R0 and infection stuff, but also the birthrate and deathrate.

I’ve putt in two dummy rates for transfer between compartments, just until I know that the model is working.

Which it is not. In the output the first simulation is just zeros for all three compartment for all days, and the other 149 look like they work, but only for days 1 and 2.

A lot of the results also don’t change between days 1 and 2, which is unlikely to be good because the distributions of the transfer parameters (i.e. dEmFu and dFuUn) shouldn’t be 0.

Well for a start the parameter values don’t seem to be taken from the beta distributions correctly. For example propFuUn is always either 0.2 or 0.

propEmFu is always either 0.2 or 1.1125E-309.

Found one error in the equations. Now the values vary more, look more reasonable.

However, we still have the issue with the 0 values after the second day:



Well if I run only one simulation I get:



Which suggests that that first blank column is not the first simulation. That’s a problem in itself, and might be a clue as to the general problem.

If I calculate the sum of the values:



I can see that on day one that all add to the correct total value, but on day two there around 9650 people missing, about 0.03% that could be a rounding thing, but more likely it’s another clue.

It looks like the column of 0s on in V1 is deliberate, in that chris creates the theResult\_Em matrices with ncol=numSim+1 and then during the simulation puts the results in i+1. I don’t know why this would be, maybe it comes in useful later. I’ll take it out for now though, as it’s messy.



So that’s the blank column mystery solved. Now what about these blank rows and missing people?

Ahah, now we’re getting somewhere. It only simulates the first day and then stops, even though simTime = 194.

It’s strange. If I have:

*for (i in 1:(simTime-1)){*

*print(paste("Simulating day",i))*

*}*

It prints up to 193, correctly. But if I have:

*for (i in 1:(simTime-1)){*

*print(paste("Simulating day",i))*

*## Fetch the existing compartment values.*

*Em<-results[i, "Em"]*

*Fu<-results[i, "Fu"]*

*U<-results[i, "Un"]*

*## Initialise the interim compartments used to calculate the daily changes.*

*Emz <- Em*

*Fuz <- Fu*

*Unz <- Un*

*## Cycle through the (fidelity) slices of a day used to numerically estimate the integrals.*

*for (j in 1:fidelity){*

*## This is where the diff equations go.*

*Emz <- Emz - Emz\*propEmFu/fidelity*

*Fuz <- Fuz + Emz\*propEmFu/fidelity - Fuz\*propFuUn/fidelity*

*Unz <- Unz + Fuz\*propFuUn/fidelity*

*}*

*## Place the new compartment values into the results object*

*results[i+1,"Em"] <-Emz*

*results[i+1,"Fu"] <-Fuz*

*results[i+1,"Un"] <-Unz*

then it only prints 1. It must be breaking out the loop somehow.

Huh, if I comment out everything but that print statement it also only prints 1.

Ah, I think it was a missing bracket issue. Or rather a bracket in the wrong place the return statement was inside the loop, which I suspect will terminate it.

There we go:



Now, about the totals…



This is pretty bad. By the end we only have around 3% of our starting number!

The obvious first place to look is the differential equations.

Found an error in the code (a typo). now comparing:



That’s way better, but still not exact. It could just be the nature of the approximation of calculations.

Lets increase the fidelity. If I’m correct then higher fidelity will reduce this difference.

Changing fidelity from 10 to 100000.



Yep, that explains it.

Ok, we’re good to continue.

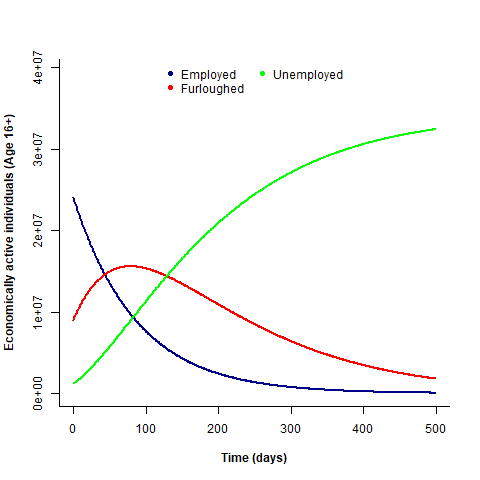
Ok, so the gap score calculation in SIRDItaly.R won’t work because theResult doesn’t contain the target values (incidentally this might explain why the scoring system didn’t work later on with the covid modelling, because theResult\_D[1:length(obs\_D), ] and theResult\_D[1:length(obs\_D), 1]

Are identical if the matrix has only one column.

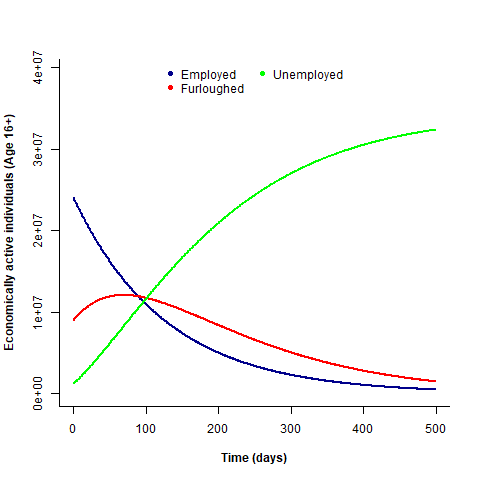
That seems to all work:



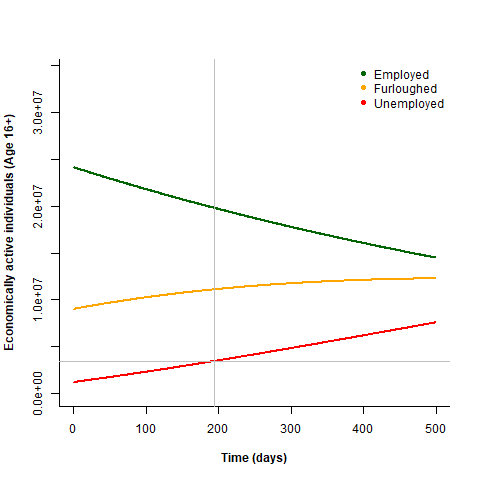
With 5 scenarios:



With 50:



Tweaking the parameters and running 5000, with lines showing the target point:



# Meeting with chris

chris has an IFOA meeting at the start of friday, so it'd be nice to have something for the end of thursday.

explanatory model is what we are after.

could use unemployment specific data points (e.g. 20% unemployment on 1st october).

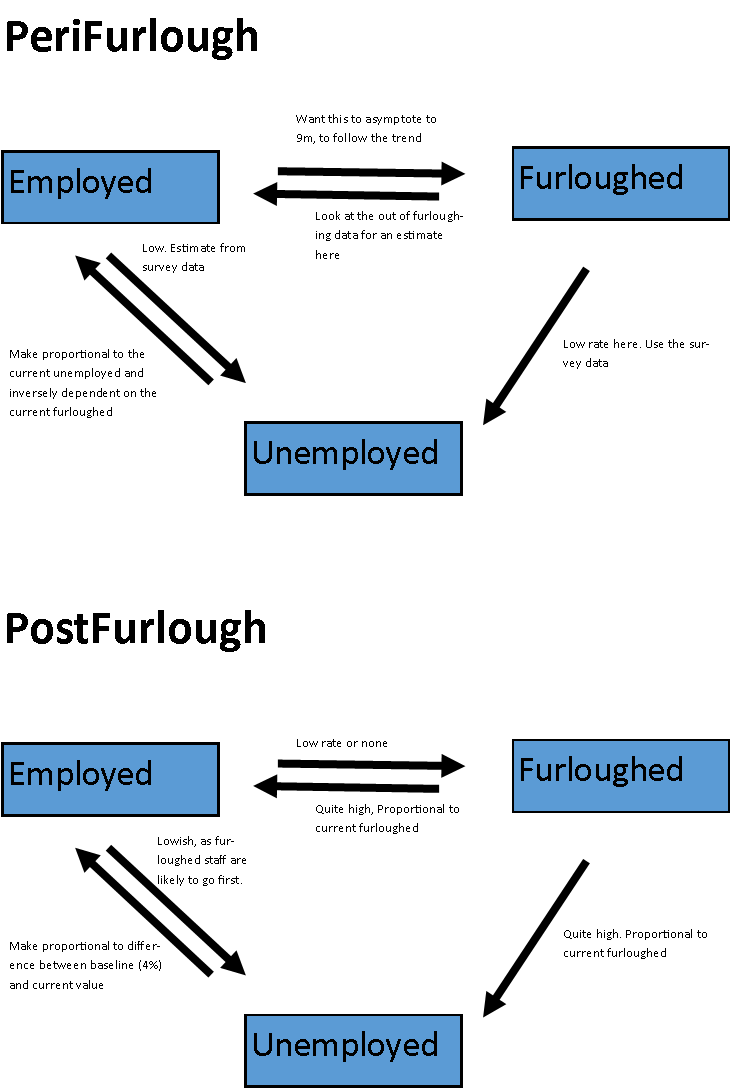
In effect what he seemed to like the idea of is me putting together a plausible model with a fixed target point (e.g. 20% unemployment on 1st october), and then run scenarios based on a fit to that model, e.g. what would change if the rate of re-employment went up, or the rate of redundancy from furloughing went down?

So now I need to make the model plausible. This needs to include:

A date for the end of furloughing

Rates for transfer between compartments pre and post furloughing. It will probably be easiest to just keep these completely separate and just use a lookup for which applies based on the current day in the simulation.

This is what I have as an idea so far:



The thing is that it’s quite complicated. We’re looking at 10 parameters right of the bat, not including any complexities of making things asymptote or proportional to baselines and things.

Well, the periFurlough phase should be a lot more predictable, so we can make that part of the model much more constrained, and can always updated if and when we get more data.

The second part can then be the freer part to explore scenarios.

# Meeting with chris

He wasn’t thrilled by the splitting into two models, but accepted my arguments.

Could then use a step function over 10,000 scenarios to create a 'smooth' function.

chris expecting a pulse of unemployment at the end of the scheme, and then a rising curve underlying progression after that.

the problem is that so much of these equations will actually depend on how the economy is doing, which isn't part of the model. Effectively for each of these transfer equations I’m thinking to myself what is the economy likely to do and so what will happen, which is just me making the model fit what I think.

For example for post-furlough Em>Un I just have to pick a number arbritrarily.

Perhaps this is something to address once the basic model is done.

What I want to avoid here is just using equations to paint a graph of what I think is likely to happen. For example I reckon that as furlough ends there will be a small pulse of people moving from furlough to employed, followed by a trickle. However, if I just write a function that does that I may as well just have drawn a graph on a whiteboard.

Also a lot of the outcome in terms of unemployment numbers will depend on the ratio of furloughed people made redundant or rehired when the scheme ends. This ratio will in turn strongly depend on the status of the economy and what lockdown restrictions are in place.

And there is likely to be a pulse of response at the end of furlough followed by a longer term trend, which arguably should be treated separately in the model. But again, this goes back to overfitting the model to my expectations.

Ok, I have expressions for the transfers between compartments before and after ending of furlough.

Now I need to update the model to include them.

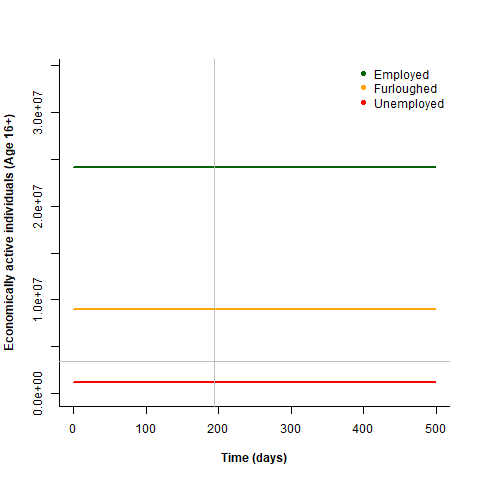
Created new copies of the sim and main R scripts, starting with today’s date (200624)

Ok, I currently have 12 parameters (including the date of endFurlough), plus starting values, target unemployed rate, target unemployed date.

It’s quite a lot…

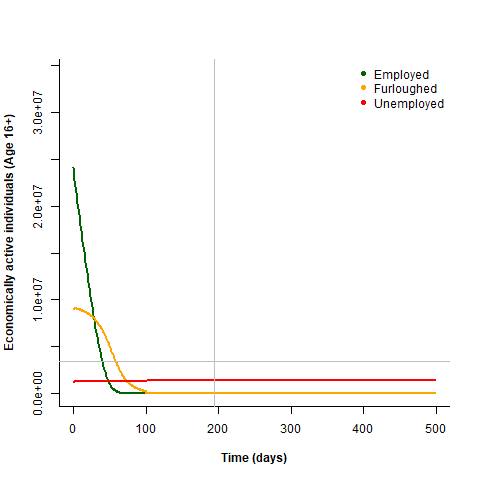
Ok, well anyway, I’ve updated the model

# Modal code ver 200624

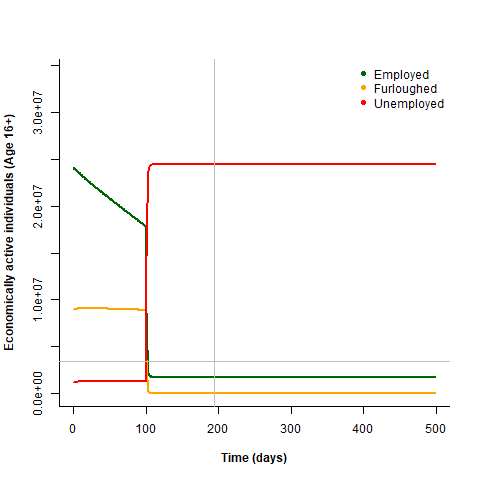


Hmm

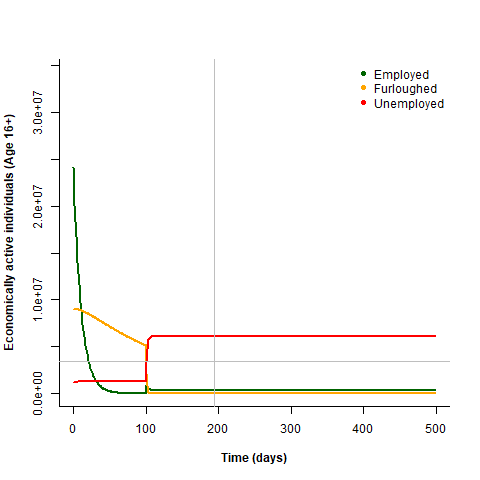
There was a problem where I’d put the operators on the next line rather than ending the previous.



Well that’s better… but they clearly don’t sum to 1. Ah part of the issue is I forgot the include the fidelity-s.

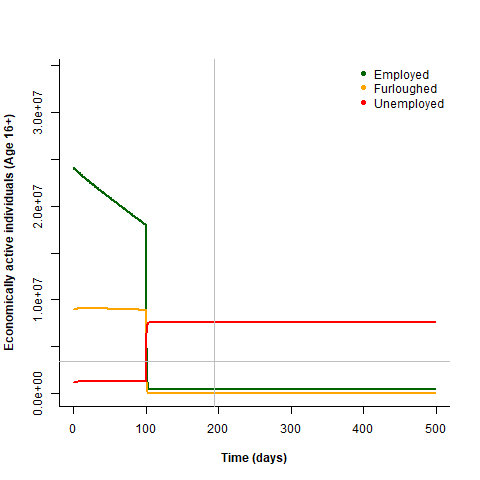


With 10,000 fidelity instead of 10



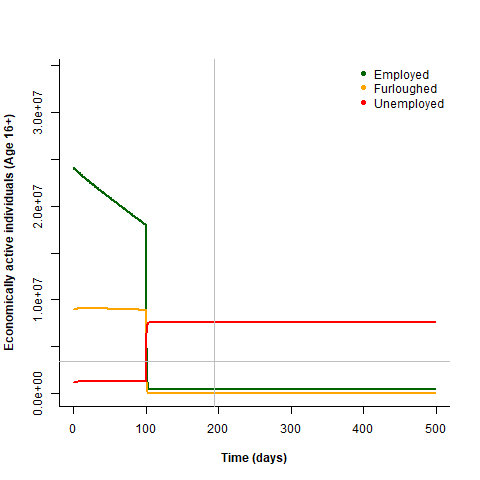
Ok, so there’s still some kind of problem with the fidelity.

Fidelity of 1:

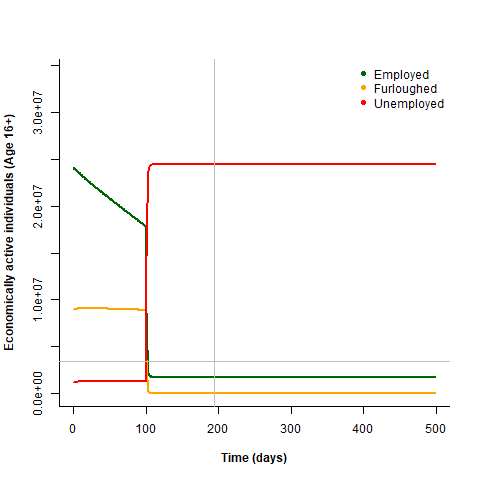


Added brackets around the expressions to which to apply fidelity to and found one where I’d missed fidelity.

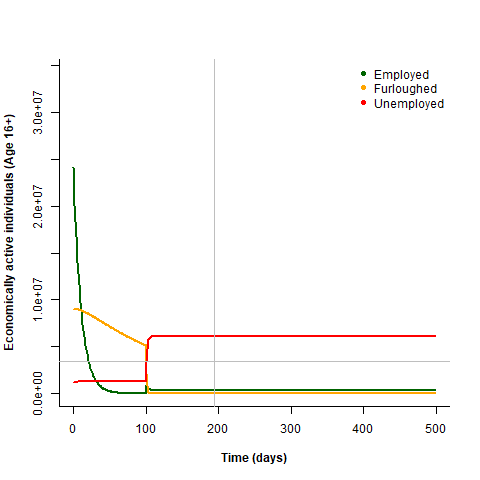
Fidelity of 1:



10:



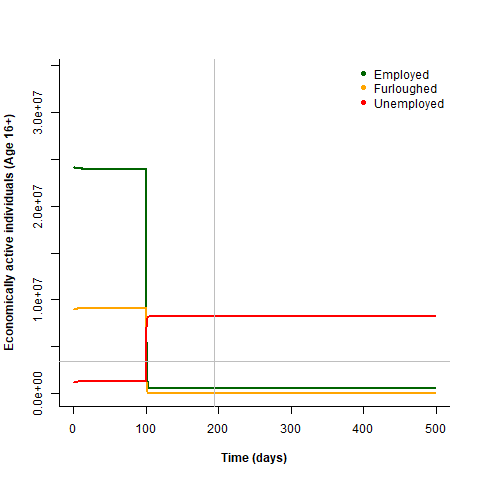
10000:



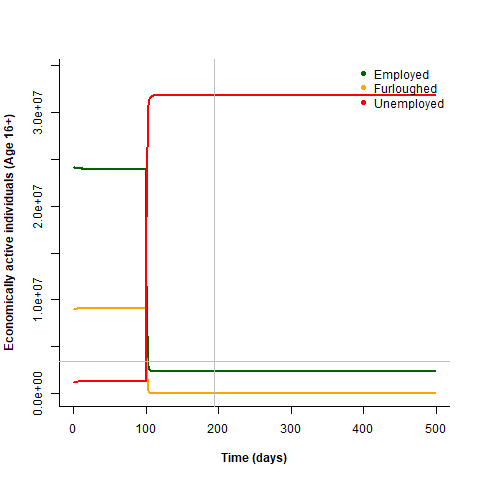
I’ve checked through the equations multiple times and they seem fine. It’s possible that this fidelity issue is related to the not summing to 1 issue and solving that will solve both.

So it’s clear from the above that the fractions do not sum to 1 this suggests that the partial equations are not balanced. I have found an unbalanced equation in the first set.

Fidelity 1:



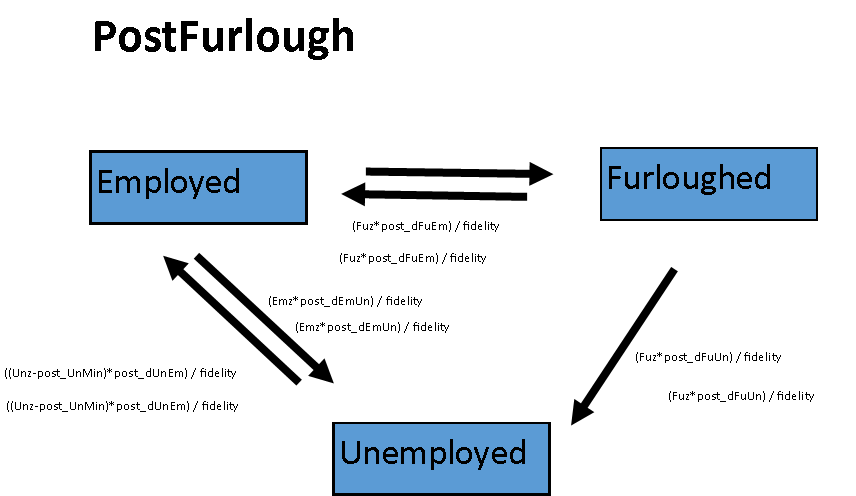
Fidelity 100:



So that may have solved the issue with the first set of equations. Let’s now look through the second.

Annoyingly the second set looks correct. But they are clearly not balanced as they don’t sum to one…

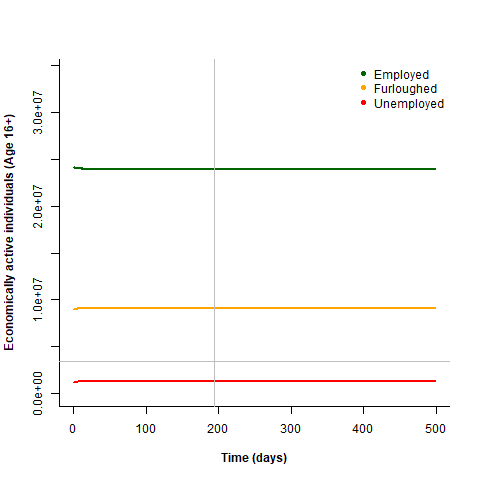
I’ve checked them again and they all seem to match:



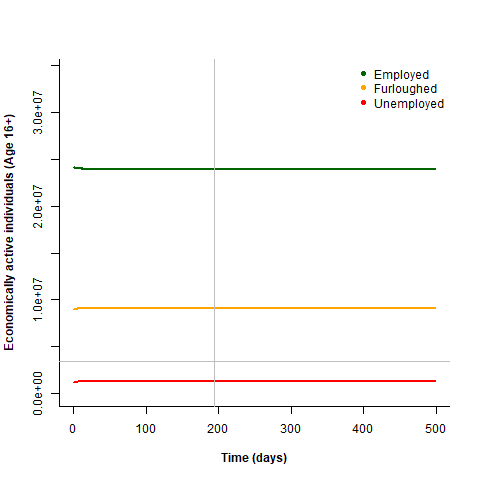
Let’s try taking pairs out and see if we see a fix arising.

Taking all pairs out of the post-furlough set:

Fidelity 1:



Fidelity 10000:

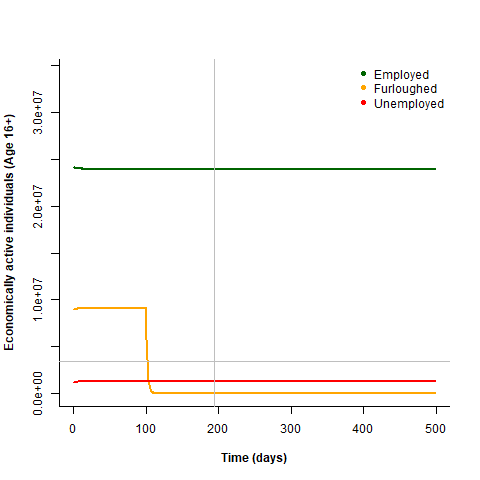


So that seems fine.

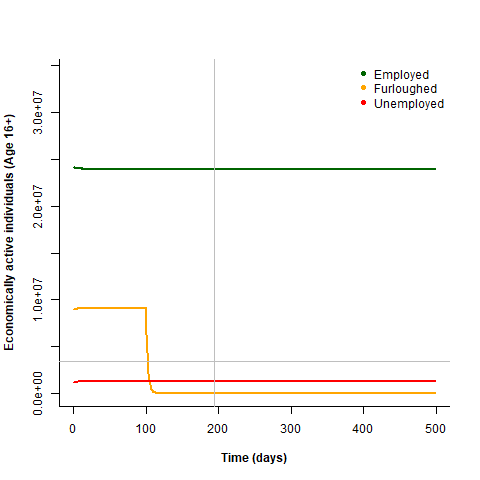
Now adding back in one pair at a time:

(Fuz\*post\_dFuEm) / fidelity:

Fidelity 1:



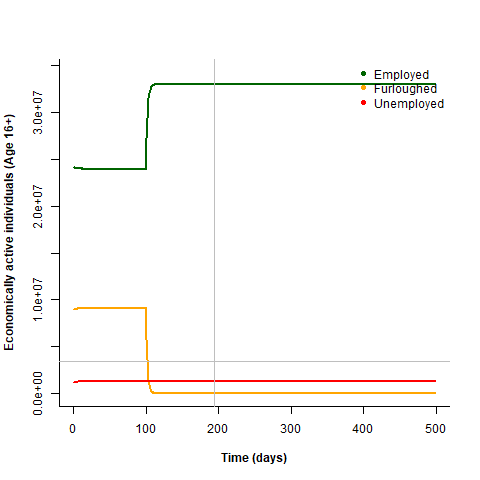
Fidelity 10,000:



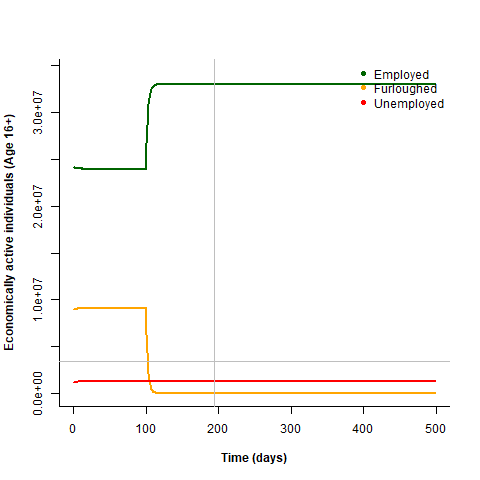
Well we’re getting the loss from furloughed, but not the gain to Employed. Ah, messed up the + and – signs.

Trying that again.

Fidelity 1:



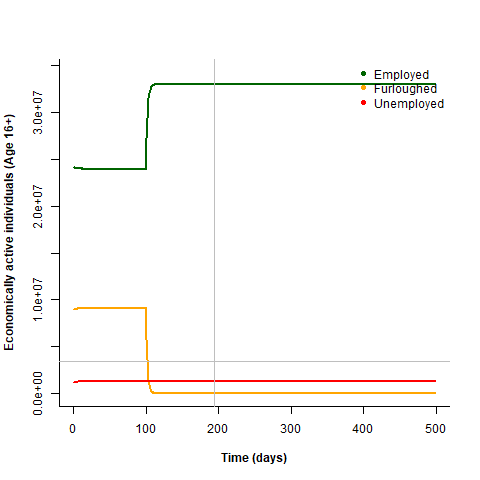
Fidelity 10000:



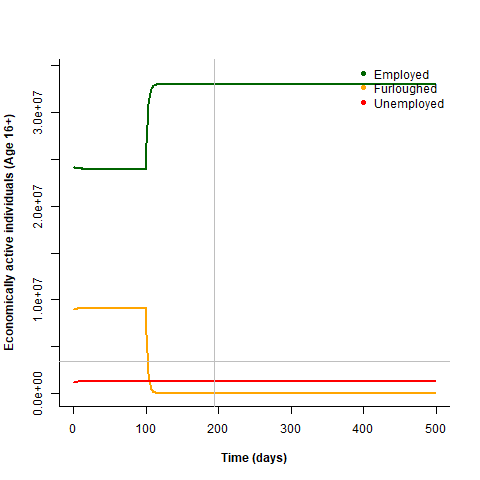
Ok, they seem to sum to 1 and they seem to match.

Adding another. ((Unz-post\_UnMin)\*post\_dUnEm) / fidelity

Fid1:



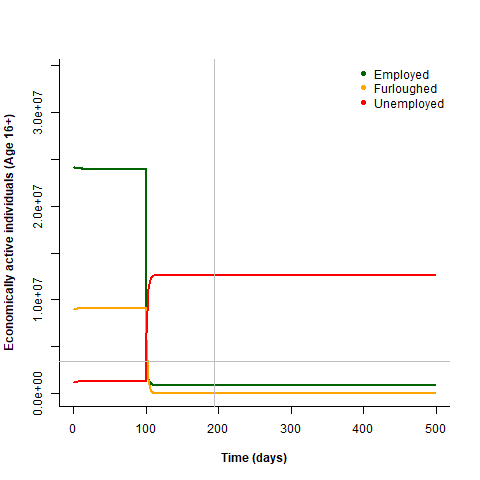
Fid10000:



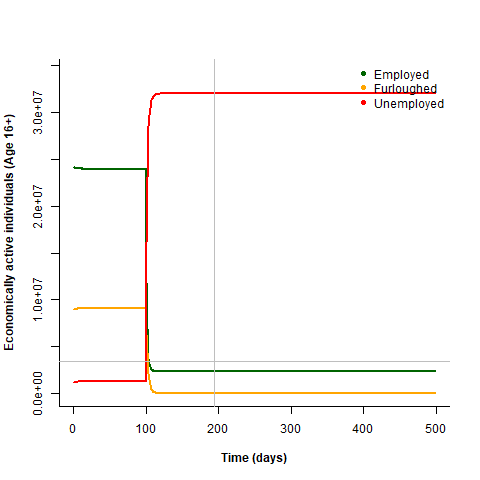
Again those match. They look pretty similar to the previous ones because when the unemployed is low then the effect gets vanishingly small.

Adding back (Emz\*post\_dEmUn) / fidelity:

Fid1:



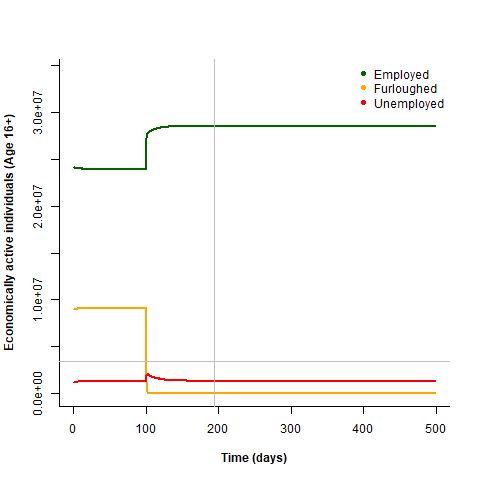
Fid 10000:



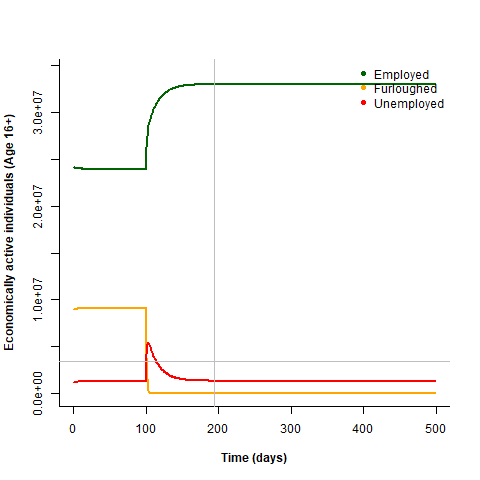
Ok here the graphs are different and only seem to sum to 1 in the latter case.

Taking that equation back out and adding the forth one instead. (Fuz\*post\_dFuUn) / fidelity

Fid1:



Fid 10000:



Ok those don’t match either.

So the equation pairs that seem to be working are:

(Fuz\*post\_dFuEm) / fidelity

((Unz-post\_UnMin)\*post\_dUnEm) / fidelity

And not working are:

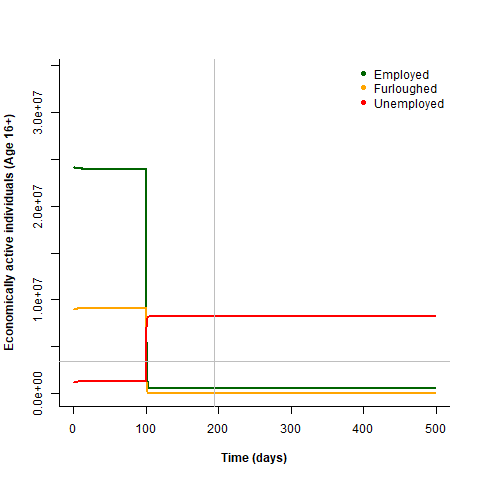
Adding back (Emz\*post\_dEmUn) / fidelity

(Fuz\*post\_dFuUn) / fidelity

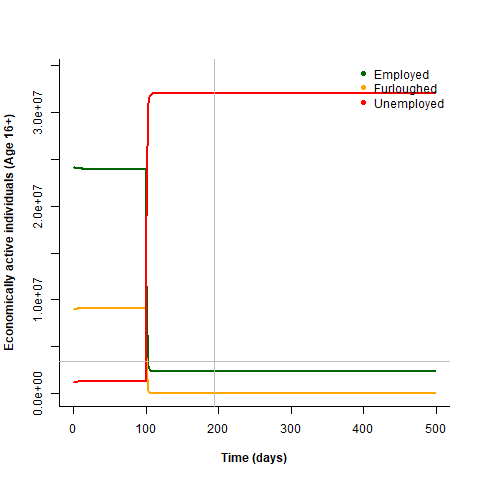
I see no obvious reason why this would be the case.

Adding all back in:

Fid1:



Fid10000:

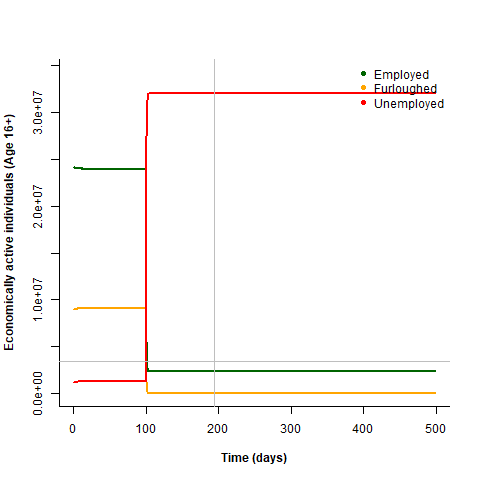


I just don’t get it.

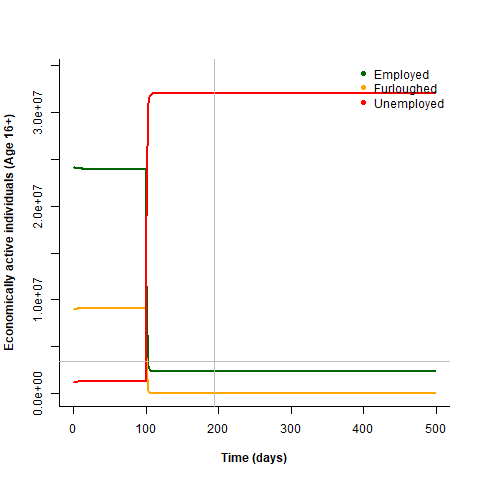
Ok rewriting the code to change the format to make it easier to see what’s going on. Instead of calculating new values for Emz, Fuz, and Unz, I’m going to calculate the transfer between compartments and then sum those transfers. effectively this is breaking it down into two steps.

Now running again

Fid1:



Fid10000:



That looks much better.

I’ll reformat the peri-fulough in the same way as it’s easier to read:

*##Proceed with peri-furlough changes.*

*## First calculate compartment transfers.*

*Em\_Fu <- (Emz\*(peri\_furMax-Fuz)\*peri\_dEmFu) / fidelity*

*Em\_Un <- (Emz\*peri\_dEmUn) / fidelity*

*Fu\_Em <- (Fuz\*peri\_dFuEm) / fidelity*

*Fu\_Un <- 0*

*Un\_Em <- ((Unz-peri\_UnMin)\*peri\_dUnEm) / fidelity*

*Um\_Fu <- 0*

*## Then combine these to calculate compartment changes.*

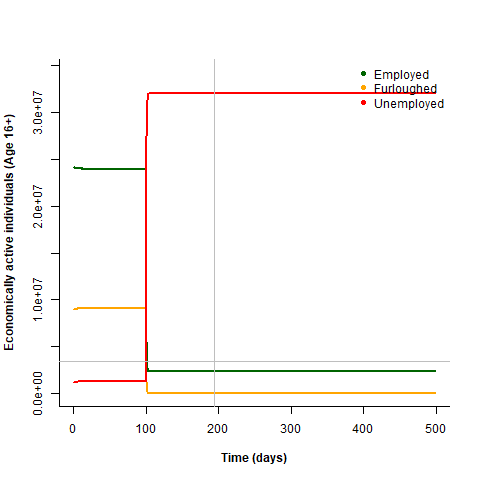
*Emz <- Emz - Em\_Un - Em\_Fu + Fu\_Em + Un\_Em*

*Fuz <- Fuz - Fu\_Em - Fu\_Un + Em\_Fu + Um\_Fu*

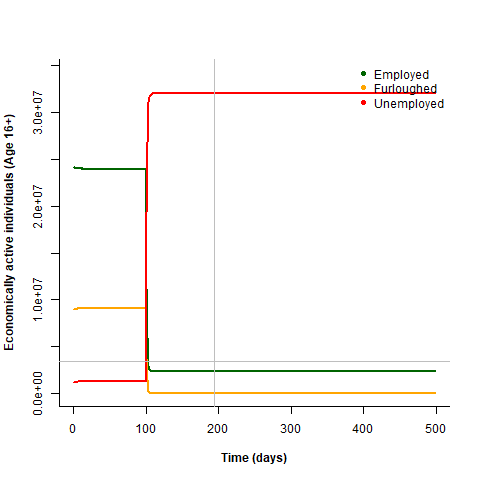
*Unz <- Unz - Un\_Em - Um\_Fu + Em\_Un + Fu\_Un*

Ok, running again at fidelity:

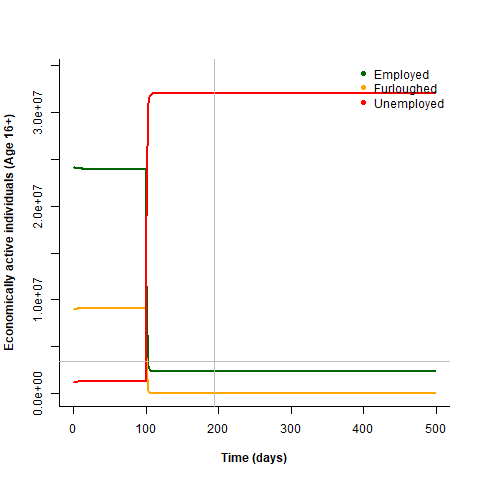
1:



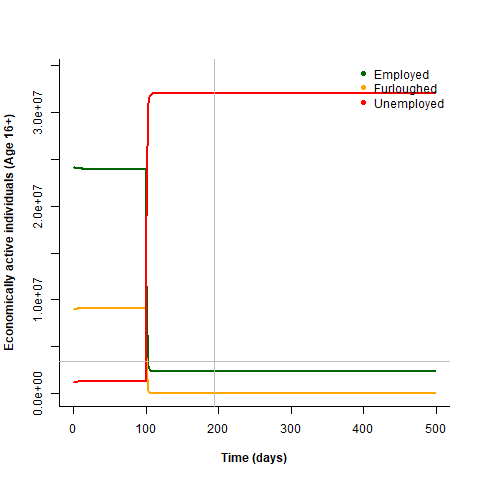
10:



1000:



100,000:



Ok great, looks like the fidelity issue is sorted.

Next we need to generate parameter values for peri-furlough that give plausible results up to october (and maybe a bit beyond). Then we can fix those and then vary the others to see what fits and looks vaguely plausible. Then we are ready to go from this base model to fitting to a certain unemployment rate at a certain date and also explore scenarios.

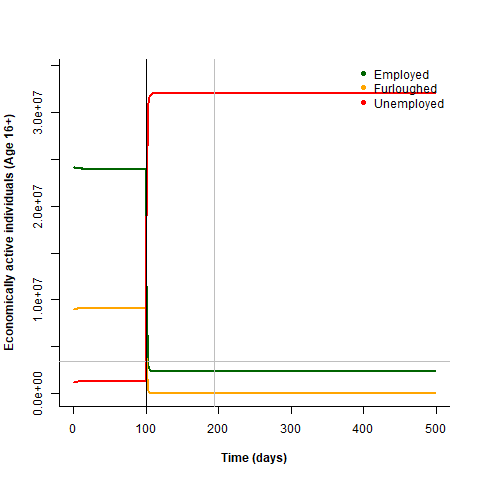
>just a reminder here that the model does not include self employed with temporary or permanent closure of own business.

# Model development – ver 200625

Ok, first I want to modify the parameters to get a more realistic scenario.

Actually just before that I want to add a line to the graph to show the transition to post-furlough.

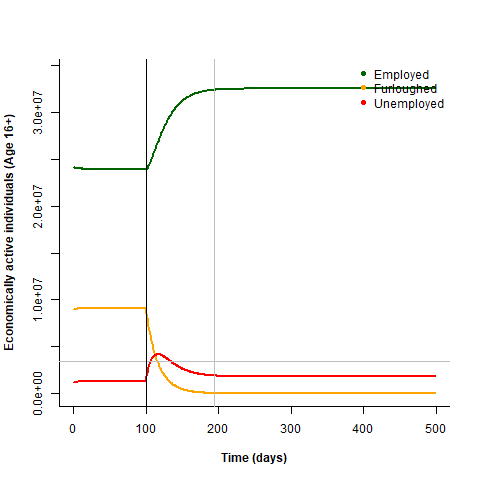
Ok done.



Ok, well this graph has some serious issues. For a start in both simulation epochs we reach a steady-state, which seems unlikely – particularly in the post-furlough time.

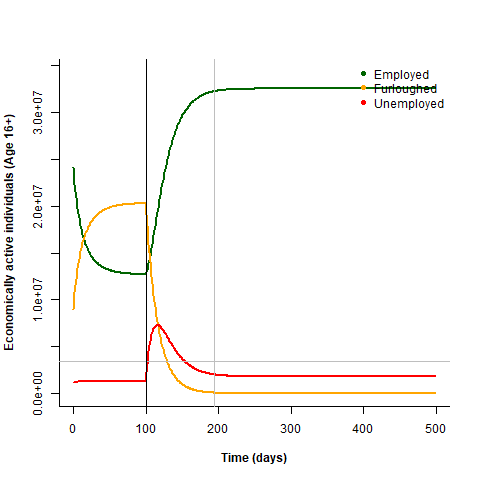
First though, why does Employed drop so massively. It really shouldn’t. mostly it’s transfer to unemployed, which is (Emz\*post\_dEmUn) / fidelity, where post\_dEmUn = 0.904. ah yes, I copied the wrong number. That should be 0.001. in fact I’m going to check them all at this point.

Also found that a couple of others in post had the wrong numbers copied in. running again:



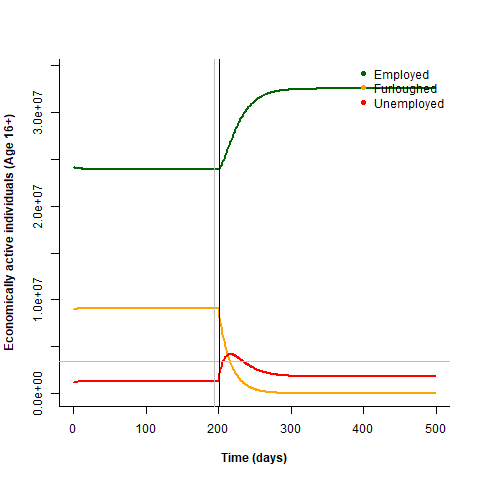
Ah, this is more like it.

Just as a demo, if I let the peri\_furMax equal 0.7 instead of 0.29 then I get this:



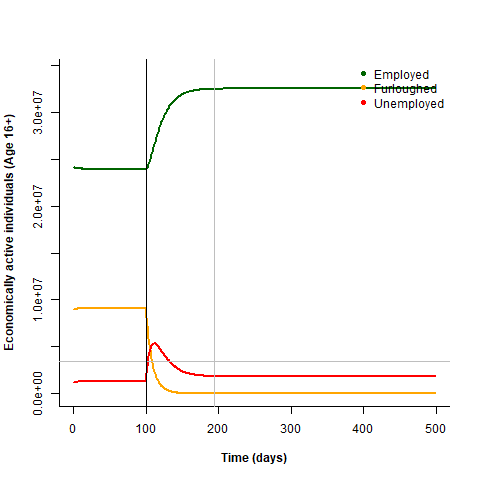
Which shows that the first half of the simulation does respond appropriately.

What if I instead extend the furlough scheme to 201 days?



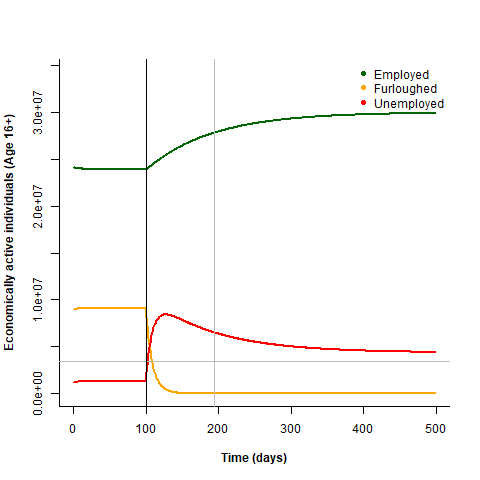
Basically the same, which isn’t a surprise as the peri simulation is already basically at steady-state. Is this reasonable? Probably not, and highlights the potential need to include lockdown paramters and/or economy parameters.

What If I increase the rate of transfer from furloughed to unemployed from 0.05 to 0.10?



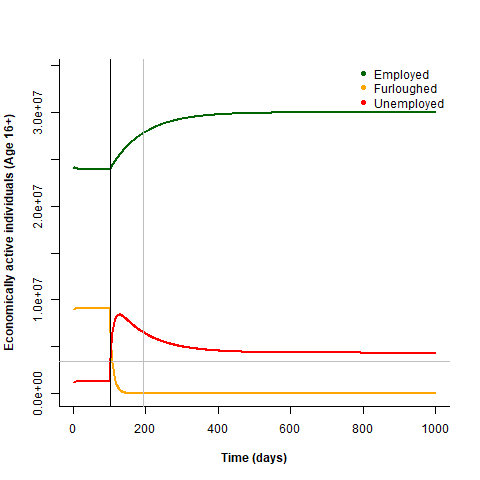
This gives bigger spike in unemployment. In the short term, but the long-term trend is the same.

What if I now also decrease the rate from unemployed to employed from 0.69 to 0.1?



A bigger spike in unemployment that goes on much longer.

If I now double the simTime:



Huh, it never returns to 3.9% why is that? Ah, probably because the rate from employed to unemployed is still 0.001 and so the equilibrium is off. This balance is actually important, because the ratio of dEmUn:dUnEm in post determines the eventual steady state unemployment. This again will depend on the lockdown and economy.

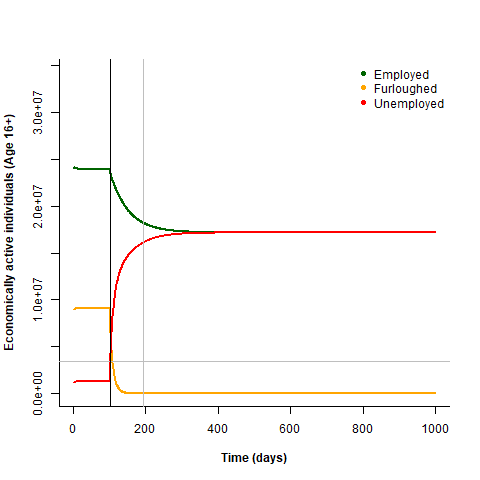
Ok. Getting off topic here. First I need to get a plausible scenario, and then I can try tweaking the variables to explore other scenarios.

So the target is 10% unemployment on the 1st January 2021, which is day 199 or something, and is marked on the figure above. In fact it might be better to have it as a point, then there are fewer lines to worry about. Leaving it for now.

Ok, so I’m going to fix the peri parameters, as the current data does suggest we’re reaching steady state (this could be wrong!). also fixing endFur as that’s the current government position. This leaves 5 post parameters. unMin in theory could be fixed. But here again unMin depends a lot on the economy and so having dEmUn, dUnEm and unMin seems like a duplication of information here. In theory picking the correct ratio of dEmUn:dUnEm would result in the unMin we want.

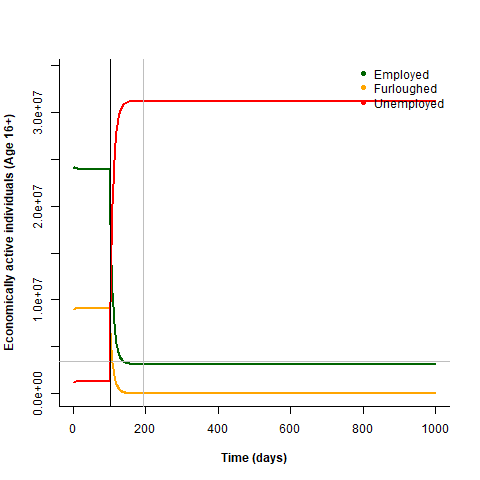
So in the post phase there is no movement into furloughing, which quickly falls to 0. That means that the long term value of Un is dependent only on Em\_Un and Un\_Em. If I remove the post\_UnMin from the partial equation, making it just a set proportion of Un then I could see what ratio of dEmUn:dUnEm gives an unemployment of 3.9%.

Running that at 0.01:0.01:



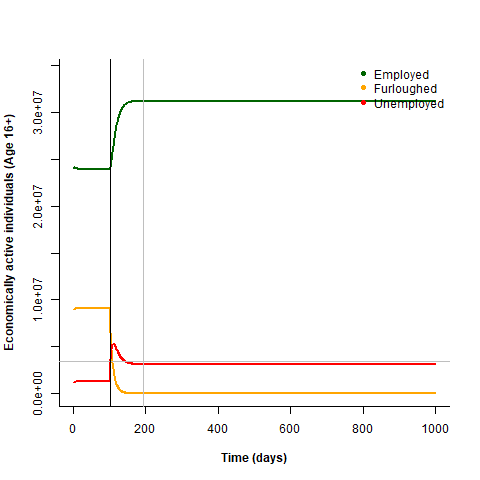
Gives a 1:1 ratio, not surprising.

What about 0.1:0.01



Gives massive unemployment, makes sense.

What about 0.01:0.1

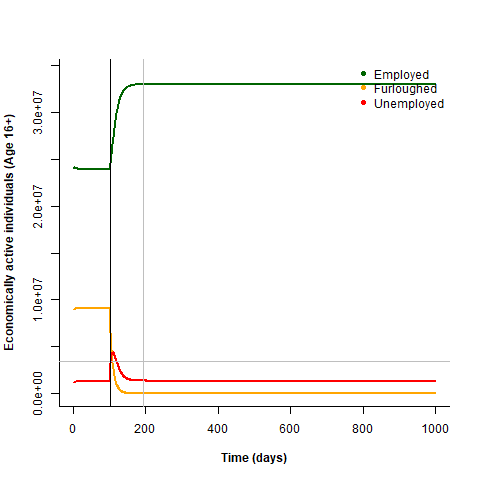


That looks more reasonable, gives around 10% stable unemployment.

Ok, after some more tests a ratio of 0.041 gives a stable Un rate of 3.9%:

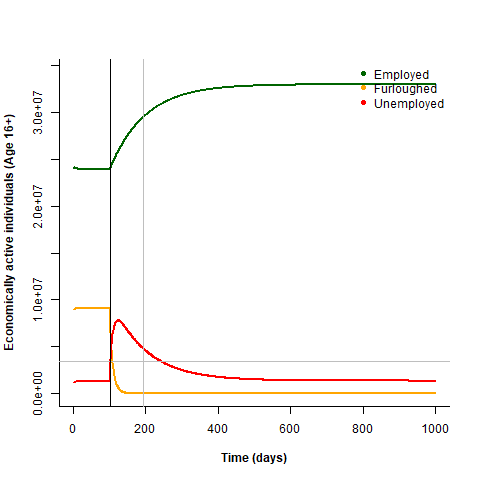


So taking value of 0.0041:0.01 gives:



Which is a very quick return.

Values of 0.00041:0.001 gives:

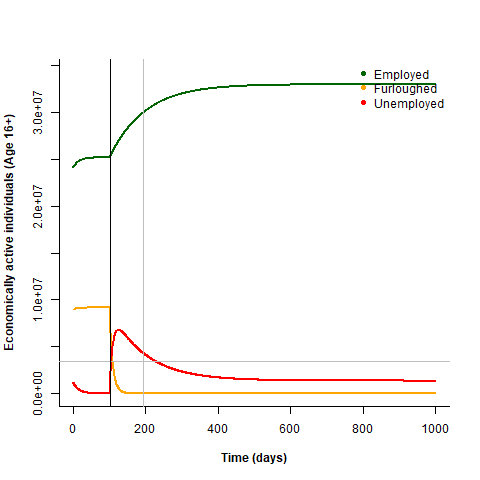


Which is much slower and more reasonable.

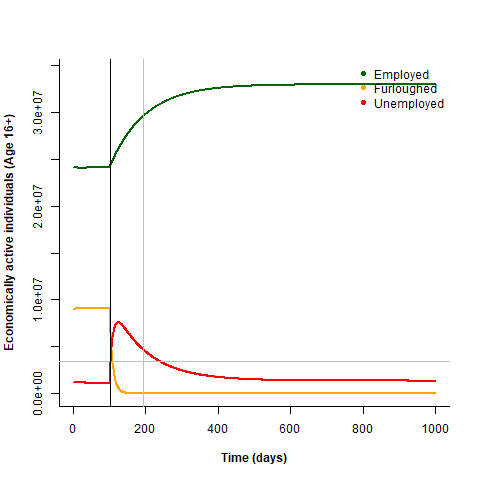
So we can get rid of the post\_UnMin parameter entirely.

The same logic should apply to the per\_UnMin parameter. We now know that without any furloughing the baseline ratio is 0.041. however, we’ve used the assumption that people are being furloughed and then made redundant and that the rate of Em\_Un is 0. What I could do instead is have the baseline ratio of 0.041, but then add on furlough-specific rates.

What happens if I just drop the peri\_UnMin parameter in the same way?

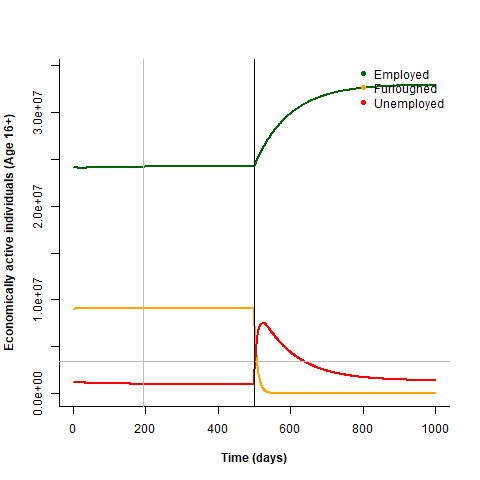


Well, unemployment drops as would be expected. But now what happens if I make the ratio of Em\_Un:Un\_Em 0.041



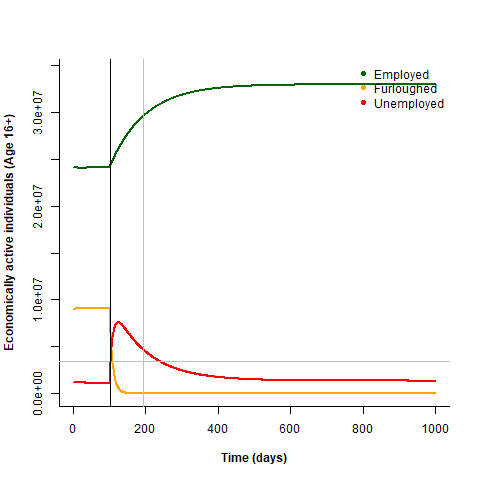
Unemployment falls less.

Just out of interest what if I now make the end of furlough much later?



So pretty much at steady state already then.

So let’s get rid of the peri\_UnMin parameter entirely as well.



That looks fine.

Really we want to then exchange those two rates for a ratio (which gives the steady-state unemployment rate) and a multiplier (which gives the rate at which you approach it). For now let’s just leave it as is.

In fact the ratio of Fu\_Em:Fu\_Un would also better be represented by a ratio and a rate, exactly as above, with the ratio determining the steady state and the rate determining how fast it gets there. Then the post model has effectively 4 paramters: the ratio of furlough transfer to employment or unemployment, the ratio of employment>unemployment and the reverse, and then the rates for both of those.

The ratio of furlough transfer to employment or unemployment is also intuitively much easier to understand. If it’s 10 then 10x as many furloughed people are being rehired as made redundant.

So how would these correspond?

If our current values are:

post\_dEmUn <- 0.00041

post\_dFuEm <- 0.01

post\_dFuUn <- 0.1

post\_dUnEm <- 0.01

then FuEm:FuUn ratio is 0.01:0.1, = 0.1

then FuEm is the ratio / 10

and FuUn is just 1 / 10

the Em\_Un:Un\_Em ratio is 0.00041:0.01 = 0.041

then Em\_un is the ratio /100

and Un\_Em is 1 / 100

so the new parameters would be:

post\_FuEm\_FuUn\_ratio = 0.1

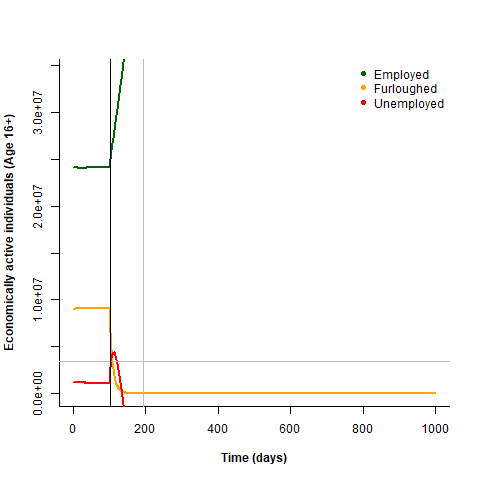
post\_FuEm\_FuUn\_damp = 10

post\_EmUn\_UnEm\_ratio = 0.041

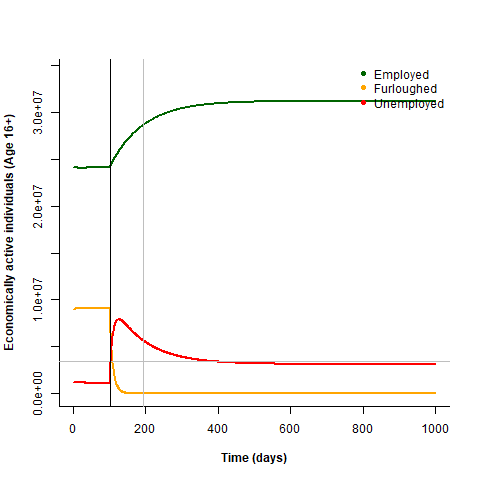
post\_EmUn\_UnEm\_damp = 100

Where ‘damp’ is short for dampening.

Substituted those in, but get this:

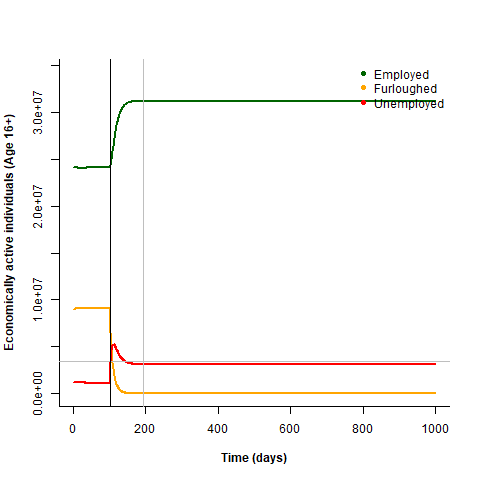


Well they likely still sum to 1, but when has unemployed gone negative. The maximum it should lose should always be dependent on it’s current size. Ah, a typo. Does it work now?



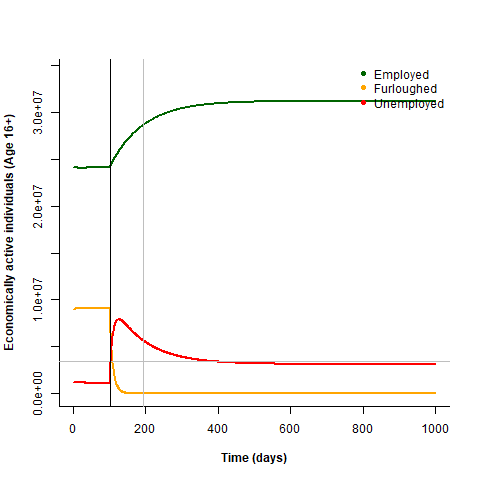
Well yes, but the steady state unemployment has increase from 3.9 to around 10%. An error with the params calculation?

What if I change the damp from 100 to 10?

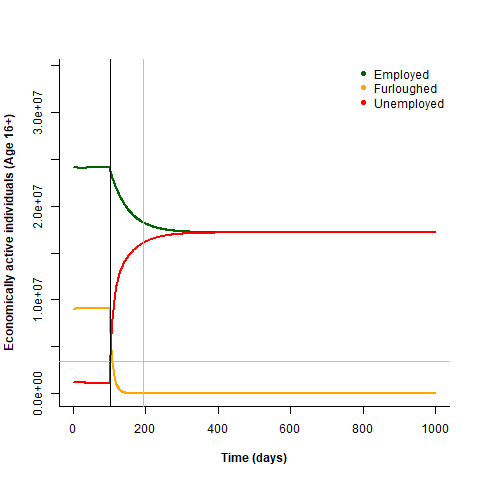


So that’s encouraging. The steady state is still the same.

What if I change the ratio from 0.041 to 1? I would expect the employment:unemployment count to have a ratio of 1:1:

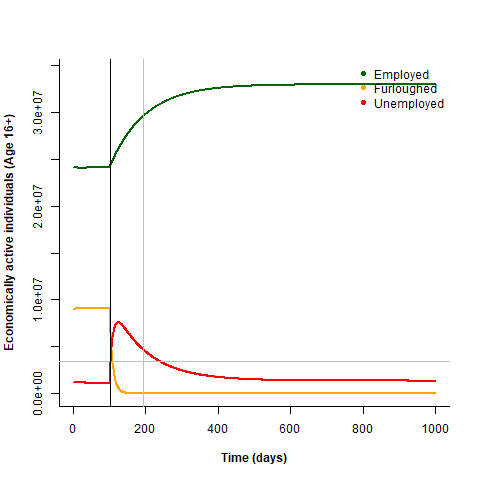


Ahah, no change, so an error with looking up the correct ratio value then?



There we go.

And returning the ratio to 0.041:



Great. Identical to before I re-parameterised.

Now we should re-parameterise the peri values in the same way.

peri\_furMax <- 0.29132

peri\_dEmFu <- 0.1

peri\_dEmUn <- 0.00041

peri\_dFuEm <- 0.006760946

peri\_dUnEm <- 0.01

Em\_Fu <- (Emz\*(peri\_furMax-Fuz)\*peri\_dEmFu) / fidelity

Em\_Un <- (Emz\*peri\_dEmUn) / fidelity

Fu\_Em <- (Fuz\*peri\_dFuEm) / fidelity

Fu\_Un <- 0

Un\_Em <- (Unz\*peri\_dUnEm) / fidelity

Um\_Fu <- 0

So peri\_dEmUn and peri\_dUnEm can actually be done in exactly the same way as with the post, because the ratio and dampening are the same.

For Fu\_Em and EM\_Fu we have the extra complexity of the furlough max, but still shouldn’t be too difficult.

peri\_dFuEm <- 0.006760946

peri\_furMax <- 0.29132

peri\_dEmFu <- 0.1

so EmFu:FuEm ratio is 0.1:0.006760946, or 1:0.06760946, = 14.7908427

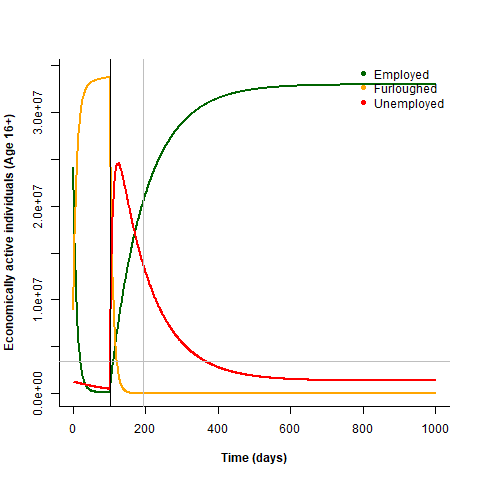
and then the damp is 147.90843

so then dEmFu = 14.7908427/147.90843

and dFuEm = 1/ 14.7908427

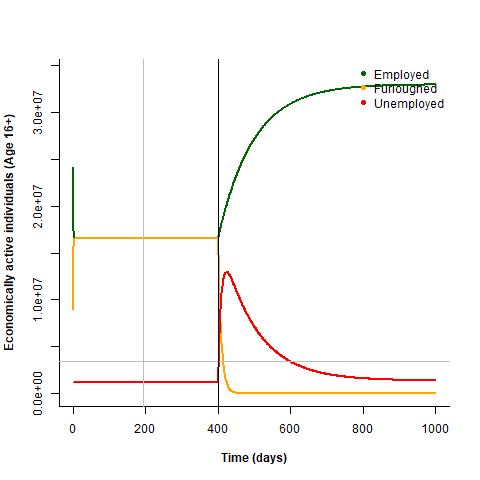
we could do the same thing as for the post period and get rid of our furMax parameter, but then it would be a case of finding the exact balance of parameters to create the steady state of 10m furloughed. I mean it could be done…

ok, done it?...



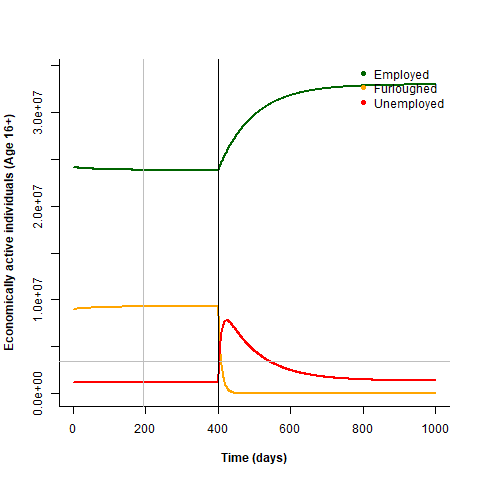
Ok that looks kind of nuts, but you’d expect it too because the values will be wrong without the furMax. What I want to do is set the damps very small and then alter the ratios until I maintain this steady state. Also extending the furlough so we have longer to get into steady state.

If I freeze unemployment and have a ratio of 1 for Em\_Fu I end up with everyone furloughed, which makes no sense. Ah, another copy-paste typo?

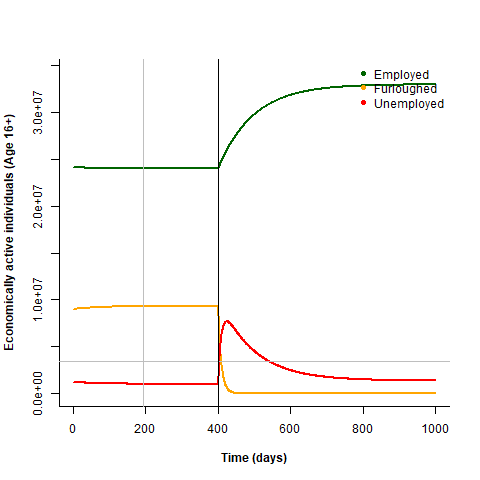


Yes, now we’re cooking.

Ok, with a ratio of 0.39 and damp of 100 we get:

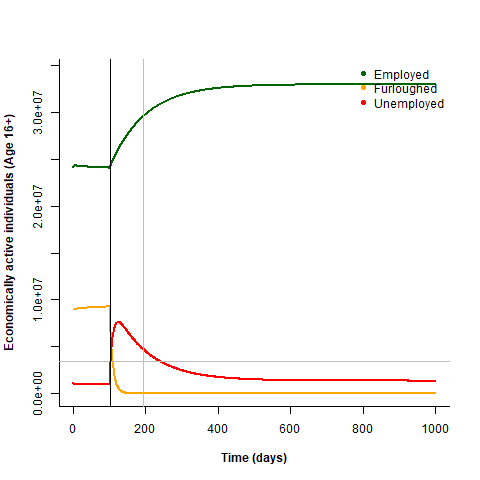


But now I need to release the unemployed again…

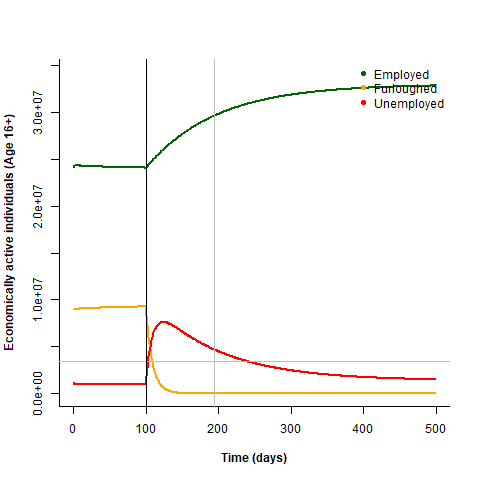


Ok, looks good, although the unemployed goes down a bit… this will be because there are fewer employed in the balance and I have no unemployed coming in from furloughing. Another thing to bear in mind for an updated model.

Returning end of furlough date to 101 days:



And shortening time frame to 500 again:



Ok, this is something like our base scenario, in which on the end of furloughing the furloughed workers move to Employed and Unemployed in a 1:10 ratio at quite a fast rate (damp=10). At the same time the unemployed move to employed at a relative rate of 1/0.041=24.39, but this transition is slower (damp=100). Therefore there is a large pulse of unemployed people, which takes a long time to return to baseline.

>incidentally the current setup of the peri params means that furlough number eventually lies at 9.3M steady state.

# Scenario sims

## Parameters

## Baseline scenario

Leaving all the peri params at:

peri\_EmFu\_FuEm\_ratio <-0.39

peri\_EmFu\_FuEm\_damp <-100

peri\_EmUn\_UnEm\_ratio <-0.041

peri\_EmUn\_UnEm\_damp <- 100

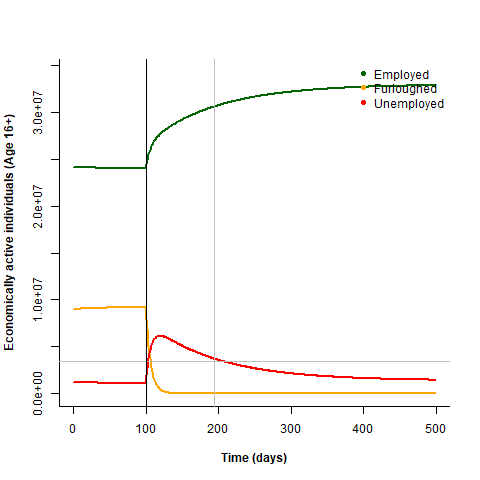
baseline post params are:

post\_FuEm\_FuUn\_ratio <- 0.5

post\_FuEm\_FuUn\_damp <- 10

post\_EmUn\_UnEm\_ratio <- 0.041

post\_EmUn\_UnEm\_damp <- 100



This results in a pulse of unemployment immediately post-furlough that gradually returns to the baseline unemployment.

## Severe social lockdown

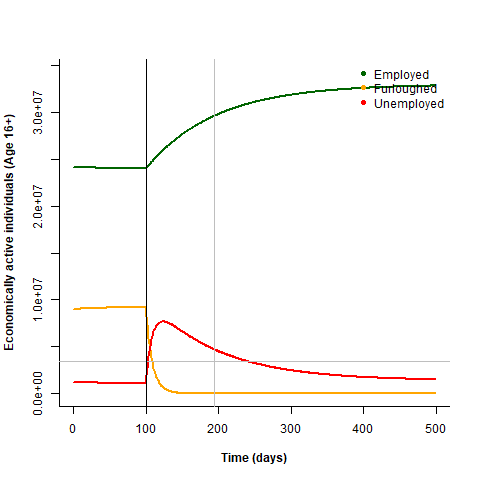
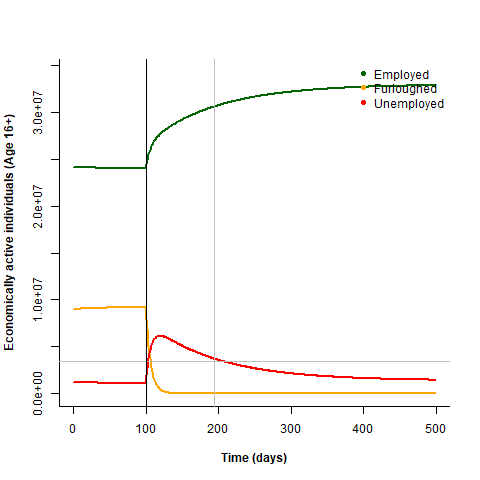
To simulate a severe lockdown policy post-furlough that prevents people from returning to work immediately I decrease the FuEm:FuUn ratio:

post\_FuEm\_FuUn\_ratio <- 0.1

post\_FuEm\_FuUn\_damp <- 10

post\_EmUn\_UnEm\_ratio <- 0.041

post\_EmUn\_UnEm\_damp <- 100



This results in a higher pulse of unemployment post-furlough.

## Relaxed social lockdown

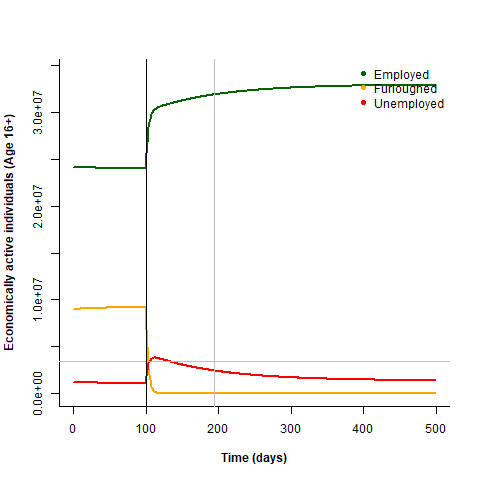
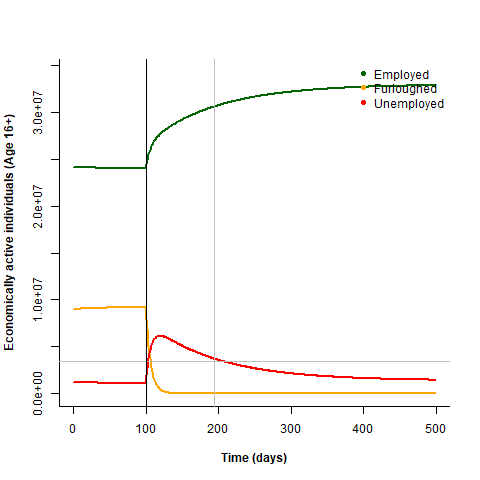
To simulate a relaxed lockdown policy that allows businesses to reopen and furloughed workers to return to work immediately I increase the FuEm:FuUn ratio:

post\_FuEm\_FuUn\_ratio <- 2

post\_FuEm\_FuUn\_damp <- 10

post\_EmUn\_UnEm\_ratio <- 0.041

post\_EmUn\_UnEm\_damp <- 100



This results in a much smaller pulse of unemployment immediately post-furlough.

## Economic recession

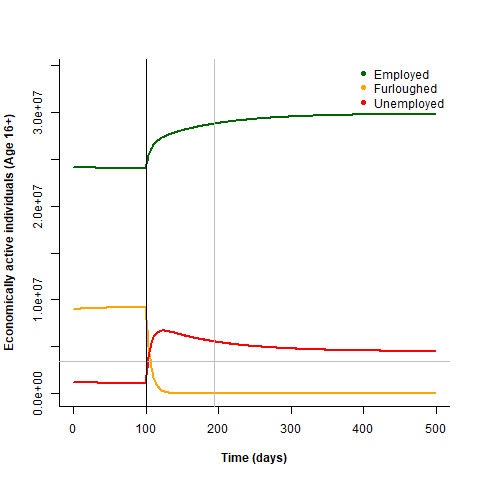
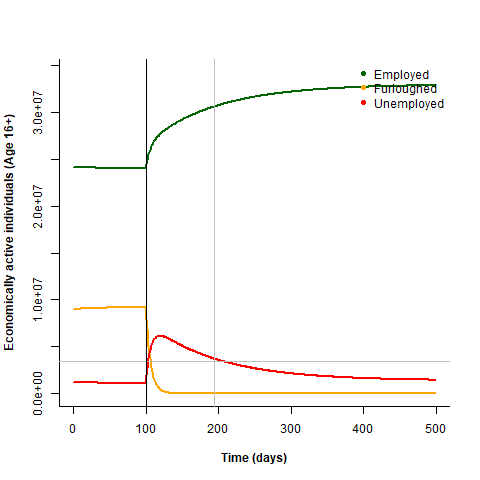
To simulate an economic recession post-furlough that reduces job opportunities in the long term I increase the EmUn:UnEm ratio:

post\_FuEm\_FuUn\_ratio <- 0.5

post\_FuEm\_FuUn\_damp <- 10

post\_EmUn\_UnEm\_ratio <- 0.15

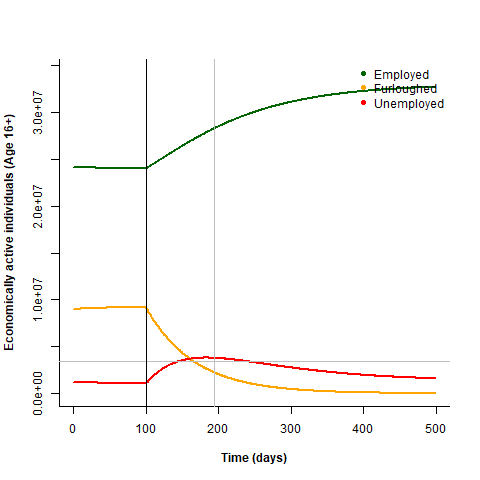
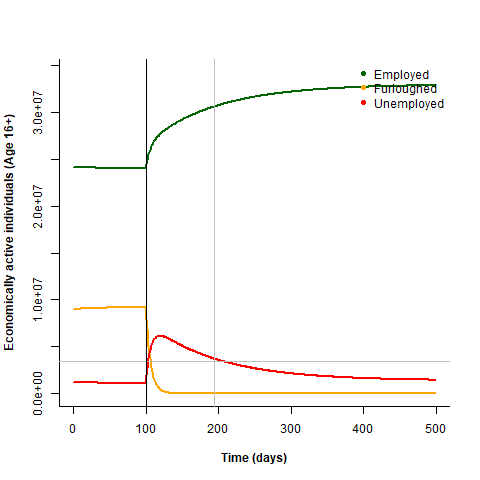
post\_EmUn\_UnEm\_damp <- 100



This results in a much weaker recovery after the pulse of unemployment.

## Phased end of furlough

To simulate a phased ending to the furlough scheme I increase the FuEm\_FuUn dampening, so that employees are rehired or made redundant gradually.



This results in a flattened peak of unemployment and a more gradual rehiring of furloughed workers.

## Faster economic recovery

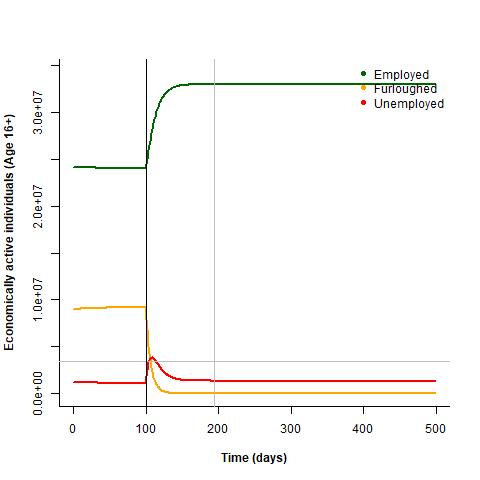
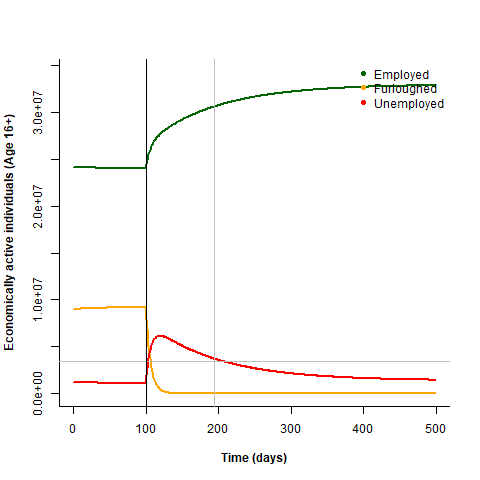
To simulate a faster economic recovery back to pre-covid levels after the end of the furlough scheme I decrease the EmUn\_UnEm dampening.

post\_FuEm\_FuUn\_ratio <- 0.5

post\_FuEm\_FuUn\_damp <- 10

post\_EmUn\_UnEm\_ratio <- 0.041

post\_EmUn\_UnEm\_damp <- 10



This results in a smaller peak of unemployment and faster return to pre-covid unemployment levels.