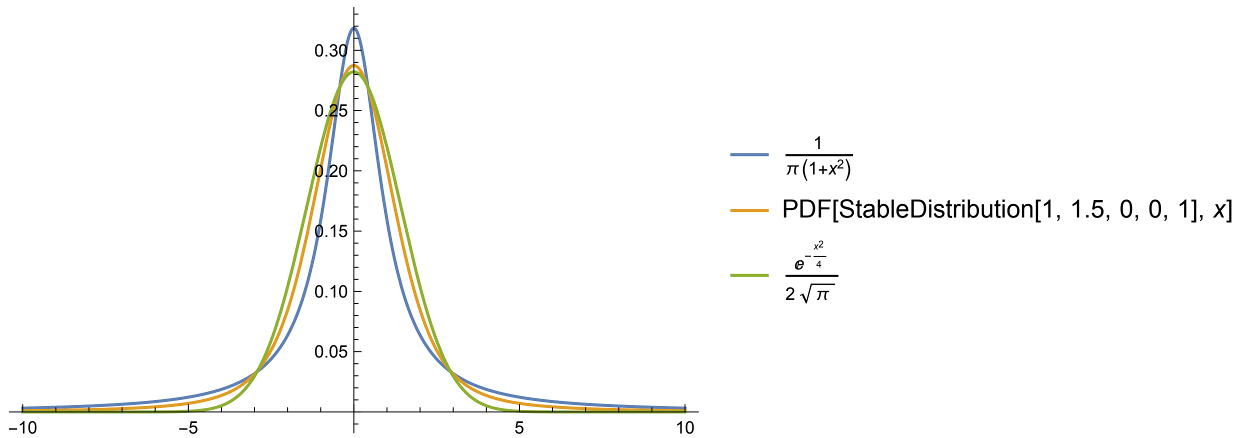
In[1]:=
```
(* Look at stable distribution and infer what
 best payoff cap is based on underlying distr params *)
```

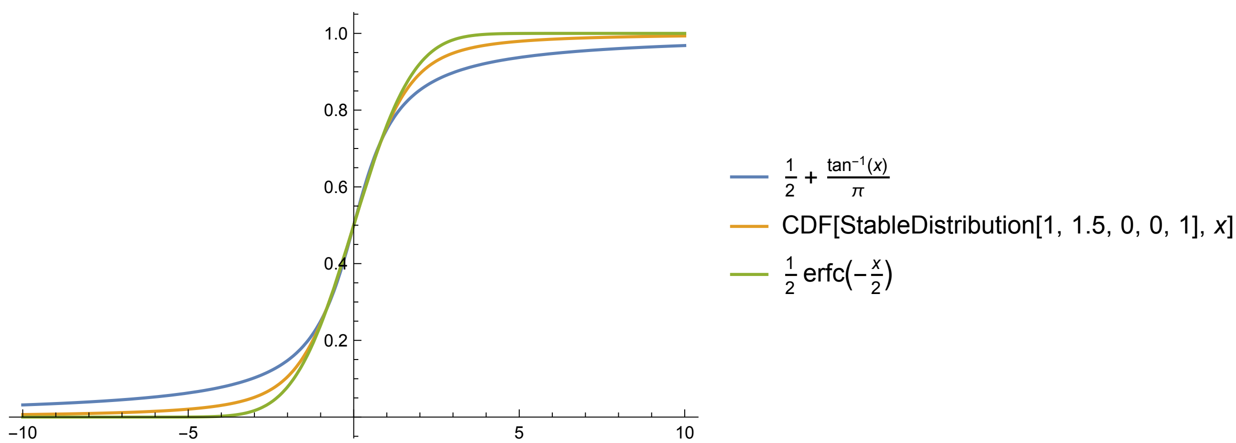
In[2]:=
```
Plot[Table[PDF[StableDistribution[1, a, 0, 0, 1], x], {a, {1.0, 1.5, 2.0}}] //
  Evaluate, {x, -10, 10}, PlotLegends → "Expressions"]
```

Out[2]=




In[3]:=
```
Plot[Table[CDF[StableDistribution[1, a, 0, 0, 1], x], {a, {1.0, 1.5, 2.0}}] //
  Evaluate, {x, -10, 10}, PlotLegends → "Expressions"]
```

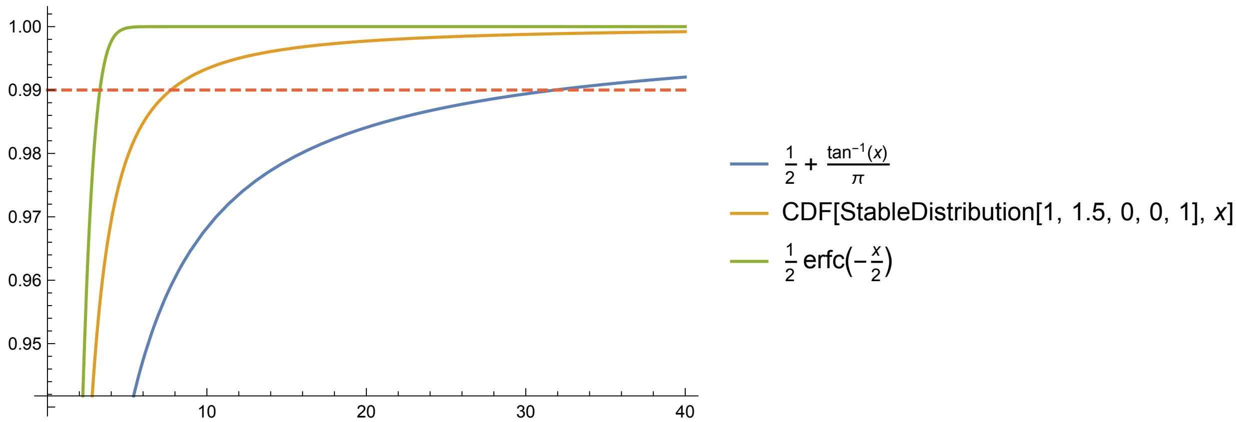
Out[3]=




In[4]:=
```
(* For alpha of 0.05, can essentially put cap on x as CDF
 (1-alpha). Which ultimately relates payoff cap to VaR metric *)
```

In[5]:=
```
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[5]=




In[6]:=
```
InverseCDF[StableDistribution[1, 2, 0, 0, 1], 0.99]
```

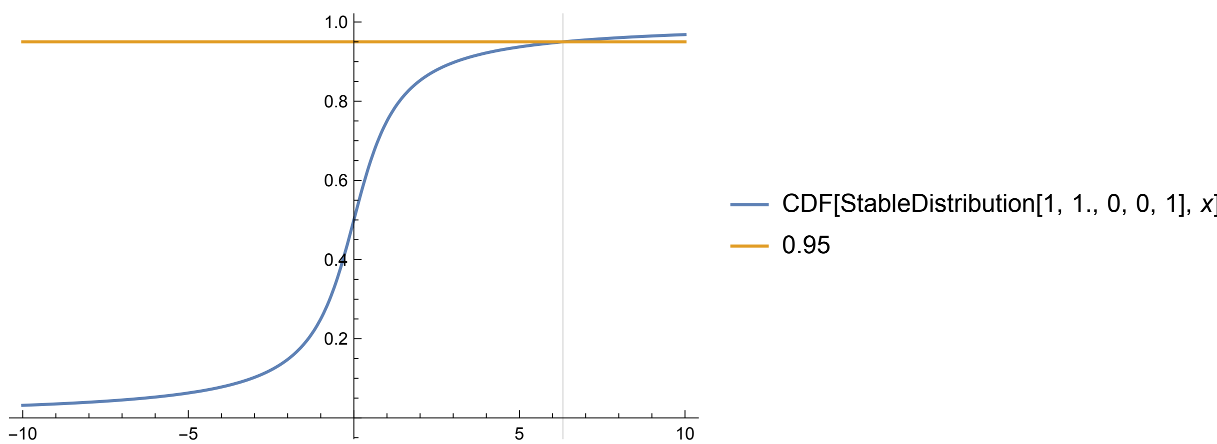
Out[6]= 3.28995

In[7]:=
```
InverseCDF[StableDistribution[1, 1, 0, 0, 1], 0.99]
```

Out[7]= 31.8205

In[8]:=
```
Plot[{CDF[StableDistribution[1, 1.0, 0, 0, 1], x], 0.95}, {x, -10, 10},
  GridLines → {{6.313751514675042`}, {}}, PlotLegends → "Expressions"]
```

Out[8]=

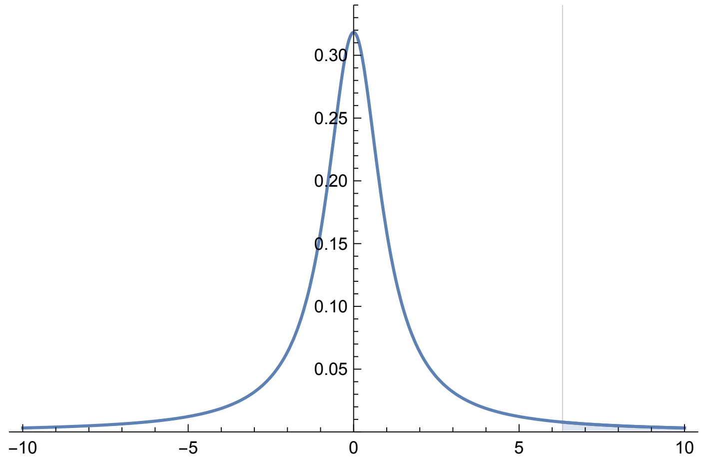



In[9]:=
```
Solve[CDF[StableDistribution[1, 1.0, 0, 0, 1], x] == 0.95, x]
```

Out[9]= {{x → 6.31375}}

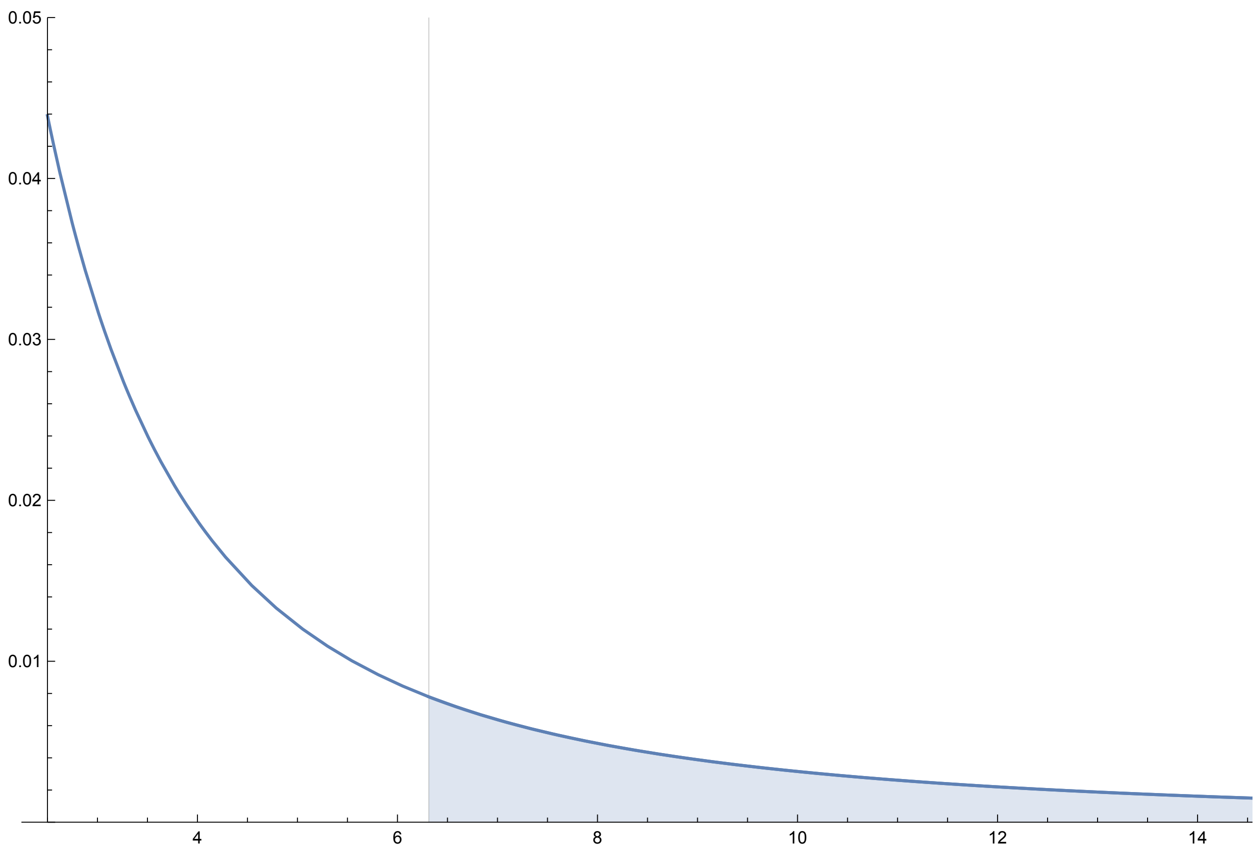
In[10]:=
```
Show[Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]}, {x, -10, 10},
  GridLines → {{6.313751514675042`}, {}}, PlotLegends → "Expressions",
  PlotRange → {0, 0.34}], Plot[{PDF[StableDistribution[1, 1.0, 0, 0, 1], x]},
  {x, 6.313751514675042`, 10}, PlotRange → {0, 0.34}, Filling → Axis]]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

Out[10]=




In[11]:=
```
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]]
```

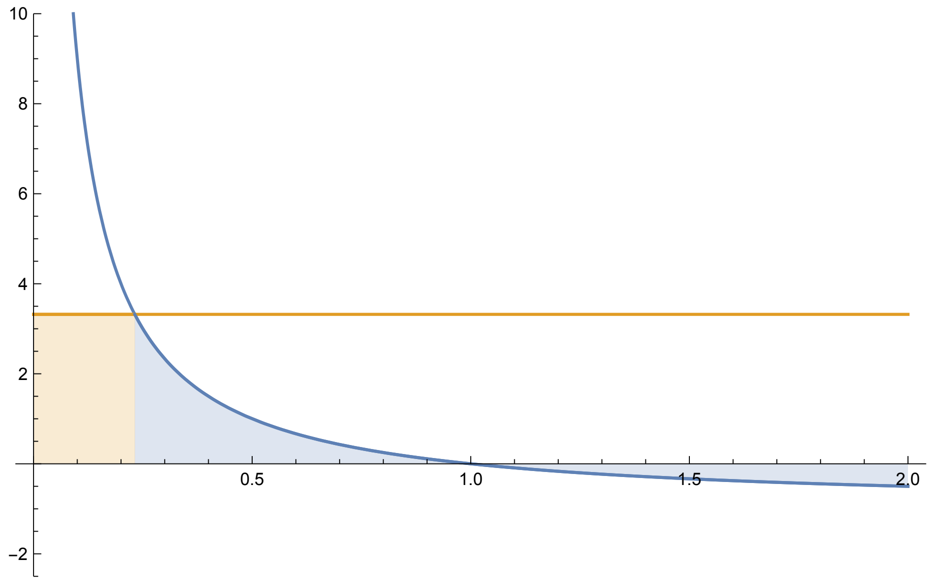
Out[11]=



In[12]:=
```
(* 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[13]:=
```
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[13]=




In[14]:=
```
1 / (Exp[1.2] + 1)
```

Out[14]= 0.231475

In[15]:=
```
Exp[1.2]
```

Out[15]= 3.32012

In[16]:=
```
Show[%86, AxesLabel → {HoldForm[P], HoldForm[PnL]},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

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

Out[16]= Show[%86, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]

In[17]:=
```
Exp[1.2]
```

Out[17]= 3.32012

In[18]:=
```
1 / (Exp[1.2] + 1)
```

Out[18]= 0.231475

In[19]:=
```
(* 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[20]:=
```
Exp[10^(-7) * 6.313751514675042 * 144 * 7]
```

Out[20]= 1.00064

In[21]:=
```
Log[10]
```

Out[21]= Log[10]

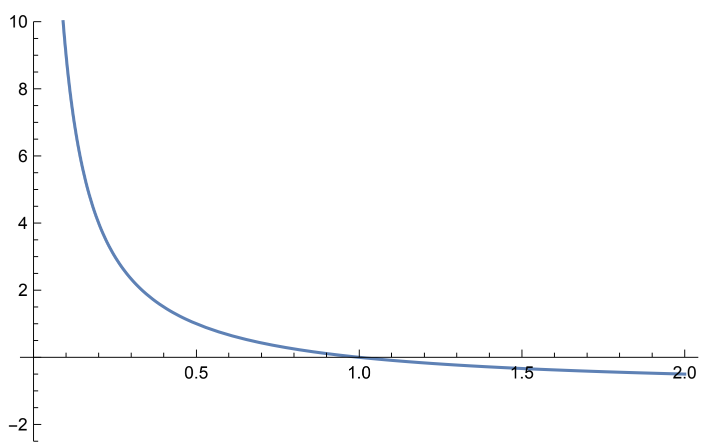
In[22]:=
```
N[Log[10]]
```

Out[22]= 2.30259

In[23]:=
```
(* Inverse market payoff *)
```

In[24]:=
```
Plot[{1 / x - 1}, {x, 0, 2}, GridLines → {{}, {}},
 PlotLegends → "Expressions", PlotRange → {10, -2.5}]
```

Out[24]=



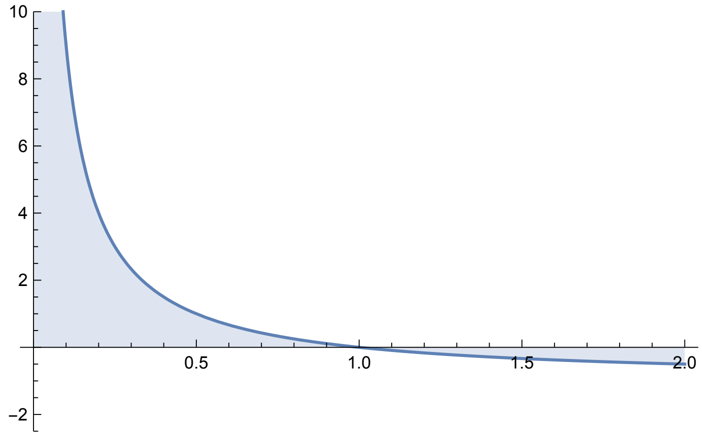

In[25]:=
```
Show[%87, AxesLabel → {HoldForm[P], HoldForm[PnL]},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Show: Out is not a type of graphics.

Out[25]= Show[%87, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]

In[26]:=
```
Plot[{1 / x - 1}, {x, 0, 2}, GridLines → {{}, {}},
 PlotLegends → "Expressions", PlotRange → {10, -2.5}, Filling → Axis]
```

Out[26]=




In[27]:=
```
Show[%93, AxesLabel → {HoldForm[P], HoldForm[PnL]},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

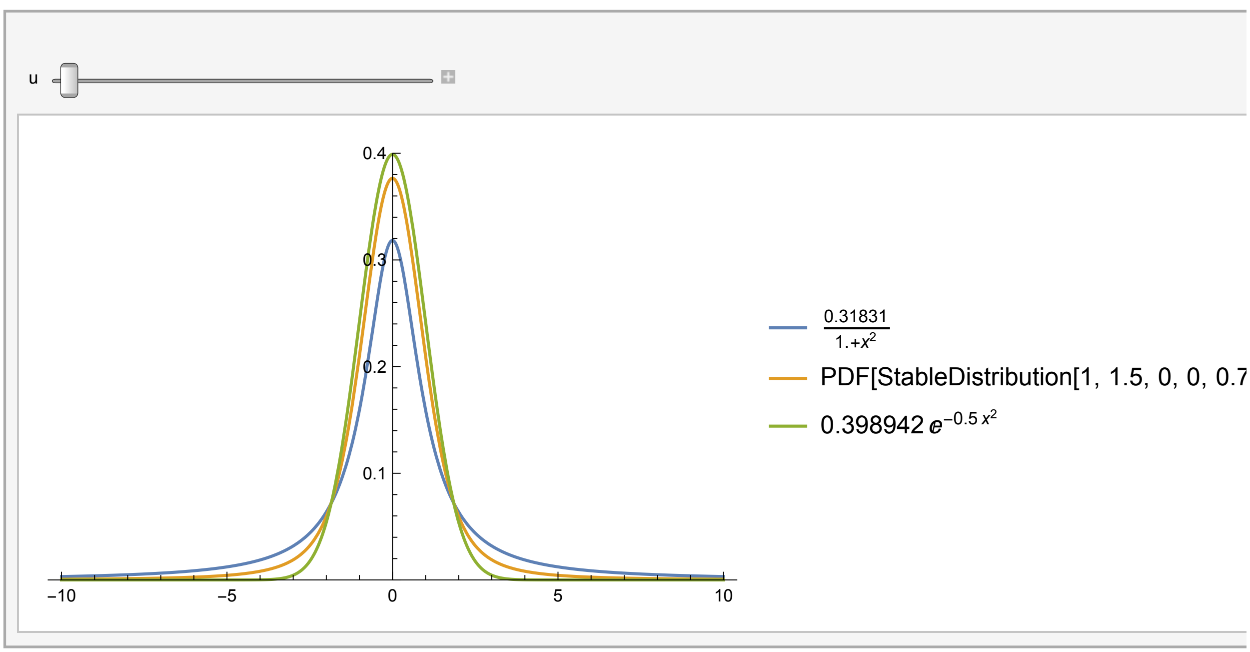
Show: Out is not a type of graphics.

Out[27]=
```
Show[%93, AxesLabel → {P, PnL}, PlotLabel → None, LabelStyle → {■}]
```

In[28]:=
```
(* Look at how PDF,
CDF evolve over time with e**(μ*t + σ*L_t) Levy stable increments *)
```

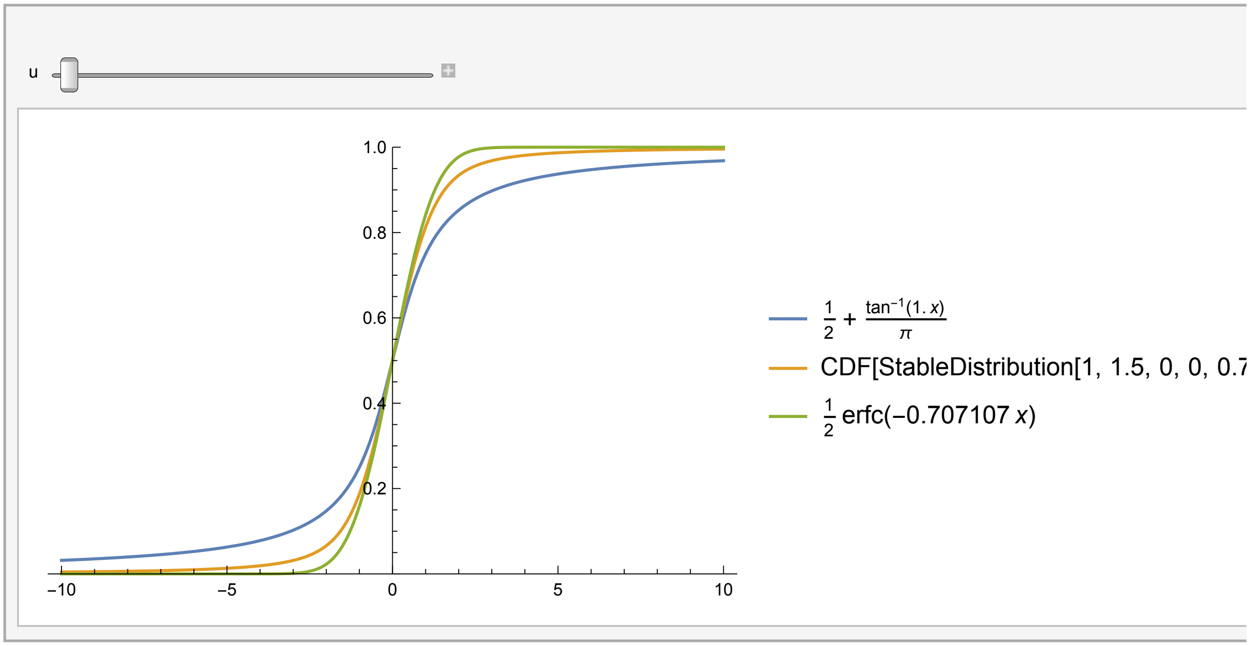
In[29]:=
```
Manipulate[Plot[Table[PDF[StableDistribution[1, a, 0, 0, (u / a) ^ (1 / a)], x],
    {a, {1.0, 1.5, 2.0}}] // Evaluate, {x, -10, 10},
  PlotRange → {0, 0.4}, PlotLegends → "Expressions"], {u, 1, 10}]
```

Out[29]=



In[30]:=
```
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[30]=




In[31]:=
```
1 / Sqrt[2 * Pi]
```

Out[31]= $\frac{1}{\sqrt{2\,\pi}}$

In[32]:= $N\left[\frac{1}{\sqrt{2\,\pi}}\right]$

Out[32]= 0.398942

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

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

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

Out[35]= /Users/personal

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

Out[36]= /Users/personal/Desktop/note7/points

In[37]:=
```
tblWethDai = Import["30/data-1624665179_weth-dai-twap.csv"]
```

Out[37]=

$\{\{, \text{timestamp}, \text{twap}\}, \{0, 1.62208 \times 10^9, 2823283256860140896256\},$
$\{1, 1.62208 \times 10^9, 2807416500389253480448\},$
$\{2, 1.62208 \times 10^9, 2795741643020610568192\},$
$\{3, 1.62208 \times 10^9, 2778509477732340465664\},$
$\{4, 1.62208 \times 10^9, 2767564848768069140480\}, \cdots 3738 \cdots,$
$\{3743, 1.62466 \times 10^9, 1840325819329555202048\},$
$\{3744, 1.62466 \times 10^9, 1832467229514366976000\},$
$\{3745, 1.62466 \times 10^9, 1825074114325430403072\},$
$\{3746, 1.62466 \times 10^9, 1817085838281425027072\},$
$\{3747, 1.62466 \times 10^9, 1810494773684469760000\},$
$\{3748, 1.62466 \times 10^9, 1810804556505733136384\}\}$

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

In[38]:=
```
Length[tblWethDai]
```

Out[38]= 3750

In[39]:=
```
tblWethDai[[2]]
```

Out[39]= $\{0, 1.62208 \times 10^9, 2823283256860140896256\}$

In[40]:=
```
tblWethDai[[2]][[2]]
```

Out[40]= $1.62208 \times 10^9$

In[41]:=
```
timesWethDai = Table[tblWethDai[[i]][[2]], {i, 2, Length[tblWethDai]}]
```

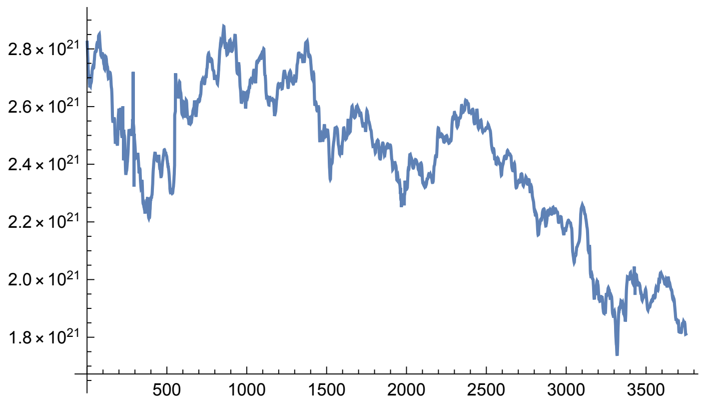
In[42]:=
```
twapsWethDai = Table[tblWethDai[[i]][[3]], {i, 2, Length[tblWethDai]}]
```

In[43]:=
```
twapsWethDai[[1]]
```

Out[43]= 2 823 283 256 860 140 896 256

In[44]:=
```
ListLinePlot[twapsWethDai, PlotRange → All]
```

Out[44]=




In[45]:=
```
rsWethDai = Differences[Log[twapsWethDai]]
```

Out[45]=
```
{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], ··· 3740 ···,
 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...

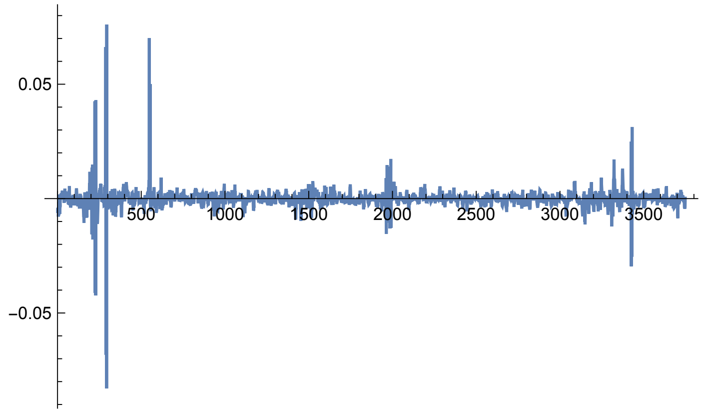
In[46]:=
```
rsWethDai[[100]]
```

Out[46]=
```
Log[2 770 690 494 359 044 358 144] - Log[2 778 801 122 423 983 833 088]
```

In[47]:=
```
N[Log[2 770 690 494 359 044 358 144] - Log[2 778 801 122 423 983 833 088]]
```

Out[47]=
```
-0.00292302
```

In[48]:= 
```
ListLinePlot[rsWethDai, PlotRange → All]
```

Out[48]=



In[49]:= 
```
edistWethDai = EstimatedDistribution[rsWethDai,
  StableDistribution[1, aWethDai, bWethDai, locWethDai, scaleWethDai]]
```

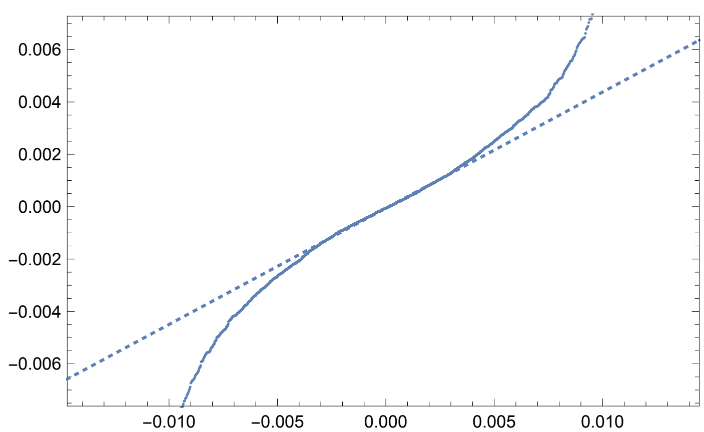
Out[49]= 
```
StableDistribution[1, 1.60712, -0.108702, -0.000176324, 0.00134358]
```

In[50]:= 
```
FindDistributionParameters[rsWethDai,
 StableDistribution[1, aWethDaif, bWethDaif, locWethDaif, scaleWethDaif]]
```

Out[50]= 
```
{aWethDaif → 1.60712, bWethDaif → -0.108702,
 locWethDaif → -0.000176324, scaleWethDaif → 0.00134358}
```

In[51]:= 
```
(* Not normal ... *)
```

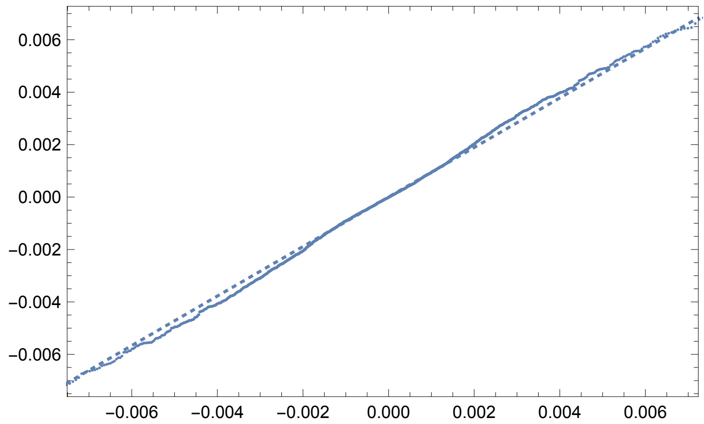
In[52]:= 
```
QuantilePlot[rsWethDai]
```

Out[52]=



In[53]:= 
```
(* Stable fit looks good for sample ... *)
```

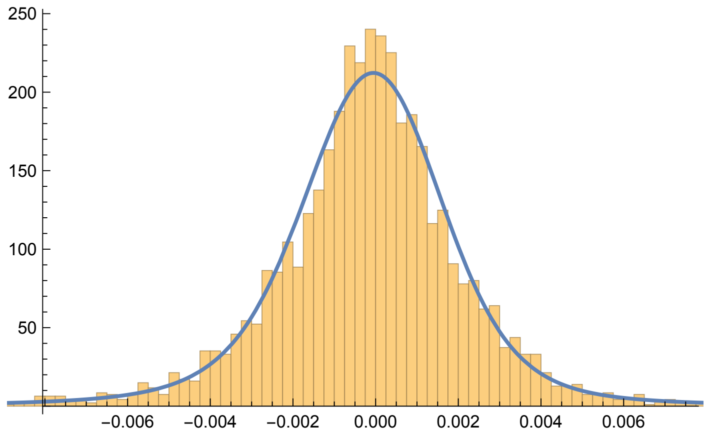
In[54]:=
```
QuantilePlot[rsWethDai, edistWethDai]
```

Out[54]=



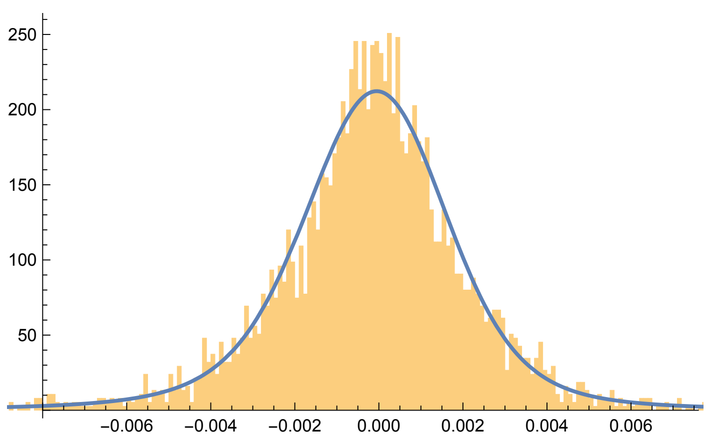

In[55]:=
```
Show[Histogram[rsWethDai, {0.00025}, "PDF"], Plot[PDF[edistWethDai, x],
  {x, Min[rsWethDai], Max[rsWethDai]}, PlotRange → All, PlotStyle → Thick]]
```

Out[55]=




In[56]:=
```
Show[Histogram[rsWethDai, {0.0001}, "PDF"], Plot[PDF[edistWethDai, x],
  {x, Min[rsWethDai], Max[rsWethDai]}, PlotRange → All, PlotStyle → Thick]]
```

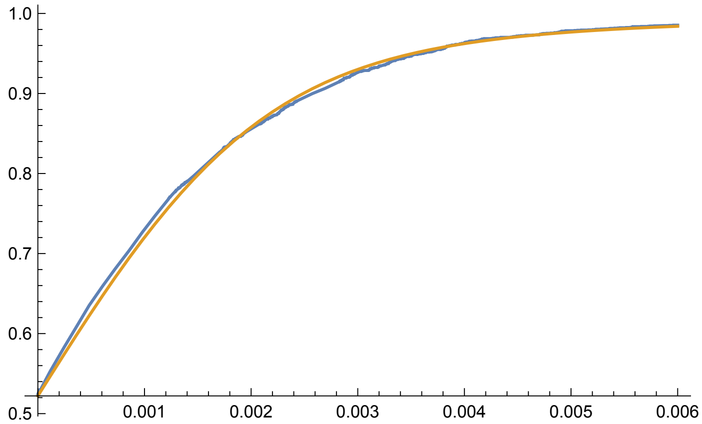
Out[56]=




In[57]:=
```
(* Compare the CDFs ... *)
```

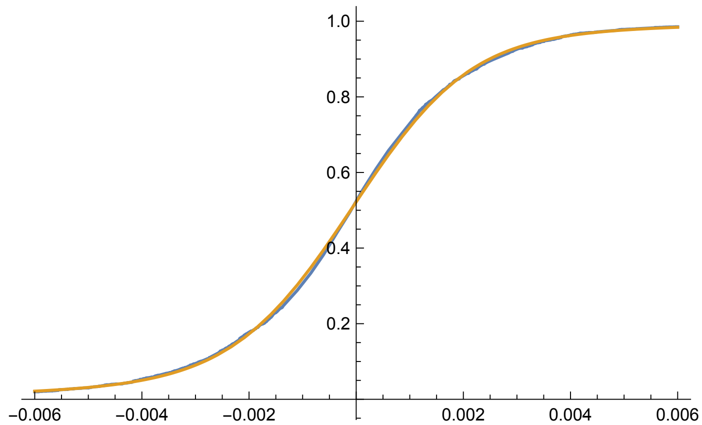
In[58]:=
```
Plot[{CDF[EmpiricalDistribution[rsWethDai], x], CDF[edistWethDai, x]}, {x, 0, 0.006}]
```

Out[58]=




In[59]:=
```
Plot[{CDF[EmpiricalDistribution[rsWethDai], x], CDF[edistWethDai, x]},
 {x, -0.006, 0.006}]
```

Out[59]=




In[60]:=
```
(* Look at histogram behavior *)
```

In[61]:=
```
rsWethDai
```

Out[61]=
```
{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], ··· 3740 ···,
 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[62]:=
```
Length[rsWethDai]
```
Out[62]= 3748

In[63]:=
```
(* N rsWethDai obs > 0.001 *)
fGt001 = # > 0.001 &;
rsWethDaiGt001 = Select[rsWethDai, fGt001]
```
In[65]:=
```
N[Length[rsWethDaiGt001] / Length[rsWethDai]]
```
Out[65]= 0.268943

In[66]:=
```
(* N rsWethDai obs > 0.002 *)
fGt002 = # > 0.002 &;
rsWethDaiGt002 = Select[rsWethDai, fGt002]
```
In[68]:=
```
N[Length[rsWethDaiGt002] / Length[rsWethDai]]
```
Out[68]= 0.14461

In[69]:=
```
(* N rsWethDai obs > 0.004 *)
fGt004 = # > 0.004 &;
rsWethDaiGt004 = Select[rsWethDai, fGt004]
```
In[71]:=
```
N[Length[rsWethDaiGt004] / Length[rsWethDai]]
```
Out[71]= 0.0368196

In[72]:=
```
(* N rsWethDai obs > 0.008 *)
fGt008 = # > 0.008 &;
rsWethDaiGt008 = Select[rsWethDai, fGt008]
```
In[74]:=
```
N[Length[rsWethDaiGt008] / Length[rsWethDai]]
```
Out[74]= 0.00773746

In[75]:=
```
(* N rsWethDai obs > 0.016 *)
fGt016 = # > 0.016 &;
rsWethDaiGt016 = Select[rsWethDai, fGt016]
```
Out[76]=
```
{-Log[2 492 649 337 524 945 682 432] + Log[2 600 896 358 324 474 216 448],
 -Log[2 413 592 805 809 920 671 744] + Log[2 519 753 151 904 606 584 832],
 -Log[2 546 001 124 171 610 849 280] + Log[2 720 404 121 477 766 447 104],
 -Log[2 322 151 657 675 870 961 664] + Log[2 505 792 289 942 667 788 288],
 -Log[2 395 965 941 133 844 938 752] + Log[2 570 155 672 313 468 551 168],
 -Log[2 582 923 918 759 191 117 824] + Log[2 715 565 447 663 336 816 640],
 -Log[2 302 931 843 464 126 791 680] + Log[2 343 196 769 077 410 398 208],
 -Log[1 786 647 008 628 442 660 864] + Log[1 817 487 287 292 930 555 904],
 -Log[1 994 839 663 308 514 000 896] + Log[2 045 194 922 429 345 431 552],
 -Log[1 946 835 571 456 962 985 984] + Log[2 008 760 898 207 868 780 544]}
```

```
In[77]:= N[Length[rsWethDaiGt016] / Length[rsWethDai]]
Out[77]= 0.00266809

In[78]:= 0.007737459978655283` / 0.0026680896478121665`
Out[78]= 2.9

In[79]:= 0.0368196371398079` / 0.007737459978655283`
Out[79]= 4.75862

In[80]:= 0.14461045891141944` / 0.0368196371398079`
Out[80]= 3.92754

In[81]:= (* 10 min periods ... *)

In[82]:= period = 600
Out[82]= 600

In[83]:= aWethDai = 1.6071199346185128`
Out[83]= 1.60712

In[84]:= bWethDai = -0.10870229617829233`
Out[84]= -0.108702

In[85]:= locWethDai = -0.00017632373873260288`
Out[85]= -0.000176324

In[86]:= scaleWethDai = 0.0013435809449464948`
Out[86]= 0.00134358

In[87]:= μWethDai = locWethDai / period
Out[87]= -2.93873 × 10^-7

In[88]:= σWethDai = scaleWethDai / (period / aWethDai) ^ (1 / aWethDai)
Out[88]= 0.0000337149

In[89]:= (* Now repeat for more volatile pair like ALCX/WETH ... *)
```

In[90]:=
```
tblAlcxWeth = Import["30/data-1624665179_alcx_weth-twap.csv"]
```

Out[90]=
```
{{, 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...

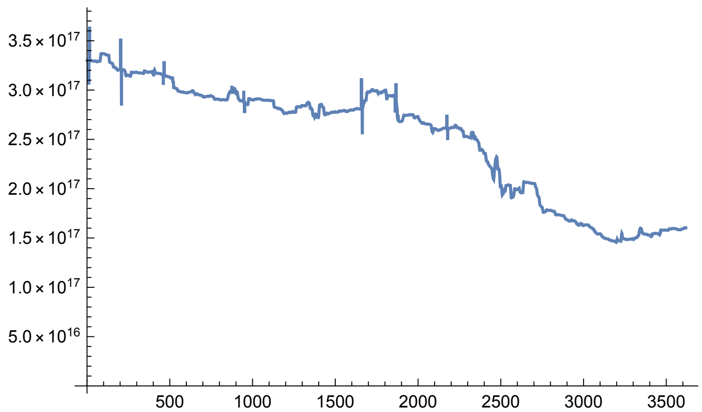
In[91]:=
```
twapsAlcxWeth = Table[tblAlcxWeth[[i]][[3]], {i, 2, Length[tblAlcxWeth]}]
```

In[92]:=
```
twapsAlcxWeth[[100]]
```

Out[92]= $3.36841 \times 10^{17}$

In[93]:=
```
ListLinePlot[twapsAlcxWeth, PlotRange → All]
```

Out[93]=



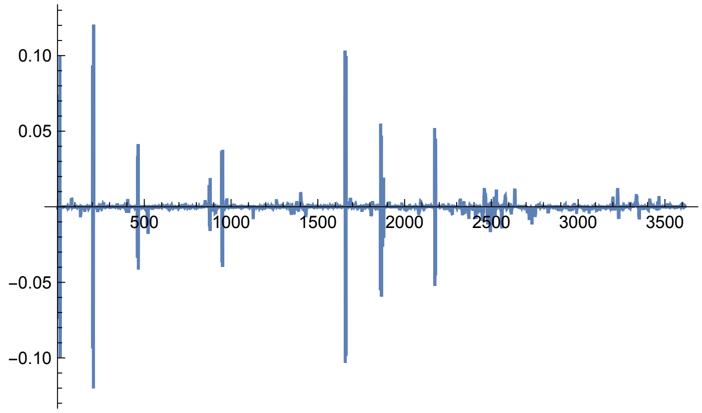

In[94]:=
```
rsAlcxWeth = Differences[Log[twapsAlcxWeth]]
```

In[95]:=
```
rsAlcxWeth[[100]]
```

Out[95]= $-9.20679 \times 10^{-6}$

In[96]:=
```
ListLinePlot[rsAlcxWeth, PlotRange → All]
```

Out[96]=

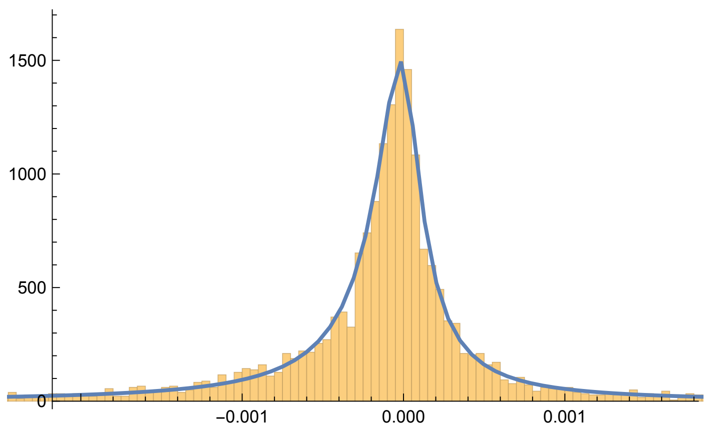



In[97]:=
```
edistAlcxWeth = EstimatedDistribution[
  rsAlcxWeth, StableDistribution[1, aAW, bAW, locAW, scaleAW]]
```

Out[97]=
```
StableDistribution[1, 0.838474, -0.181726, 0.00011192, 0.000231752]
```

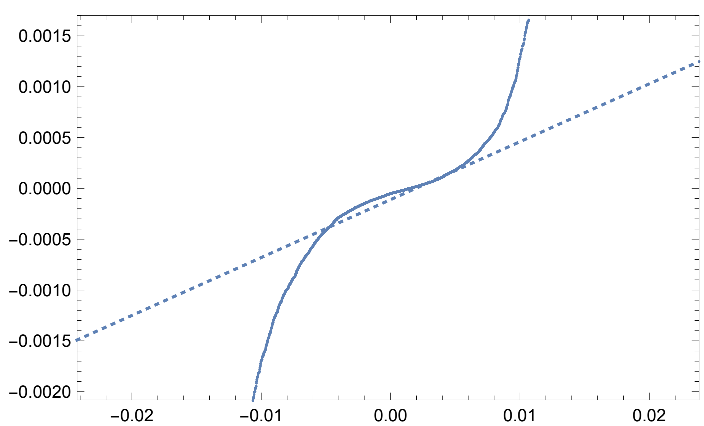
In[98]:=
```
Show[Histogram[rsAlcxWeth, {0.00005}, "PDF"], Plot[PDF[edistAlcxWeth, x],
  {x, Min[rsAlcxWeth], Max[rsAlcxWeth]}, PlotRange → All, PlotStyle → Thick]]
```

Out[98]=



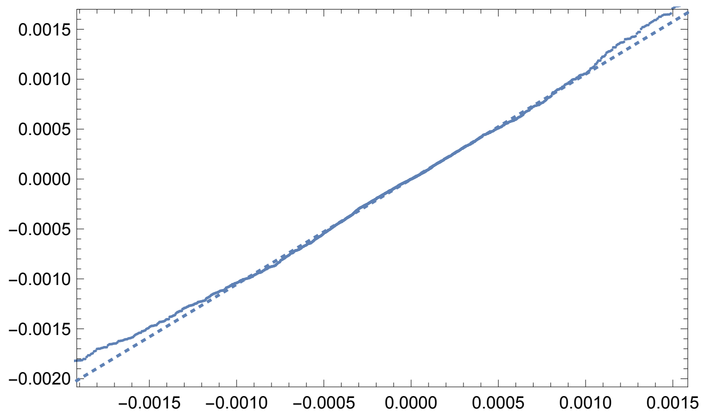

In[99]:=
```
QuantilePlot[rsAlcxWeth]
```

Out[99]=



In[100]:=
```
QuantilePlot[rsAlcxWeth, edistAlcxWeth]
```

Out[100]=




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

In[102]:=
```
(* And just to go through them all, look at UNI/WETH ... How bad? *)
```

In[110]:=
```
tblUniWeth = Import["30/data-1624665179_uni_weth-twap.csv"]
```

Out[110]=

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

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

In[111]:=
```
twapsUniWeth = Table[tblUniWeth[[i]][[3]], {i, 2, Length[tblUniWeth]}]
```

In[120]:=
```
twapsUniWeth[[100]] / 10^18 * tblWethDai[[100]][[3]] / 10^18
```

Out[120]= 28.8615

In[143]:=
```
FromUnixTime[tblUniWeth[[2]][[2]]]
```

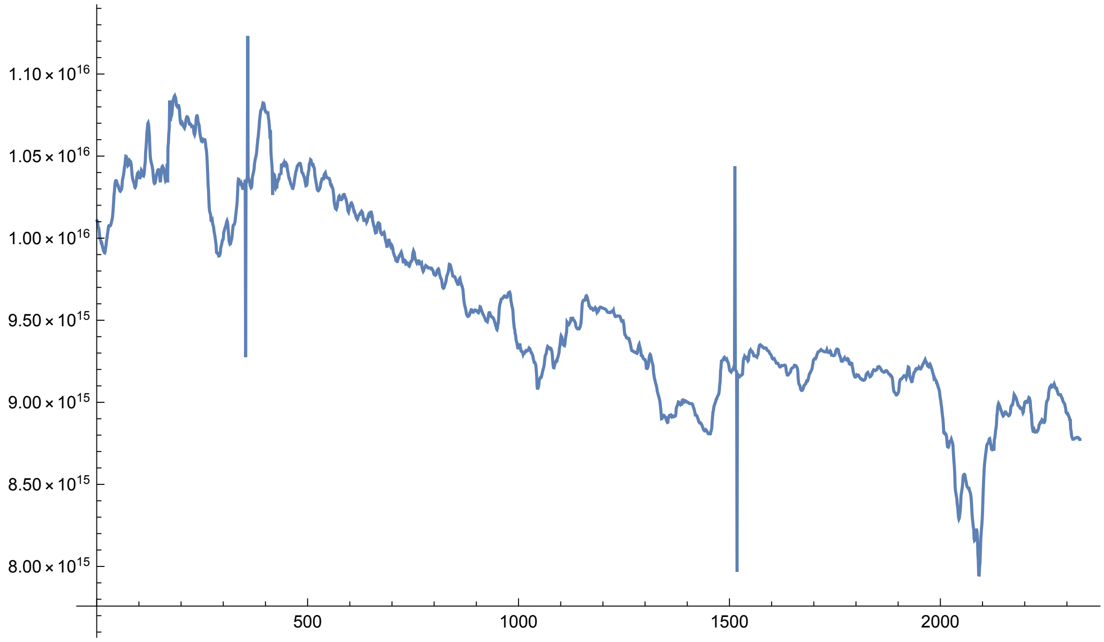
Out[143]= Wed 26 May 2021 21:04:01 GMT-4.

In[146]:=
```
FromUnixTime[tblUniWeth[[Length[tblUniWeth]]][[2]]]
```

Out[146]= Fri 25 Jun 2021 19:49:12 GMT-4.

In[121]:= `ListLinePlot[twapsUniWeth, PlotRange → All]`

Out[121]=



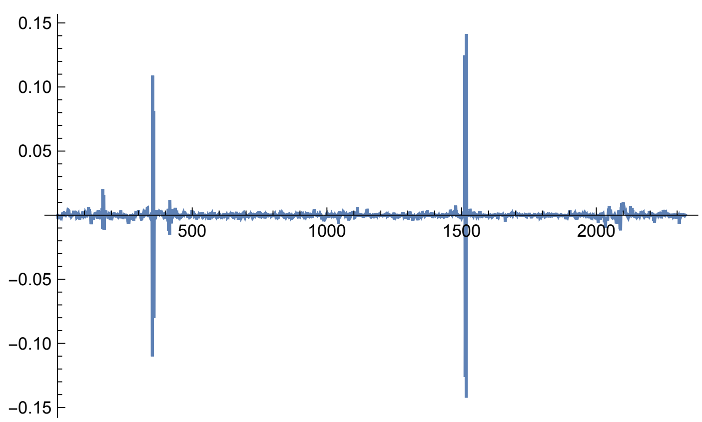

In[122]:= `rsUniWeth = Differences[Log[twapsUniWeth]]`

In[123]:= `rsUniWeth[[100]]`

Out[123]= -0.000807644

In[124]:= `ListLinePlot[rsUniWeth, PlotRange → All]`

Out[124]=

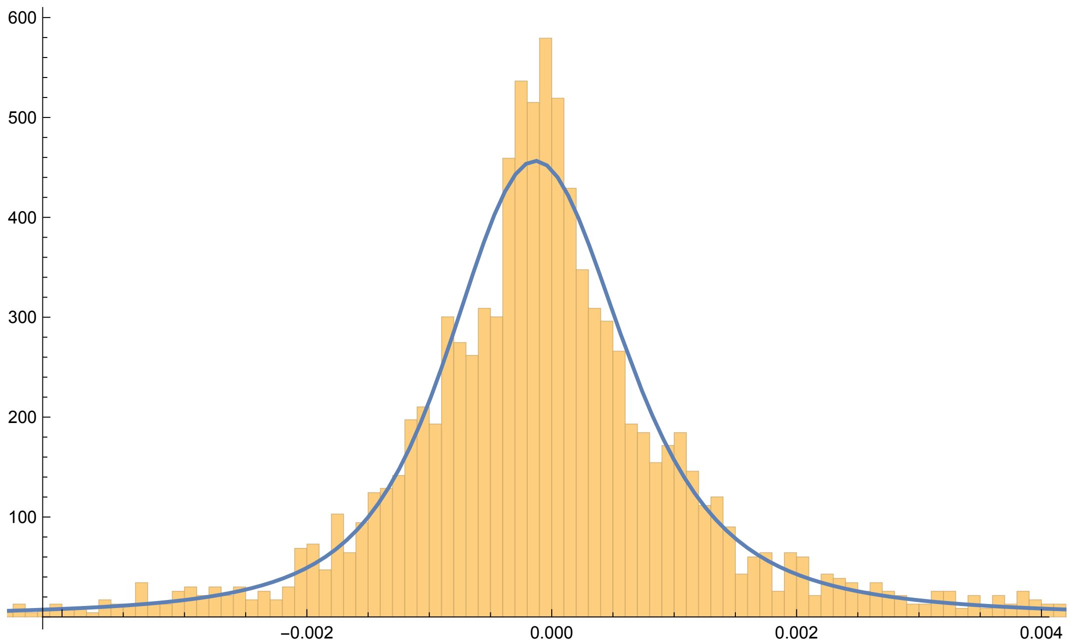



`(* Yea, UNI/WETH definitely not looking as extreme as ALCX/WETH ... :) *)`

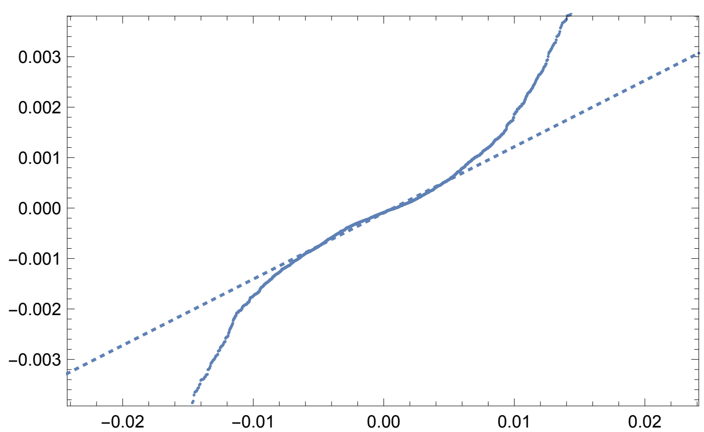
In[125]:= `edistUniWeth =`
`EstimatedDistribution[rsUniWeth, StableDistribution[1, aUW, bUW, locUW, scaleUW]]`

Out[125]= StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]

In[127]:= 
```
Show[Histogram[rsUniWeth, {0.0001}, "PDF"], Plot[PDF[edistUniWeth, x],
  {x, Min[rsUniWeth], Max[rsUniWeth]}, PlotRange → All, PlotStyle → Thick]]
```

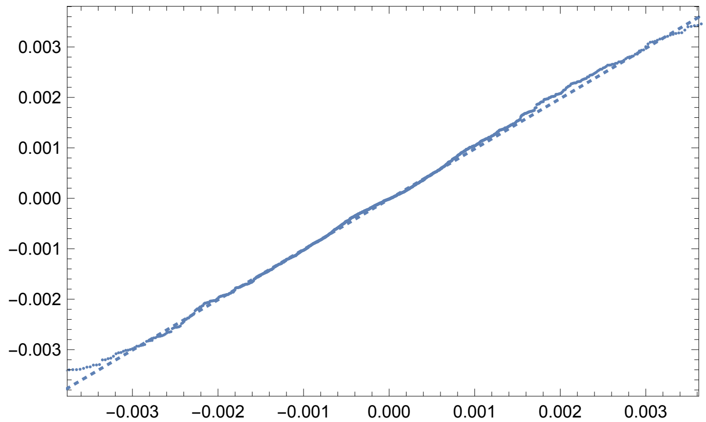
Out[127]=



In[128]:= 
```
QuantilePlot[rsUniWeth]
```

Out[128]=


In[129]:=
```
QuantilePlot[rsUniWeth, edistUniWeth]
```

Out[129]=




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

In[140]:=
```
tblUniWeth90d = Import["90/data-1625069716_uni-weth-twap.csv"]
```

Out[140]=

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

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

In[141]:=
```
Length[tblUniWeth90d]
```

Out[141]= 5952

In[142]:=
```
FromUnixTime[tblUniWeth90d[[2]][[2]]]
```

Out[142]= Sun 18 Apr 2021 17:11:37 GMT−4.

In[144]:=
```
FromUnixTime[tblUniWeth90d[[Length[tblUniWeth90d]]][[2]]]
```

Out[144]= Wed 30 Jun 2021 11:12:11 GMT−4.

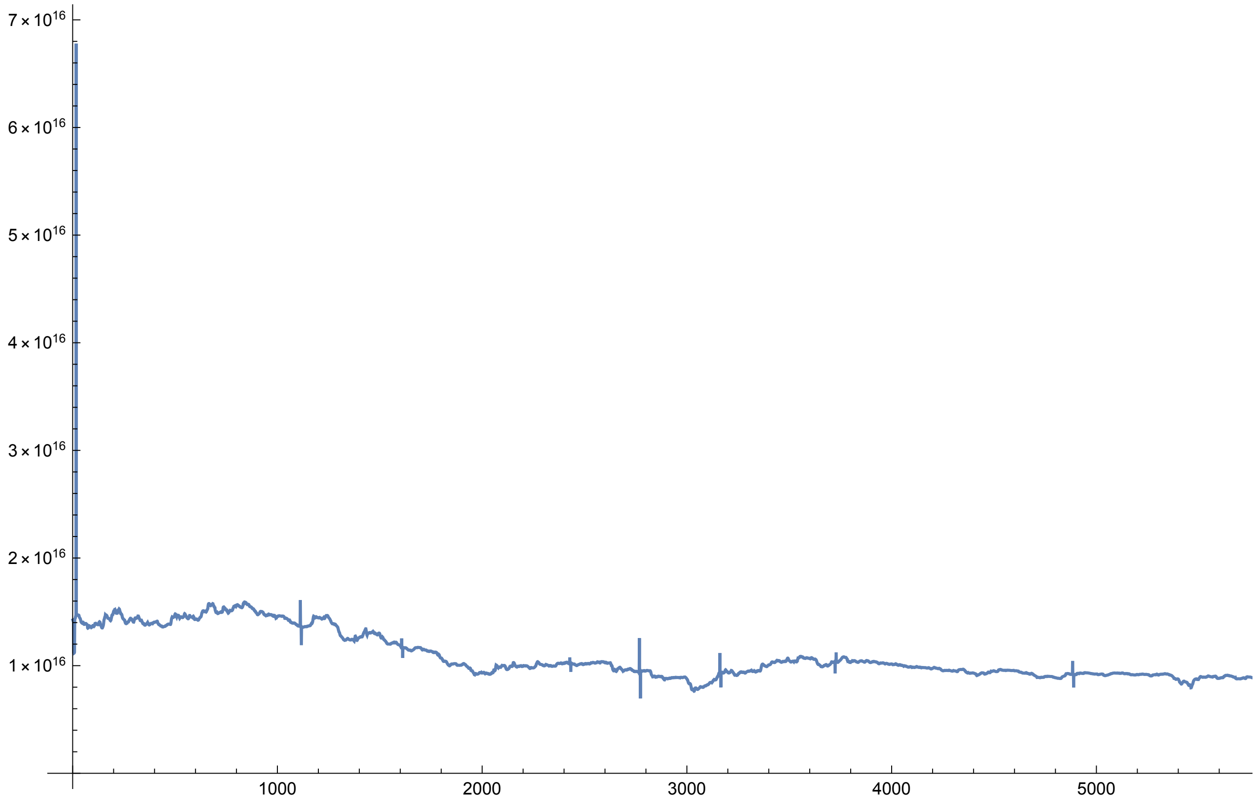
In[155]:=
```
twapsUniWeth90d = Table[tblUniWeth90d[[i]][[3]], {i, 2, Length[tblUniWeth90d]}]
```

In[160]:=
```
twapsUniWeth90d[[2]] / 10^18
```

Out[160]= 0.0142041

In[161]:=
```
ListLinePlot[twapsUniWeth90d, PlotRange → All]
```

Out[161]=

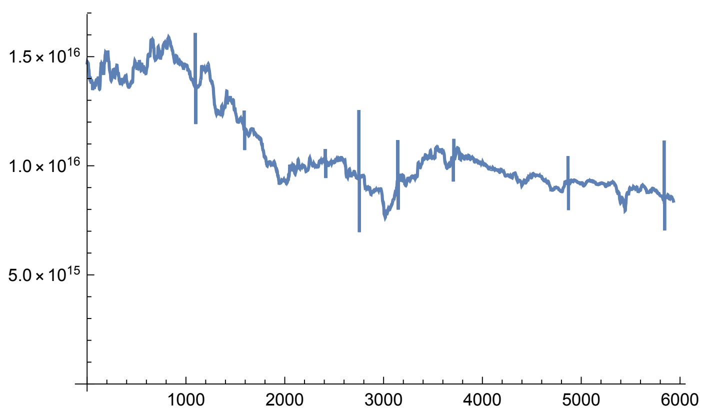



```
(* Something's up with the TWAP calc in early days =>
  related to cron issues ? Likely related to difference in
   metrics Deep was seeing. For now, ignore first 20 elements *)
```

In[166]:=
```
twapsUniWeth90dFiltered =
 Table[tblUniWeth90d[[i]][[3]], {i, 20, Length[tblUniWeth90d]}]
```

In[167]:=
```
ListLinePlot[twapsUniWeth90dFiltered, PlotRange → All]
```

Out[167]=

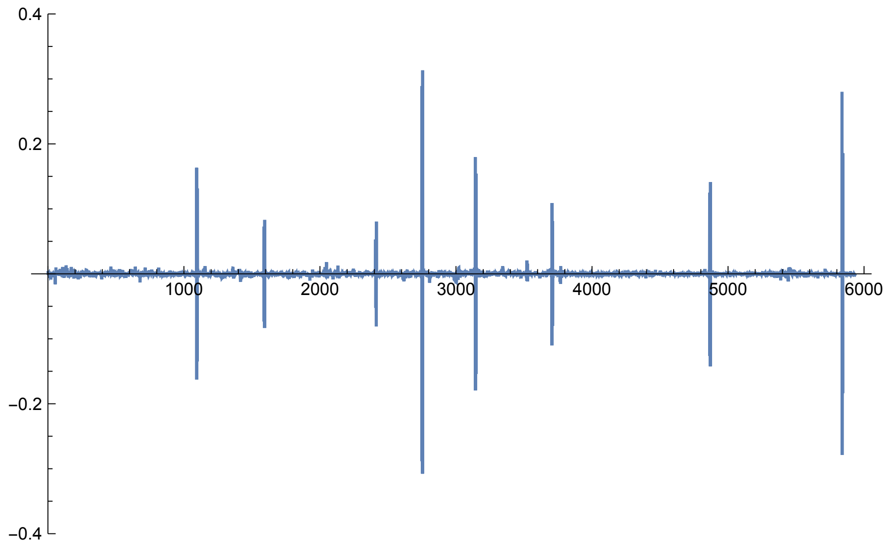



In[168]:=
```
Length[twapsUniWeth90dFiltered ]
```

Out[168]= 5933

```
(* So cuts out a day ... *)

In[170]:= FromUnixTime[tblUniWeth90d[[2]][[2]]]

Out[170]= Sun 18 Apr 2021 17:11:37 GMT-4.

In[169]:= FromUnixTime[tblUniWeth90d[[20]][[2]]]

Out[169]= Mon 19 Apr 2021 15:31:16 GMT-4.

In[171]:= rsUniWeth90dFiltered = Differences[Log[twapsUniWeth90dFiltered]]

In[172]:= rsUniWeth90dFiltered[[100]]

Out[172]= -0.000738056

In[179]:= ListLinePlot[rsUniWeth90dFiltered, PlotRange → {-0.4, 0.4}]
```

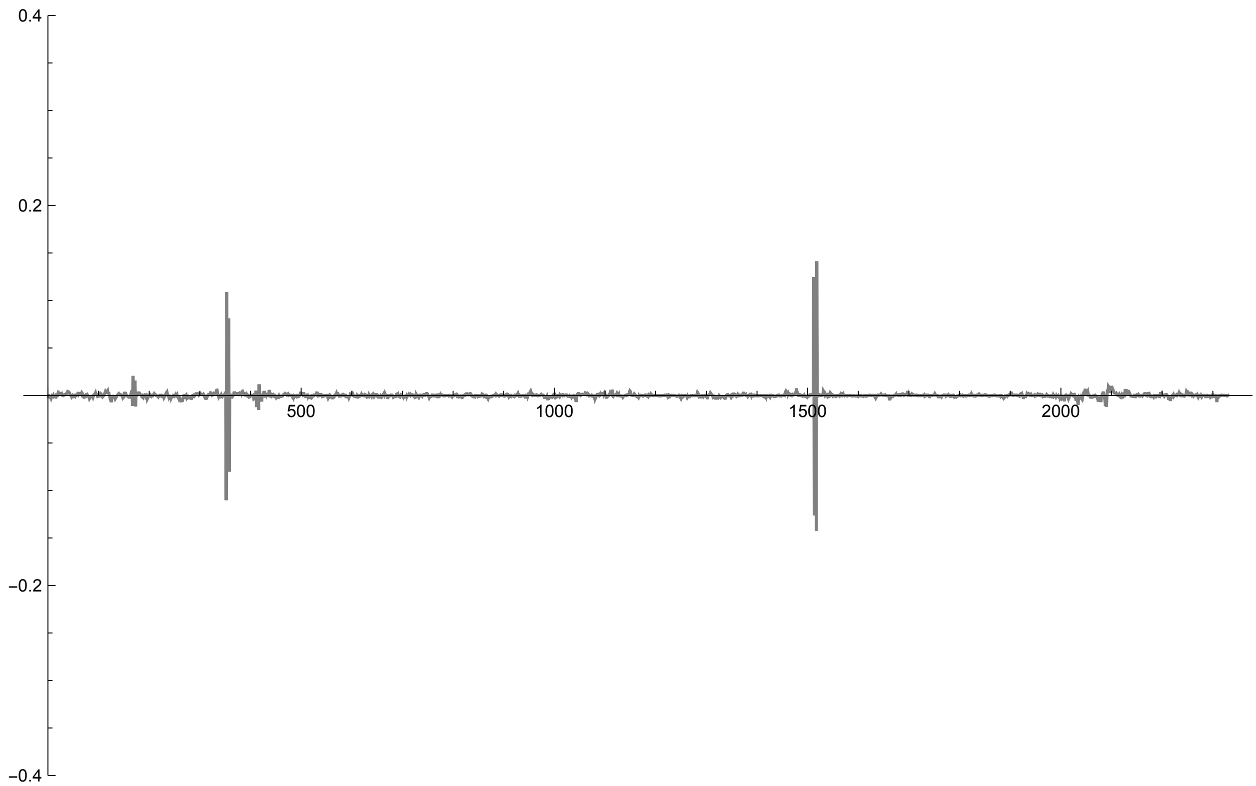
Out[179]=




```
(* Look at it in comparison to the 30d sampling for UNI/WETH ... *)
```

In[180]:=
```
ListLinePlot[rsUniWeth, PlotRange → {-0.4, 0.4}, PlotStyle → Gray]
```

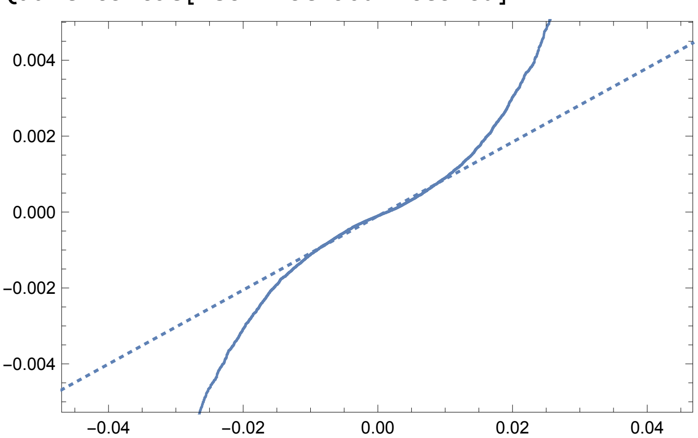
Out[180]=




```
(* 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). *)

(* Check again we're not normal ... *)
```

In[181]:=
```
QuantilePlot[rsUniWeth90dFiltered]
```

Out[181]=




```
(* Let's fit the 90d data ... *)
```

In[182]:=
```
edistUniWeth90dFiltered = EstimatedDistribution[rsUniWeth90dFiltered,
  StableDistribution[1, aUW90d, bUW90d, locUW90d, scaleUW90d]]
```

Out[182]=
```
StableDistribution[1, 1.29905, 0.00317474, -0.00011575, 0.000868971]
```

```
(* And compare with prior 30d fit *)
```

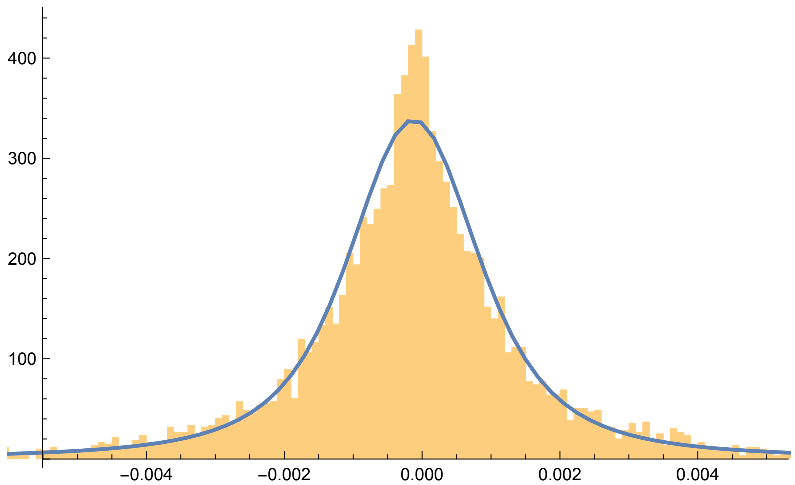
In[183]:=
```
edistUniWeth
```

Out[183]=
```
StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]
```

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

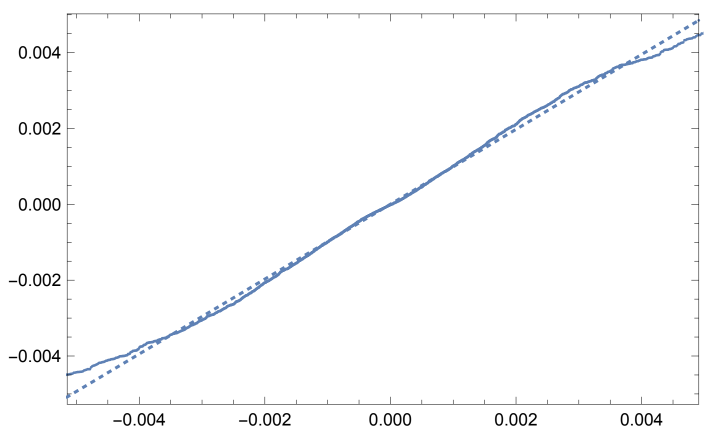
In[186]:=
```
Show[Histogram[rsUniWeth90dFiltered, {0.0001}, "PDF"],
 Plot[PDF[edistUniWeth90dFiltered, x], {x, Min[rsUniWeth90dFiltered],
   Max[rsUniWeth90dFiltered]}, PlotRange → All, PlotStyle → Thick]]
```

Out[186]=




In[187]:=
```
QuantilePlot[rsUniWeth90dFiltered, edistUniWeth90dFiltered]
```

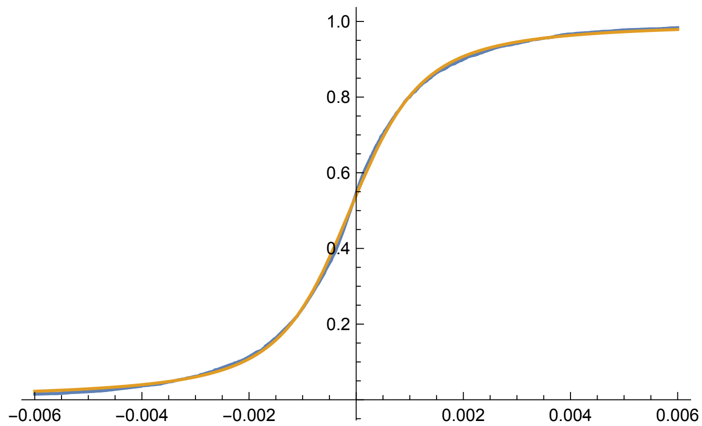
Out[187]=




```
(* hmm still slightly off at ends but not terrible *)

(* Compare the CDFs *)
```

In[188]:=
```
Plot[{CDF[EmpiricalDistribution[rsUniWeth90dFiltered], x],
  CDF[edistUniWeth90dFiltered, x]}, {x, -0.006, 0.006}]
```

Out[188]=




```
(* Nice ... *)

(* Filter out last month, so only analyzing first month of data *)
```

In[191]:=
```
Length[tblUniWeth90d] / 2
```

Out[191]= 2976

In[192]:=
```
tblUniWeth90d[[2976]]
```

Out[192]= $\{2975, 1.62173 \times 10^9, 8.912 \times 10^{15}\}$

In[197]:=
```
FromUnixTime[tblUniWeth90d[[2]][[2]]]
```

Out[197]= Sun 18 Apr 2021 17:11:37 GMT−4.

In[193]:=
```
FromUnixTime[tblUniWeth90d[[2976]][[2]]]
```

Out[193]= Sat 22 May 2021 21:52:37 GMT−4.

In[201]:=
```
rsUniWeth90dFiltered1stMo = Table[rsUniWeth90dFiltered[[i]], {i, 1, 2976}]

(* Fit the first mo of 90d data *)
```

In[202]:=
```
edistUniWeth90dFiltered1stMo = EstimatedDistribution[rsUniWeth90dFiltered1stMo,
  StableDistribution[1, aUW90d1m, bUW90d1m, locUW90d1m, scaleUW90d1m]]
```

Out[202]= StableDistribution[1, 1.38642, -0.0595957, -0.000254308, 0.00107224]

In[203]:=
```
edistUniWeth
```

Out[203]= StableDistribution[1, 1.32783, 0.0816836, -0.0000252748, 0.000640944]

```
(* Not terrible .... *)

(* Go back and download WETH/USDC data from cron. Look at that *)
```