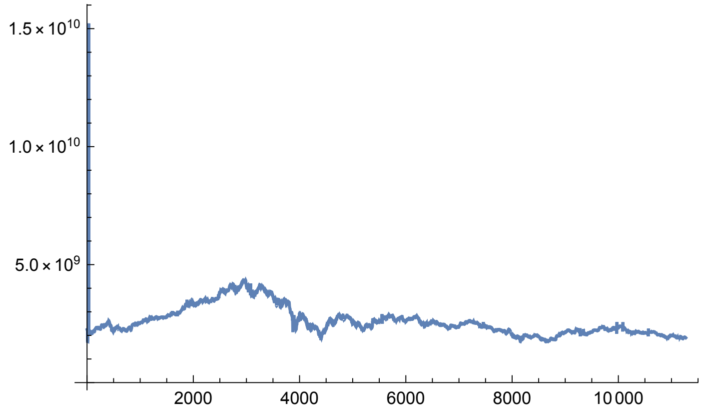
```
In[280]:= (* Fits to log stable below. Starting
            with 120d worth of data from WETH/USDC .... *)

In[2]:= (* Import from csv *)

In[3]:= Directory[]

Out[3]= /Users/personal

In[281]:= Module[{directory = SystemDialogInput["Directory"]},
            If[directory =!= $Canceled, SetDirectory[directory]]]

Out[281]= /Users/personal/Desktop/note7/points

In[5]:= (* Go back and download WETH/USDC data from cron. Look at that *)

In[294]:= tblWethUsdc120d = Import["120/data-1626472963_weth-usdc-twap.csv"]

Out[294]= {{, timestamp, twap}, {0, 1.61878 × 10^9, 2.23957 × 10^9},
            {1, 1.61878 × 10^9, 2.24054 × 10^9}, {2, 1.61878 × 10^9, 2.23817 × 10^9},
            {3, 1.61878 × 10^9, 2.25083 × 10^9}, {4, 1.61878 × 10^9, 2.25656 × 10^9}, … 11 259 …,
            {11 266, 1.62647 × 10^9, 1.91147 × 10^9}, {11 267, 1.62647 × 10^9, 1.90971 × 10^9},
            {11 268, 1.62647 × 10^9, 1.90596 × 10^9}, {11 269, 1.62647 × 10^9, 1.90286 × 10^9},
            {11 270, 1.62647 × 10^9, 1.90034 × 10^9}, {11 271, 1.62647 × 10^9, 1.89838 × 10^9}}

          large output | show less | show more | show all | set size limit...

In[295]:= Length[tblWethUsdc120d]

Out[295]= 11 271

In[296]:= FromUnixTime[tblWethUsdc120d[[2]][[2]]]

Out[296]= Sun 18 Apr 2021 16:50:52 GMT–4.

In[299]:= FromUnixTime[tblWethUsdc120d[[Length[tblWethUsdc120d]]][[2]]]

Out[299]= Fri 16 Jul 2021 18:01:02 GMT–4.

In[300]:= twapsWethUsdc120d =
            Table[tblWethUsdc120d[[i]][[3]], {i, 2, Length[tblWethUsdc120d]}]
```

In[301]:=
```
ListLinePlot[twapsWethUsdc120d, PlotRange → All]
```

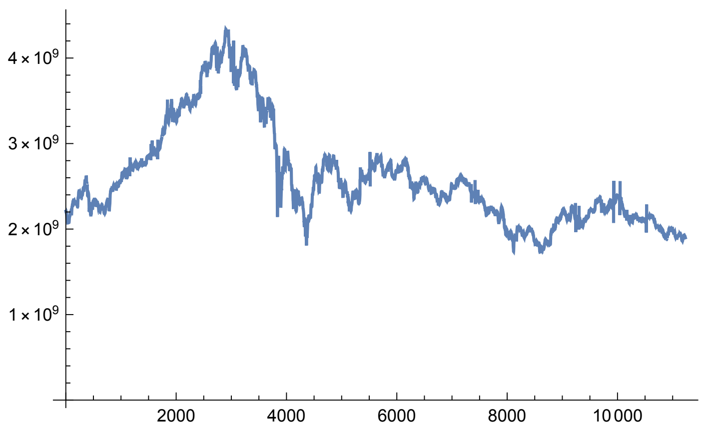
Out[301]=



In[302]:=
```
(* Same issues here for early data as UNI/WETH. Cut off first 40 elements *)
twapsWethUsdc120dFiltered =
 Table[twapsWethUsdc120d[[i]], {i, 40, Length[twapsWethUsdc120d]}]
```

In[303]:=
```
ListLinePlot[twapsWethUsdc120dFiltered, PlotRange → All]
```

Out[303]=



In[304]:=
```
FromUnixTime[tblWethUsdc120d[[40]][[2]]]
```

Out[304]= Mon 19 Apr 2021 17:14:44 GMT−4.

In[305]:=
```
FromUnixTime[tblWethUsdc120d[[Length[twapsWethUsdc120dFiltered]]][[2]]]
```

Out[305]= Fri 16 Jul 2021 10:27:07 GMT−4.

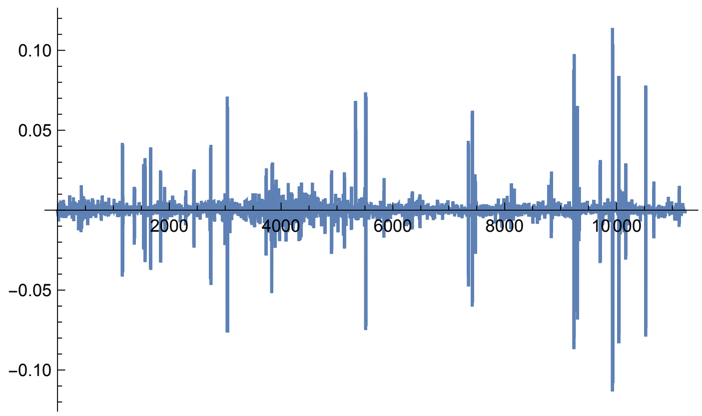
In[16]:=
```
(* Calculate the rs ... *)
```

```
In[306]:= rsWethUsdc120dFiltered = Differences[Log[twapsWethUsdc120dFiltered]]

Out[306]= {-0.00120673, 0.000176452, -0.0000603568, -0.000373722, -0.000389599,
  0.000559858, -0.00092094, -0.00154871, -0.000905547, -0.000584311,
  … 11 210 …, -0.000975245, -0.000888855, -0.00068242, -0.000128976,
  -0.00026784, -0.000921873, -0.00196508, -0.00162411, -0.00132945, -0.00102799}
```

large output | show less | show more | show all | set size limit...

```
In[307]:= rsWethUsdc120dFiltered[[100]]

Out[307]= -0.00443888

In[308]:= ListLinePlot[rsWethUsdc120dFiltered, PlotRange → All]
```

Out[308]=




```
In[309]:= Length[rsWethUsdc120dFiltered]

Out[309]= 11 230

In[311]:= edistWethUsdc120dFiltered = EstimatedDistribution[rsWethUsdc120dFiltered,
  StableDistribution[1, aWU120d, bWU120d, locWU120d, scaleWU120d]]
```

Out[311]= StableDistribution[1, 1.41834, -0.0286247, -6.93658×10^{-6}, 0.00140585]

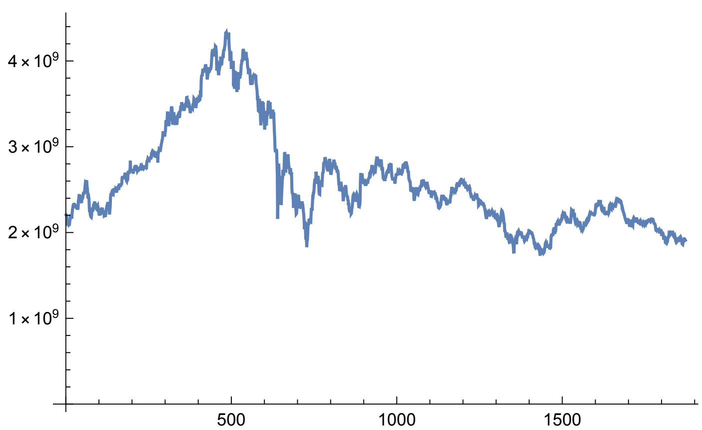
```
In[312]:= FromUnixTime[tblWethUsdc120d[[5000]][[2]]]

Out[312]= Fri 28 May 2021 06:16:07 GMT-4.

In[313]:= FromUnixTime[tblWethUsdc120d[[Length[tblWethUsdc120d]]][[2]]]

Out[313]= Fri 16 Jul 2021 18:01:02 GMT-4.

In[25]:= (* Seems a good place to estimate k
  values would be 1h candles (from note-7.nb) *)

In[26]:=

In[27]:= (* Some concrete numbers below in terms of funding rate ... *)
```

```
In[28]:= (* What does e^{mu * T + sig * (T/a)^(1/a) * F^{-1}(1-alpha)}
          translate to for ETH/USDC fit? *)

In[29]:= (* And if d = f * e^{mu * T + sig * (T/a)^(1/a) * F^{-1}(1-alpha)},
         what f value should we use across the board
           to get k max interest rates on order of 1-10% daily? *)

In[30]:= (* e.x. d = 1.0014649 for T=10m results in 10% per day funding rate *)

In[314]:= edistWethUsdc120dFiltered
```

Out[314]= StableDistribution[1, 1.41834, −0.0286247, -6.93658×10^{-6}, 0.00140585]

```
In[318]:= InverseCDF[edistWethUsdc120dFiltered, 0.99]

Out[318]= 0.0127859

In[320]:= InverseCDF[edistWethUsdc120dFiltered, 0.95]

Out[320]= 0.00460265

In[321]:= InverseCDF[edistWethUsdc120dFiltered, 0.90]

Out[321]= 0.00300704

In[322]:= Exp[InverseCDF[edistWethUsdc120dFiltered, 0.95]]

Out[322]= 1.00461

In[323]:= Exp[InverseCDF[edistWethUsdc120dFiltered, 0.99]]

Out[323]= 1.01287

In[37]:= (* Produce 1 hour candles from 10min cron data to use in k analysis ... *)

In[324]:= twapsWethUsdc120dFiltered1HourCandle = Table[
            twapsWethUsdc120dFiltered[[i]], {i, 1, Length[twapsWethUsdc120dFiltered], 6}]

In[325]:= ListLinePlot[twapsWethUsdc120dFiltered1HourCandle]
```

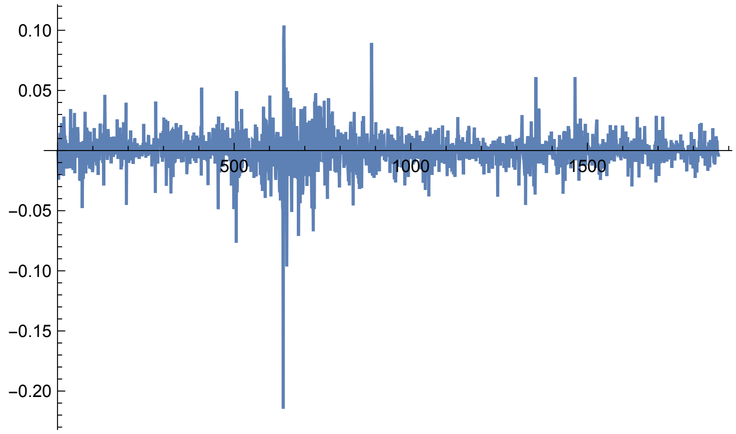
Out[325]=




```
In[326]:= rsWethUsdc120dFiltered1HourCandle =
           Differences[Log[twapsWethUsdc120dFiltered1HourCandle]]
```

In[327]:=
```
Length[rsWethUsdc120dFiltered1HourCandle]
```
Out[327]= 1871

In[328]:=
```
ListLinePlot[rsWethUsdc120dFiltered1HourCandle, PlotRange → All]
```
Out[328]=



In[333]:=
```
edistWethUsdc120dFiltered1HourCandle =
 EstimatedDistribution[rsWethUsdc120dFiltered1HourCandle,
  StableDistribution[1, aWU120dC1h, bWU120dC1h, locWU120dC1h, scaleWU120dC1h]]
```
Out[333]= StableDistribution[1, 1.5843, -0.0493885, -0.000101694, 0.0072772]

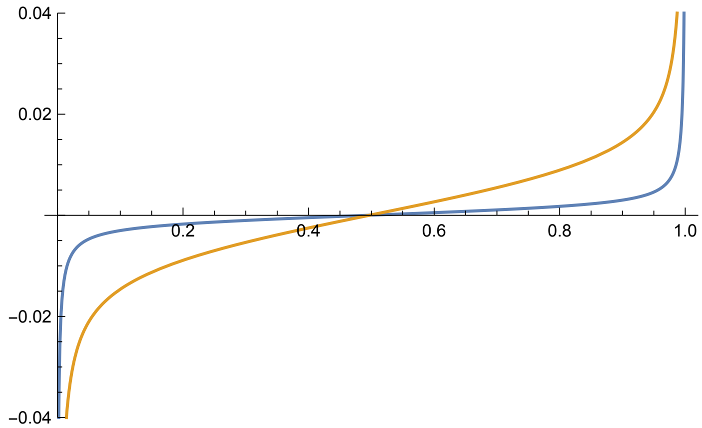
```
(* Check how similar compared to previous fit without new data ... *)

edistWethUsdc120dFiltered1HourCandle
```
Out[338]= StableDistribution[1, 1.59768, -0.0971292, -0.000157254, 0.00790013]

```
(* Not bad, especially on sigma and alpha ... *)
```

In[43]:=
```
(* Interesting ... compound less on the funding payments,
have more leeway. How does that make sense wrt inverse cdf exponentiation? *)
```

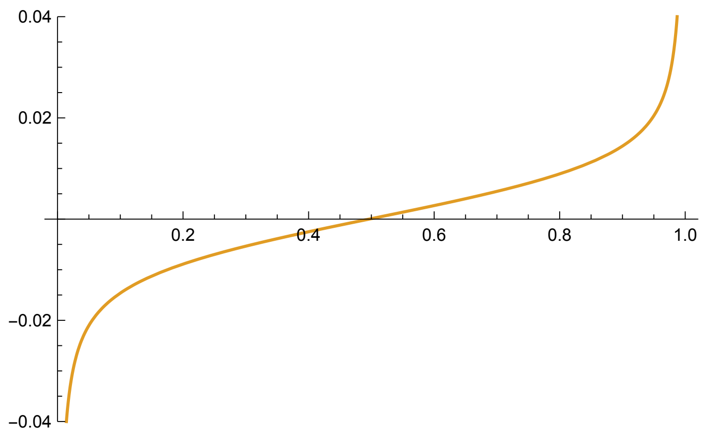
In[334]:=
```
Plot[{InverseCDF[edistWethUsdc120dFiltered, x],
  InverseCDF[edistWethUsdc120dFiltered1HourCandle, x],
  InverseCDF[edistWethUsdc120dFiltered4HourCandle, x]},
 {x, 0, 1.0}, PlotRange → {-0.04, 0.04}]
```

Out[334]=



In[335]:=
```
Plot[{, InverseCDF[edistWethUsdc120dFiltered1HourCandle, x]},
 {x, 0, 1.0},  PlotRange → {-0.04, 0.04}]
```

Out[335]=



In[336]:=
```
InverseCDF[edistWethUsdc120dFiltered1HourCandle, 0.99]
```

Out[336]= `0.0459063`

In[337]:=
```
InverseCDF[edistWethUsdc120dFiltered, 0.99]
```

Out[337]= `0.0127859`

In[339]:=
```
Exp[0.012785934108095686`]
```

Out[339]= `1.01287`

In[50]:=
```
(* Hmmm, so maybe plot e^(mu * T + sig * (T/a)^(1/a) * F^{-1}(1-alpha)) as a
 function of update times T. What does this tell us vs d we're looking for? *)
```

In[340]:=
```
edistWethUsdc120dFiltered1HourCandle
```

Out[340]=
```
StableDistribution[1, 1.5843, -0.0493885, -0.000101694, 0.0072772]
```

```
In[52]:= (* Define mu, sig functions for exponential term in d *)

In[53]:= mu[a_, loc_] := loc

In[54]:= sig[a_, scale_] := scale / (1 / a) ^ (1 / a)

In[55]:= (* Apply to our case *)

In[342]:= muWethUsdc120dFiltered1HourCandle =
          mu[1.5843003587383822`, -0.00010169418797317535`]

Out[342]= -0.000101694

In[343]:= sigWethUsdc120dFiltered1HourCandle =
          sig[1.5843003587383822`, 0.00727719566405363`]

Out[343]= 0.00972972

In[341]:= aWethUsdc120dFiltered1HourCandle = 1.5843003587383822`

Out[341]= 1.5843

In[345]:= edistWethUsdc120dFiltered1HourCandleNormalized =
          StableDistribution[1, aWethUsdc120dFiltered1HourCandle, 0, 0, 1]

Out[345]= StableDistribution[1, 1.5843, 0, 0, 1]

In[346]:= InverseCDF[edistWethUsdc120dFiltered1HourCandleNormalized, 0.90]

Out[346]= 1.99592

In[347]:= InverseCDF[edistWethUsdc120dFiltered1HourCandleNormalized, 0.95]

Out[347]= 2.84713

In[348]:= InverseCDF[edistWethUsdc120dFiltered1HourCandleNormalized, 0.99]

Out[348]= 6.48811

In[349]:= factorWethUsdc120dFiltered1HourCandle[t_, alpha_] :=
          Exp[muWethUsdc120dFiltered1HourCandle * t + sigWethUsdc120dFiltered1HourCandle *
             (t / aWethUsdc120dFiltered1HourCandle) ^ (1 / aWethUsdc120dFiltered1HourCandle) *
             InverseCDF[edistWethUsdc120dFiltered1HourCandleNormalized, 1 - alpha]]

In[350]:= factorWethUsdc120dFiltered1HourCandle[1, 0.01]

Out[350]= 1.04824

In[65]:= (* TODO: Redo this analysis! ... Below is in line with 4 hour candle above :) *)

In[351]:= factorWethUsdc120dFiltered1HourCandle[4, 0.01]

Out[351]= 1.11947

In[352]:= factorWethUsdc120dFiltered1HourCandle[8, 0.01]

Out[352]= 1.19079
```

In[68]:= (* Over 24 hours you can see the difference substantially at different confidence levels ... *)

In[353]:= factorWethUsdc120dFiltered1HourCandle[24, 0.01]

Out[353]= 1.41697

In[354]:= factorWethUsdc120dFiltered1HourCandle[24, 0.10]

Out[354]= 1.11129

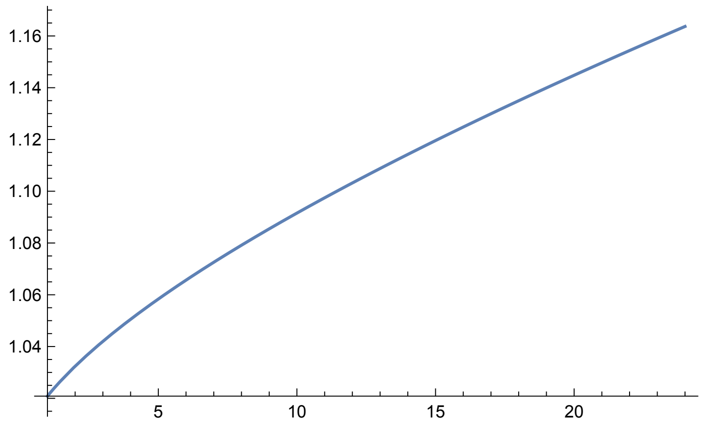
In[355]:= factorWethUsdc120dFiltered1HourCandle[24, 0.05]

Out[355]= 1.16366

In[72]:= (* VaR at 95% seems good here in terms of usability for trading. 15% draw down in a day worst case on ETH/USDC is not terrible in terms of funding rate max *)

In[356]:= Plot[factorWethUsdc120dFiltered1HourCandle[t, 0.05], {t, 1, 24}]

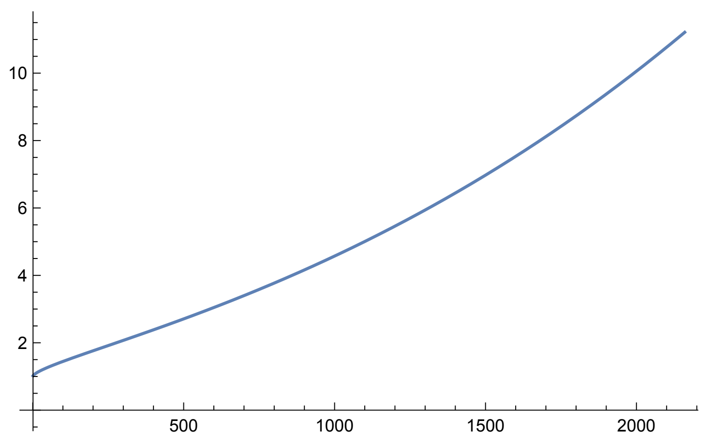
Out[356]=



In[74]:= (* Interesting S shape here ... Due to the Exp[t^(1/1.5)] term *)

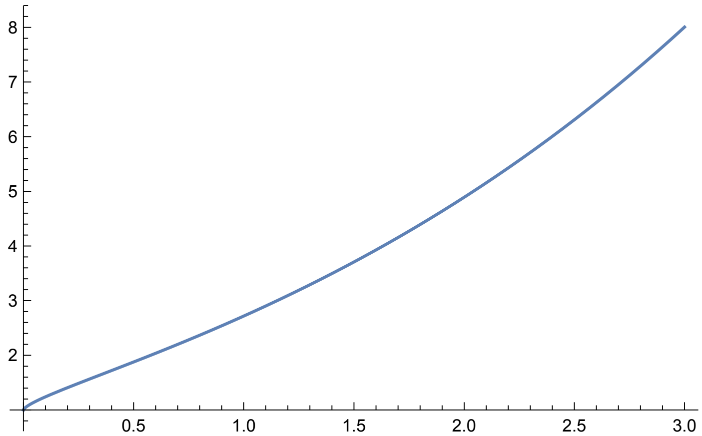
In[359]:= Plot[factorWethUsdc120dFiltered1HourCandle[t, 0.05], {t, 1, 24 * 30 * 3}]

Out[359]=



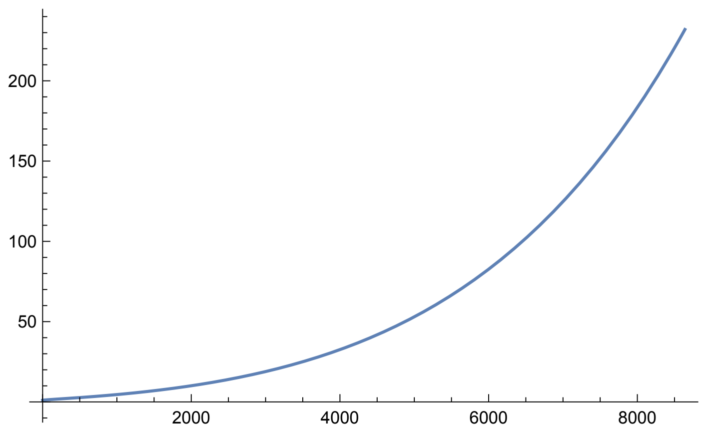
(* So takes on order of 2 months to reach a 5x price cap on price bracket amount *)

In[360]:=
```
Plot[Exp[(t)^(1/1.5)], {t, 0, 3}]
```

Out[360]=


In[77]:=
```
(* This is where caps become very good :) *)
```

In[361]:=
```
Plot[factorWethUsdc120dFiltered1HourCandle[t, 0.05], {t, 1, 24*30*12}]
```

Out[361]=


In[79]:=
```
(* Plot k AND (1-k)^m where m is diff depending on number of times we compound *)
```

In[80]:=
```
(* Assume k min is determined by factor value
 at 95% confidence for different number of periods T *)
```

In[357]:=
```
dmin95WethUsdc120dFiltered1HourCandle[t_] :=
 factorWethUsdc120dFiltered1HourCandle[t, 0.05]
```

In[358]:=
```
kmin95WethUsdc120dFiltered1HourCandle[t_] :=
 (1 - 1 / dmin95WethUsdc120dFiltered1HourCandle[t]) / 2
```

In[83]:=
```
(* Look at different time periods ... 1 (1h),
4 (4h), 6 (6h), 8 (8h), 12 (12h), 24 (1d), 168 (7d) *)
```

In[362]:=
```
kmin95T1hWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[1]
```

Out[362]= 0.0102032

In[363]:=
```
kmin95T4hWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[4]
```
Out[363]= 0.0240508

In[364]:=
```
kmin95T6hWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[6]
```
Out[364]= 0.0308052

In[365]:=
```
kmin95T8hWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[8]
```
Out[365]= 0.0366706

In[366]:=
```
kmin95T12hWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[12]
```
Out[366]= 0.0467734

In[367]:=
```
kmin95T1dWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[24]
```
Out[367]= 0.0703214

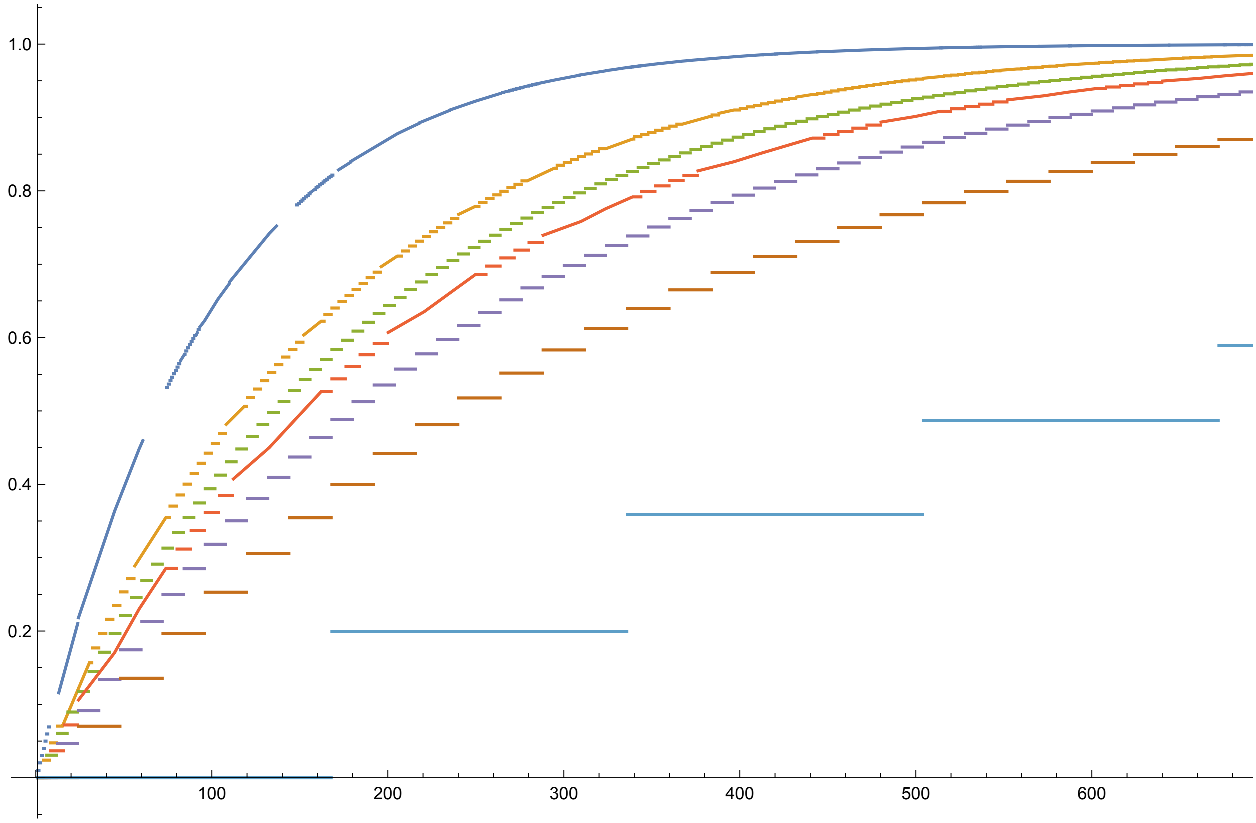
In[368]:=
```
kmin95T7dWethUsdc120dFiltered1HourCandle =
 kmin95WethUsdc120dFiltered1HourCandle[168]
```
Out[368]= 0.199422

In[369]:=
```
(* Interesting ... around minimum of 3.26% every 6 hr
 compounding. Can work with that to have some buffer in event bad fit *)
```

In[370]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandle[t_, m_] :=
 1 - (1 - kmin95WethUsdc120dFiltered1HourCandle[m]) ^ (Floor[t / m])
```

In[371]:=
```
Plot[{drawDownMin95WethUsdc120dFiltered1HourCandle[t, 1],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 4],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 6],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 8],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 12],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 24],
  drawDownMin95WethUsdc120dFiltered1HourCandle[t, 168]}, {t, 1, 24 * 30}]
```

Out[371]=

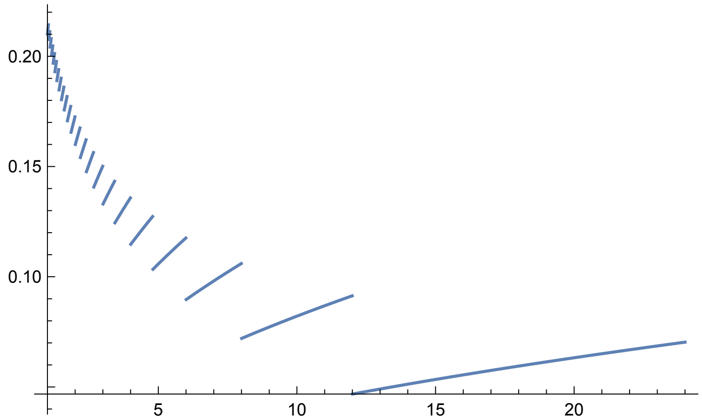


In[94]:=
```
(* Should like compare ratio of price bracket to d^(t/m). When set m for d,
setting point at which d^m =
 price bracket[m].. But d is NOT an interest rate... k = (1-1/d)/2 makes this
    interesting in terms of traders view on draw down to their OI ... *)
```

In[95]:=
```
(* Plot the actual interest rates seen per day for different values of m *)
```

In[374]:=
```
Plot[drawDownMin95WethUsdc120dFiltered1HourCandle[24, m], {m, 1, 24}]
```

Out[374]=



In[375]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandle[24, 1]
```

Out[375]=
```
0.218183
```

In[376]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandle[24, 4]
```

Out[376]=
```
0.135901
```

In[377]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandle[24, 8]
```

Out[377]=
```
0.106027
```

In[378]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandle[24, 12]
```

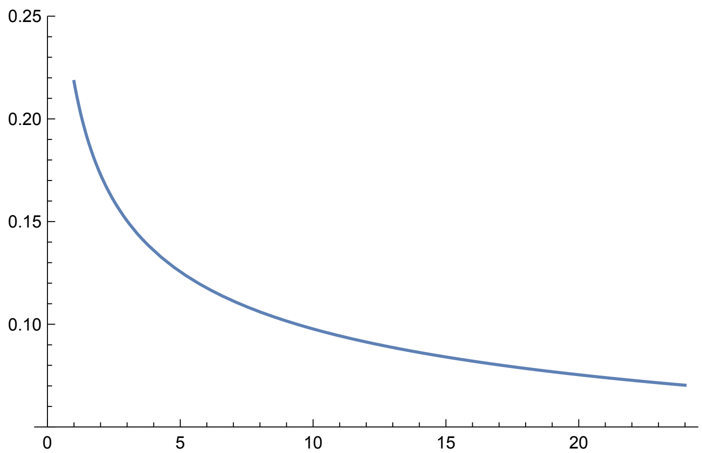
Out[378]=
```
0.091359
```

In[101]:=
```
(* Floor is making this visual
   difficult. Make continuous for plot purposes ... *)
```

In[379]:=
```
drawDownMin95WethUsdc120dFiltered1HourCandleContinuous[t_, m_] :=
 1 - (1 - kmin95WethUsdc120dFiltered1HourCandle[m]) ^ (t / m)
```

In[103]:=
```
(* Below is min funding rate required extended to per day rate,
if apply funding payment every m hours  *)
```

In[380]:=
```
Plot[drawDownMin95WethUsdc120dFiltered1HourCandleContinuous[24, m],
 {m, 1, 24}, PlotRange → {0.05, 0.25}]
```

Out[380]=



```
(* TODO: generalize this plot to 3d with alpha confidence on the other axis *)
```

In[105]:=
```
(* Less we compound, smaller max DAILY funding rate needed to overcome
 price bracket changes. But more short term (<1d) we likely take *)
```

In[381]:=
```
(* Generalize this for any confidence level (not just 95%) *)
```

In[382]:=
```
dminWethUsdc120dFiltered1HourCandle[t_, alpha_] :=
 factorWethUsdc120dFiltered1HourCandle[t, alpha]
```

In[383]:=
```
kminWethUsdc120dFiltered1HourCandle[t_, alpha_] :=
 (1 - 1 / dminWethUsdc120dFiltered1HourCandle[t, alpha]) / 2
```

In[384]:=
```
drawDownMinWethUsdc120dFiltered1HourCandle[t_, alpha_, m_] :=
 1 - (1 - kminWethUsdc120dFiltered1HourCandle[m, alpha]) ^ (Floor[t / m])
```

In[385]:=
```
drawDownMinWethUsdc120dFiltered1HourCandleContinuous[t_, alpha_, m_] :=
 1 - (1 - kminWethUsdc120dFiltered1HourCandle[m, alpha]) ^ (t / m)
```

In[386]:=
```
(* If fit to 8h compounded rate but still pay every 10min,
how much is k? (k paid every 10 min but use 8h compounding amount) *)
```

In[387]:=
```
(* Function to rescale k min calibrated using factor's value at
  time m in the future. But then applied over shorter time span < m *)
```

In[388]:=
```
(* d_shorter^(#periods of short in long) = d_longer; m > t here *)
```

In[389]:=
```
rescaledKminWethUsdc120dFiltered1HourCandle[t_, alpha_, m_] :=
 (1 - 1 / (dminWethUsdc120dFiltered1HourCandle[m, alpha]) ^ (t / m)) / 2
```

In[390]:=
```
rescaledKminWethUsdc120dFiltered1HourCandle[1 / 6, 0.05, 8]
```

Out[390]= 0.000792806

In[116]:=
```
(* Calculate effective funding rate after 8 hours applied every 10 min *)
```

```
In[397]:= 1 - (1 - rescaledKminWethUsdc120dFiltered1HourCandle[1/6, 0.05, 8]) ^ (6 * 8)

Out[397]= 0.0373542

In[398]:= (* Compare with draw down applied only once every 8 hours *)
          drawDownMinWethUsdc120dFiltered1HourCandle[8, 0.05, 8]

Out[398]= 0.0366706

In[119]:= (* Looks good *)

In[120]:= (* Plot price bracket * d^(t/m) over time t *)

In[121]:= (* Look at funding rates needed over different confidence levels, to compare *)

In[402]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.01, 8]

Out[402]= 0.221594

In[403]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 8]

Out[403]= 0.106027

In[404]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.10, 8]

Out[404]= 0.0756562

In[405]:= (* Last 95 → 99% is what ramps it up significantly *)

In[406]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.01, 1]

Out[406]= 0.42805

In[407]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 1]

Out[407]= 0.218183

In[408]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.10, 1]

Out[408]= 0.158401

In[409]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 24]

Out[409]= 0.0703214

In[410]:= (* Similar with rescale. See if applied every 10 min
           for 8 hour value but after 1 d, 7d, 14d what drawdown is *)

In[411]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 8]

Out[411]= 0.106027

In[412]:= 1 - (1 - rescaledKminWethUsdc120dFiltered1HourCandle[1/6, 0.05, 8]) ^ (6 * 24)

Out[412]= 0.107929

In[413]:= rescaledKminWethUsdc120dFiltered1HourCandle[1/6, 0.05, 8]

Out[413]= 0.000792806
```

In[414]:=
```
drawDownMinWethUsdc120dFiltered1HourCandle[24 * 7, 0.05, 8]
```
Out[414]=
```
0.543678
```

In[415]:=
```
1 - (1 - rescaledKminWethUsdc120dFiltered1HourCandle[1 / 6, 0.05, 8]) ^ (6 * 24 * 7)
```
Out[415]=
```
0.550431
```

In[135]:=
```
(* Beautiful. Anchoring at factor value at m sets funding rate to
 be min. Does compound surpass price bracket at the 8 hour mark? *)

(* Look at rescaled draw down first to make sure consistent *)
```

In[416]:=
```
rescaledDrawDownMinWethUsdc120dFiltered1HourCandle[t_, alpha_, m_, anchor_ ] := 1 -
   (1 - rescaledKminWethUsdc120dFiltered1HourCandle[m, alpha, anchor]) ^ (Floor[t / m])
```

In[417]:=
```
rescaledDrawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 1, 8]
```
Out[417]=
```
0.107727
```

In[418]:=
```
rescaledDrawDownMinWethUsdc120dFiltered1HourCandle[24 * 7, 0.05, 1, 8]
```
Out[418]=
```
0.549717
```

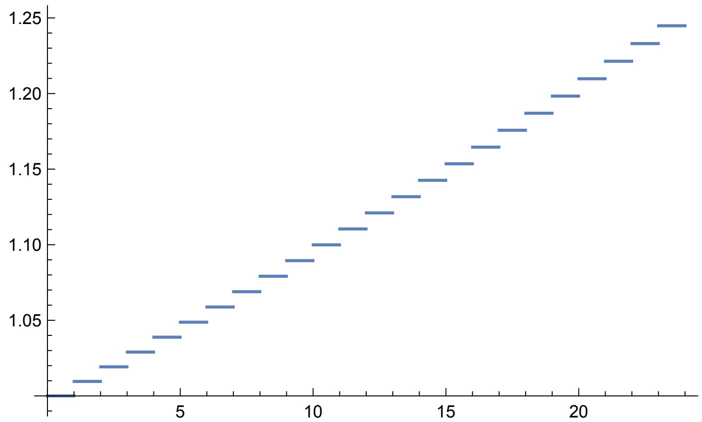
```
(* Great. 1h funding payments on 8h anchor look great *)
```

In[136]:=
```
(* Look at (1 - 2k)^m over time .... Compound
 factor for different compound intervals. Assume funding
 payments applied every 1 h for simplicity with fits. *)
```

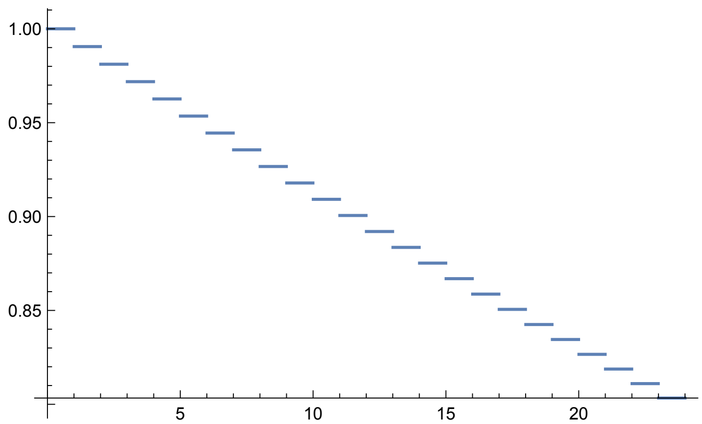
In[419]:=
```
compoundFactorMinWethUsdc120dFiltered1HourCandle[t_, alpha_, m_, anchor_] :=
  (1 - 2 * rescaledKminWethUsdc120dFiltered1HourCandle[m, alpha, anchor]) ^
   (Floor[t / m])
```

In[138]:=
```
(* 8h payment at 95% confidence of VaR led to
   drawdown of 11% worst case due to funding. Use alpha=0.05,
m=8. Rescaled funding st paid every 1 h *)
```

In[420]:= Plot[
{1 / compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t, 0, 24}]

Out[420]=


In[421]:= Plot[
{compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t, 0, 24}]

Out[421]=


In[422]:= drawDownMinWethUsdc120dFiltered1HourCandle[24, 0.05, 8]

Out[422]= 0.106027

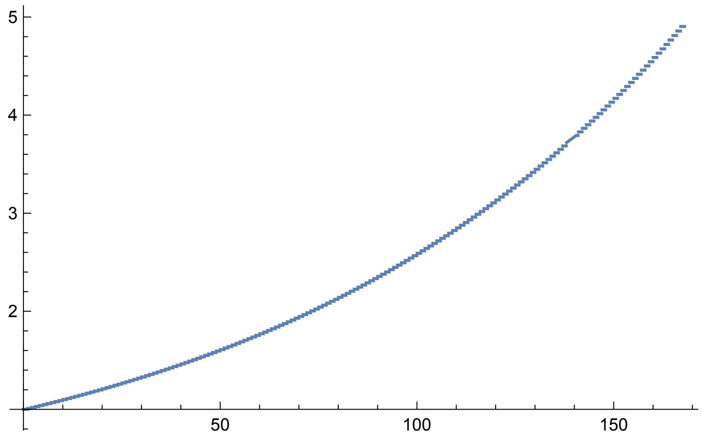
In[423]:= 1 - compoundFactorMinWethUsdc120dFiltered1HourCandle[24, 0.05, 1, 8]

Out[423]= 0.204281

In[424]:= (* Perfect. Draw down to OI imbalance
should be about 2 times draw down to OI on a side *)

In[425]:=
```
Plot[1 / compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
  {t, 0, 24 * 7}]
```

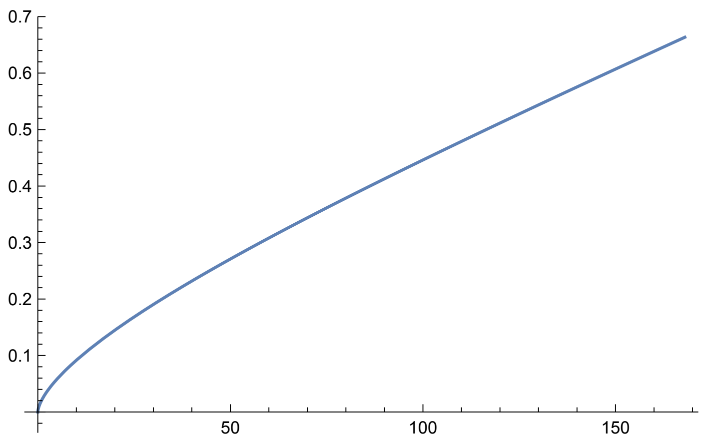
Out[425]=



In[426]:=
```
(* Plot price bracket ...  *)
```

In[427]:=
```
Plot[factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1, {t, 0, 24 * 7}]
```

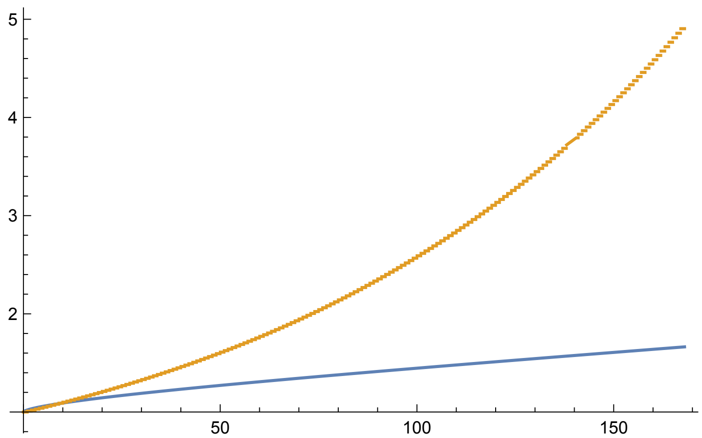
Out[427]=



In[428]:=
```
(* Plot 1/compoundFactor vs price bracket *)
```

```
In[429]:= Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05] ,
   1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t,
   0, 24*7}]
```

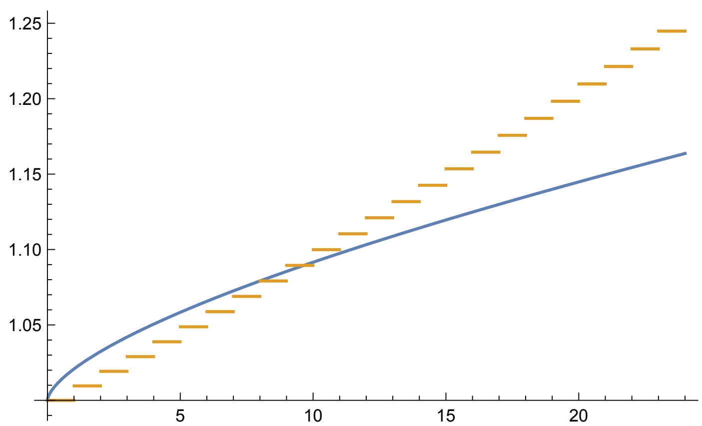
Out[429]=




```
In[430]:= (* Look at the earlier times ... *)
```

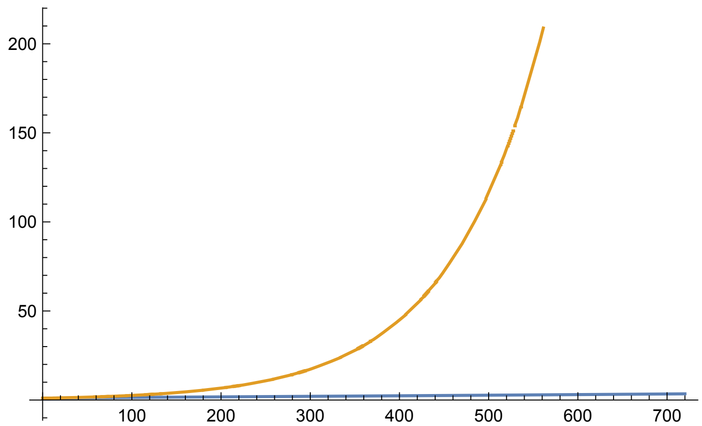
```
In[431]:= Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05] ,
   1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t, 0, 24}]
```

Out[431]=



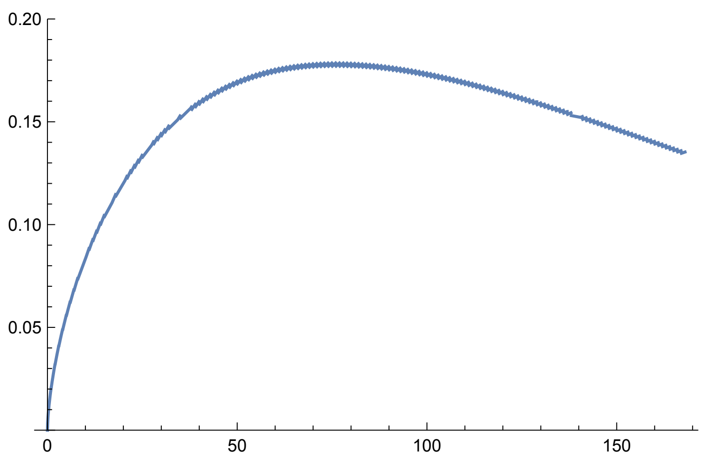

```
In[432]:= (* Look at later times ... *)
```

In[433]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05] ,
  1 / compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t,
  0, 24 * 30}]
```

Out[433]=


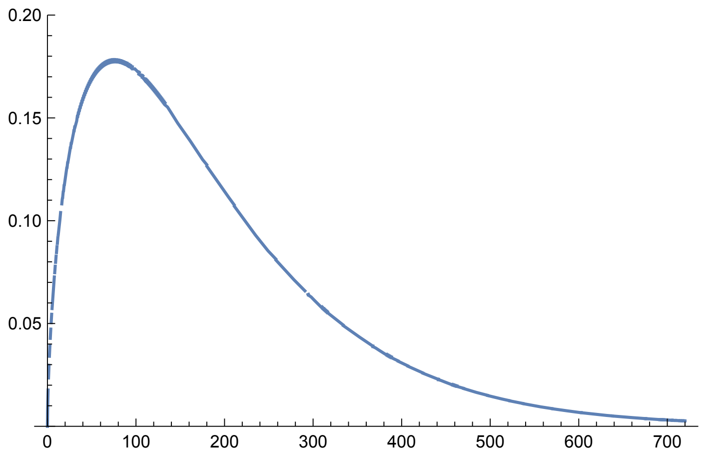
In[434]:=
```
(* and the product of the two to make sure decreasing over time ... *)
```

In[435]:=
```
Plot[(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
  compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
 {t, 0, 24 * 7}, PlotRange → {0, 0.2}]
```

Out[435]=


In[436]:=
```
Plot[(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
  compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
 {t, 0, 24 * 30}, PlotRange → {0, 0.2}]
```

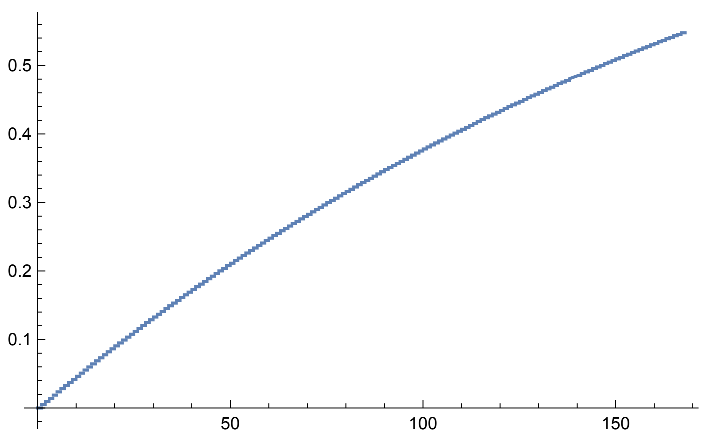
Out[436]=



In[437]:=
```
(* Compare with draw down over time *)
```

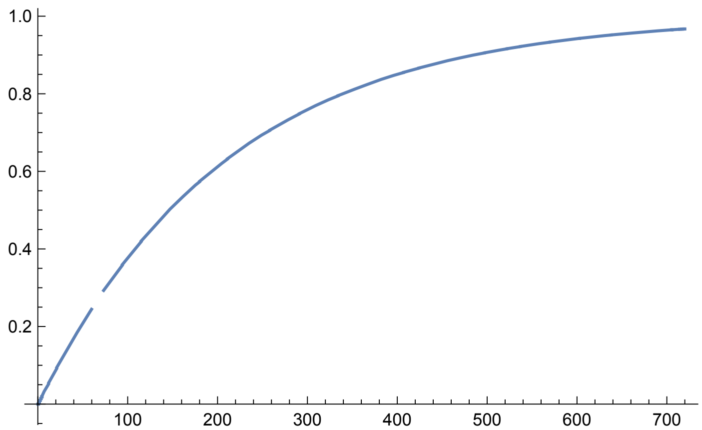
In[438]:=
```
Plot[{rescaledDrawDownMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]},
 {t, 0, 24 * 7}]
```

Out[438]=



In[439]:=
```
Plot[{rescaledDrawDownMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]},
 {t, 0, 24 * 30}]
```

Out[439]=



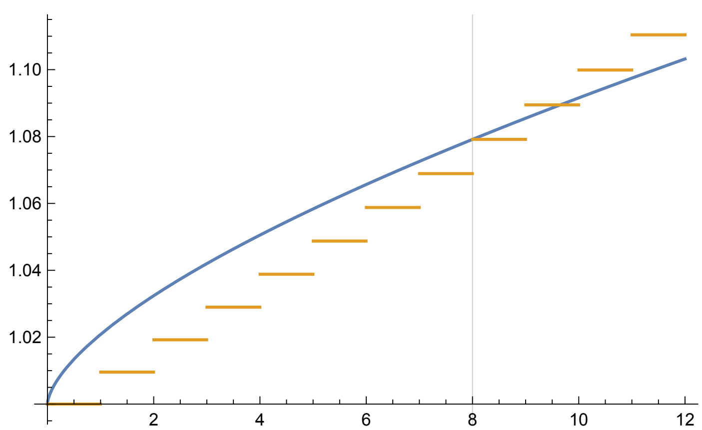
In[440]:=

In[441]:=
```
(* Anchor used in k is everything. What is the
  interpretation? Time when d^m is exactly equal to first term in
  price bracket[m]. Then d^m surpasses greatly in size as m → infty *)
```

In[442]:=
```
(* So look at 1/compoundFactor vs price factor BEFORE hit the anchor to
  make sure 1/compoundFactor is less but we cross at anchor time ... *)
```

In[443]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05],
  1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]},
 {t, 0, 12}, GridLines → {{8}}]
```

Out[443]=



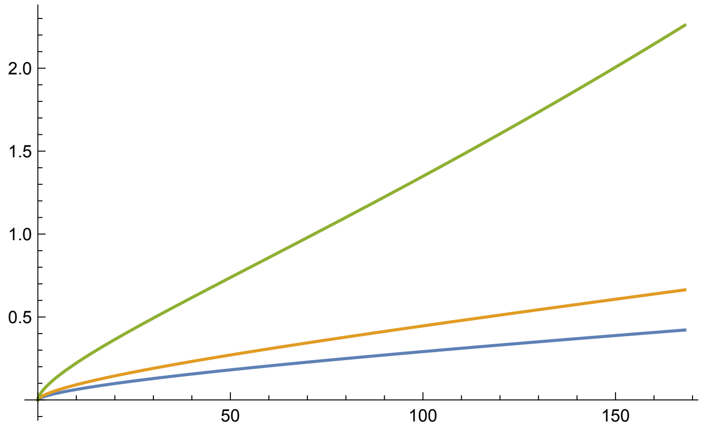

```
(* Perfect. Interpretation is correct. *)
```

```
(* Then product of price bracket and compound factor is playing catch up for
 the first 8 hours in order to get to point where decreasing over time? *)
```

```
(* Look at different price bracket
 VaR growth over time without compound factor *)
```

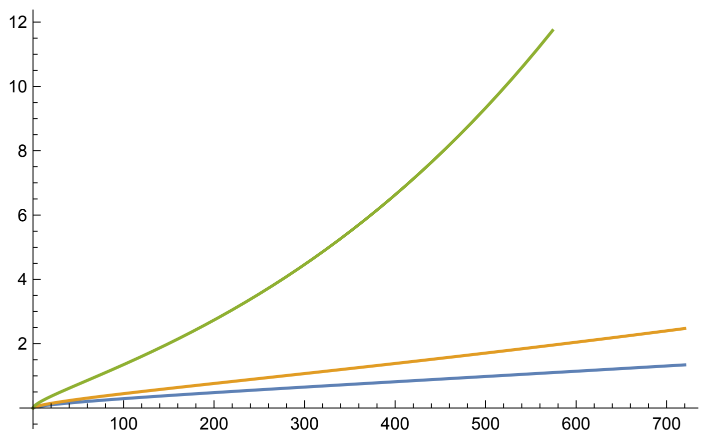
In[446]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.10] - 1),
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1),
  (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1)}, {t, 0, 24 * 7}]
```

Out[446]=



In[445]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.10] - 1),
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1),
  (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1)}, {t, 0, 24 * 30}]
```

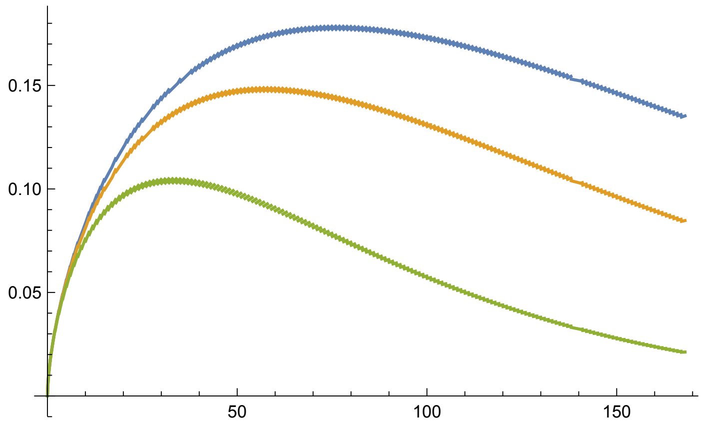
Out[445]=




```
(* Try product of price bracket and compound
  factor for different anchor times in k calibration ... *)
```

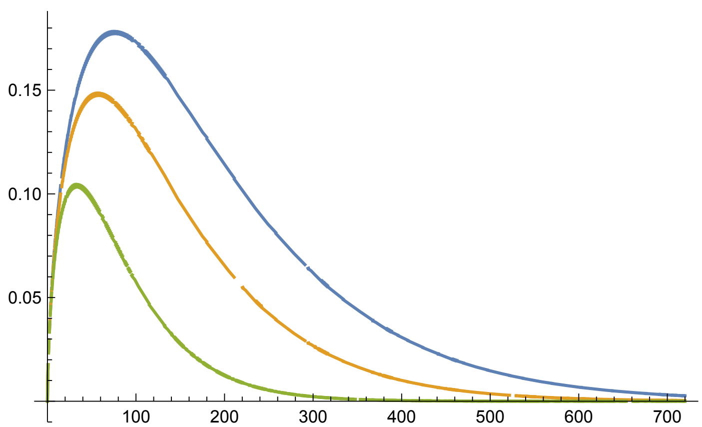
In[447]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[
     t, 0.05, 1, 1]}, {t, 0, 24 * 7}]
```

Out[447]=



In[448]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[
     t, 0.05, 1, 1]}, {t, 0, 24 * 30}]
```

Out[448]=



In[449]:=
```
(* So most VaR to passive holders happens over
 shorter term when imbalance first created. Longer users hold
 positions and imbalance gets drawn down through funding,
the more VaR gets curbed and eventually brought to zero *)
```

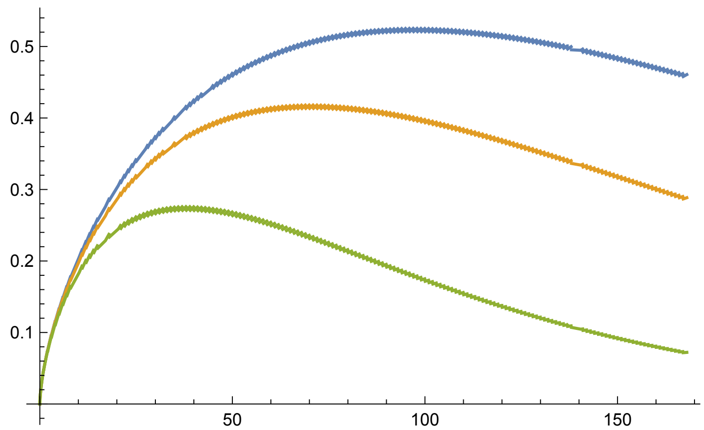
In[450]:=

In[451]:=

In[452]:=
```
(* What if we underestimate funding rate needed? As in should have used alpha=
 0.01 instead of alpha=0.05? what does this do to VaR over time? *)
```

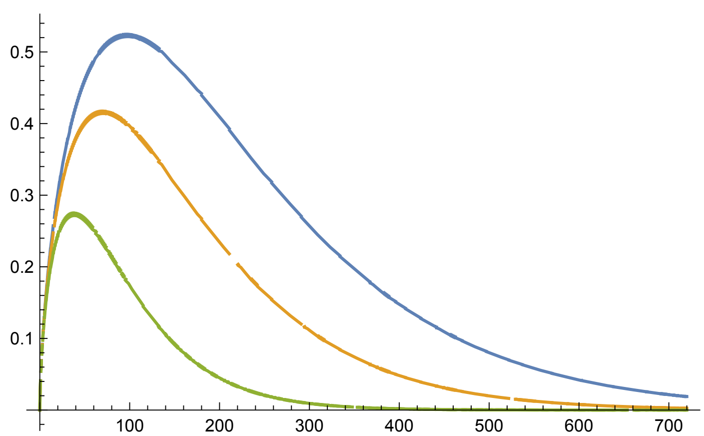
In[453]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[
     t, 0.05, 1, 1]}, {t, 0, 24 * 7}]
```

Out[453]=



In[454]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
   (factorWethUsdc120dFiltered1HourCandle[t, 0.01] - 1) *
    compoundFactorMinWethUsdc120dFiltered1HourCandle[
     t, 0.05, 1, 1]}, {t, 0, 24 * 30}]
```

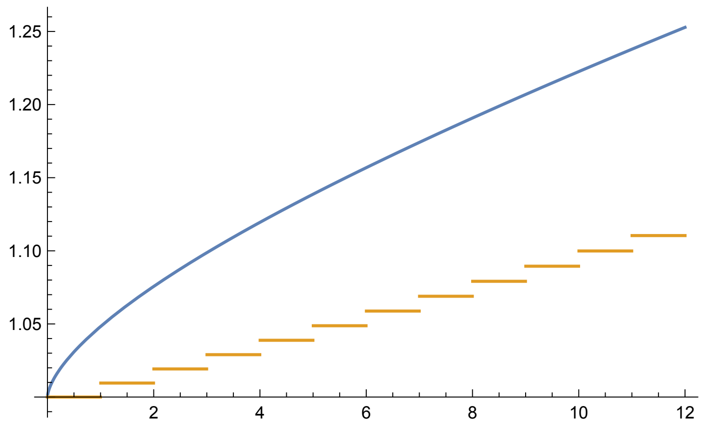
Out[454]=



In[455]:=
```
(* Look at 1/compoundFactor (larger alpha than required) vs price
  factor BEFORE hit the anchor to see what new anchor time is ... *)
```

In[456]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.01],
  1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t, 0, 12}]
```

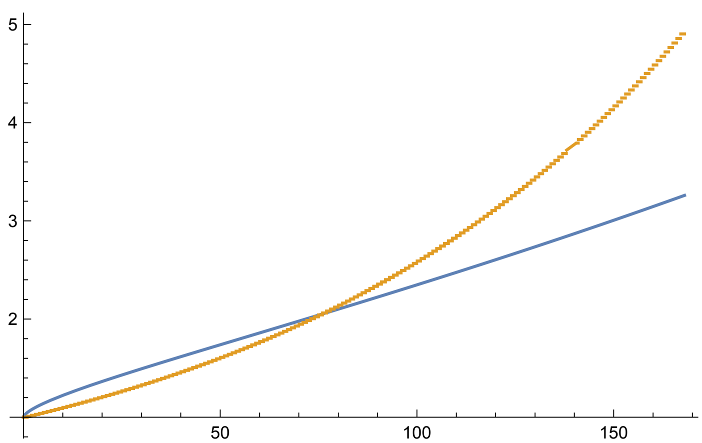
Out[456]=



In[457]:=
```
(* Interesting, anchor time comes much later. Not great,
but still exists ... However, VaR still seems to decrease tho, just more drawn
 out AND significantly higher max value/peak (about 2x for alpha=0.05). *)
```

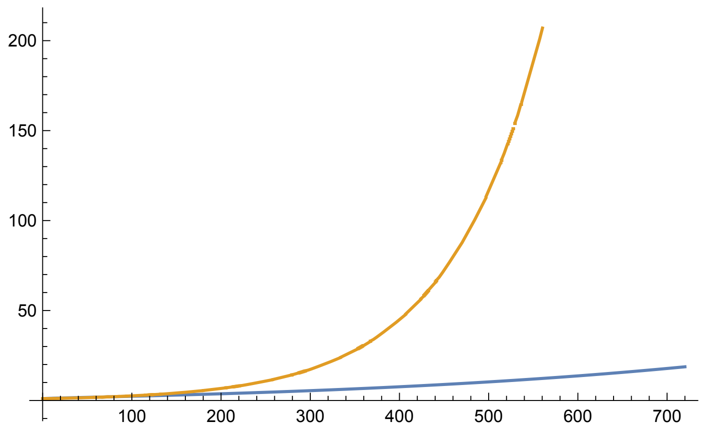
In[458]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.01],
  1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t,
  0, 24 * 7}]
```

Out[458]=



In[459]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.01],
  1/compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8]}, {t,
  0, 24*30}]
```

Out[459]=



In[460]:=
```
(* Instead of anticipated anchor of 8h, get anchor around 72h (3d). *)
```

In[461]:=

In[462]:=
```
(* What about set funding factor to not exp^{(m*t)^{1/a}}
 but exp^{m*(t)^(1/a)}? What's difference in interest rate? *)
```

In[463]:=
```
factorLargerWethUsdc120dFiltered1HourCandle[t_, alpha_] := Exp[
  muWethUsdc120dFiltered1HourCandle * t + sigWethUsdc120dFiltered1HourCandle * t *
    (1/aWethUsdc120dFiltered1HourCandle)^(1/aWethUsdc120dFiltered1HourCandle) *
    InverseCDF[edistWethUsdc120dFiltered1HourCandleNormalized, 1 - alpha]]
```

In[464]:=
```
factorLargerWethUsdc120dFiltered1HourCandle[8, 0.05]
```

Out[464]= 1.17932

In[465]:=
```
factorWethUsdc120dFiltered1HourCandle[8, 0.05]
```

Out[465]= 1.07915

In[466]:=
```
factorLargerWethUsdc120dFiltered1HourCandle[24, 0.05]
```

Out[466]= 1.6402

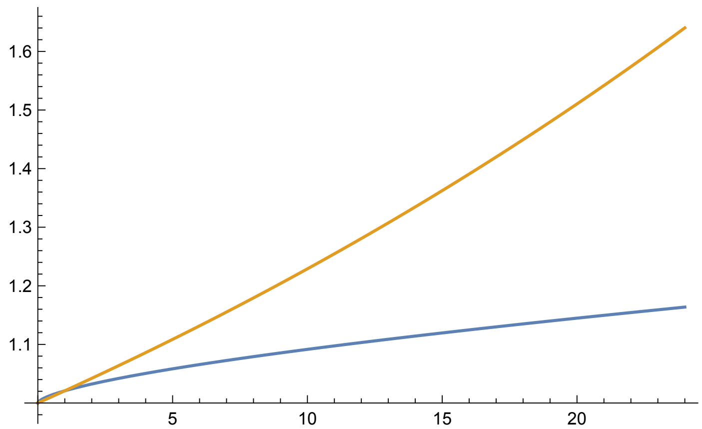
In[467]:=
```
factorWethUsdc120dFiltered1HourCandle[24, 0.05]
```

Out[467]= 1.16366

In[468]:=
```
(* Yea this gets out of control fast ... *)
```

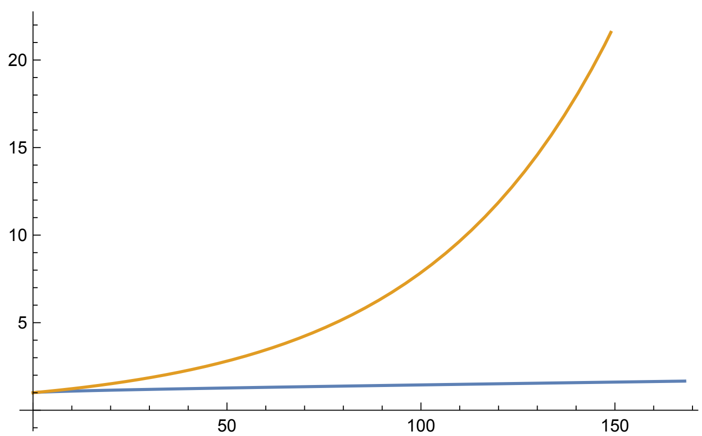
In[469]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05],
  factorLargerWethUsdc120dFiltered1HourCandle[t, 0.05]}, {t, 0, 24}]
```

Out[469]=



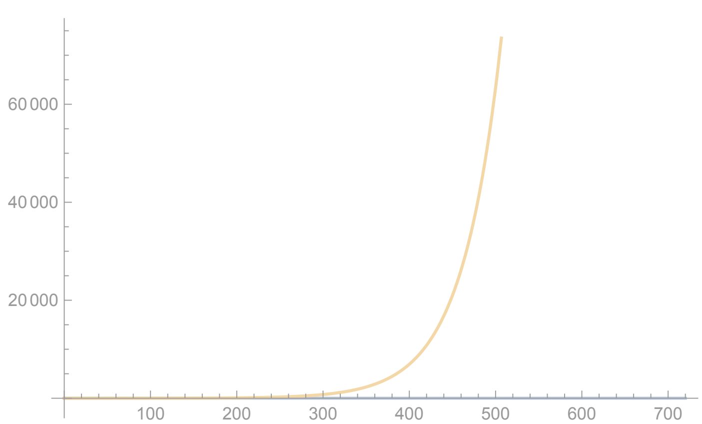
In[470]:=
```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05],
  factorLargerWethUsdc120dFiltered1HourCandle[t, 0.05]}, {t, 0, 24 * 7}]
```

Out[470]=




```
Plot[{factorWethUsdc120dFiltered1HourCandle[t, 0.05],
  factorLargerWethUsdc120dFiltered1HourCandle[t, 0.05]}, {t, 0, 24 * 30}]
```

Out[184]=

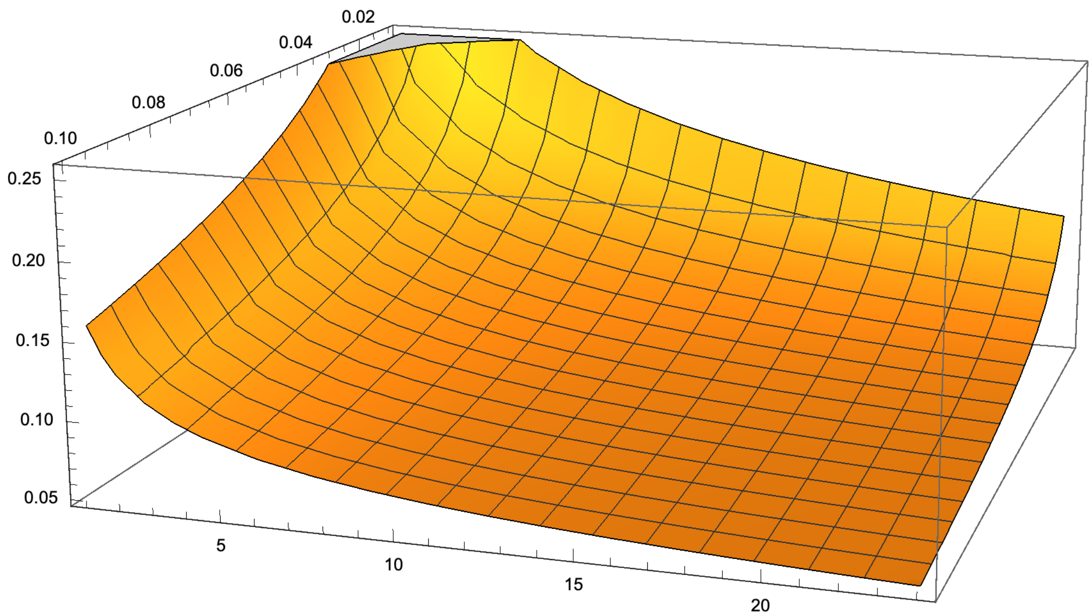



```
(* We can't use the larger factor ... Actually
 a good preview for fits that are closer to Cauchy :/ *)
```

```
(* Plot continuous draw down as 3d with alpha varying as well *)
```

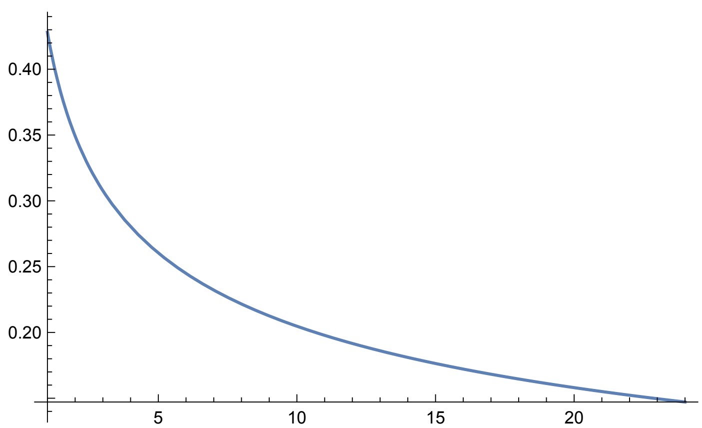
In[541]:=
```
Plot3D[drawDownMinWethUsdc120dFiltered1HourCandleContinuous[24, a, m],
 {a, 0.01, 0.1}, {m, 1, 24}]
```

Out[541]=



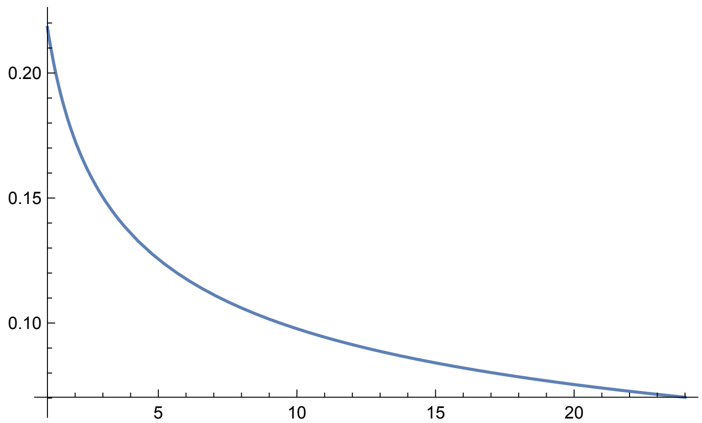
In[542]:=
```
Plot[
 drawDownMinWethUsdc120dFiltered1HourCandleContinuous[24, 0.01, m], {m, 1, 24}]
```

Out[542]=



In[543]:=
```
Plot[
 drawDownMinWethUsdc120dFiltered1HourCandleContinuous[24, 0.05, m], {m, 1, 24}]
```

Out[543]=




```
(* About double if look at 99% confidence level vs 95% *)
```

In[538]:=
```
drawDownMinWethUsdc120dFiltered1HourCandleContinuous[24, 0.01, 5]
```

Out[538]= 0.260379

```
(* What about underestimating the stable distribution fit params? *)
(* Import data for UNI/WETH and convert to 1h candles *)
```

In[471]:=
```
tblUniWeth120d = Import["120/data-1626472963_uni-weth-twap.csv"]
```

Out[471]=

{{, timestamp, twap}, {0, 1.61878×10^9, 1.4244×10^{16}},
{1, 1.61878×10^9, 1.42041×10^{16}}, {2, 1.61878×10^9, 1.41202×10^{16}},
{3, 1.61878×10^9, 1.41232×10^{16}}, {4, 1.61885×10^9, 1.11905×10^{16}}, … 6779 …,
{6785, 1.62646×10^9, 8.75082×10^{15}}, {6786, 1.62647×10^9, 8.75708×10^{15}},
{6787, 1.62647×10^9, 8.75789×10^{15}}, {6788, 1.62647×10^9, 8.76048×10^{15}},
{6789, 1.62647×10^9, 8.75972×10^{15}}, {6790, 1.62647×10^9, 8.75626×10^{15}}}

large output | show less | show more | show all | set size limit...

In[472]:=
```
Length[tblUniWeth120d]
```

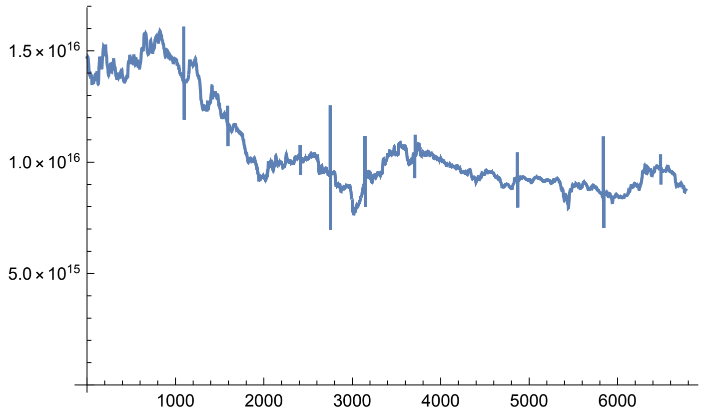
Out[472]= 6791

In[473]:=
```
twapsUniWeth120dFiltered =
 Table[tblUniWeth120d[[i]][[3]], {i, 20, Length[tblUniWeth120d]}]
```

In[474]:=
```
Length[twapsUniWeth120dFiltered]
```

Out[474]= 6772

In[475]:= ListLinePlot[twapsUniWeth120dFiltered , PlotRange → All]

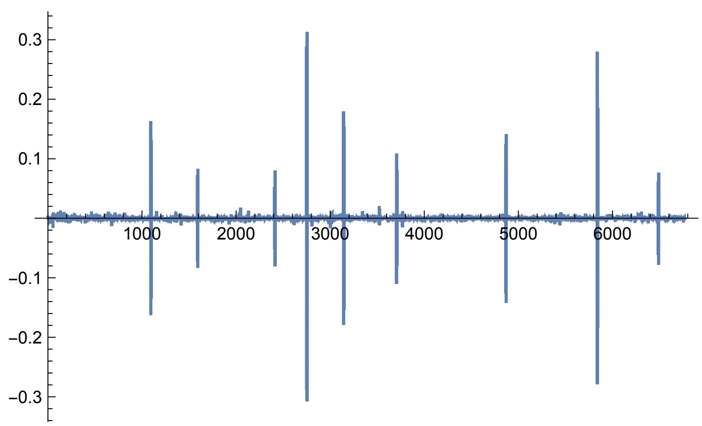
Out[475]=


In[476]:= rsUniWeth120dFiltered = Differences[Log[twapsUniWeth120dFiltered]]

Out[476]=
{0.00192969, -0.000607667, -0.00173262, -0.000132253, -0.00086947, -0.000234575, 0.00029493, 0.000681779, -0.000561976, -0.0000244193, -0.000716273, ⋯ 6749 ⋯, -0.000151326, -0.000615037, 0.0018207, 0.00197972, 0.000885439, 0.000325081, 0.000715271, 0.0000924127, 0.000294919, -0.0000864918, -0.000395188}

large output | show less | show more | show all | set size limit...

In[477]:= ListLinePlot[rsUniWeth120dFiltered, PlotRange → All]

Out[477]=


In[478]:= edistUniWeth120dFiltered = EstimatedDistribution[rsUniWeth120dFiltered,
StableDistribution[1, aUW120d, bUW120d, locUW120d, scaleUW120d]]

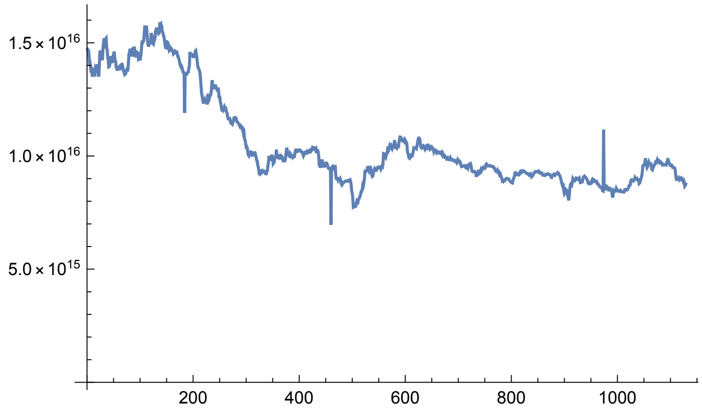
Out[478]= StableDistribution[1, 1.28413, 0.00661751, -0.0000924243, 0.000844373]

In[479]:= twapsUniWeth120dFiltered1HourCandle =
Table[twapsUniWeth120dFiltered[[i]], {i, 1, Length[twapsUniWeth120dFiltered], 6}]

In[480]:= Length[twapsUniWeth120dFiltered1HourCandle]

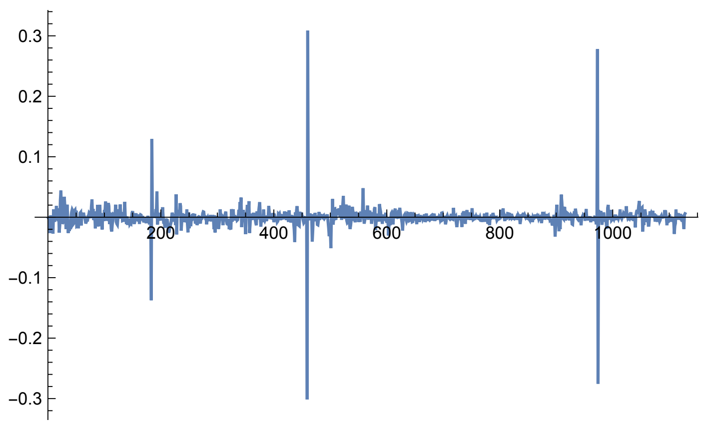
Out[480]= 1129

In[481]:= ListLinePlot[twapsUniWeth120dFiltered1HourCandle, PlotRange → All]

Out[481]=


In[482]:= rsUniWeth120dFiltered1HourCandle =
 Differences[Log[twapsUniWeth120dFiltered1HourCandle]]

In[483]:= ListLinePlot[rsUniWeth120dFiltered1HourCandle, PlotRange → All]

Out[483]=


In[484]:= edistUniWeth120dFiltered1HourCandle =
 EstimatedDistribution[rsUniWeth120dFiltered1HourCandle,
  StableDistribution[1, aUW120dC1h, bUW120dC1h, locUW120dC1h, scaleUW120dC1h]]

Out[484]= StableDistribution[1, 1.44302, 0.0429383, -0.000419763, 0.00474788]

In[485]:= (* Compare w WETH/USDC *)

In[486]:= edistWethUsdc120dFiltered1HourCandle

Out[486]= StableDistribution[1, 1.5843, -0.0493885, -0.000101694, 0.0072772]

In[487]:= (* Interesting ... larger scale on WETH/USDC but
 smaller alpha. Translates to in terms of InverseCDF ? *)

In[488]:= InverseCDF[edistWethUsdc120dFiltered1HourCandle, 0.99]

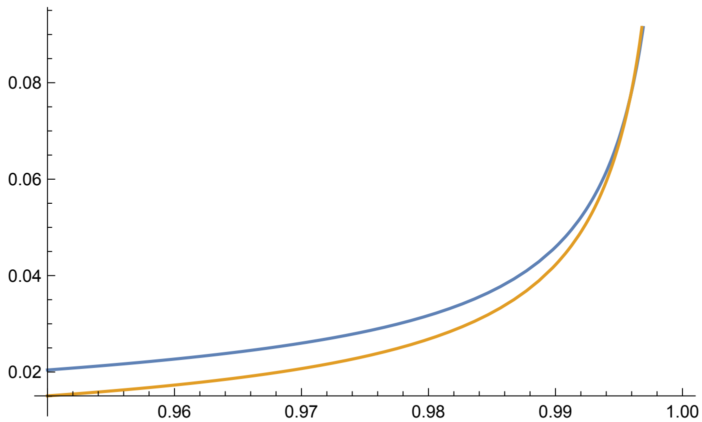
Out[488]= 0.0459063

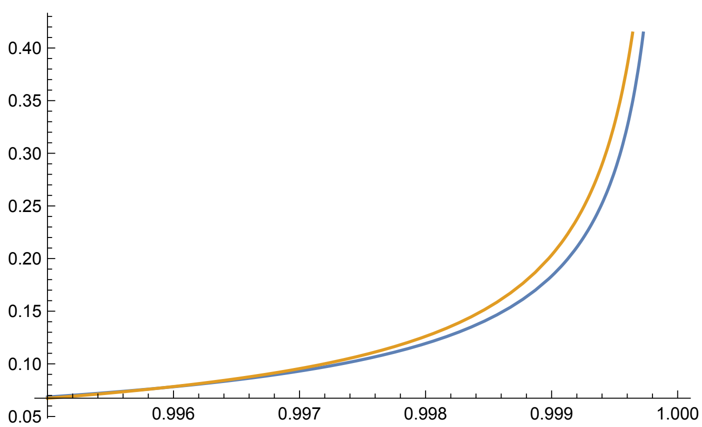
In[489]:=
```
InverseCDF[edistWethUsdc120dFiltered1HourCandle, 0.999]
```
Out[489]= 0.183327

In[490]:=
```
InverseCDF[edistUniWeth120dFiltered1HourCandle, 0.99]
```
Out[490]= 0.0422306

In[491]:=
```
InverseCDF[edistUniWeth120dFiltered1HourCandle, 0.999]
```
Out[491]= 0.203319

In[492]:=
```
(* Slight difference in alpha starts to
 become apparent at super extreme ends of cdf *)
```

In[493]:=
```
Plot[{InverseCDF[edistWethUsdc120dFiltered1HourCandle, q],
  InverseCDF[edistUniWeth120dFiltered1HourCandle, q]}, {q, 0.95, 1}]
```
Out[493]=


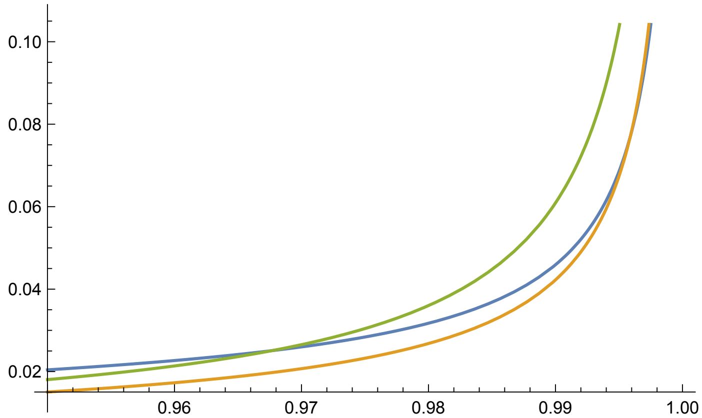
In[494]:=
```
Plot[{InverseCDF[edistWethUsdc120dFiltered1HourCandle, q],
  InverseCDF[edistUniWeth120dFiltered1HourCandle, q]}, {q, 0.995, 1}]
```
Out[494]=


In[495]:=
```
(* What about a=1.3? *)
```

In[496]:=
```
edistUniWeth120dFiltered1HourCandle
```
Out[496]= StableDistribution[1, 1.44302, 0.0429383, -0.000419763, 0.00474788]

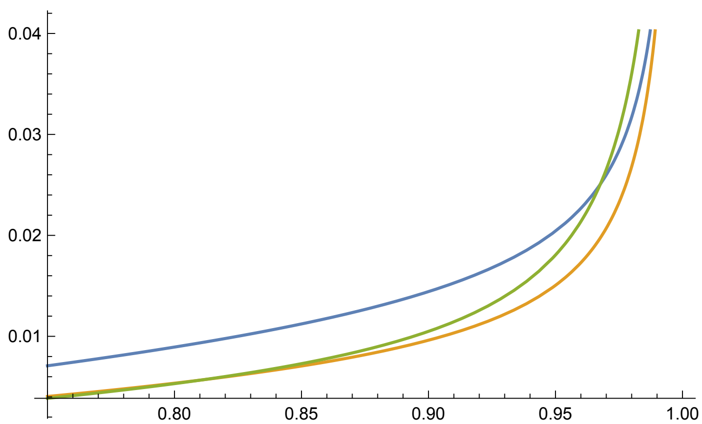
In[497]:=
```
Plot[{InverseCDF[edistWethUsdc120dFiltered1HourCandle, q],
  InverseCDF[edistUniWeth120dFiltered1HourCandle, q],
  InverseCDF[StableDistribution[1, 1.3, 0.05160448032592783`,
    -0.0005180625308600968`, 0.004823033038389856`], q]}, {q, 0.95, 1}]
```

Out[497]=



In[498]:=
```
Plot[{InverseCDF[edistWethUsdc120dFiltered1HourCandle, q],
  InverseCDF[edistUniWeth120dFiltered1HourCandle, q],
  InverseCDF[StableDistribution[1, 1.3, 0.05160448032592783`,
    -0.0005180625308600968`, 0.004823033038389856`], q]}, {q, 0.75, 1}]
```

Out[498]=




```
(* Yea so miscalc on a greatly changes things up, as expected. Lower a,
means higher inverse CDF earlier in upper range we care about *)

(* Set up factor calcs *)

edistUniWeth120dFiltered1HourCandle
```

Out[504]= `StableDistribution[1, 1.43975, 0.0516045, -0.000518063, 0.00482303]`

In[503]:= `edistUniWeth120dFiltered1HourCandle`

Out[503]= `StableDistribution[1, 1.44302, 0.0429383, -0.000419763, 0.00474788]`

```
(* Looking goood ... *)
```

```
In[505]:= muUniWeth120dFiltered1HourCandle =
  mu[1.4430192097069168`, -0.0004197632840157609`]

Out[505]= -0.000419763

In[506]:= sigUniWeth120dFiltered1HourCandle =
  sig[1.4430192097069168`, 0.004747879945834797`]

Out[506]= 0.00612173

In[507]:= aUniWeth120dFiltered1HourCandle = 1.4430192097069168`

Out[507]= 1.44302

In[508]:= edistUniWeth120dFiltered1HourCandleNormalized =
  StableDistribution[1, aUniWeth120dFiltered1HourCandle, 0, 0, 1]

Out[508]= StableDistribution[1, 1.44302, 0, 0, 1]

In[509]:= factorUniWeth120dFiltered1HourCandle[t_, alpha_] :=
  Exp[muUniWeth120dFiltered1HourCandle * t + sigUniWeth120dFiltered1HourCandle *
     (t / aUniWeth120dFiltered1HourCandle) ^ (1 / aUniWeth120dFiltered1HourCandle) *
     InverseCDF[edistUniWeth120dFiltered1HourCandleNormalized, 1 - alpha]]

In[510]:= factorUniWeth120dFiltered1HourCandle[8, 0.05]

Out[510]= 1.06319

In[511]:= factorWethUsdc120dFiltered1HourCandle[8, 0.05]

Out[511]= 1.07915

In[512]:= dminUniWeth120dFiltered1HourCandle[t_, alpha_] :=
  factorUniWeth120dFiltered1HourCandle[t, alpha]

In[513]:= rescaledKminUniWeth120dFiltered1HourCandle[t_, alpha_, m_] :=
  (1 - 1 / (dminUniWeth120dFiltered1HourCandle[m, alpha]) ^ (t / m)) / 2

In[514]:= rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t_, alpha_, m_, anchor_ ] := 1 -
   (1 - rescaledKminUniWeth120dFiltered1HourCandle[m, alpha, anchor]) ^ (Floor[t / m])

In[515]:= rescaledDrawDownMinUniWeth120dFiltered1HourCandle[24, 0.05, 1, 8]

Out[515]= 0.0876495

In[516]:= (* Make sure anchor is consistent again ... *)

In[517]:= compoundFactorMinUniWeth120dFiltered1HourCandle[t_, alpha_, m_, anchor_] :=
  (1 - 2 * rescaledKminUniWeth120dFiltered1HourCandle[m, alpha, anchor]) ^
   (Floor[t / m])

In[518]:= 1 - compoundFactorMinUniWeth120dFiltered1HourCandle[24, 0.05, 1, 8]

Out[518]= 0.167909
```

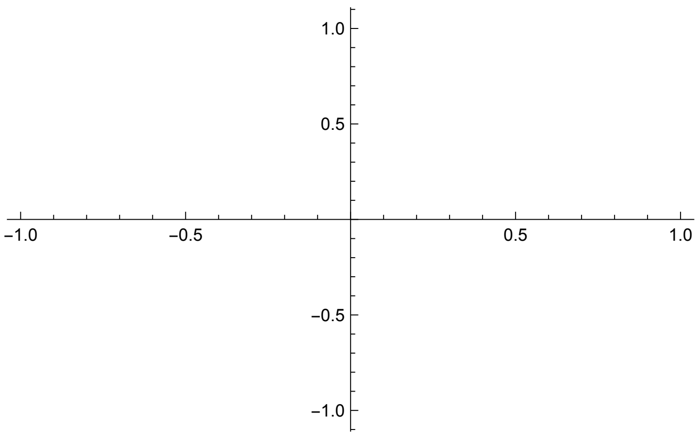
In[519]:=
```
rescaledDrawDownMinUniWeth120dFiltered1HourCandleContinuous[
  t_, alpha_, m_, anchor_] :=
 1 - (1 - rescaledKminUniWeth120dFiltered1HourCandle[m, alpha, anchor]) ^ (t / m)
```

In[520]:=
```
(* UNI/WETH at 8% versus 11% for WETH/USDC ...
  so around 10% seems to be fine for a=~1.4 *)
```

In[521]:=
```
(* Below is min funding rate required extended to per day rate,
if apply funding payment every m hours  *)
```

In[522]:=
```
Plot[rescaledDrawDownMinUniWeth120dFiltered1HourCandleContinuous[24, 0.05,  m],
 {m, 1, 24}, PlotRange → All]
```

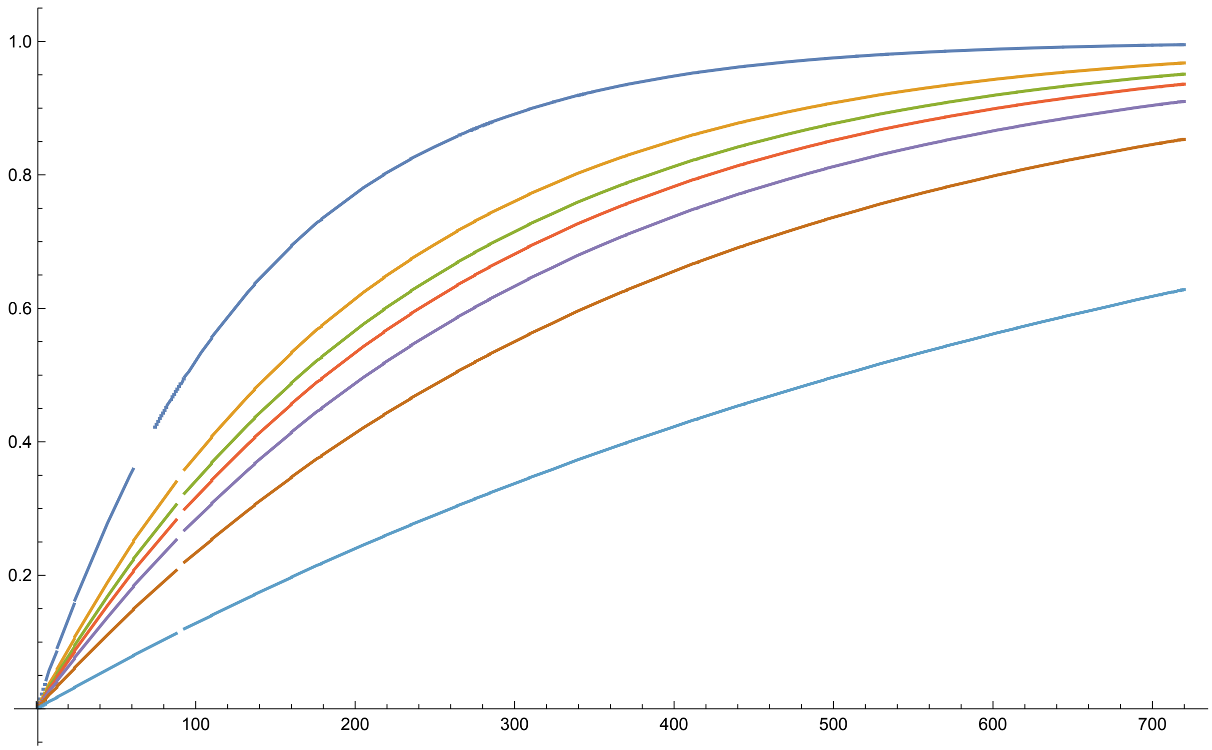
Out[522]=



In[523]:=
```
(* Plot rescaled draw down over time *)
```

In[534]:=
```
Plot[{rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 1],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 4],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 6],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 8],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 12],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 24],
  rescaledDrawDownMinUniWeth120dFiltered1HourCandle[
   t, 0.05, 1, 168]}, {t, 1, 24 * 30}]
```

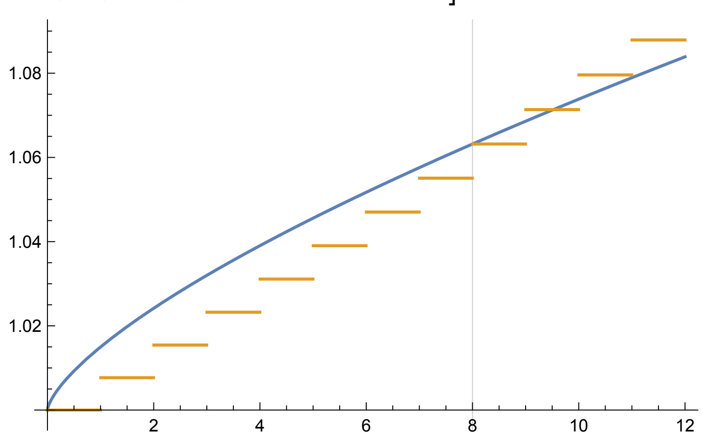
Out[534]=



In[525]:=
```
(* And anchor ... *)
```

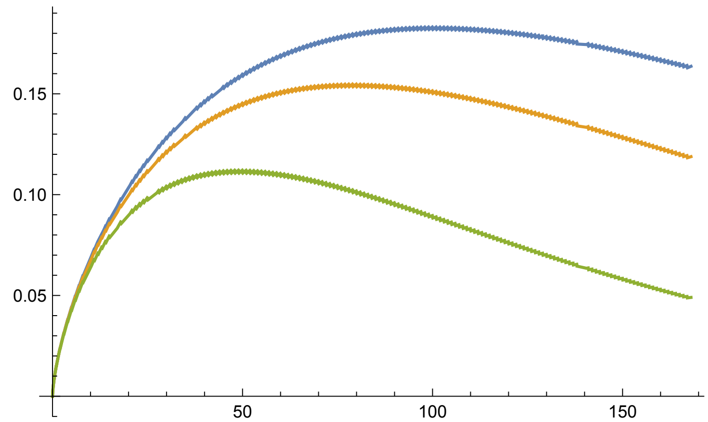
In[526]:=
```
Plot[{factorUniWeth120dFiltered1HourCandle[t, 0.05],
  1 / compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 8]},
 {t, 0, 12}, GridLines → {{8}}]
```

Out[526]=



In[527]:=
```
Plot[{(factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 8],
  (factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 4],
  (factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 1]}, {t, 0, 24 * 7}]
```

Out[527]=

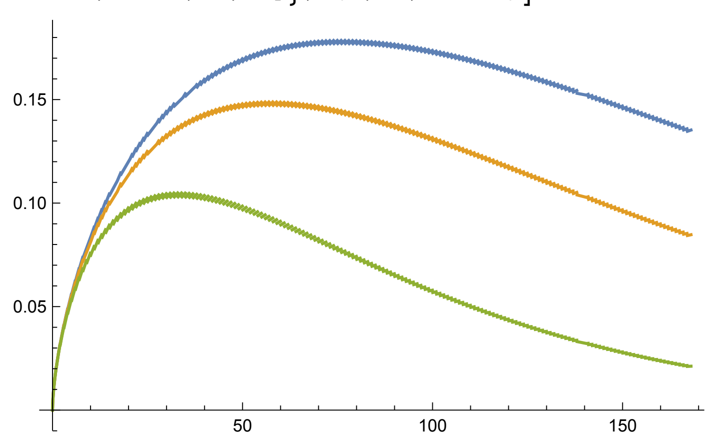


In[528]:=
```
(* This is kind of bad. Lower funding rate is translating to
 slower draw down in VaR over time, even with same framework. *)
```

In[529]:=
```
(* TODO:
 Should have a framework that has similar curves regardless of inverse CDF *)
```

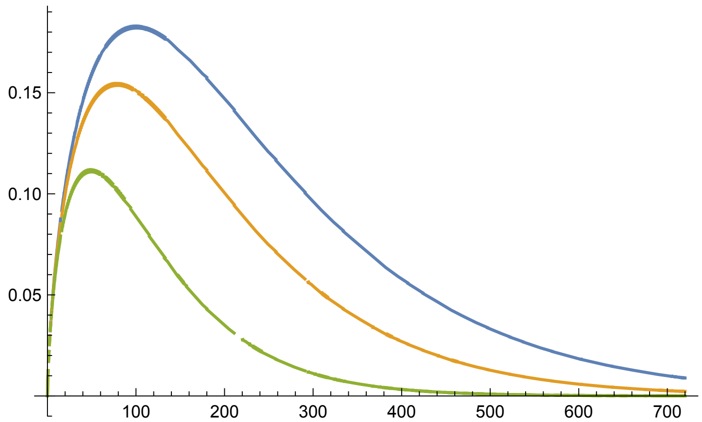
In[530]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[
    t, 0.05, 1, 1]}, {t, 0, 24 * 7}]
```

Out[530]=



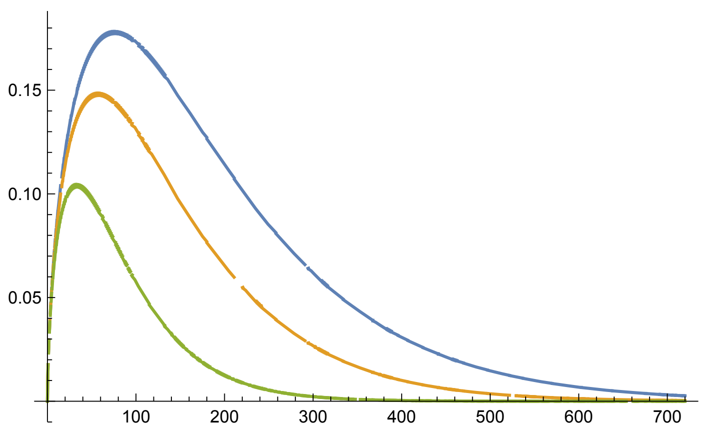
In[531]:=
```
Plot[{(factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 8],
  (factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[t, 0.05, 1, 4],
  (factorUniWeth120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinUniWeth120dFiltered1HourCandle[
    t, 0.05, 1, 1]}, {t, 0, 24 * 30}]
```

Out[531]=



In[532]:=
```
Plot[{(factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 8],
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[t, 0.05, 1, 4],
  (factorWethUsdc120dFiltered1HourCandle[t, 0.05] - 1) *
   compoundFactorMinWethUsdc120dFiltered1HourCandle[
    t, 0.05, 1, 1]}, {t, 0, 24 * 30}]
```

Out[532]=




```
(* Let's amp things up with something like ALCX/WETH which has 10m a < 1 ...
 Likely closer to what OVL/ETH and OVL/DAI will be. *)
```