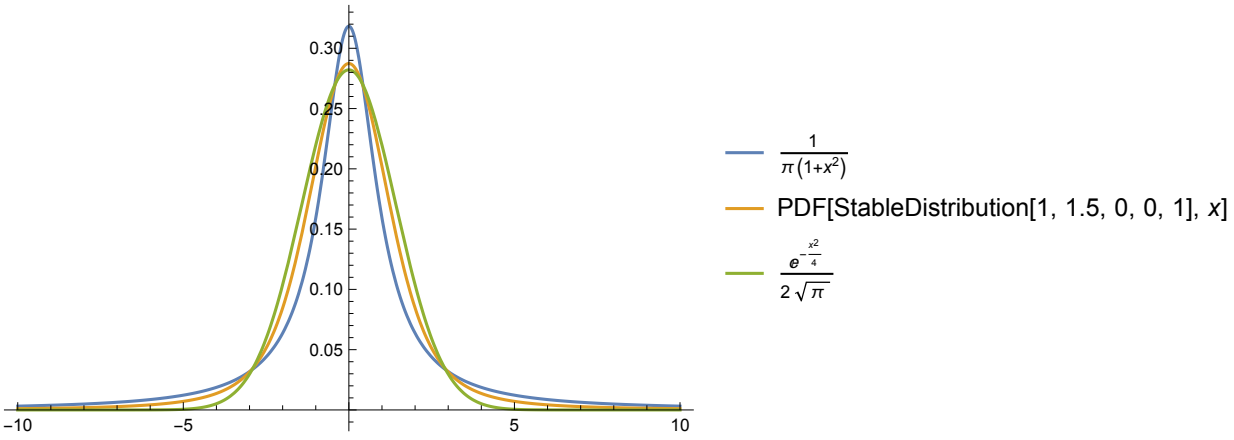
In[]:=
```
(* Look at stable distribution and infer what
 best payoff cap is based on underlying distr params *)
```

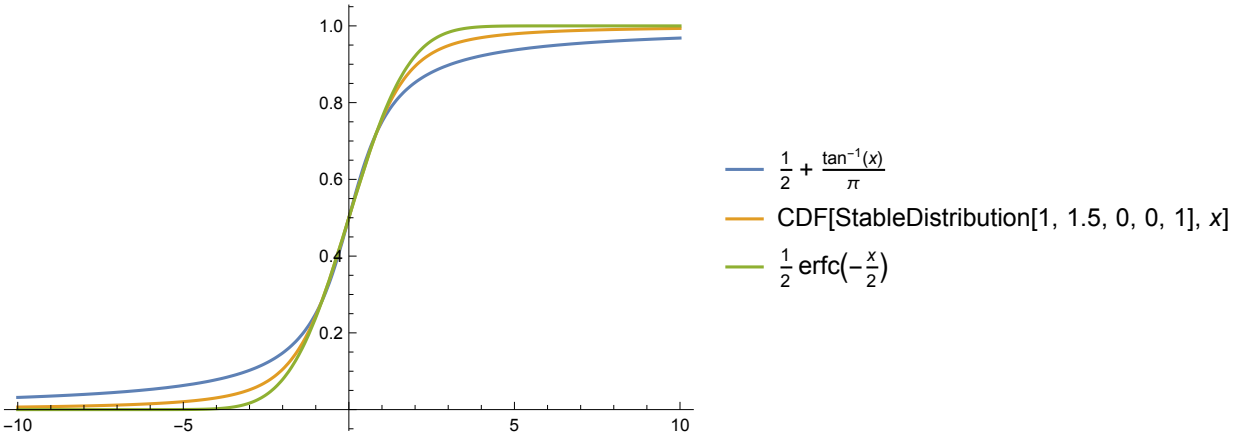
In[]:=
```
Plot[Table[PDF[StableDistribution[1, a, 0, 0, 1], x], {a, {1.0, 1.5, 2.0}}] //
  Evaluate, {x, -10, 10}, PlotLegends → "Expressions"]
```

Out[]=



In[]:=
```
Plot[Table[CDF[StableDistribution[1, a, 0, 0, 1], x], {a, {1.0, 1.5, 2.0}}] //
  Evaluate, {x, -10, 10}, PlotLegends → "Expressions"]
```

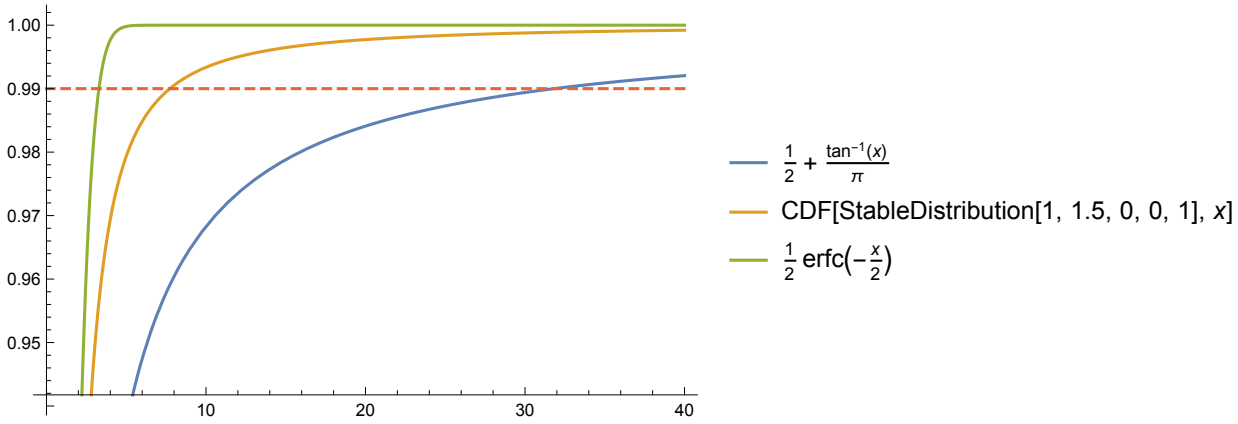
Out[]=



In[]:=
```
(* For alpha of 0.05, can essentially put cap on x as CDF
 (1-alpha). Which ultimately relates payoff cap to VaR metric *)
```

In[]:=
```
Show[Plot[{Table[CDF[StableDistribution[1, a, 0, 0, 1], x], {a, {1.0, 1.5, 2.0}}] //
    Evaluate}, {x, 0, 40}, GridLines → {{}, {}}, PlotLegends → "Expressions"],
 Plot[{ , , , 0.99}, {x, 0, 40}, PlotStyle → Dashed]]
```

Out[]=



In[]:=
```
InverseCDF[StableDistribution[1, 2, 0, 0, 1], 0.99]
```

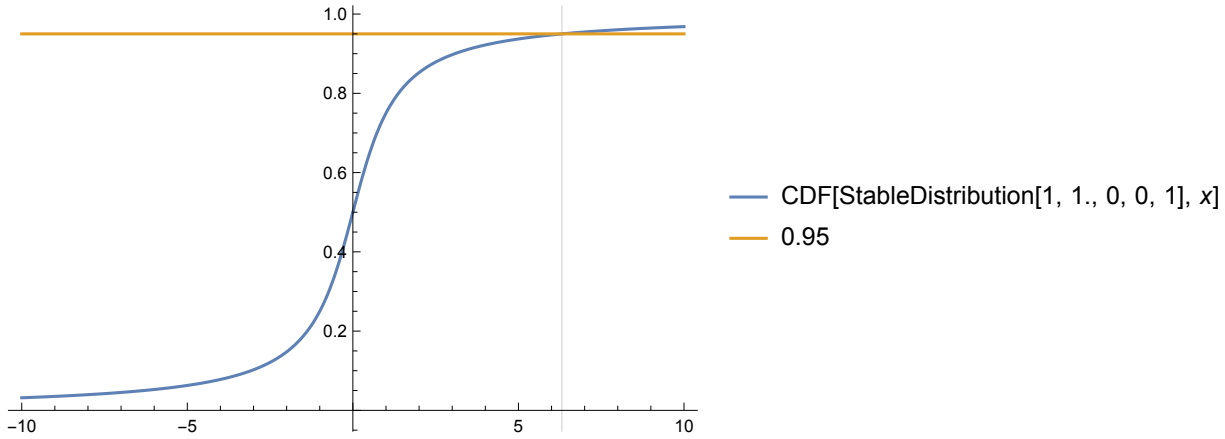
Out[]=
```
3.28995
```

In[]:=
```
InverseCDF[StableDistribution[1, 1, 0, 0, 1], 0.99]
```

Out[]=
```
31.8205
```

In[]:=
```
Plot[{CDF[StableDistribution[1, 1.0, 0, 0, 1], x], 0.95}, {x, -10, 10},
 GridLines → {{6.313751514675042`}, {}}, PlotLegends → "Expressions"]
```

Out[]=

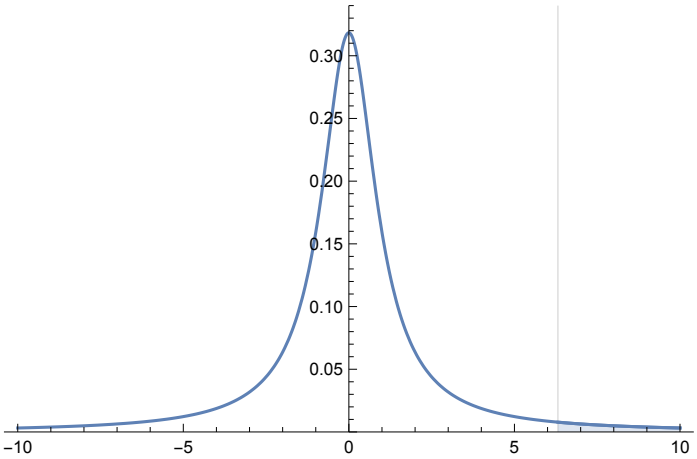


In[]:=
```
Solve[CDF[StableDistribution[1, 1.0, 0, 0, 1], x] == 0.95, x]
```

Out[]=
```
{{x → 6.31375}}
```

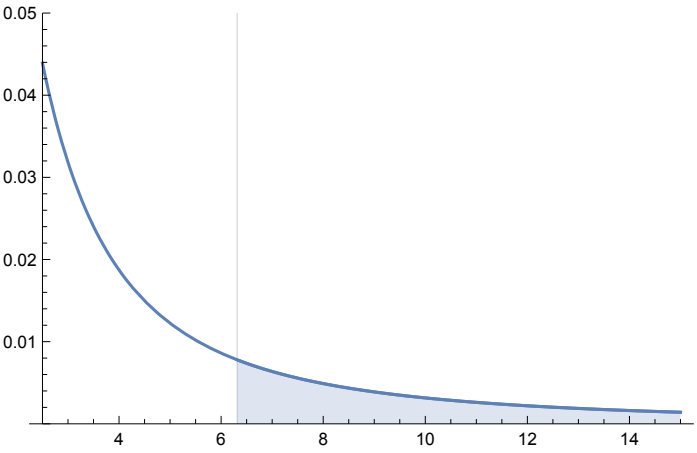
In[•]:=
```
Show[Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]}, {x, -10, 10},
  GridLines → {{6.313751514675042`}, {}}, PlotLegends → "Expressions",
  PlotRange → {0, 0.34}], Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]},
  {x, 6.313751514675042`, 10}, PlotRange → {0, 0.34}, Filling → Axis]]
```

Out[•]=



In[•]:=
```
Show[Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]}, {x, 2.5, 15},
  GridLines → {{6.313751514675042`}, {}}, PlotLegends → "Expressions",
  PlotRange → {0, 0.05}], Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]},
  {x, 6.313751514675042`, 15}, PlotRange → {0, 0.05}, Filling → Axis]]
```

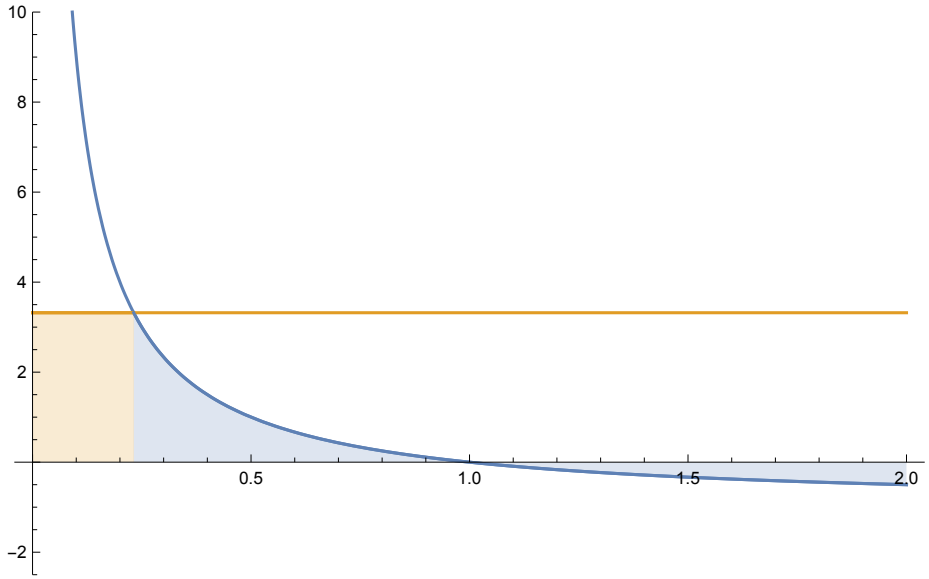
Out[•]=



In[•]:=
```
(* Payoff to a shorter for a bearish OVL trade on inverse market: PnL =
 N * P(0) * [1/P(t) - 1/P(0)]; where P is # ETH / # OVL *)
```

In[•]:=
```
Show[Plot[{1 / x - 1, Exp[1.2]}, {x, 0, 2}, PlotRange → {10, -2.5}],
 Plot[{, Exp[1.2]}, {x, 0, 1 / (Exp[1.2] + 1)}, Filling → Axis],
 Plot[{1 / x - 1}, {x, 1 / (Exp[1.2] + 1), 2}, Filling → Axis]]
```

Out[•]=



In[•]:=
```
1 / (Exp[1.2] + 1)
```

Out[•]= 0.231475

In[•]:=
```
Exp[1.2]
```

Out[•]= 3.32012

In[•]:=
```
Show[%86, AxesLabel → {HoldForm[P], HoldForm[PnL]},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

... Show: Out is not a type of graphics.

Out[•]= Show[%86, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]

In[•]:=
```
Exp[1.2]
```

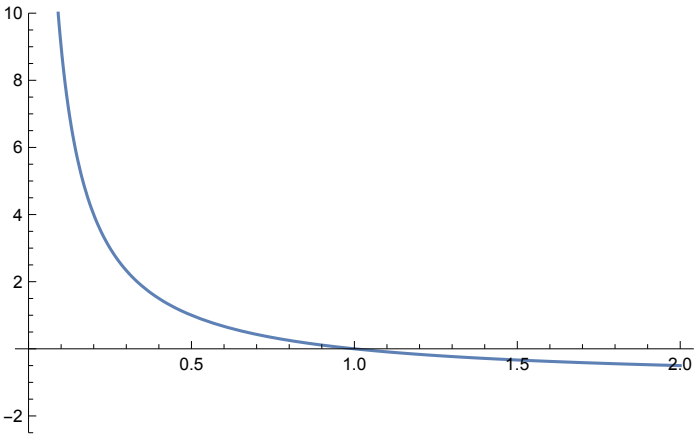
Out[•]= 3.32012

In[•]:=
```
1 / (Exp[1.2] + 1)
```

Out[•]= 0.231475

```
In[•]:= (* TODO: Get some actual numbers for e^X Levy process from fitting. smaller
          uncertainty level alpha used, means greater confidence in X max,
        and larger payoff cap to cut off tails with. This we might need
           considering vol factors from norm fitting were on order of 1e-
          7. Exp[1e-7 * 6.313751514675042 * 144 * 7] =
         1.00063663 is a very small 7d payoff cap of about 100% on
          position. Likely want payoff cap of around 10x over span of 7 days,
        which gives Log[10] of 2.30259 max in exponent. Then know that
         10x * OI imb cap is max amount we can print for one cycle,
        then we adjust cap down accordingly depending on prior printing on
         rolling 7d basis or something to strictly meet inflation goals *)

In[•]:= Exp[10^(-7) * 6.313751514675042 * 144 * 7]

Out[•]= 1.00064

In[•]:= Log[10]

Out[•]= Log[10]

In[•]:= N[Log[10]]

Out[•]= 2.30259

In[•]:= (* Inverse market payoff *)

In[•]:= Plot[{1 / x - 1}, {x, 0, 2}, GridLines → {{}, {}},
         PlotLegends → "Expressions", PlotRange → {10, -2.5}]
```

Out[•]=



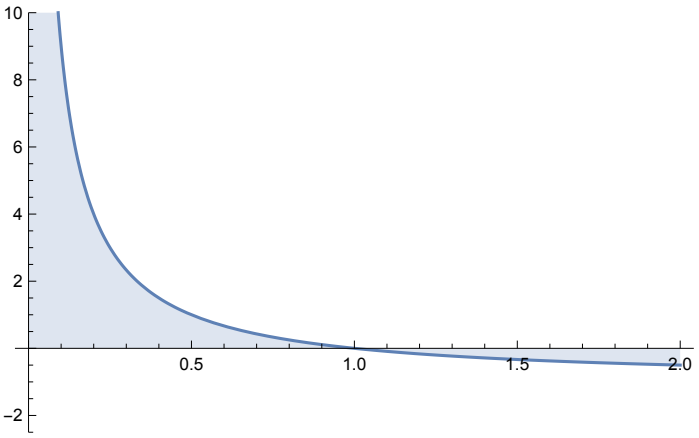

```
In[•]:= Show[%87, AxesLabel → {HoldForm[P], HoldForm[PnL]},
         PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Show: Out is not a type of graphics.

```
Out[•]= Show[%87, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]
```

In[]:=
```
Plot[{1 / x - 1}, {x, 0, 2}, GridLines → {{}, {}},
 PlotLegends → "Expressions", PlotRange → {10, -2.5}, Filling → Axis]
```

Out[]=



In[]:=
```
Show[%93, AxesLabel → {HoldForm[P], HoldForm[PnL]},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

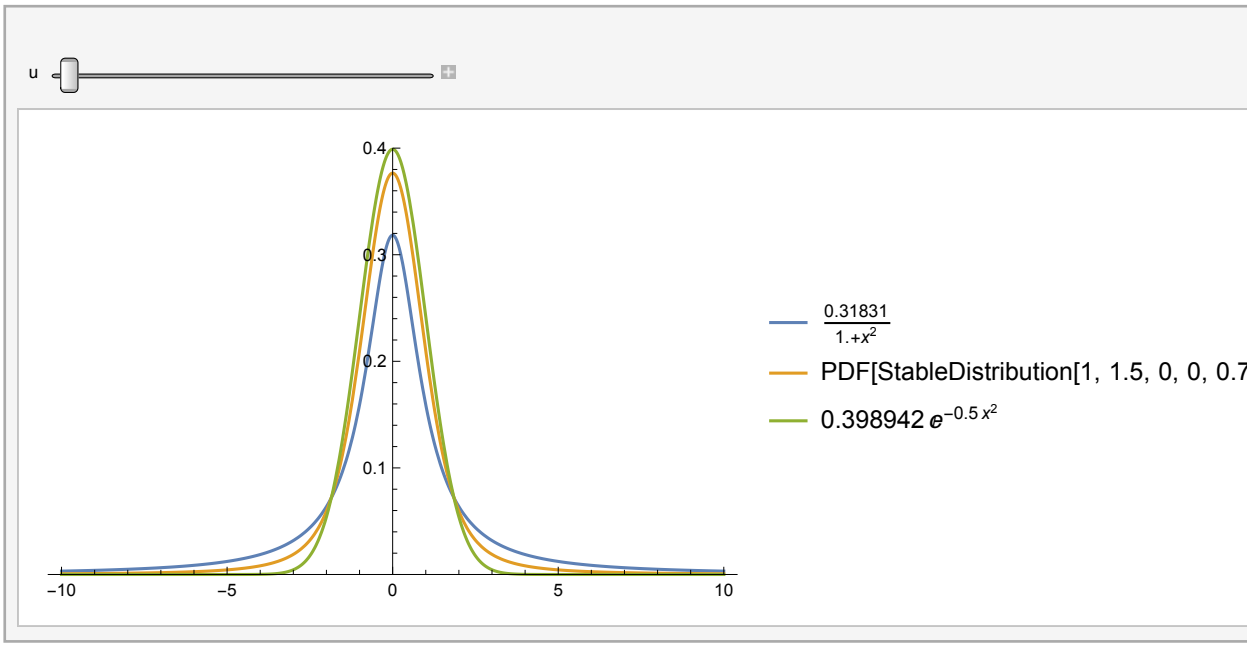
... Show: Out is not a type of graphics.

Out[]=
```
Show[%93, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]
```

In[]:=
```
(* Look at how PDF,
CDF evolve over time with e**(μ*t + σ*L_t) Levy stable increments *)
```

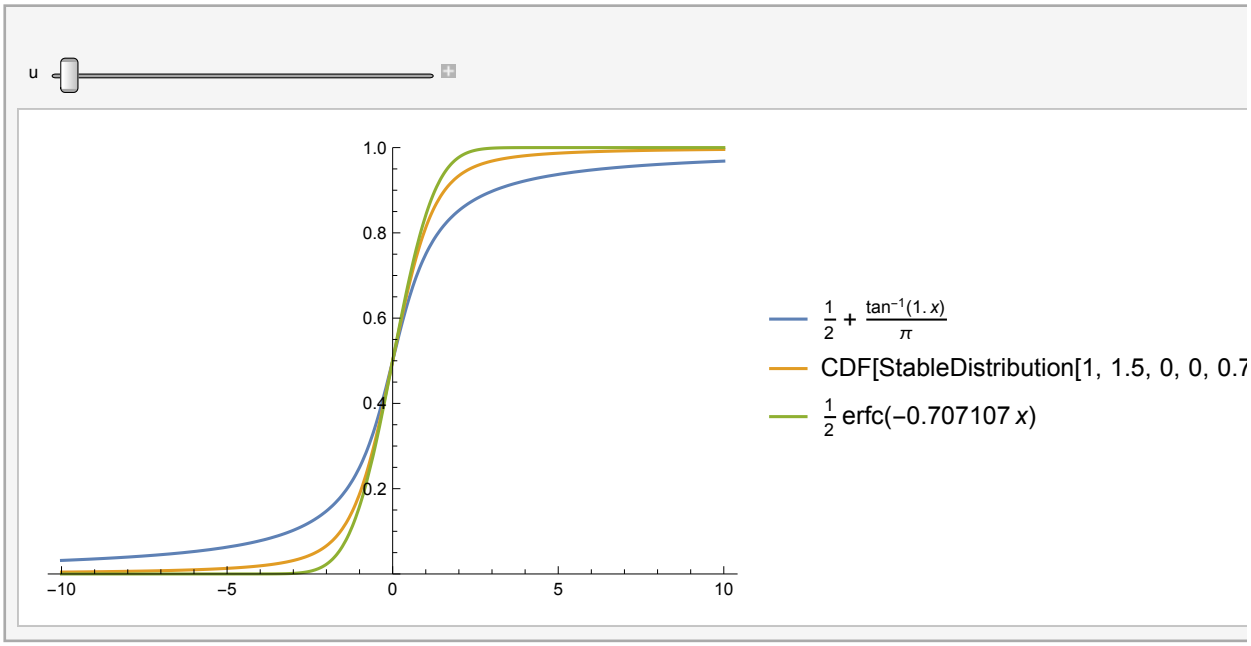
In[]:=
```
Manipulate[Plot[Table[PDF[StableDistribution[1, a, 0, 0, (u / a) ^ (1 / a)], x],
    {a, {1.0, 1.5, 2.0}}] // Evaluate, {x, -10, 10},
  PlotRange → {0, 0.4}, PlotLegends → "Expressions"], {u, 1, 10}]
```

Out[]=



In[ ]:=
```
Manipulate[Plot[Table[CDF[StableDistribution[1, a, 0, 0, (u / a) ^ (1 / a)], x],
    {a, {1.0, 1.5, 2.0}}] // Evaluate, {x, -10, 10},
  PlotRange → {0, 1}, PlotLegends → "Expressions"], {u, 1, 10}]
```

Out[ ]=



In[ ]:=
```
1 / Sqrt[2 * Pi]
```

Out[ ]=
$$\frac{1}{\sqrt{2\,\pi}}$$

In[ ]:=
```
N[1/Sqrt[2 π]]
```

Out[ ]=
```
0.398942
```

In[33]:=
```
(* Fits to log stable below. Starting with 30d worth of data .... *)
```

In[34]:=
```
(* Import from csv *)
```

In[35]:=
```
Directory[]
```

Out[ ]=
```
/Users/personal
```

In[36]:=
```
Module[{directory = SystemDialogInput["Directory"]},
 If[directory =!= $Canceled, SetDirectory[directory]]]
```

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

In[•]:=
```
tblWethDai = Import["30/data-1624665179_weth-dai-twap.csv"]
```

Out[•]=
```
{{, timestamp, twap}, {0, 1.62208 × 10^9, 2 823 283 256 860 140 896 256},
 {1, 1.62208 × 10^9, 2 807 416 500 389 253 480 448},
 {2, 1.62208 × 10^9, 2 795 741 643 020 610 568 192},
 {3, 1.62208 × 10^9, 2 778 509 477 732 340 465 664},
 {4, 1.62208 × 10^9, 2 767 564 848 768 069 140 480}, ··· 3738 ···,
 {3743, 1.62466 × 10^9, 1 840 325 819 329 555 202 048},
 {3744, 1.62466 × 10^9, 1 832 467 229 514 366 976 000},
 {3745, 1.62466 × 10^9, 1 825 074 114 325 430 403 072},
 {3746, 1.62466 × 10^9, 1 817 085 838 281 425 027 072},
 {3747, 1.62466 × 10^9, 1 810 494 773 684 469 760 000},
 {3748, 1.62466 × 10^9, 1 810 804 556 505 733 136 384}}
```

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

In[•]:=
```
Length[tblWethDai]
```

Out[•]=
```
3750
```

In[•]:=
```
FromUnixTime[tblWethDai[[2]][[2]]]
```

Out[•]=
```
Wed 26 May 2021 21:04:33 GMT−4.
```

In[•]:=
```
FromUnixTime[tblWethDai[[Length[tblWethDai]]][[2]]]
```

Out[•]=
```
Fri 25 Jun 2021 19:49:11 GMT−4.
```

In[•]:=
```
tblWethDai[[2]]
```

Out[•]=
```
{0, 1.62208 × 10^9, 2 823 283 256 860 140 896 256}
```

In[•]:=
```
tblWethDai[[2]][[2]]
```

Out[•]=
```
1.62208 × 10^9
```

In[•]:=
```
timesWethDai = Table[tblWethDai[[i]][[2]], {i, 2, Length[tblWethDai]}]
```

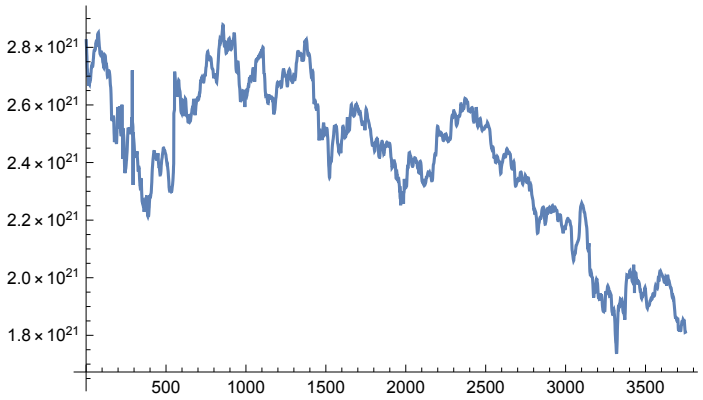
In[•]:=
```
twapsWethDai = Table[tblWethDai[[i]][[3]], {i, 2, Length[tblWethDai]}]
```

In[•]:=
```
twapsWethDai[[1]]
```

Out[•]=
```
2 823 283 256 860 140 896 256
```

In[•]:=
```
ListLinePlot[twapsWethDai, PlotRange → All]
```

Out[•]=



In[•]:=
```
rsWethDai = Differences[Log[twapsWethDai]]
```

Out[•]=
```
{Log[2 807 416 500 389 253 480 448] - Log[2 823 283 256 860 140 896 256],
 Log[2 795 741 643 020 610 568 192] - Log[2 807 416 500 389 253 480 448],
 Log[2 778 509 477 732 340 465 664] - Log[2 795 741 643 020 610 568 192],
 Log[2 767 564 848 768 069 140 480] - Log[2 778 509 477 732 340 465 664],
 Log[2 745 828 354 088 193 490 944] - Log[2 767 564 848 768 069 140 480], ··· 3739 ···,
 Log[1 825 074 114 325 430 403 072] - Log[1 832 467 229 514 366 976 000],
 Log[1 817 085 838 281 425 027 072] - Log[1 825 074 114 325 430 403 072],
 Log[1 810 494 773 684 469 760 000] - Log[1 817 085 838 281 425 027 072],
 -Log[1 810 494 773 684 469 760 000] + Log[1 810 804 556 505 733 136 384]}
```

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

In[•]:=
```
rsWethDai[[100]]
```

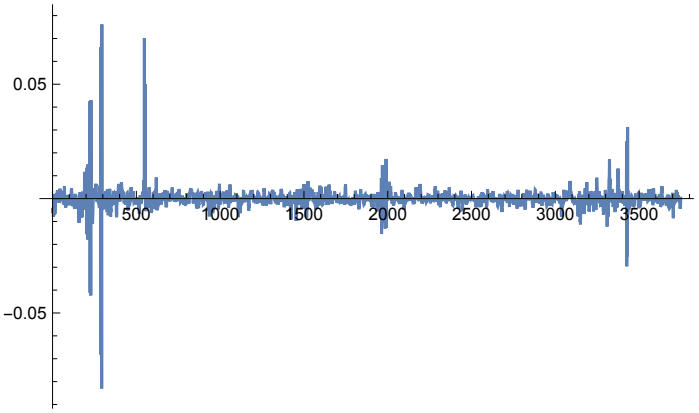
Out[•]=
```
Log[2 770 690 494 359 044 358 144] - Log[2 778 801 122 423 983 833 088]
```

In[•]:=
```
N[Log[2 770 690 494 359 044 358 144] - Log[2 778 801 122 423 983 833 088]]
```

Out[•]=
```
-0.00292302
```

In[•]:=
```
ListLinePlot[rsWethDai, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethDai = EstimatedDistribution[rsWethDai,
  StableDistribution[1, aWethDai, bWethDai, locWethDai, scaleWethDai]]
```

Out[•]=
```
StableDistribution[1, 1.60712, -0.108702, -0.000176324, 0.00134358]
```

In[•]:=
```
FindDistributionParameters[rsWethDai,
 StableDistribution[1, aWethDaif, bWethDaif, locWethDaif, scaleWethDaif]]
```

Out[•]=
```
{aWethDaif → 1.60712, bWethDaif → -0.108702,
 locWethDaif → -0.000176324, scaleWethDaif → 0.00134358}
```

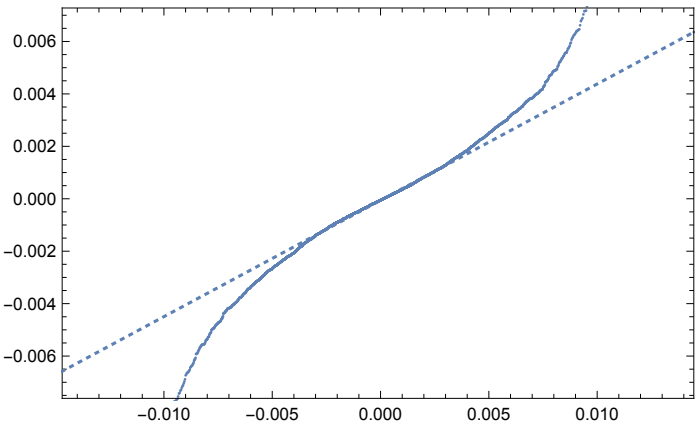
In[•]:=
```
(* Not normal ... *)
```

In[•]:=
```
eNormdistWethDai =
 EstimatedDistribution[rsWethDai, NormalDistribution[uWethDai, sWethDai]]
```

Out[•]=
```
NormalDistribution[-0.000118498, 0.00409742]
```

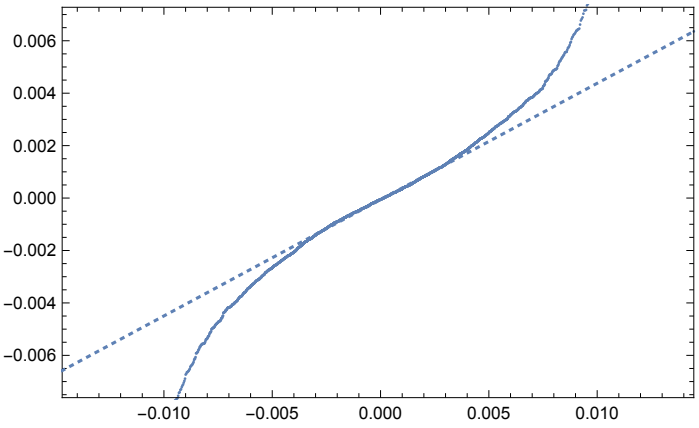
In[•]:=
```
QuantilePlot[rsWethDai]
```

Out[•]=



In[]:=
```
QuantilePlot[rsWethDai, eNormdistWethDai]
```

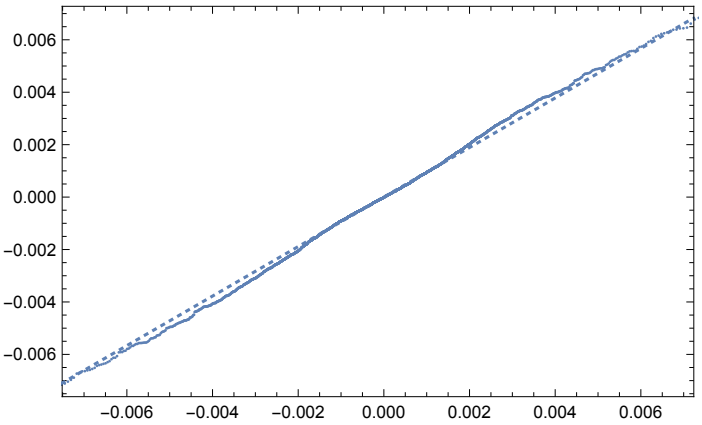
Out[]=



In[]:=
```
(* Stable fit looks good for sample ... *)
```

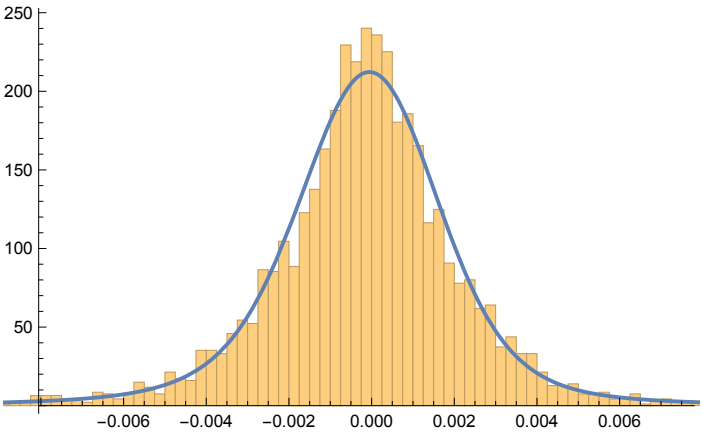
In[]:=
```
QuantilePlot[rsWethDai, edistWethDai]
```

Out[]=



In[]:=
```
Show[Histogram[rsWethDai, {0.00025}, "PDF"], Plot[PDF[edistWethDai, x],
  {x, Min[rsWethDai], Max[rsWethDai]}, PlotRange → All, PlotStyle → Thick]]
```

Out[]=



In[]:=
```
Min[rsWethDai]
```

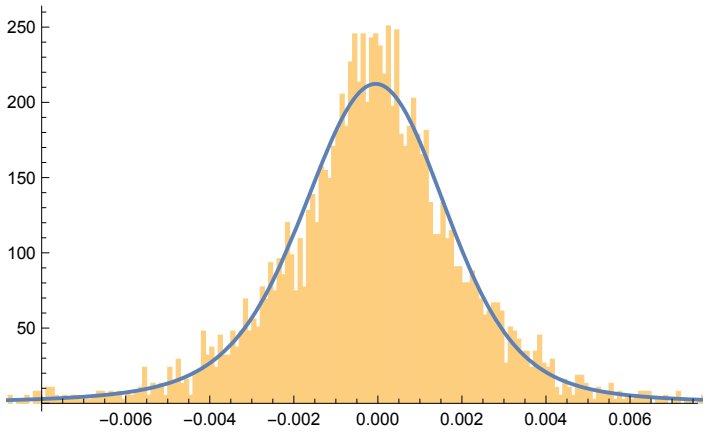
Out[]=
```
Log[2 322 151 657 675 870 961 664] - Log[2 523 108 235 309 858 422 784]
```

In[]:=
```
Max[rsWethDai]
```

Out[]=
$$-\text{Log}[2\,322\,151\,657\,675\,870\,961\,664] + \text{Log}[2\,505\,792\,289\,942\,667\,788\,288]$$

In[]:=
```
Show[Histogram[rsWethDai, {0.0001}, "PDF"], Plot[PDF[edistWethDai, x],
  {x, Min[rsWethDai], Max[rsWethDai]}, PlotRange → All, PlotStyle → Thick]]
```

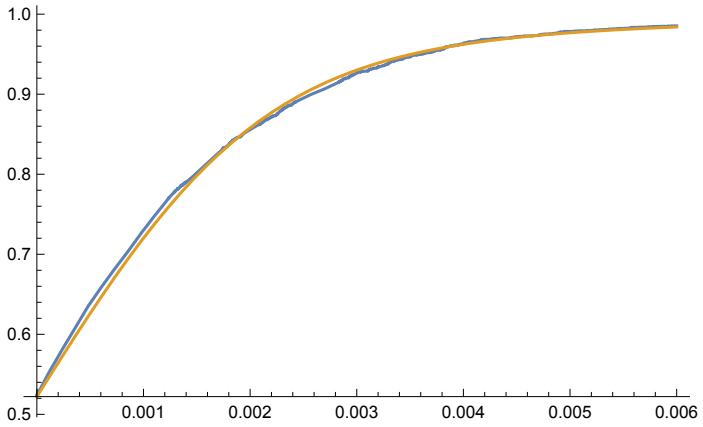
Out[]=



In[]:=
```
(* Compare the CDFs ... *)
```

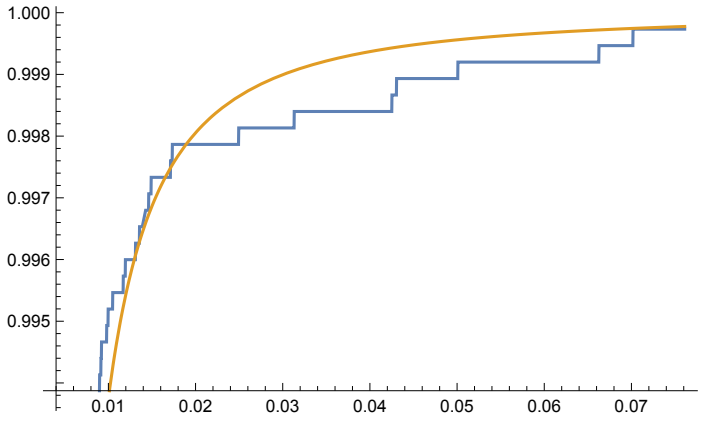
In[]:=
```
Plot[{CDF[EmpiricalDistribution[rsWethDai], x], CDF[edistWethDai, x]}, {x, 0, 0.006}]
```

Out[]=



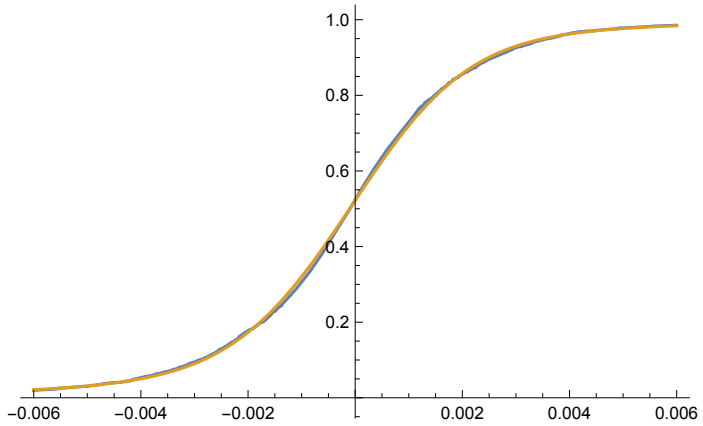
In[•]:=
```
Plot[{CDF[EmpiricalDistribution[rsWethDai], x], CDF[edistWethDai, x]},
 {x, 0.004, Max[rsWethDai]}]
```

Out[•]=



In[•]:=
```
Plot[{CDF[EmpiricalDistribution[rsWethDai], x], CDF[edistWethDai, x]},
 {x, -0.006, 0.006}]
```

Out[•]=



In[•]:=
```
(* Look at histogram behavior *)
```

In[•]:=
```
rsWethDai
```

Out[•]=
```
{Log[2 807 416 500 389 253 480 448] - Log[2 823 283 256 860 140 896 256],
 Log[2 795 741 643 020 610 568 192] - Log[2 807 416 500 389 253 480 448],
 Log[2 778 509 477 732 340 465 664] - Log[2 795 741 643 020 610 568 192],
 Log[2 767 564 848 768 069 140 480] - Log[2 778 509 477 732 340 465 664],
 Log[2 745 828 354 088 193 490 944] - Log[2 767 564 848 768 069 140 480], ··· 3739 ··· ,
 Log[1 825 074 114 325 430 403 072] - Log[1 832 467 229 514 366 976 000],
 Log[1 817 085 838 281 425 027 072] - Log[1 825 074 114 325 430 403 072],
 Log[1 810 494 773 684 469 760 000] - Log[1 817 085 838 281 425 027 072],
 - Log[1 810 494 773 684 469 760 000] + Log[1 810 804 556 505 733 136 384]}
```

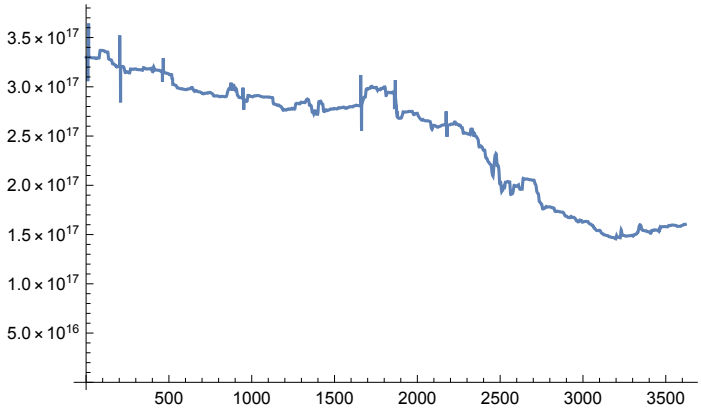
large output | show less | show more | show all | set size limit...

```
In[•]:= Length[rsWethDai]
Out[•]= 3748

In[•]:= (* N rsWethDai obs > 0.001 *)
        fGt001 = # > 0.001 &;
        rsWethDaiGt001 = Select[rsWethDai, fGt001]

In[•]:= N[Length[rsWethDaiGt001] / Length[rsWethDai]]
Out[•]= 0.268943

In[•]:= (* N rsWethDai obs > 0.002 *)
        fGt002 = # > 0.002 &;
        rsWethDaiGt002 = Select[rsWethDai, fGt002]

In[•]:= N[Length[rsWethDaiGt002] / Length[rsWethDai]]
Out[•]= 0.14461

In[•]:= (* N rsWethDai obs > 0.004 *)
        fGt004 = # > 0.004 &;
        rsWethDaiGt004 = Select[rsWethDai, fGt004]

In[•]:= N[Length[rsWethDaiGt004] / Length[rsWethDai]]
Out[•]= 0.0368196

In[•]:= (* N rsWethDai obs > 0.008 *)
        fGt008 = # > 0.008 &;
        rsWethDaiGt008 = Select[rsWethDai, fGt008]

In[•]:= N[Length[rsWethDaiGt008] / Length[rsWethDai]]
Out[•]= 0.00773746

In[•]:= (* N rsWethDai obs > 0.016 *)
        fGt016 = # > 0.016 &;
        rsWethDaiGt016 = Select[rsWethDai, fGt016]

In[•]:= N[Length[rsWethDaiGt016] / Length[rsWethDai]]
Out[•]= 0.00266809

In[•]:= 0.007737459978655283` / 0.0026680896478121665`
Out[•]= 2.9

In[•]:= 0.0368196371398079` / 0.007737459978655283`
Out[•]= 4.75862

In[•]:= 0.14461045891141944` / 0.0368196371398079`
Out[•]= 3.92754

In[•]:= (* 10 min periods ... *)
```

```
In[•]:= period = 600
Out[•]= 600

In[•]:= aWethDai = 1.6071199346185128`
Out[•]= 1.60712

In[•]:= bWethDai = -0.10870229617829233`
Out[•]= -0.108702

In[•]:= locWethDai = -0.00017632373873260288`
Out[•]= -0.000176324

In[•]:= scaleWethDai = 0.0013435809449464948`
Out[•]= 0.00134358

In[•]:= μWethDai = locWethDai / period
Out[•]= -2.93873 × 10^-7

In[•]:= σWethDai = scaleWethDai / (period / aWethDai) ^ (1 / aWethDai)
Out[•]= 0.0000337149

In[•]:= (* Now repeat for more volatile pair like ALCX/WETH ... *)

In[•]:= tblAlcxWeth = Import["30/data-1624665179_alcx_weth-twap.csv"]
Out[•]= {{, timestamp, twap}, {0, 1.62208 × 10^9, 3.29724 × 10^17},
  {1, 1.62208 × 10^9, 3.29397 × 10^17}, {2, 1.62208 × 10^9, 3.29296 × 10^17},
  {3, 1.62208 × 10^9, 3.29306 × 10^17}, {4, 1.62208 × 10^9, 3.29321 × 10^17}, ··· 3607 ···,
  {3612, 1.62466 × 10^9, 1.60227 × 10^17}, {3613, 1.62466 × 10^9, 1.60103 × 10^17},
  {3614, 1.62466 × 10^9, 1.60128 × 10^17}, {3615, 1.62466 × 10^9, 1.60188 × 10^17},
  {3616, 1.62466 × 10^9, 1.60262 × 10^17}, {3617, 1.62466 × 10^9, 1.60305 × 10^17}}

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

In[•]:= twapsAlcxWeth = Table[tblAlcxWeth[[i]][[3]], {i, 2, Length[tblAlcxWeth]}]

In[•]:= twapsAlcxWeth[[100]]
Out[•]= 3.36841 × 10^17
```

In[•]:=
```
ListLinePlot[twapsAlcxWeth, PlotRange → All]
```

Out[•]=



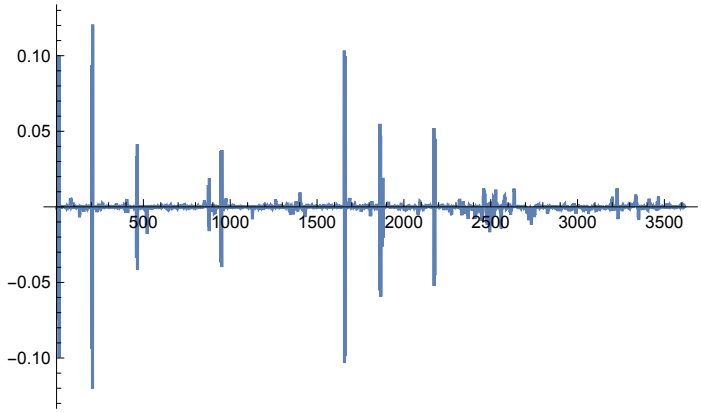
In[•]:=
```
rsAlcxWeth = Differences[Log[twapsAlcxWeth]]
```

In[•]:=
```
rsAlcxWeth[[100]]
```

Out[•]=
$$-9.20679 \times 10^{-6}$$

In[•]:=
```
ListLinePlot[rsAlcxWeth, PlotRange → All]
```

Out[•]=

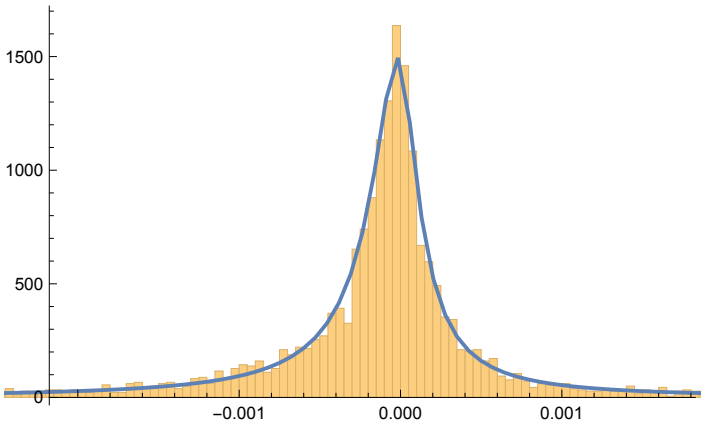


In[•]:=
```
edistAlcxWeth = EstimatedDistribution[
  rsAlcxWeth, StableDistribution[1, aAW, bAW, locAW, scaleAW]]
```

Out[•]=
```
StableDistribution[1, 0.838474, -0.181726, 0.00011192, 0.000231752]
```

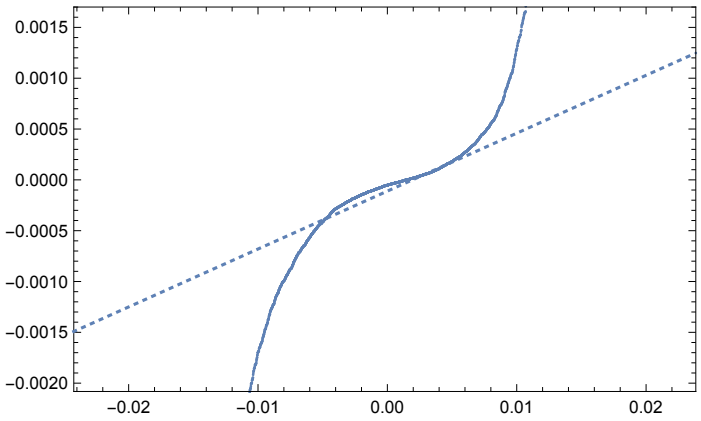
In[•]:=
```
Show[Histogram[rsAlcxWeth, {0.00005}, "PDF"], Plot[PDF[edistAlcxWeth, x],
  {x, Min[rsAlcxWeth], Max[rsAlcxWeth]}, PlotRange → All, PlotStyle → Thick]]
```

Out[•]=



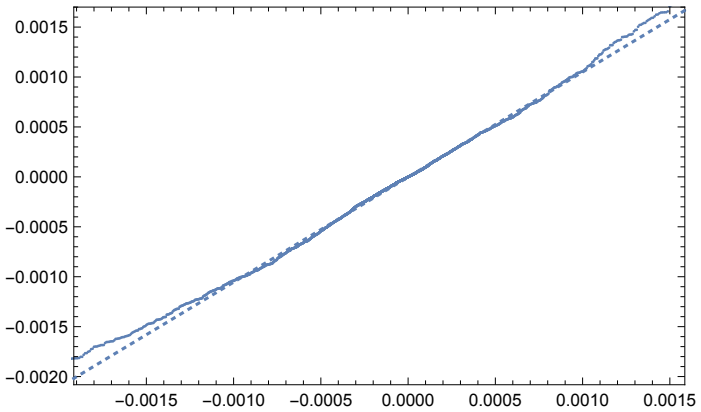
In[•]:=
```
QuantilePlot[rsAlcxWeth]
```

Out[•]=



In[•]:=
```
QuantilePlot[rsAlcxWeth, edistAlcxWeth]
```

Out[•]=



In[•]:=
```
(* That's scary. This is literally fuhgetaboutdit
  territory => Caps on payoff obviously very necessary ... *)
```

In[•]:=
```
(* And just to go through them all, look at UNI/WETH ... How bad? *)
```

In[•]:=
```
tblUniWeth = Import["30/data-1624665179_uni_weth-twap.csv"]
```

Out[•]=
$$\{\{\text{, timestamp, twap}\}, \{0, 1.62208\times10^{9}, 1.01055\times10^{16}\},$$
$$\{1, 1.62208\times10^{9}, 1.01007\times10^{16}\}, \{2, 1.62208\times10^{9}, 1.00728\times10^{16}\},$$
$$\{3, 1.62208\times10^{9}, 1.00671\times10^{16}\}, \{4, 1.62208\times10^{9}, 1.00592\times10^{16}\}, \cdots 2320 \cdots,$$
$$\{2325, 1.62465\times10^{9}, 8.78565\times10^{15}\}, \{2326, 1.62465\times10^{9}, 8.78524\times10^{15}\},$$
$$\{2327, 1.62466\times10^{9}, 8.78139\times10^{15}\}, \{2328, 1.62466\times10^{9}, 8.77619\times10^{15}\},$$
$$\{2329, 1.62466\times10^{9}, 8.77331\times10^{15}\}, \{2330, 1.62466\times10^{9}, 8.77378\times10^{15}\}\}$$

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

In[•]:=
```
twapsUniWeth = Table[tblUniWeth[[i]][[3]], {i, 2, Length[tblUniWeth]}]
```

In[•]:=
```
twapsUniWeth[[100]] / 10^18 * tblWethDai[[100]][[3]] / 10^18
```

Out[•]=
```
28.8615
```

In[•]:=
```
FromUnixTime[tblUniWeth[[2]][[2]]]
```

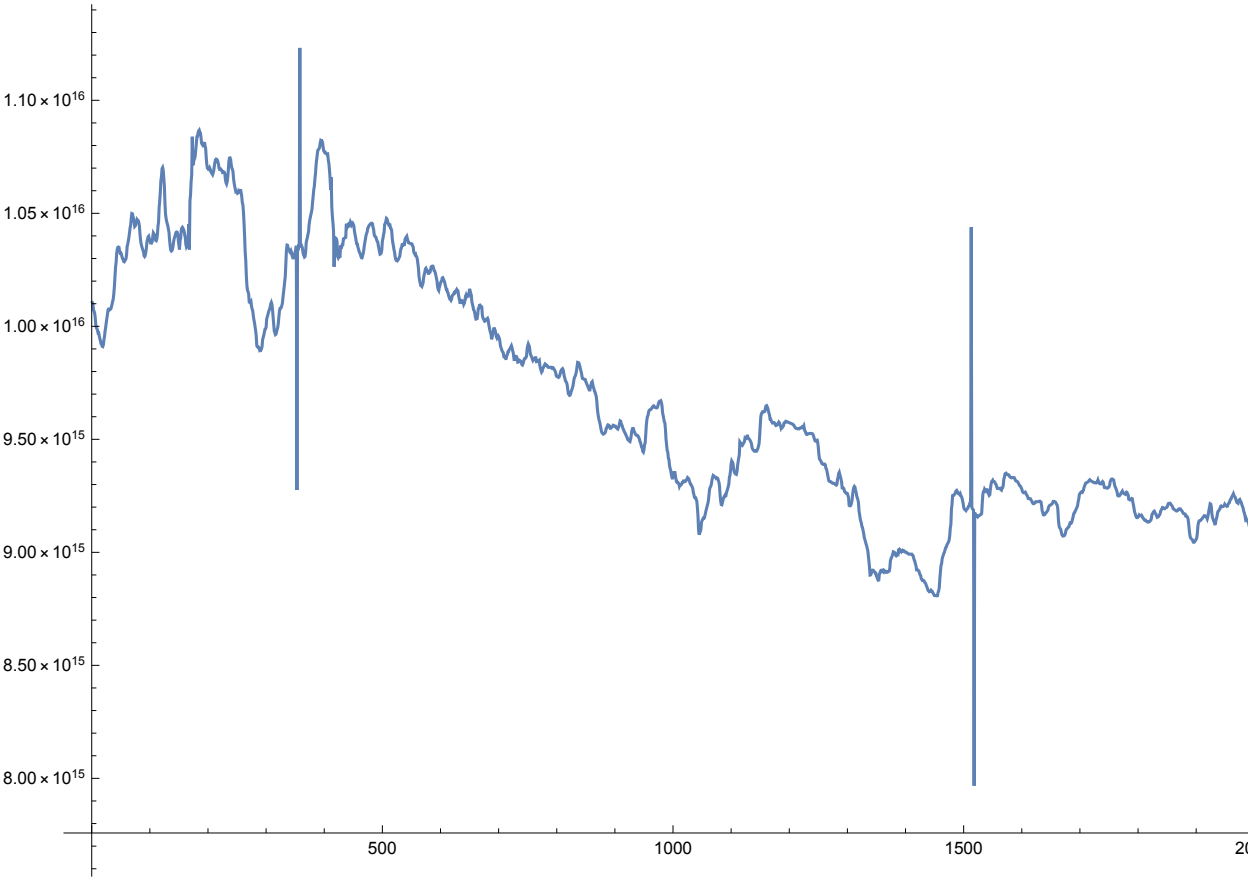
Out[•]=
```
Wed 26 May 2021 21:04:01 GMT–4.
```

In[•]:=
```
FromUnixTime[tblUniWeth[[Length[tblUniWeth]]][[2]]]
```

Out[•]=
```
Fri 25 Jun 2021 19:49:12 GMT–4.
```

In[•]:=
```
ListLinePlot[twapsUniWeth, PlotRange → All]
```

Out[•]=



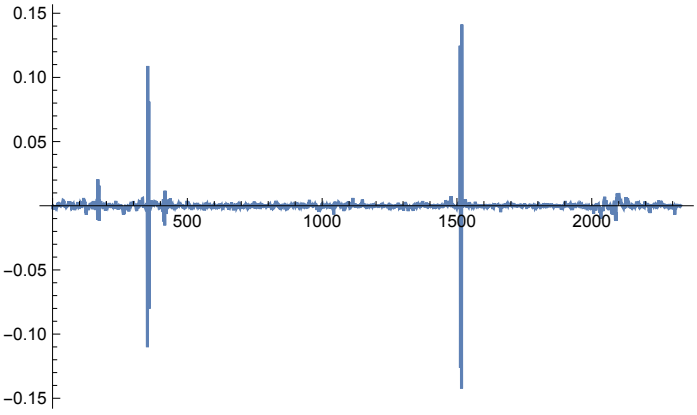
In[•]:=
```
rsUniWeth = Differences[Log[twapsUniWeth]]
```

In[•]:=
```
rsUniWeth[[100]]
```

Out[•]=
```
-0.000807644
```

In[•]:=
```
ListLinePlot[rsUniWeth, PlotRange → All]
```

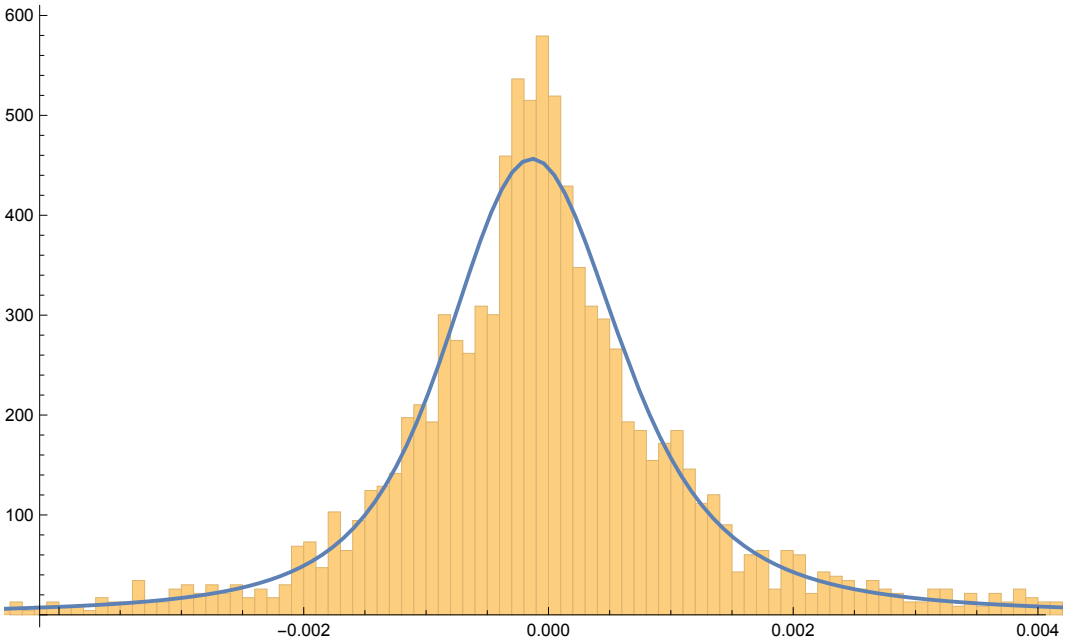
Out[•]=



In[•]:=
```
(* Yea, UNI/WETH definitely not looking as extreme as ALCX/WETH ... :) *)
```

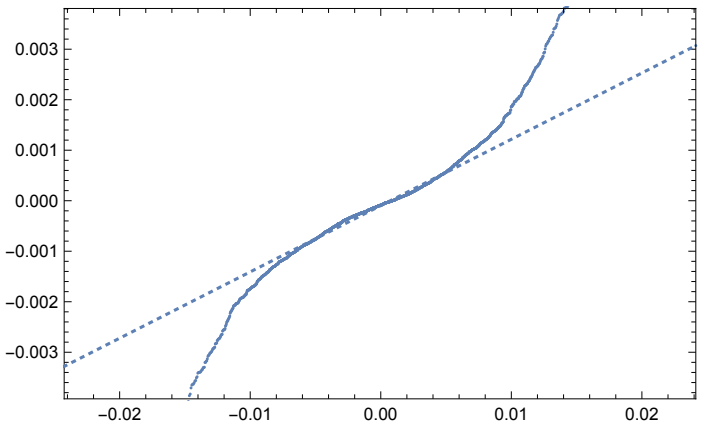
```
In[•]:= edistUniWeth =
         EstimatedDistribution[rsUniWeth, StableDistribution[1, aUW, bUW, locUW, scaleUW]]
Out[•]= StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]

In[•]:= f = {1.3278285879842862`, 0.0816835526225623`,
           -0.0000252748167384907`, 0.0006409442772706084`}
Out[•]= {1.32783, 0.0816836, -0.0000252748, 0.000640944}

In[•]:= Export["fit.csv", f]
Out[•]= fit.csv

In[•]:= Show[Histogram[rsUniWeth, {0.0001}, "PDF"], Plot[PDF[edistUniWeth, x],
          {x, Min[rsUniWeth], Max[rsUniWeth]}, PlotRange → All, PlotStyle → Thick]]
```

Out[•]=



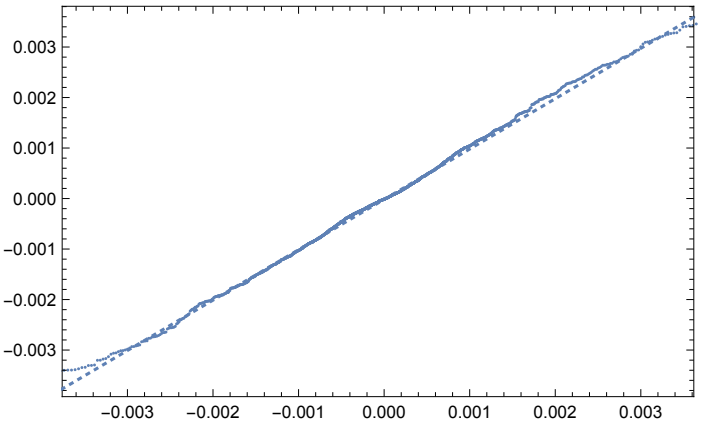

```
In[•]:= QuantilePlot[rsUniWeth]
```

Out[•]=



In[•]:=
```
QuantilePlot[rsUniWeth, edistUniWeth]
```

Out[•]=



In[•]:=
```
(* Examine more data ... ~ 2.5 months. Don't use
 WETH/DAI since cron didn't start til May 19th for this pair *)
```

In[•]:=
```
tblUniWeth90d = Import["90/data-1625069716_uni-weth-twap.csv"]
```

Out[•]=

$$\{\{\text{, timestamp, twap}\}, \{0, 1.61878\times10^{9}, 1.4244\times10^{16}\},$$
$$\{1, 1.61878\times10^{9}, 1.42041\times10^{16}\}, \{2, 1.61878\times10^{9}, 1.41202\times10^{16}\},$$
$$\{3, 1.61878\times10^{9}, 1.41232\times10^{16}\}, \{4, 1.61885\times10^{9}, 1.11905\times10^{16}\}, \text{⋯ 5940 ⋯},$$
$$\{5946, 1.62505\times10^{9}, 8.45706\times10^{15}\}, \{5947, 1.62505\times10^{9}, 8.45105\times10^{15}\},$$
$$\{5948, 1.62505\times10^{9}, 8.43944\times10^{15}\}, \{5949, 1.62506\times10^{9}, 8.40945\times10^{15}\},$$
$$\{5950, 1.62506\times10^{9}, 8.39535\times10^{15}\}, \{5951, 1.62507\times10^{9}, 8.38014\times10^{15}\}\}$$

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

In[•]:=
```
Length[tblUniWeth90d]
```

Out[•]=
```
5952
```

In[•]:=
```
FromUnixTime[tblUniWeth90d[[2]][[2]]]
```

Out[•]=
```
Sun 18 Apr 2021 17:11:37 GMT−4.
```

In[•]:=
```
FromUnixTime[tblUniWeth90d[[Length[tblUniWeth90d]]][[2]]]
```

Out[•]=
```
Wed 30 Jun 2021 11:12:11 GMT−4.
```

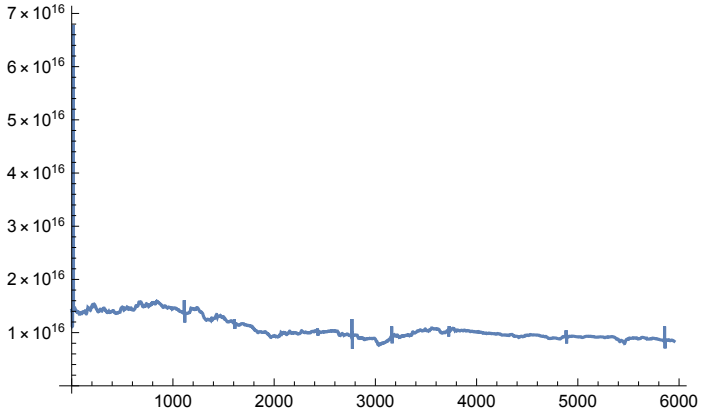
In[•]:=
```
twapsUniWeth90d = Table[tblUniWeth90d[[i]][[3]], {i, 2, Length[tblUniWeth90d]}]
```

In[•]:=
```
twapsUniWeth90d[[2]] / 10^18
```

Out[•]=
```
0.0142041
```

In[]:=
```
ListLinePlot[twapsUniWeth90d, PlotRange → All]
```

Out[]=

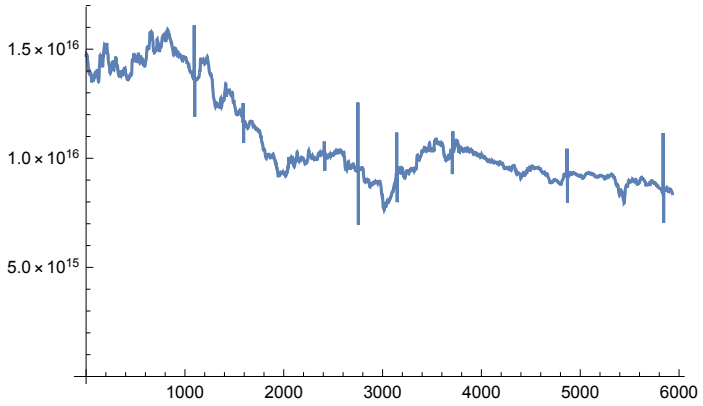


In[]:=
```
(* Something's up with the TWAP calc in early days =>
  related to cron issues ? Likely related to difference in
   metrics Deep was seeing. For now, ignore first 20 elements *)
```

In[]:=
```
twapsUniWeth90dFiltered =
 Table[tblUniWeth90d[[i]][[3]], {i, 20, Length[tblUniWeth90d]}]
```

In[]:=
```
ListLinePlot[twapsUniWeth90dFiltered, PlotRange → All]
```

Out[]=



In[]:=
```
Length[twapsUniWeth90dFiltered ]
```

Out[]=
```
5933
```

In[]:=
```
(* So cuts out a day ... *)
```

In[]:=
```
FromUnixTime[tblUniWeth90d[[2]][[2]]]
```

Out[]=
```
Sun 18 Apr 2021 17:11:37 GMT−4.
```

In[]:=
```
FromUnixTime[tblUniWeth90d[[20]][[2]]]
```

Out[]=
```
Mon 19 Apr 2021 15:31:16 GMT−4.
```

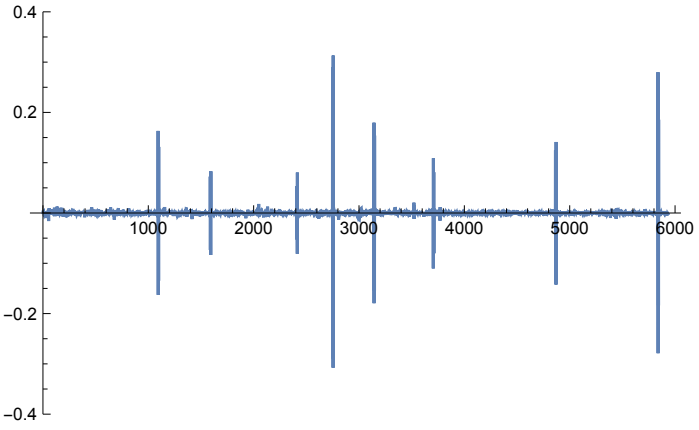
In[]:=
```
rsUniWeth90dFiltered = Differences[Log[twapsUniWeth90dFiltered]]
```

In[•]:=
```
rsUniWeth90dFiltered[[100]]
```

Out[•]=
```
-0.000738056
```

In[•]:=
```
ListLinePlot[rsUniWeth90dFiltered, PlotRange → {-0.4, 0.4}]
```

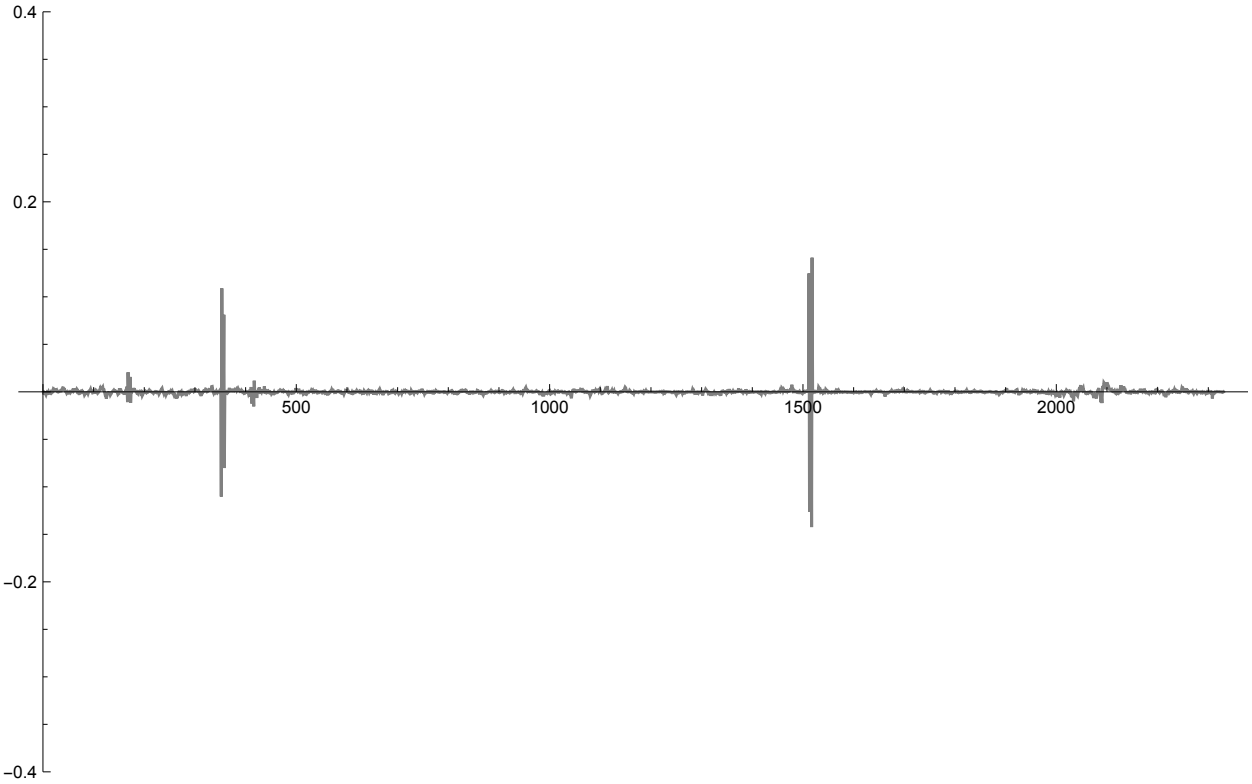
Out[•]=



In[•]:=
```
(* Look at it in comparison to the 30d sampling for UNI/WETH ... *)
```

In[•]:=
```
ListLinePlot[rsUniWeth, PlotRange → {-0.4, 0.4}, PlotStyle → Gray]
```

Out[•]=

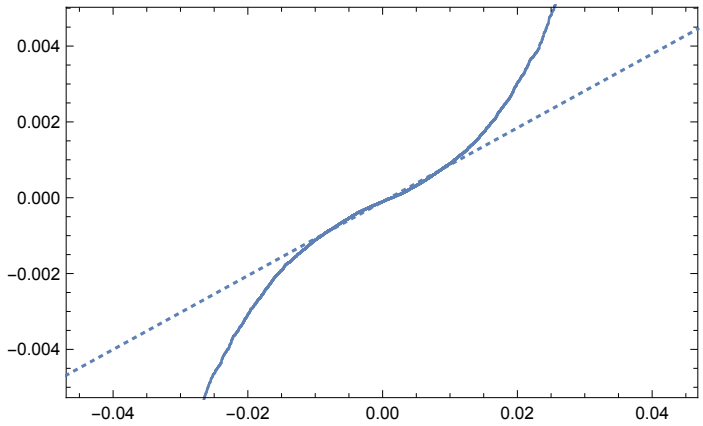


In[•]:=
```
(* Note, the prior 30d sampling doesn't include the last few days at end of June,
which had a large bump up. Included in the 90d pull (see ~ 6000 element). *)
```

In[•]:=
```
(* Check again we're not normal ... *)
```

In[•]:=
```
QuantilePlot[rsUniWeth90dFiltered]
```

Out[•]=



In[•]:=
```
(* Let's fit the 90d data ... *)
```

In[•]:=
```
edistUniWeth90dFiltered = EstimatedDistribution[rsUniWeth90dFiltered,
  StableDistribution[1, aUW90d, bUW90d, locUW90d, scaleUW90d]]
```

Out[•]=
```
StableDistribution[1, 1.29905, 0.00317474, -0.00011575, 0.000868971]
```

In[•]:=
```
(* And compare with prior 30d fit *)
```

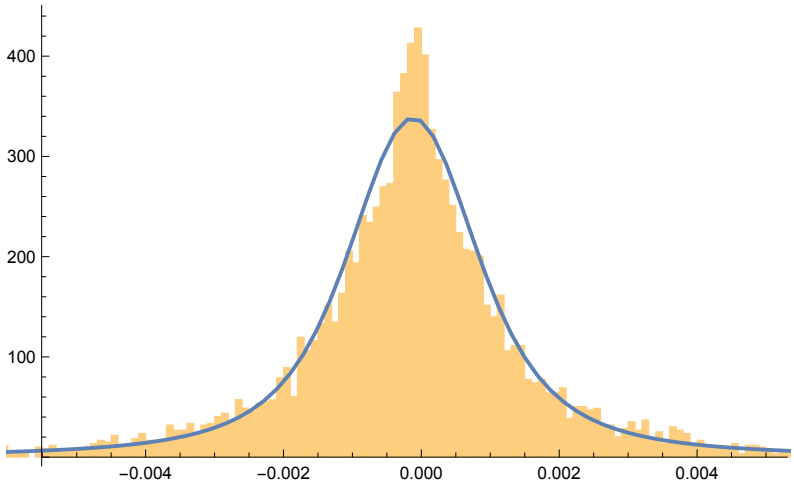
In[•]:=
```
edistUniWeth
```

Out[•]=
```
StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]
```

In[•]:=
```
(* Actually, this isn't terribly different which is great. alpha
 estimation looks consistent at first glance for UNI/WETH *)
```

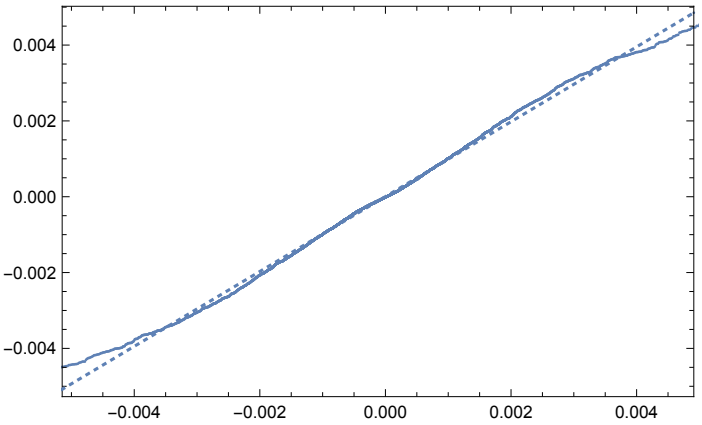
In[•]:=
```
Show[Histogram[rsUniWeth90dFiltered, {0.0001}, "PDF"],
 Plot[PDF[edistUniWeth90dFiltered, x], {x, Min[rsUniWeth90dFiltered],
   Max[rsUniWeth90dFiltered]}, PlotRange → All, PlotStyle → Thick]]
```

Out[•]=



In[]:=
```
QuantilePlot[rsUniWeth90dFiltered, edistUniWeth90dFiltered]
```

Out[]=

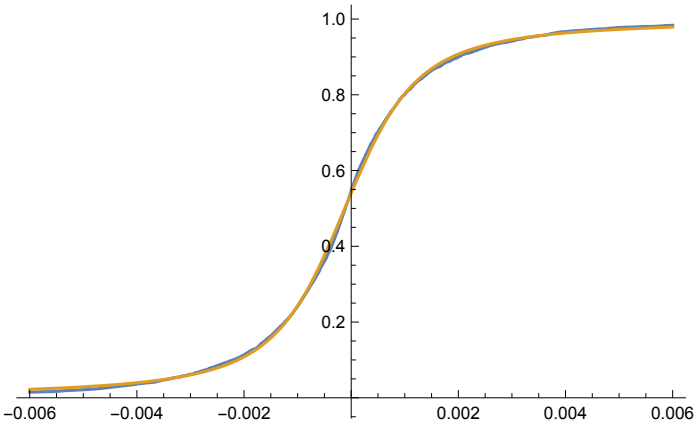


In[]:=
```
(* hmm still slightly off at ends but not terrible *)
```

In[]:=
```
(* Compare the CDFs *)
```

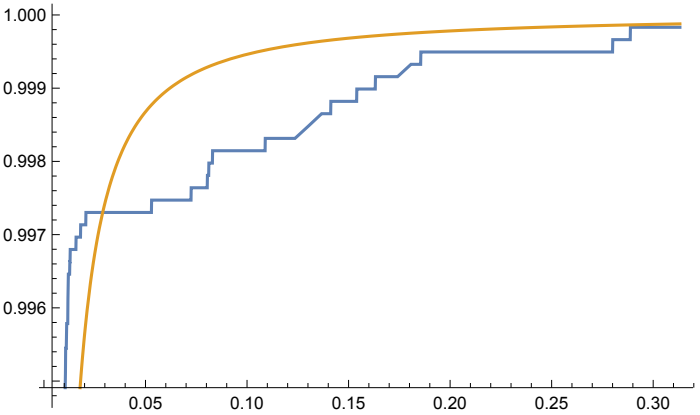
In[]:=
```
Plot[{CDF[EmpiricalDistribution[rsUniWeth90dFiltered], x],
  CDF[edistUniWeth90dFiltered, x]}, {x, -0.006, 0.006}]
```

Out[]=



In[]:=
```
Plot[{CDF[EmpiricalDistribution[rsUniWeth90dFiltered], x],
  CDF[edistUniWeth90dFiltered, x]}, {x, 0.004, Max[rsUniWeth90dFiltered]}]
```

Out[]=



In[]:=
```
(* Nice ... *)
```

In[•]:=
```
(* Filter out last month, so only analyzing first month of data *)
```

In[•]:=
```
Length[tblUniWeth90d] / 2
```

Out[•]=
```
2976
```

In[•]:=
```
tblUniWeth90d[[2976]]
```

Out[•]=
```
{2975, 1.62173 × 10^9, 8.912 × 10^15}
```

In[•]:=
```
FromUnixTime[tblUniWeth90d[[2]][[2]]]
```

Out[•]=
```
Sun 18 Apr 2021 17:11:37 GMT−4.
```

In[•]:=
```
FromUnixTime[tblUniWeth90d[[2976]][[2]]]
```

Out[•]=
```
Sat 22 May 2021 21:52:37 GMT−4.
```

In[•]:=
```
rsUniWeth90dFiltered1stMo = Table[rsUniWeth90dFiltered[[i]], {i, 1, 2976}]
```

In[•]:=
```
(* Fit the first mo of 90d data *)
```

In[•]:=
```
edistUniWeth90dFiltered1stMo = EstimatedDistribution[rsUniWeth90dFiltered1stMo,
  StableDistribution[1, aUW90d1m, bUW90d1m, locUW90d1m, scaleUW90d1m]]
```

Out[•]=
```
StableDistribution[1, 1.38642, -0.0595957, -0.000254308, 0.00107224]
```

In[•]:=
```
edistUniWeth
```

Out[•]=
```
StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]
```

In[•]:=
```
(* Not terrible .... *)
```

In[211]:=
```
(* Go back and download WETH/USDC data from cron. Look at that *)
```

In[212]:=
```
tblWethUsdc90d = Import["90/data-1625069716_weth-usdc-twap.csv"]
```

Out[•]=
```
{{, 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},
 {5, 1.61878 × 10^9, 2.25775 × 10^9}, {6, 1.61878 × 10^9, 2.25804 × 10^9},
 ··· 9164 ···, {9173, 1.62507 × 10^9, 2.13479 × 10^9},
 {9174, 1.62507 × 10^9, 2.12846 × 10^9}, {9175, 1.62507 × 10^9, 2.1222 × 10^9},
 {9176, 1.62507 × 10^9, 2.11587 × 10^9}, {9177, 1.62507 × 10^9, 2.11137 × 10^9},
 {9178, 1.62507 × 10^9, 2.10661 × 10^9}, {9179, 1.62507 × 10^9, 2.10415 × 10^9}}

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

In[213]:=
```
Length[tblWethUsdc90d]
```

Out[•]=
```
9179
```

In[214]:=
```
FromUnixTime[tblWethUsdc90d[[2]][[2]]]
```

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

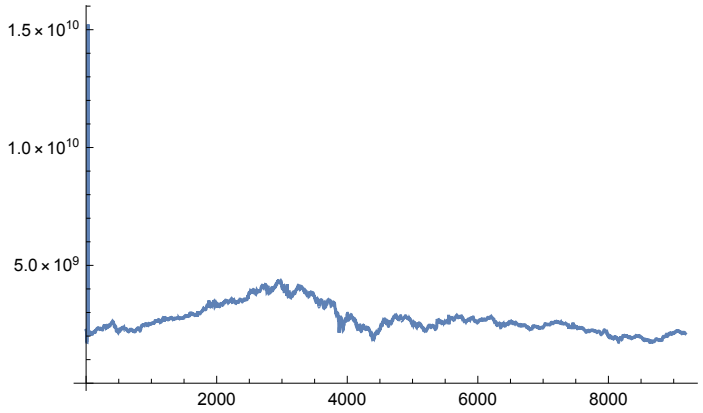
In[215]:=
```
FromUnixTime[tblWethUsdc90d[[Length[tblWethUsdc90d]]][[2]]]
```

Out[]= Wed 30 Jun 2021 12:09:58 GMT−4.

In[216]:=
```
twapsWethUsdc90d = Table[tblWethUsdc90d[[i]][[3]], {i, 2, Length[tblWethUsdc90d]}]
```

In[217]:=
```
ListLinePlot[twapsWethUsdc90d, PlotRange → All]
```

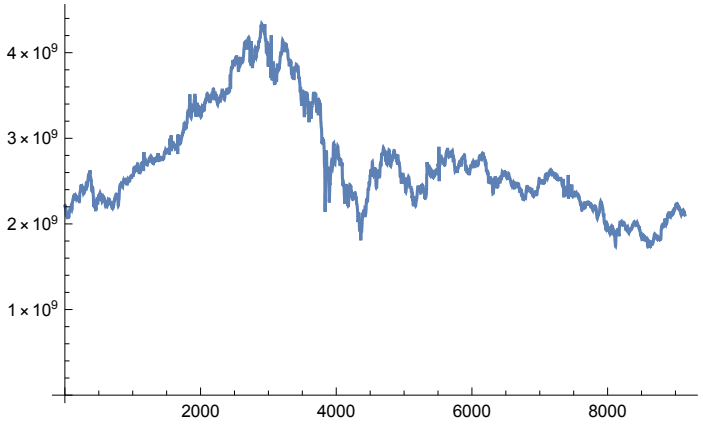
Out[]=



In[218]:=
```
(* Same issues here for early data as UNI/WETH. Cut off first 40 elements *)
twapsWethUsdc90dFiltered =
 Table[twapsWethUsdc90d[[i]], {i, 40, Length[twapsWethUsdc90d]}]
```

In[219]:=
```
ListLinePlot[twapsWethUsdc90dFiltered, PlotRange → All]
```

Out[]=



In[220]:=
```
FromUnixTime[tblWethUsdc90d[[40]][[2]]]
```

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

In[221]:=
```
FromUnixTime[tblWethUsdc90d[[Length[twapsWethUsdc90dFiltered]]][[2]]]
```

Out[]= Wed 30 Jun 2021 04:33:51 GMT−4.

In[222]:=
```
(* Calculate the rs ... *)
```

In[223]:=
```
rsWethUsdc90dFiltered = Differences[Log[twapsWethUsdc90dFiltered]]
```

Out[•]=
```
{-0.00120673, 0.000176452, -0.0000603568, -0.000373722,
 -0.000389599, 0.000559858, -0.00092094, -0.00154871, -0.000905547,
 -0.000584311, -0.00172231, -0.00371162, ··· 9115 ···, 0.00150055,
 0.000591025, 0.000421125, -0.000241228, -0.000868266, -0.00297129,
 -0.00294238, -0.00298715, -0.00213101, -0.00225849, -0.00116666}
```

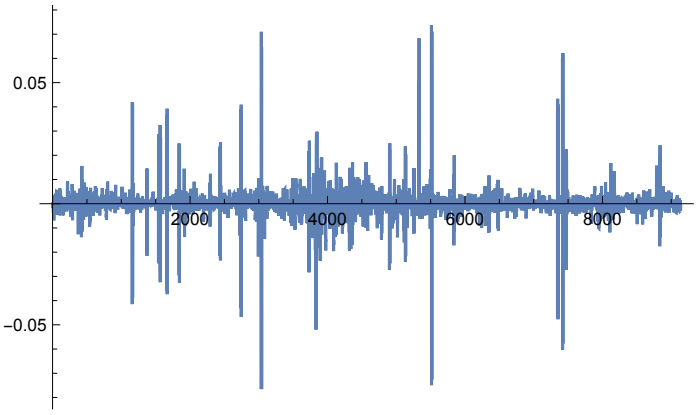
large output | show less | show more | show all | set size limit...

In[224]:=
```
rsWethUsdc90dFiltered[[100]]
```

Out[•]=
```
-0.00443888
```

In[225]:=
```
ListLinePlot[rsWethUsdc90dFiltered, PlotRange → All]
```

Out[•]=



In[226]:=
```
edistWethUsdc90dFiltered = EstimatedDistribution[rsWethUsdc90dFiltered,
  StableDistribution[1, aWU90d, bWU90d, locWU90d, scaleWU90d]]
```

Out[•]=
```
StableDistribution[1, 1.46465, -0.0496207, -0.000010553, 0.00154277]
```

In[227]:=
```
(* This seems more reasonable. more data from weth/usdc lead
 to decrease in alpha because included massive run up to $4k *)
```

In[228]:=
```
FromUnixTime[tblWethUsdc90d[[5000]][[2]]]
```

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

In[229]:=
```
FromUnixTime[tblWethUsdc90d[[Length[tblWethUsdc90d]]][[2]]]
```

Out[•]=
```
Wed 30 Jun 2021 12:09:58 GMT-4.
```

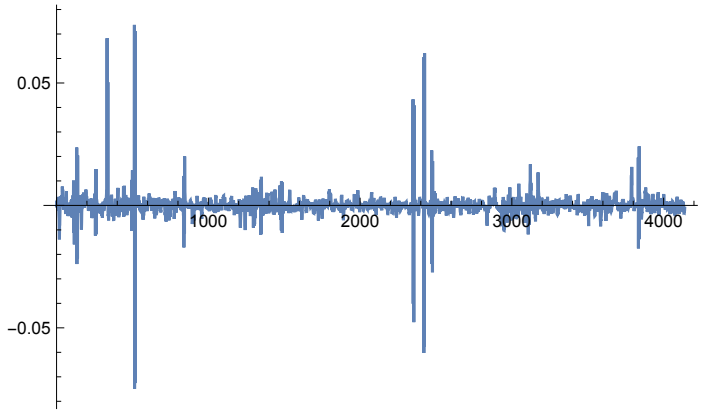
In[•]:=
```
(* Compare with last month which should be close to same fit as ethDai above *)
rsWethUsdc90dFilteredLastMo =
 Table[rsWethUsdc90dFiltered[[i]], {i, 5000, Length[rsWethUsdc90dFiltered]}]
```

In[]:=
```
Length[rsWethUsdc90dFilteredLastMo]
```

Out[]=
```
4139
```

In[]:=
```
ListLinePlot[rsWethUsdc90dFilteredLastMo, PlotRange → All]
```

Out[]=



In[]:=
```
edistWethUsdc90dFilteredLastMo =
 EstimatedDistribution[rsWethUsdc90dFilteredLastMo,
  StableDistribution[1, aWU90dLM, bWU90dLM, locWU90dLM, scaleWE90dLM]]
```

Out[]=
```
StableDistribution[1, 1.58793, -0.0706768, -0.000101131, 0.00135632]
```

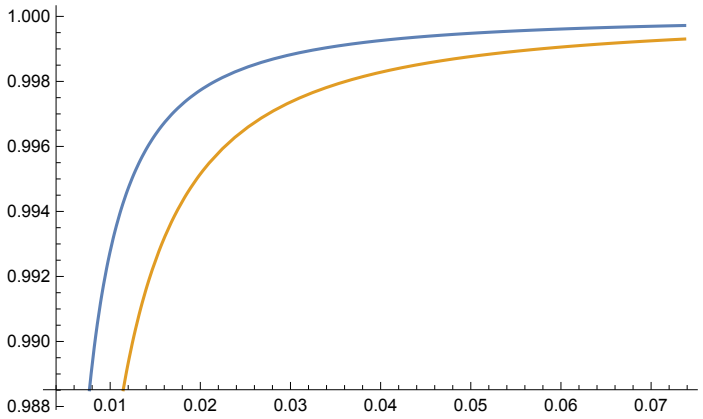
In[]:=
```
edistWethDai
```

Out[]=
```
StableDistribution[1, 1.60712, -0.108702, -0.000176324, 0.00134358]
```

In[]:=
```
(* It's not terrible in terms of alpha difference
 but likely manifests in extreme fashion for inverse CDF? *)
```

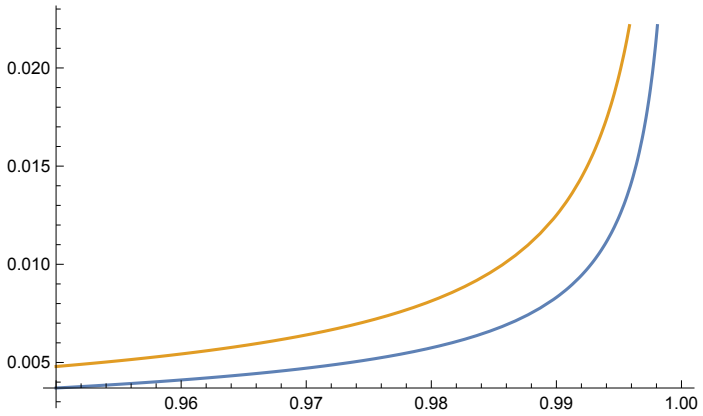
In[]:=
```
Plot[{CDF[edistWethUsdc90dFilteredLastMo, x], CDF[edistWethUsdc90dFiltered, x]},
 {x, 0.004, Max[rsWethUsdc90dFiltered]}]
```

Out[]=



In[]:=
```
Plot[{InverseCDF[edistWethUsdc90dFilteredLastMo, x],
  InverseCDF[edistWethUsdc90dFiltered, x]}, {x, 0.95, 1.0}]
```

Out[]=



In[]:=
```
InverseCDF[edistWethUsdc90dFiltered, 0.99]
```

Out[]=
```
0.0124991
```

In[]:=
```
InverseCDF[edistWethUsdc90dFilteredLastMo, 0.99]
```

Out[]=
```
0.00831473
```

In[]:=
```
InverseCDF[edistWethUsdc90dFiltered, 0.99] /
 InverseCDF[edistWethUsdc90dFilteredLastMo, 0.99]
```

Out[]=
```
1.50325
```

In[]:=
```
(* Factor of 1.5 difference ... Not terrible especially if
 we're conservative with k values: k ~ e**(F^{-1}(1-alpha)) *)
```

In[]:=
```
Exp[InverseCDF[edistWethUsdc90dFiltered, 0.99]]
```

Out[]=
```
1.01258
```

In[]:=
```
Exp[InverseCDF[edistWethUsdc90dFilteredLastMo, 0.99]]
```

Out[]=
```
1.00835
```

In[]:=
```
(Exp[InverseCDF[edistWethUsdc90dFiltered, 0.99]] -
   Exp[InverseCDF[edistWethUsdc90dFilteredLastMo, 0.99]]) /
 Exp[InverseCDF[edistWethUsdc90dFilteredLastMo, 0.99]]
```

Out[]=
```
0.00419318
```

In[]:=
```
(* 0.4% difference in value we care about ... Yea not massive *)
```

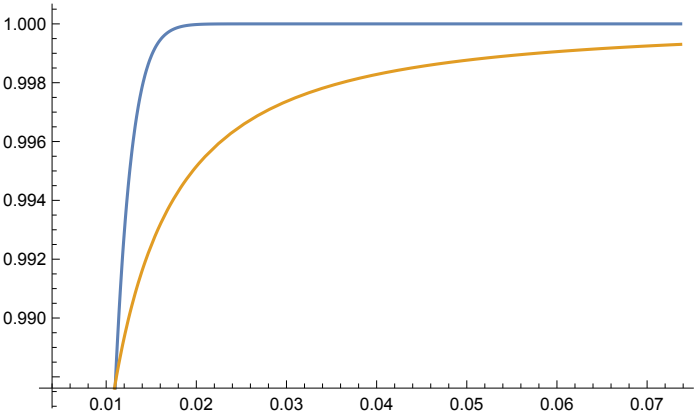
In[]:=
```
(* Compared with normal dist fit? *)
```

In[]:=
```
eNormdistWethUsdc90dFiltered = EstimatedDistribution[
  rsWethUsdc90dFiltered, NormalDistribution[uWethUsdc, sWethUsdc]]
```

Out[]=
$$\text{NormalDistribution}\left[-5.27153\times 10^{-6},\ 0.00489882\right]$$

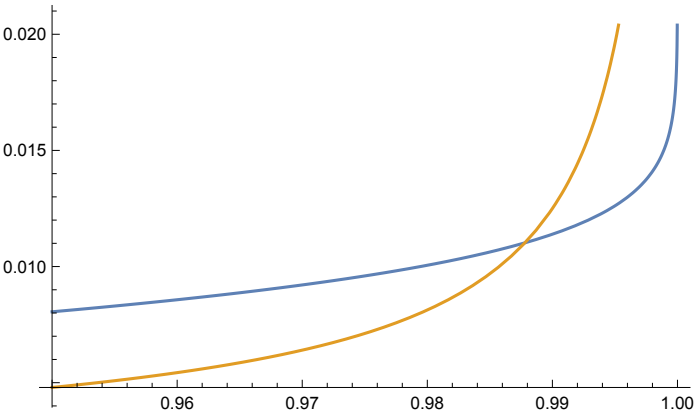
```
In[•]:= InverseCDF[eNormdistWethUsdc90dFiltered, 0.99]
Out[•]= 0.0113911

In[•]:= (* Interesting, close ... *)

In[•]:= Plot[{CDF[eNormdistWethUsdc90dFiltered, x], CDF[edistWethUsdc90dFiltered, x]},
         {x, 0.004, Max[rsWethUsdc90dFiltered]}]
```

Out[•]=



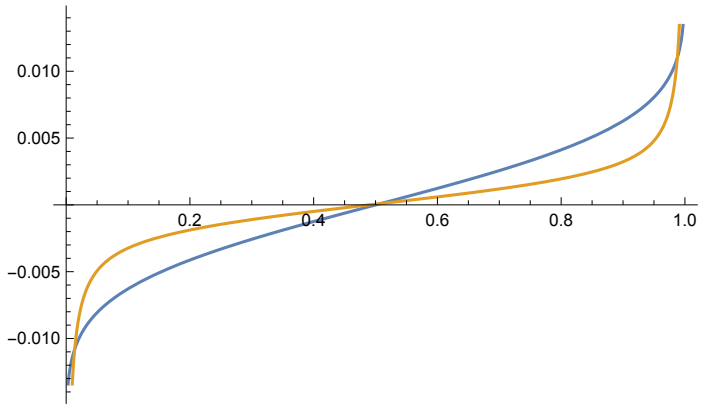

```
In[•]:= Plot[{InverseCDF[eNormdistWethUsdc90dFiltered, x],
           InverseCDF[edistWethUsdc90dFiltered, x]}, {x, 0.95, 1.0}]
```

Out[•]=



In[•]:=
```
Plot[{InverseCDF[eNormdistWethUsdc90dFiltered, x],
  InverseCDF[edistWethUsdc90dFiltered, x]}, {x, 0, 1.0}]
```

Out[•]=



In[•]:=
```
(* More around 0.999 mark where divergence
 is bad. Less so for stable v. stable w diff alpha *)
```

In[•]:=
```
InverseCDF[edistWethUsdc90dFiltered, 0.999]
```

Out[•]= 0.0576731

In[•]:=
```
InverseCDF[edistWethUsdc90dFilteredLastMo, 0.999]
```

Out[•]= 0.0332197

In[•]:=
```
InverseCDF[eNormdistWethUsdc90dFiltered, 0.999]
```

Out[•]= 0.0151332

In[•]:=
```
(* yes, much bigger difference here ... and even worse at 0.9999 *)
```

In[•]:=
```
InverseCDF[edistWethUsdc90dFiltered, 0.9999]
```

Out[•]= 0.276565

In[•]:=
```
InverseCDF[edistWethUsdc90dFilteredLastMo, 0.9999]
```

Out[•]= 0.140854

In[•]:=
```
InverseCDF[eNormdistWethUsdc90dFiltered, 0.9999]
```

Out[•]= 0.0182135

In[•]:=
```
(* Still remains about a factor of 2
 between stable v stable w differing alphas. HOWEVER,
becomes factor of 10 at 99.99% confidence bw normal and stable *)
```

In[•]:=
```
Exp[InverseCDF[eNormdistWethUsdc90dFiltered, 0.9999]]
```

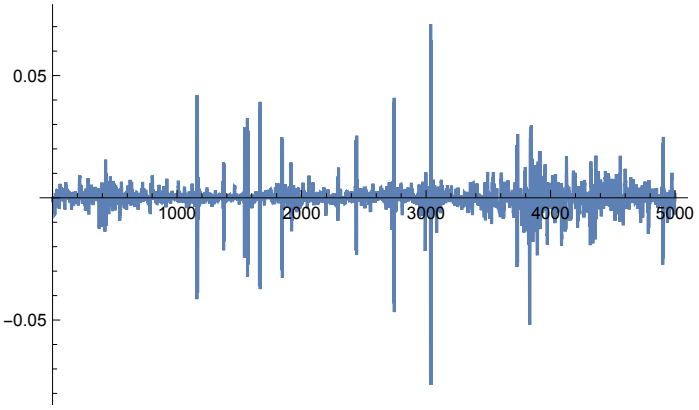
Out[•]= 1.01838

In[•]:=
```
Exp[InverseCDF[edistWethUsdc90dFilteredLastMo, 0.9999]]
```

Out[•]= 1.15126

```
In[•]:= Exp[InverseCDF[edistWethUsdc90dFiltered, 0.9999]]
Out[•]= 1.31859

In[•]:= (* which in VaR calc of Exp[F^{-1}] -
      1 translates to k jumping from 0.018 up to 0.15 / 0.32 .... *)

In[•]:= (* So yes we likely want more data and use stable. Report fits on
     both rolling 30d basis and full set of data (or longer time frame). *)

In[•]:= (* Compare first month w last month of ETH/USDC *)

In[•]:= rsWethUsdc90dFilteredFirstMo = Table[rsWethUsdc90dFiltered[[i]], {i, 1, 5000}]

In[•]:= ListLinePlot[rsWethUsdc90dFilteredFirstMo, PlotRange → All]
```

Out[•]=




```
In[•]:= edistWethUsdc90dFilteredLastMo
Out[•]= StableDistribution[1, 1.58793, -0.0706768, -0.000101131, 0.00135632]

In[•]:= edistWethUsdc90dFilteredFirstMo =
     EstimatedDistribution[rsWethUsdc90dFilteredFirstMo,
      StableDistribution[1, aWU90dFM, bWU90dFM, locWU90dFM, scaleWE90dFM]]
Out[•]= StableDistribution[1, 1.42463, -0.0683195, 0.0000414126, 0.00174465]

In[•]:= (* Compare w last month *)

In[•]:= edistWethUsdc90dFilteredLastMo
Out[•]= StableDistribution[1, 1.58793, -0.0706768, -0.000101131, 0.00135632]

In[•]:= (* Similar but not completely exactly distributed. Ideally would have
     access to uncertainty in fit. scales are close tho which is good *)

In[•]:= (* TODO: Calc margin required given conditional prob breached level =>
      want to min chance get to negative value position *)

In[•]:= (* Look at probabilities of different cap values
     being breached in t number of update periods into future. We
     use 10min here as our update period from the cron *)
```

```
(* C_p = e^{mu*t + sig*(t/a)^(1/a) * F^{-1}(1-alpha)} - 1 *)

Cp[t_, alpha_, a_, b_, mu_, sig_] := Exp[mu * t +
    sig * (t / a) ^ (1 / a) * InverseCDF[StableDistribution[1, a, b, 0, 1], 1 - alpha]] - 1

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

mu[a_, loc_] := loc

alpha[t_, cp_, a_, b_, mu_, sig_] :=
 1 - CDF[StableDistribution[1, a, b, 0, 1], (Log[1 + cp] - mu * t) / (sig * (t / a) ^ (1 / a))]

(* Partial for wethusdc 90d params ... *)

edistWethUsdc90dFiltered
```

```
StableDistribution[1, 1.46465, -0.0496207, -0.000010553, 0.00154277]
```

```
mu[1.4646547203677354`, -0.000010553047315527044`]
```

```
-0.000010553
```

```
sig[1.4646547203677354`, 0.0015427669619594313`]
```

```
0.00200197
```

```
CpWethUsdc90dFiltered[t_, alpha_] := Cp[t, alpha, 1.4646547203677354`,
  -0.04962071457296463`, mu[1.4646547203677354`, -0.000010553047315527044`],
  sig[1.4646547203677354`, 0.0015427669619594313`]]

alphaWethUsdc90dFiltered[t_, cp_] := alpha[t, cp, 1.4646547203677354`,
  -0.04962071457296463`, mu[1.4646547203677354`, -0.000010553047315527044`],
  sig[1.4646547203677354`, 0.0015427669619594313`]]

(* for 10m update periods of cron, have 144 periods in 1d. *)

CpWethUsdc90dFiltered[144, 0.001]
```

```
4.55748
```

```
alphaWethUsdc90dFiltered[144, 4.55747500266425]
```

```
0.001
```

```
(* works :) *)

(* 99.9% of time on WETH/USDC, price won't increase 5.5x in a day *)

6 * 24
```

```
144
```

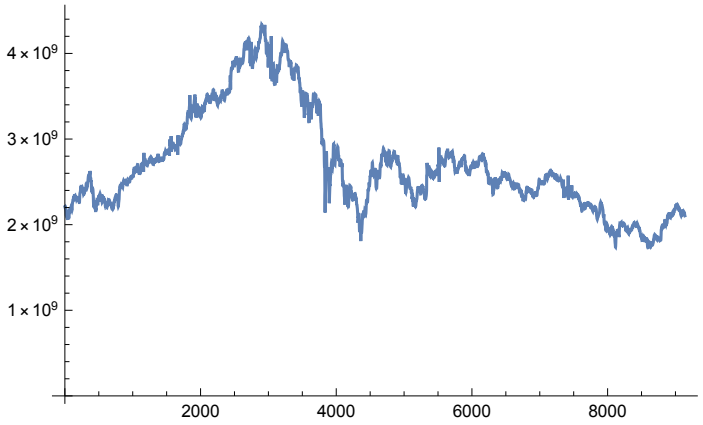
```
(* and for 7d ? *)

6 * 24 * 7
```

```
1008
```

```
In[•]:= CpWethUsdc90dFiltered[1008, 0.01]
Out[•]= 3.03498

In[•]:= (* 99% of time on WETH/USDC, price won't increase 4x in a week *)

In[•]:= (* TODO: We have two time horizons to consider:
 1. Cooldown on trading period (how often does this happen?);
2. Lookback rolling window to assess whether to cooldown ...  *)

In[•]:= (* ... 2. should be dictated by average time user is in
 a position so it's related to cap on payoff. 1. is related to
 inflation targets and OI definitive max to reach target *)
(* If we enforce Cp at certain level alpha over period t,
then we should likely have t as the lookback window *)

In[•]:= CpWethUsdc90dFiltered[1008, 0.03]
Out[•]= 1.03352

In[•]:= (* and for 14d ? *)

In[•]:= 6 * 24 * 14
Out[•]= 2016

In[•]:= CpWethUsdc90dFiltered[2016, 0.01]
Out[•]= 8.34704

In[•]:= (* So: Assuming users immediately unwind once cap hits, a) only 1% of the
   time 9.34x cap will be breached over 2 week holding span; b) only 1
 % of the time 4.03x cap will be breached over 1 week holding span *)

In[•]:= (* But we want a relationship s.t. lower alpha stable a values
 produce lower cap in OI. If greater percent of time hit the Cp cap,
counteract with lowering Coi accordingly. So e.g. if alpha
 jumps from 1% to 5% in frequency of breaching a 4x Cp cap,
expected shortfall goes 5x assuming worst case OI amount and same OI cap
 for each case. We can therefore reduce the OI cap 1/5 to be inline. *)

In[•]:= (* This gives us the relationship we want: more intense tails,
leads to more conservative markets => we take less risk by lowering OI cap *)

In[•]:= (* Then what's the function for determining alpha ? *)

In[•]:= alphaWethUsdc90dFiltered[144, 4.55747500266425]
Out[•]= 0.001

In[•]:= alphaWethUsdc90dFiltered[1008, 5]
Out[•]= 0.00679635
```

```
In[•]:= alphaWethUsdc90dFiltered[2016, 9]
Out[•]= 0.009548

In[•]:= (* 5x within 1 week only happens 0.6% of time with WETH/USDC ... seems good *)

In[•]:= (* Check fit distribution over daily and weekly
      to see whether c is correct when extend: L_1 + L_2 = L_3 *)

In[•]:= tsWethUsdc90dFiltered =
     Table[tblWethUsdc90d[[i]][[2]], {i, 40, Length[tblWethUsdc90d]}]

In[•]:= tsWethUsdc90dFiltered

In[•]:= dtsWethUsdc90dFiltered = Differences[tsWethUsdc90dFiltered]

In[•]:= (* relatively consistent with 600 intervals *)

In[•]:= (* Hourly is value every 6, Daily is value every 144 *)

In[•]:= dtsWethUsdc90dFiltered[[100]]
Out[•]= 581.

In[•]:= Length[twapsWethUsdc90dFiltered]
Out[•]= 9139

In[•]:= (* Instead of splitting month by month,
     look at data for multiple periods legnth started w first data point *)

In[•]:= ListLinePlot[twapsWethUsdc90dFiltered, PlotRange → All]
```

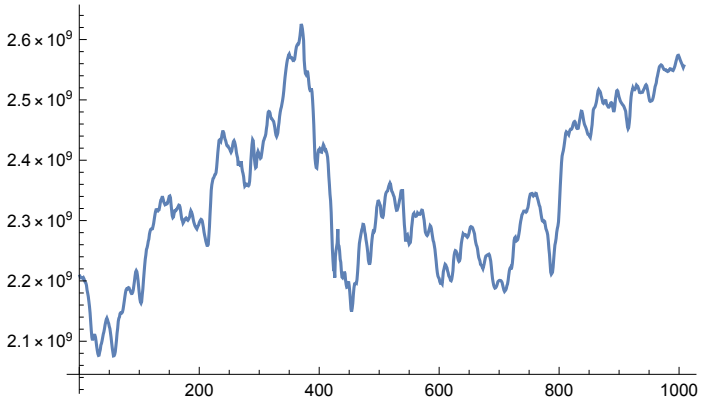
Out[•]=




```
In[•]:= twapsWethUsdc90dFiltered1Week =
     Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 6 * 24 * 7}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered1Week, PlotRange → All]
```

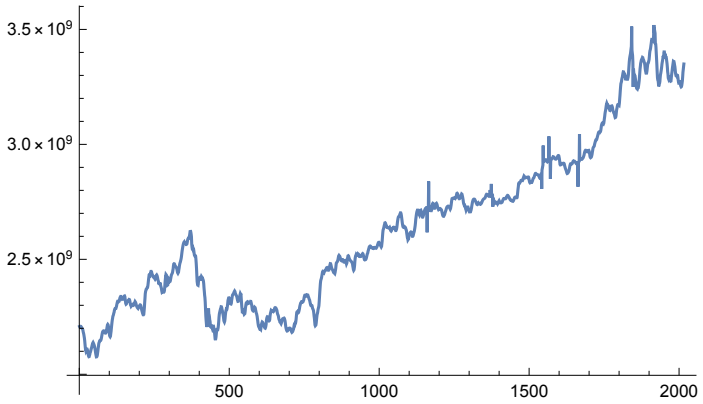
Out[•]=



In[•]:=
```
twapsWethUsdc90dFiltered2Week =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 2 * 6 * 24 * 7}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered2Week , PlotRange → All]
```

Out[•]=

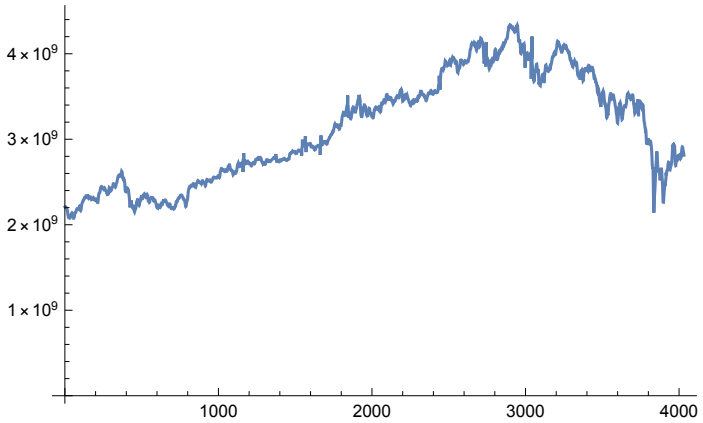


In[•]:=
```
twapsWethUsdc90dFiltered2Week =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 2 * 6 * 24 * 7}]
```

In[•]:=
```
twapsWethUsdc90dFiltered4Week =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 4 * 6 * 24 * 7}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered4Week, PlotRange → All]
```

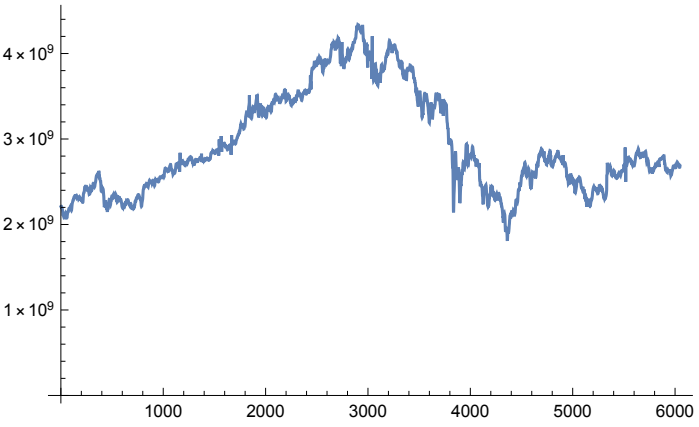
Out[•]=



In[•]:=
```
twapsWethUsdc90dFiltered6Week =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 6 * 6 * 24 * 7}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered6Week, PlotRange → All]
```

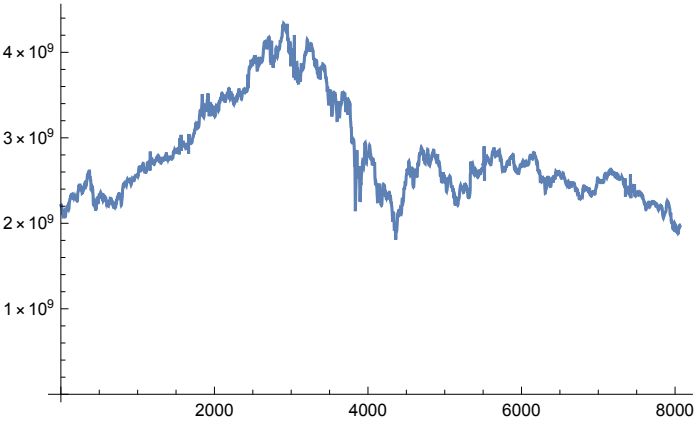
Out[•]=



In[•]:=
```
twapsWethUsdc90dFiltered8Week =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, 8 * 6 * 24 * 7}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered8Week, PlotRange → All]
```

Out[•]=

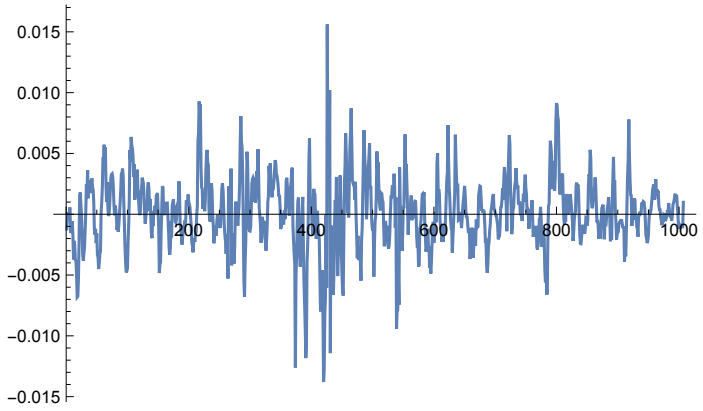


In[•]:=
```
(* Estimate fits for each span of time *)
```

In[•]:=
```
rsWethUsdc90dFiltered1Week = Differences[Log[twapsWethUsdc90dFiltered1Week]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered1Week, PlotRange → All]
```

Out[•]=



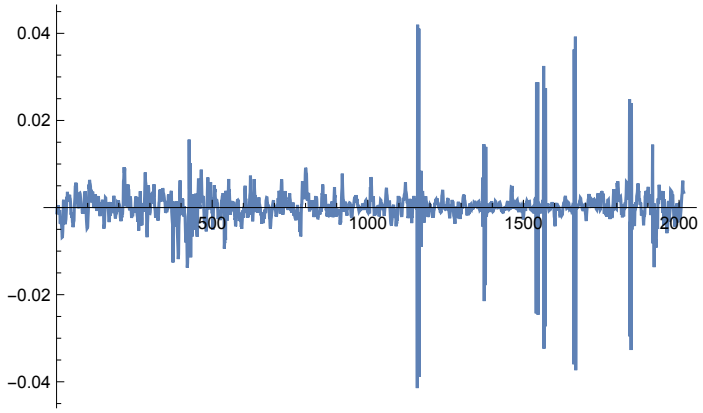
In[•]:=
```
edistWethUsdc90dFiltered1Week = EstimatedDistribution[rsWethUsdc90dFiltered1Week,
  StableDistribution[1, aWU90dF1W, bWU90dF1W, locWU90dF1W, scaleWE90dF1W]]
```

Out[•]=
```
StableDistribution[1, 1.69578, -0.00768849, 0.000175309, 0.0016727]
```

In[•]:=
```
rsWethUsdc90dFiltered2Week = Differences[Log[twapsWethUsdc90dFiltered2Week]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered2Week, PlotRange → All]
```

Out[•]=



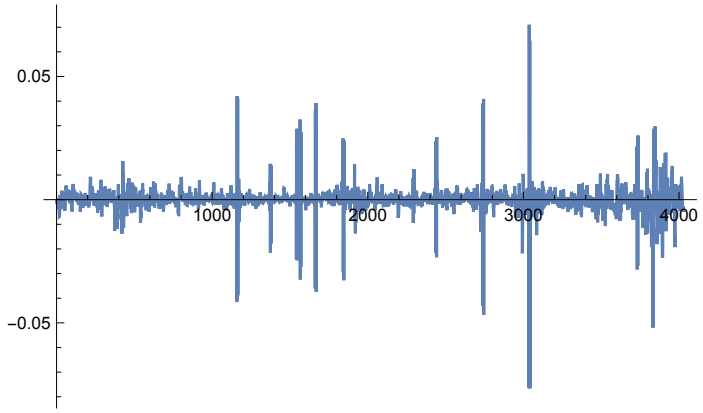
In[•]:=
```
edistWethUsdc90dFiltered2Week = EstimatedDistribution[rsWethUsdc90dFiltered2Week,
  StableDistribution[1, aWU90dF2W, bWU90dF2W, locWU90dF2W, scaleWE90dF2W]]
```

Out[•]=
```
StableDistribution[1, 1.5094, 0.0433026, 0.00027506, 0.00136471]
```

In[•]:=
```
rsWethUsdc90dFiltered4Week = Differences[Log[twapsWethUsdc90dFiltered4Week]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered4Week, PlotRange → All]
```

Out[•]=



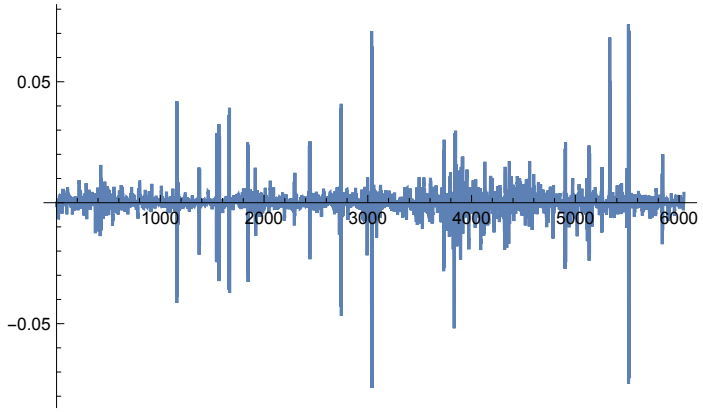
In[•]:=
```
edistWethUsdc90dFiltered4Week = EstimatedDistribution[rsWethUsdc90dFiltered4Week,
  StableDistribution[1, aWU90dF4W, bWU90dF4W, locWU90dF4W, scaleWE90dF4W]]
```

Out[•]=
```
StableDistribution[1, 1.42855, -0.0680586, 0.0000962224, 0.00156667]
```

In[•]:=
```
rsWethUsdc90dFiltered6Week = Differences[Log[twapsWethUsdc90dFiltered6Week]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered6Week, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethUsdc90dFiltered6Week = EstimatedDistribution[rsWethUsdc90dFiltered6Week,
  StableDistribution[1, aWU90dF6W, bWU90dF6W, locWU90dF6W, scaleWE90dF6W]]
```

Out[•]=
```
StableDistribution[1, 1.44156, -0.0954883, -5.04204 × 10^-6, 0.00170617]
```

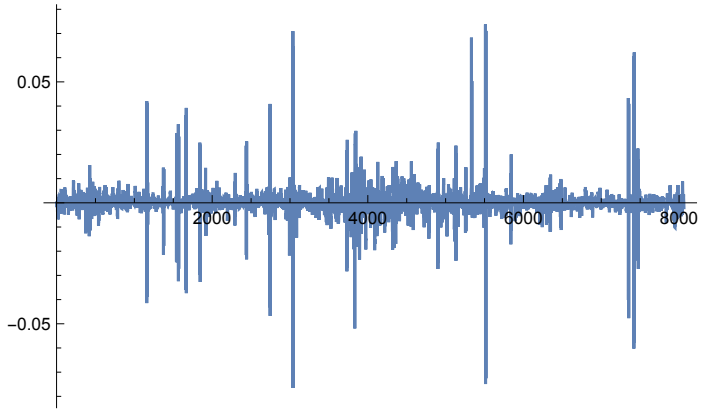
In[•]:=
```
rsWethUsdc90dFiltered8Week = Differences[Log[twapsWethUsdc90dFiltered8Week]]
```

Out[•]=
```
{-0.00120673, 0.000176452, -0.0000603568, -0.000373722, -0.000389599,
 0.000559858, -0.00092094, -0.00154871, -0.000905547, -0.000584311, -0.00172231,
 ··· 8041 ···, -0.000801922, 0.000895879, 0.00162265, 0.00187851, 0.00260868,
 0.00351812, 0.00180984, 0.00092817, 0.000511447, -0.000752754, -0.00200428}
```

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

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered8Week, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethUsdc90dFiltered8Week = EstimatedDistribution[rsWethUsdc90dFiltered8Week,
  StableDistribution[1, aWU90dF8W, bWU90dF8W, locWU90dF8W, scaleWE90dF8W]]
```

Out[•]=
```
StableDistribution[1, 1.44755, -0.0635521, -0.0000205867, 0.00156225]
```

In[•]:=
```
(* Not terrible in terms of consistency: particularly on alpha and scale,
across different time horizons. Ideally we'd have error estimates here. *)
```

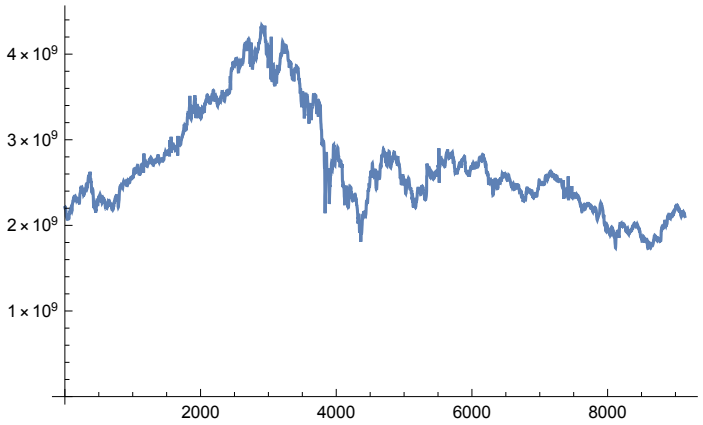
In[•]:=
```
(* Look for stationarity X_t - X_s depends on t-
 s. So want candles at different time intervals: 10m candles,
30m candles, ... And make sure e.g. scale_30m ^ alpha = 3 * scale_10m ^ alpha *)
```

In[•]:=
```
(* Use the "close" value for "candles" *)
```

In[•]:=
```
(* 10 min candles *)
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered, PlotRange → All]
```

Out[•]=



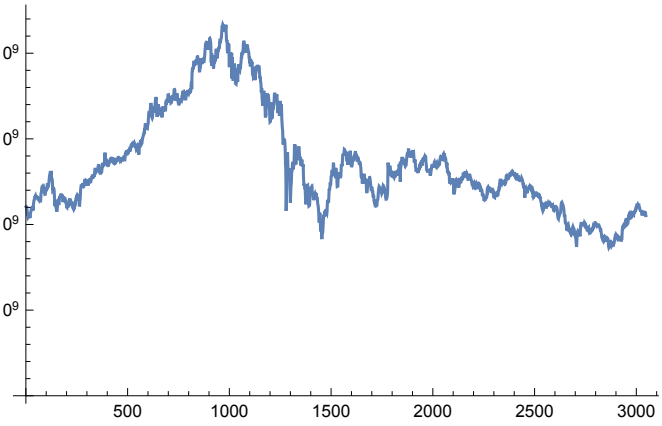
In[•]:=
```
Length[twapsWethUsdc90dFiltered]
```

Out[•]=
```
9139
```

In[•]:=
```
twapsWethUsdc90dFiltered30MinCandle =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, Length[twapsWethUsdc90dFiltered], 3}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered30MinCandle]
```

Out[•]=



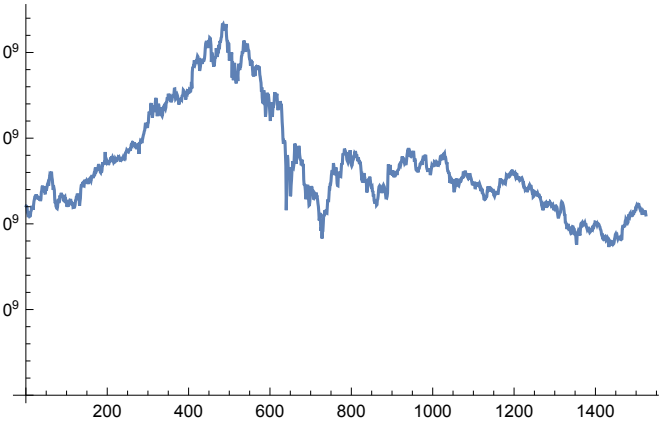
In[•]:=
```
Length[twapsWethUsdc90dFiltered30MinCandle]
```

Out[•]=
```
3047
```

In[•]:=
```
twapsWethUsdc90dFiltered1HourCandle =
 Table[twapsWethUsdc90dFiltered[[i]], {i, 1, Length[twapsWethUsdc90dFiltered], 6}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered1HourCandle]
```

Out[•]=



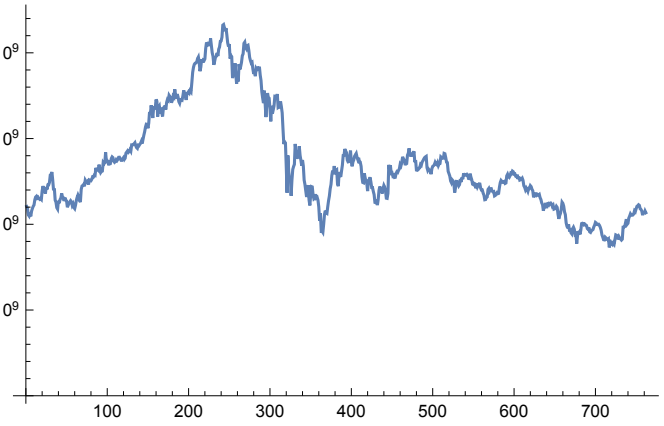
In[•]:=
```
Length[twapsWethUsdc90dFiltered1HourCandle]
```

Out[•]=
```
1524
```

In[•]:=
```
twapsWethUsdc90dFiltered2HourCandle = Table[
  twapsWethUsdc90dFiltered[[i]], {i, 1, Length[twapsWethUsdc90dFiltered], 12}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered2HourCandle, PlotRange → All]
```

Out[•]=



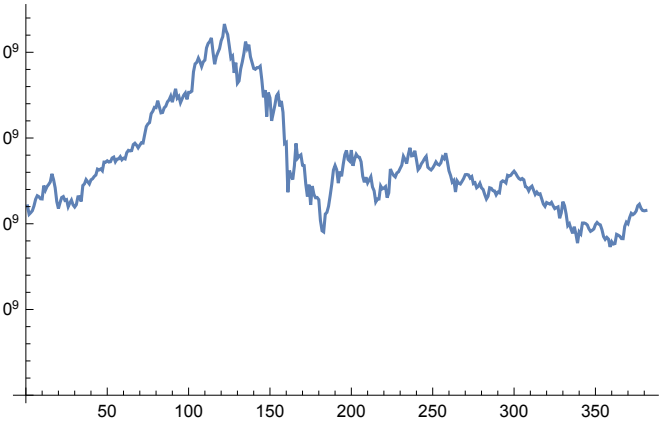
In[•]:=
```
Length[twapsWethUsdc90dFiltered2HourCandle]
```

Out[•]= 762

In[•]:=
```
twapsWethUsdc90dFiltered4HourCandle = Table[
  twapsWethUsdc90dFiltered[[i]], {i, 1, Length[twapsWethUsdc90dFiltered], 24}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered4HourCandle, PlotRange → All]
```

Out[•]=



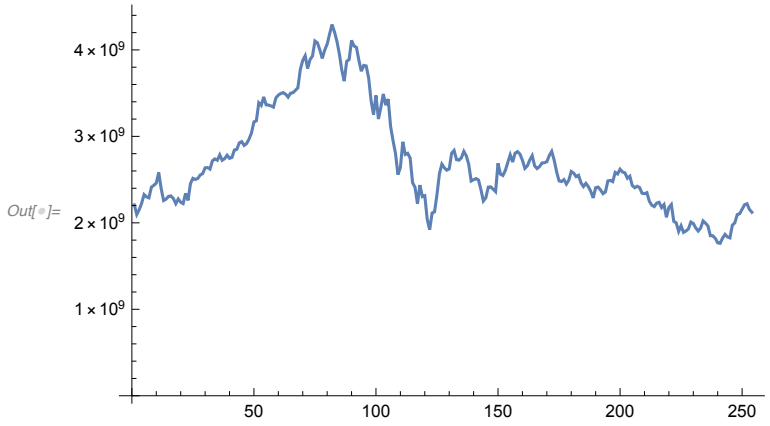
In[•]:=
```
Length[twapsWethUsdc90dFiltered4HourCandle]
```

Out[•]= 381

In[•]:=
```
twapsWethUsdc90dFiltered6HourCandle = Table[
  twapsWethUsdc90dFiltered[[i]], {i, 1, Length[twapsWethUsdc90dFiltered], 36}]
```

In[•]:=
```
ListLinePlot[twapsWethUsdc90dFiltered6HourCandle, PlotRange → All]
```

Out[•]=



In[•]:=
```
Length[twapsWethUsdc90dFiltered6HourCandle]
```

Out[•]=
```
254
```

In[•]:=
```
(* Estimate distributional params for each candle set *)
```

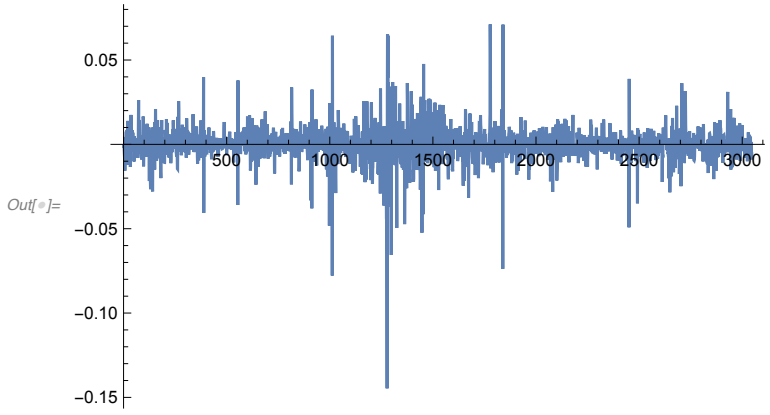
In[•]:=
```
edistWethUsdc90dFiltered
```

Out[•]=
```
StableDistribution[1, 1.46465, -0.0496207, -0.000010553, 0.00154277]
```

In[•]:=
```
rsWethUsdc90dFiltered30MinCandle =
 Differences[Log[twapsWethUsdc90dFiltered30MinCandle]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered30MinCandle, PlotRange → All]
```

Out[•]=



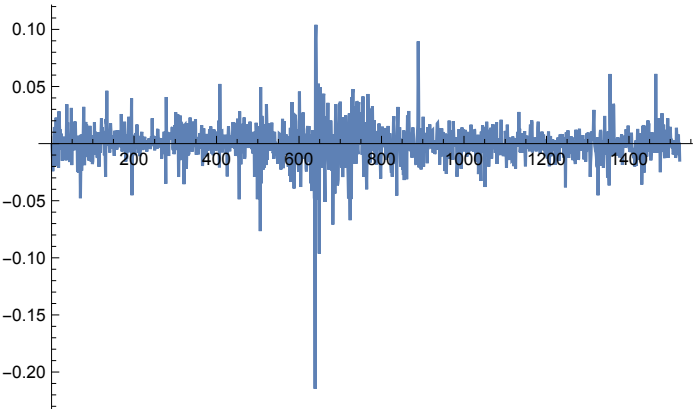
In[•]:=
```
edistWethUsdc90dFiltered30MinCandle =
 EstimatedDistribution[rsWethUsdc90dFiltered30MinCandle,
  StableDistribution[1, aWU90dC30m, bWU90dC30m, locWU90dC30m, scaleWU90dC30m]]
```

Out[•]=
```
StableDistribution[1, 1.55263, -0.0735773, -0.0000556332, 0.00439438]
```

In[•]:=
```
rsWethUsdc90dFiltered1HourCandle =
 Differences[Log[twapsWethUsdc90dFiltered1HourCandle]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered1HourCandle, PlotRange → All]
```

Out[•]=



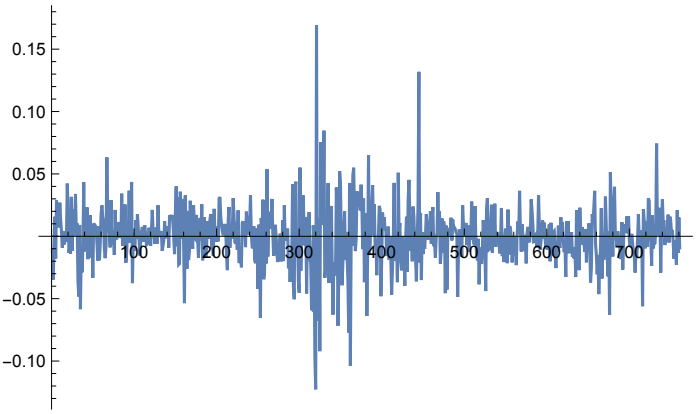
In[•]:=
```
edistWethUsdc90dFiltered1HourCandle =
 EstimatedDistribution[rsWethUsdc90dFiltered1HourCandle,
  StableDistribution[1, aWU90dC1h, bWU90dC1h, locWU90dC1h, scaleWU90dC1h]]
```

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

In[•]:=
```
rsWethUsdc90dFiltered2HourCandle =
 Differences[Log[twapsWethUsdc90dFiltered2HourCandle]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered2HourCandle, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethUsdc90dFiltered2HourCandle =
 EstimatedDistribution[rsWethUsdc90dFiltered2HourCandle,
  StableDistribution[1, aWU90dC2h, bWU90dC2h, locWU90dC2h, scaleWU90dC2h]]
```

Out[•]=
```
StableDistribution[1, 1.66239, -0.0643151, -0.000079592, 0.0126146]
```

In[•]:=
```
0.007900125269736371` * (2^(1/1.6))
```

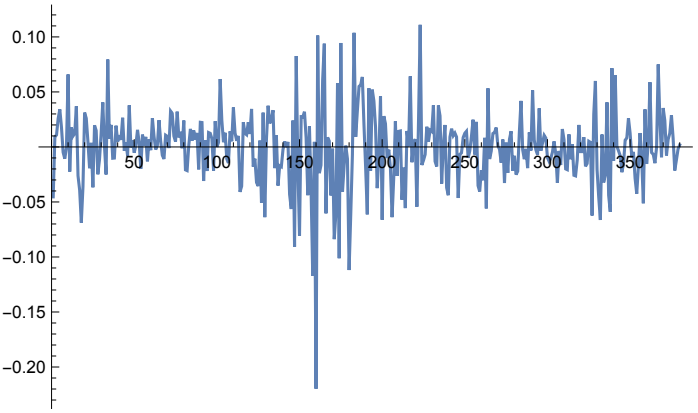
Out[•]=
```
0.0121837
```

In[•]:=
```
(* :) ... 1h → 2h looks way more stationary *)
```

In[•]:=
```
rsWethUsdc90dFiltered4HourCandle =
 Differences[Log[twapsWethUsdc90dFiltered4HourCandle]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered4HourCandle, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethUsdc90dFiltered4HourCandle =
 EstimatedDistribution[rsWethUsdc90dFiltered4HourCandle,
  StableDistribution[1, aWU90dC4h, bWU90dC4h, locWU90dC4h, scaleWU90dC4h]]
```

Out[•]=
```
StableDistribution[1, 1.56948, -0.158903, -0.000646165, 0.0176279]
```

In[•]:=
```
0.007900125269736371` * (4^(1/1.6))
```

Out[•]=
```
0.0187898
```

In[•]:=
```
0.012614644025489237` * (2^(1/1.6))
```

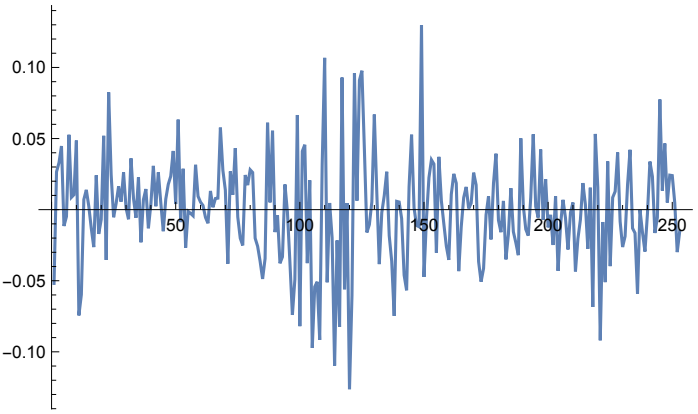
Out[•]=
```
0.0194544
```

In[•]:=
```
(* 1h → 4h looks good as well for c_long = N^(1/alpha) * c_short *)
```

In[•]:=
```
rsWethUsdc90dFiltered6HourCandle =
 Differences[Log[twapsWethUsdc90dFiltered6HourCandle]]
```

In[•]:=
```
ListLinePlot[rsWethUsdc90dFiltered6HourCandle, PlotRange → All]
```

Out[•]=



In[•]:=
```
edistWethUsdc90dFiltered6HourCandle =
 EstimatedDistribution[rsWethUsdc90dFiltered6HourCandle,
  StableDistribution[1, aWU90dC6h, bWU90dC6h, locWU90dC6h, scaleWU90dC6h]]
```

Out[•]=
```
StableDistribution[1, 1.73971, -0.0744163, -0.000470773, 0.0227901]
```

```
In[•]:= (* Not a good alpha here for 6h candle ... (not enough data?) *)

In[•]:= 0.007900125269736371` / (6^(1/1.6))
Out[•]= 0.00257804

In[•]:= edistWethUsdc90dFiltered
Out[•]= StableDistribution[1, 1.46465, -0.0496207, -0.000010553, 0.00154277]

In[•]:= edistWethUsdc90dFiltered30MinCandle
Out[•]= StableDistribution[1, 1.55263, -0.0735773, -0.0000556332, 0.00439438]

In[•]:= edistWethUsdc90dFiltered1HourCandle
Out[•]= StableDistribution[1, 1.59768, -0.0971292, -0.000157254, 0.00790013]

In[•]:= edistWethUsdc90dFiltered2HourCandle
Out[•]= StableDistribution[1, 1.66239, -0.0643151, -0.000079592, 0.0126146]

In[•]:= edistWethUsdc90dFiltered4HourCandle
Out[•]= StableDistribution[1, 1.56948, -0.158903, -0.000646165, 0.0176279]

In[•]:= edistWethUsdc90dFiltered6HourCandle
Out[•]= StableDistribution[1, 1.73971, -0.0744163, -0.000470773, 0.0227901]

(* Beta should linearly add up like loc as
 increase candle size if not consistent with zero. Unclear,
but not behaving according to linear combo props of stable dist as nicely as
 scale is: so maybe zero? Enable fitting in pystable that fixes params? *)

In[•]:= 0.007900125269736371` / (2^(1/1.6))
Out[•]= 0.0051226

In[•]:= (* Actually the scale down from 1h → 30m isn't terrible as well. *)

(* Seems a good place to estimate k values would be 1h candles ... see note-
 4.nb for k analysis there *)
```