In[1]:=
```
SetDirectory[NotebookDirectory[]];
```

In[2]:=
```
x[1] := 10 000 - w;
x[t_] := (1 + r[t - 1]) * x[t - 1] - w;
Table[r[i] = 3 / 100, {i, 1, 9}];
(* Find the withdrawal corresponding to a 3% arithmetic crediting rate,
which is TIAA's contractual minimum for many contracts. *)
```

In[5]:=
```
FindRoot[x[10], {w, 1100}, WorkingPrecision → 8]
```

Out[5]=
```
{w → 1138.1603}
```

In[6]:=
```
wTIAA = w /. %;
w = w /. %%; x[10]
```

Out[7]=
```
0. × 10^-4
```

In[8]:=
```
Table[x[t], {t, 1, 10}] // TableForm
```

Out[8]//TableForm=
```
8861.8397
7989.5347
7091.0605
6165.6320
5212.4407
4230.6537
3219.4130
2177.835
1105.010
0. × 10^-4
```

In[9]:=
```
Clear[w]
```

In[10]:=
```
(* switch to continuously-compounded returns *)
x[t_] := Exp[r[t - 1]] * x[t - 1] - w;
```

In[11]:=
```
(* TIAA's guaranteed minimum 3% in continuously-compounded terms. *)
guaranteeTIAA = Log[1 + 3 / 100];
N[%]
Table[r[i] = guaranteeTIAA, {i, 1, 9}];
FindRoot[x[10], {w, 1100}, WorkingPrecision → 8]
wTIAA = w /. %
w = w /. %%; x[10]
Clear[w]
```

Out[12]=
```
0.0295588
```

Out[14]=
```
{w → 1138.1603}
```

Out[15]=
```
1138.1603
```

Out[16]=
```
0. × 10^-4
```

In[18]:=
```
(* continuously-compounded returns,
indirectly from portfoliovisualizer.com's annual balances; 1987 to 2017 *)
ThreemLogRet = {1.35, 9.06, 13.22, 7.28, 17.61, 8.21, 9.78, -2.92, 19.17,
   3.69, 10.87, 7.66, -2.20, 10.18, 8.05, 10.23, 4.93, 6.27, 4.03, 3.88,
```

```
    5.19, 3.47, 8.39, 8.58, 11.11, 6.79, -3.43, 10.76, -0.29, 4.88, 6.21};

ThreetLogRet = {1.32, 7.99, 12.35, 6.81, 16.10, 7.92, 10.76, -3.25, 18.32,
    3.39, 10.09, 8.20, -1.83, 10.07, 8.00, 8.58, 5.30, 5.47, 3.03, 4.15,
    5.70, 2.56, 8.40, 7.56, 9.30, 6.28, -2.65, 8.10, -0.36, 4.24, 5.46};

TwomLogRet = {1.28, 8.81, 13.93, 8.17, 17.04, 7.83, 9.19, -2.91, 19.14,
    3.40, 10.78, 7.76, -2.38, 10.86, 8.33, 10.70, 4.15, 6.07, 4.08, 3.64,
    5.43, 5.21, 6.71, 8.30, 11.21, 6.21, -3.87, 10.98, -0.17, 4.40, 6.01};

TwotLogRet = {1.24, 7.57, 13.09, 7.81, 15.25, 7.43, 10.19, -3.29, 18.19,
    3.02, 9.89, 8.37, -2.00, 10.86, 8.33, 8.95, 4.42, 5.14, 2.97, 3.90,
    6.04, 4.50, 6.41, 7.12, 9.22, 5.55, -3.09, 8.06, -0.23, 3.61, 5.14};

FourmtLogRet = {1.33, 8.56, 12.88, 7.13, 16.88, 8.05, 10.19, -3.07, 18.79,
    3.53, 10.51, 7.92, -2.05, 10.18, 8.05, 9.52, 5.04, 5.89, 3.58, 3.99,
    5.44, 3.19, 8.27, 8.10, 10.30, 6.52, -3.11, 9.57, -0.32, 4.55, 5.86};

CashLogRet = {5.33, 6.17, 8.04, 7.55, 5.45, 3.44, 2.86, 3.82,
    5.45, 5.07, 5.12, 4.74, 4.58, 5.72, 3.75, 1.62, 1.02, 1.18, 2.94, 4.70,
    4.56, 1.58, 0.09, 0.10, 0.04, 0.06, 0.00, 0.00, 0.01, 0.21, 0.79};

TotalLogRet = {1.53, 7.09, 12.79, 8.30, 14.19, 6.90, 9.24, -2.70,
  16.71, 3.52, 9.02, 8.23, -0.76, 10.79, 8.09, 7.93, 3.90, 4.15, 2.37,
  4.18, 6.69, 4.93, 5.77, 6.22, 7.29, 3.97, -2.29, 5.60, 0.30, 2.47, 3.40}

(*Store these numbers internally as exact fractions
instead of as approximate decimals.*)
MakeExact[logreturns_] := Map[IntegerPart, 1000 * logreturns] / 100 000

ThreemLogRet = MakeExact[ThreemLogRet];
N[%]
Print[{"3m", Mean[%], StandardDeviation[%]}]

ThreetLogRet = MakeExact[ThreetLogRet];
Print[{"3t", N[Mean[%]], N[StandardDeviation[%]]}]

TwomLogRet = MakeExact[TwomLogRet];
Print[{"2m", N[Mean[%]], N[StandardDeviation[%]]}]

TwotLogRet = MakeExact[TwotLogRet];
Print[{"2t", N[Mean[%]], N[StandardDeviation[%]]}]

FourmtLogRet = MakeExact[FourmtLogRet];
Print[{"4mt", N[Mean[%]], N[StandardDeviation[%]]}]

CashLogRet = MakeExact[CashLogRet];
N[%]
Print[{"Cash", N[Mean[%]], N[StandardDeviation[%]]}]
```

```
TotalLogRet = MakeExact[TotalLogRet];
N[%]
Print[{"Total", N[Mean[%]], N[StandardDeviation[%]]}]
```

Out[24]= {1.53, 7.09, 12.79, 8.3, 14.19, 6.9, 9.24, -2.7, 16.71,
3.52, 9.02, 8.23, -0.76, 10.79, 8.09, 7.93, 3.9, 4.15, 2.37, 4.18,
6.69, 4.93, 5.77, 6.22, 7.29, 3.97, -2.29, 5.6, 0.3, 2.47, 3.4}

Out[27]= {0.0135, 0.0906, 0.1322, 0.0728, 0.1761, 0.0821, 0.0978,
-0.0292, 0.1917, 0.0369, 0.1087, 0.0766, -0.022, 0.1018, 0.0805,
0.1023, 0.0493, 0.0627, 0.0403, 0.0388, 0.0519, 0.0347, 0.0839,
0.0858, 0.1111, 0.0679, -0.0343, 0.1076, -0.0029, 0.0488, 0.0621}

{3m, 0.0683903, 0.0521768}

{3t, 0.0636645, 0.0487326}

{2m, 0.0678355, 0.0525902}

{2t, 0.062471, 0.0486972}

{4mt, 0.0662161, 0.0505021}

Out[38]= {0.0533, 0.0617, 0.08039, 0.0755, 0.0545, 0.0344, 0.0286, 0.0382, 0.0545, 0.0507,
0.0512, 0.0474, 0.0458, 0.0572, 0.0375, 0.0162, 0.0102, 0.0118, 0.0294, 0.047,
0.0456, 0.0158, 0.0009, 0.001, 0.0004, 0.0006, 0., 0., 0.0001, 0.0021, 0.0079}

{Cash, 0.0309642, 0.024954}

Out[41]= {0.0153, 0.0709, 0.1279, 0.083, 0.1419, 0.069, 0.0924, -0.027, 0.1671, 0.0352, 0.0902,
0.0823, -0.0076, 0.1079, 0.0809, 0.0793, 0.039, 0.0415, 0.0237, 0.0418, 0.0669,
0.0493, 0.0577, 0.0622, 0.0729, 0.0397, -0.0229, 0.056, 0.003, 0.0247, 0.034}

{Total, 0.0580065, 0.0443287}

In[43]:=
```
Length[CashLogRet] - 9 + 1
```

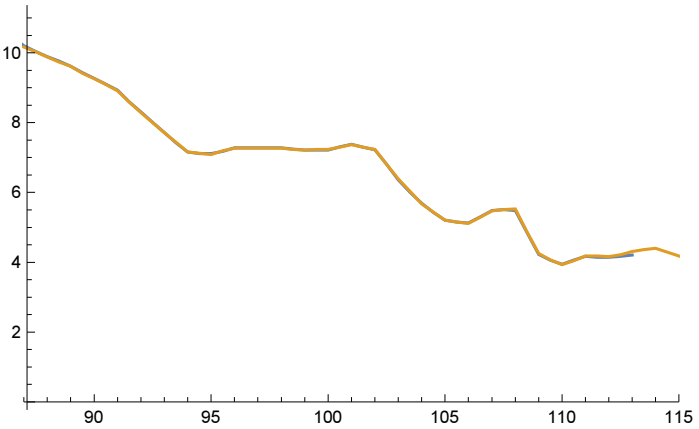
Out[43]= 23

In[44]:=
```
TradArith2 = Import["WPD_87_15.csv"];
(*y=Import["WebPlotDigitizer1964_86-2013.csv"];*)
TradArith1 = Sort[Import["WPD1964-2013.csv"]];
ListLinePlot[{TradArith1, TradArith2}, PlotRange → {{87, 115}, All}]
TradArith1 = Select[TradArith1, #[[1]] > 86.9 &];
TradArith1ReturnsOnly = TradArith1[[All, 2]];
Short[%]
TradArith2ReturnsOnly = TradArith2[[All, 2]];
Short[%]

TradArith1ReturnsOnly - Drop[TradArith2ReturnsOnly, -4]
(* drop last 4 half-years *)

ListPlot[%, Joined → True,
 PlotMarkers → {"•", 18}, GridLines → {None, {-.048, .032}}]
(* The lines agree within +3.2 and -4.8 basis points, except for 2013. *)
RootMeanSquare[%%]
RootMeanSquare[Drop[%%%, -1]]
```

Out[46]=

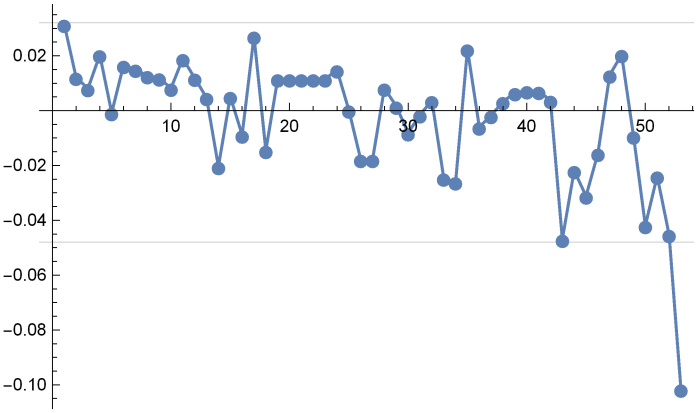


Out[49]//Short= {10.2025, 10.0391, 9.89205, 9.76132, <<45>>, 4.14011, 4.14011, 4.1728, 4.20548}

Out[51]//Short= {10.171, 10.027, 9.884, 9.741, 9.615, <<47>>, 4.307, 4.361, 4.397, 4.289, 4.182}

Out[52]= {0.03152, 0.01211, 0.00805, 0.02032, -0.00075, 0.01651, 0.0151, 0.01269,
0.01194, 0.00813, 0.019, 0.01186, 0.00473, -0.0204, 0.00515, -0.00888,
0.02712, -0.01451, 0.01153, 0.01153, 0.01153, 0.01153, 0.01153, 0.01485,
0.00017, -0.01783, -0.01783, 0.00821, 0.00158, -0.00813, -0.00149,
0.00365, -0.02455, -0.02604, 0.02248, -0.00597, -0.00174, 0.00324,
0.00656, 0.0073, 0.00705, 0.00373, -0.04695, -0.02189, -0.03118, -0.01559,
0.01303, 0.02041, -0.0092, -0.04189, -0.02389, -0.0452, -0.10152}

Out[53]=



Out[54]= 0.0227342

Out[55]= 0.0181269

In[56]:=
```
TradArith1 = Select[TradArith1, IntegerQ[#[[1]]] &]
TradArith2 = Select[TradArith2, IntegerQ[#[[1]]] &]
Drop[TradArith1, -1]
Drop[TradArith2, -3]
(% + %%) / 2 (* take the mean of the two lists without 2013/2014/2015 *)
Take[TradArith2, -3]
TradArith = Flatten[{%%, %}, 1];
(* Use the average of the readings, except for 2013, 2014,
and 2015, when you use the white paper's readings. *)
temp = TradArith;
temp[[All, 1]] = temp[[All, 1]] + 1900;
temp
Map[({#1[[1]], N[Round[100 * #1[[2]]] / 100, 4]}) &, temp] // TableForm // TeXForm
Clear[temp]
```

Out[56]= {{87, 10.2025}, {88, 9.89205}, {89, 9.61425}, {90, 9.2711}, {91, 8.92794},
{92, 8.307}, {93, 7.71873}, {94, 7.16315}, {95, 7.11412}, {96, 7.27753},
{97, 7.27753}, {98, 7.27753}, {99, 7.21217}, {100, 7.21217}, {101, 7.37558},
{102, 7.22851}, {103, 6.36245}, {104, 5.69248}, {105, 5.20226},
{106, 5.12056}, {107, 5.48005}, {108, 5.48005}, {109, 4.22182},
{110, 3.94403}, {111, 4.1728}, {112, 4.14011}, {113, 4.20548}}

Out[57]= {{87, 10.171}, {88, 9.884}, {89, 9.615}, {90, 9.256}, {91, 8.916}, {92, 8.288},
{93, 7.714}, {94, 7.158}, {95, 7.087}, {96, 7.266}, {97, 7.266}, {98, 7.266},
{99, 7.212}, {100, 7.23}, {101, 7.374}, {102, 7.23}, {103, 6.387}, {104, 5.67},
{105, 5.204}, {106, 5.114}, {107, 5.473}, {108, 5.527}, {109, 4.253}, {110, 3.931},
{111, 4.182}, {112, 4.164}, {113, 4.307}, {114, 4.397}, {115, 4.182}}

Out[58]= {{87, 10.2025}, {88, 9.89205}, {89, 9.61425}, {90, 9.2711}, {91, 8.92794}, {92, 8.307},
{93, 7.71873}, {94, 7.16315}, {95, 7.11412}, {96, 7.27753}, {97, 7.27753},
{98, 7.27753}, {99, 7.21217}, {100, 7.21217}, {101, 7.37558}, {102, 7.22851},
{103, 6.36245}, {104, 5.69248}, {105, 5.20226}, {106, 5.12056}, {107, 5.48005},
{108, 5.48005}, {109, 4.22182}, {110, 3.94403}, {111, 4.1728}, {112, 4.14011}}

Out[59]=
```
{{87, 10.171}, {88, 9.884}, {89, 9.615}, {90, 9.256}, {91, 8.916},
 {92, 8.288}, {93, 7.714}, {94, 7.158}, {95, 7.087}, {96, 7.266}, {97, 7.266},
 {98, 7.266}, {99, 7.212}, {100, 7.23}, {101, 7.374}, {102, 7.23},
 {103, 6.387}, {104, 5.67}, {105, 5.204}, {106, 5.114}, {107, 5.473},
 {108, 5.527}, {109, 4.253}, {110, 3.931}, {111, 4.182}, {112, 4.164}}
```

Out[60]=
```
{{87, 10.1868}, {88, 9.88802}, {89, 9.61463}, {90, 9.26355},
 {91, 8.92197}, {92, 8.2975}, {93, 7.71637}, {94, 7.16058}, {95, 7.10056},
 {96, 7.27177}, {97, 7.27177}, {98, 7.27177}, {99, 7.21209}, {100, 7.22109},
 {101, 7.37479}, {102, 7.22926}, {103, 6.37473}, {104, 5.68124},
 {105, 5.20313}, {106, 5.11728}, {107, 5.47653}, {108, 5.50353},
 {109, 4.23741}, {110, 3.93752}, {111, 4.1774}, {112, 4.15206}}
```

Out[61]=
```
{{113, 4.307}, {114, 4.397}, {115, 4.182}}
```

Out[65]=
```
{{1987, 10.1868}, {1988, 9.88802}, {1989, 9.61463}, {1990, 9.26355}, {1991, 8.92197},
 {1992, 8.2975}, {1993, 7.71637}, {1994, 7.16058}, {1995, 7.10056}, {1996, 7.27177},
 {1997, 7.27177}, {1998, 7.27177}, {1999, 7.21209}, {2000, 7.22109}, {2001, 7.37479},
 {2002, 7.22926}, {2003, 6.37473}, {2004, 5.68124}, {2005, 5.20313}, {2006, 5.11728},
 {2007, 5.47653}, {2008, 5.50353}, {2009, 4.23741}, {2010, 3.93752},
 {2011, 4.1774}, {2012, 4.15206}, {2013, 4.307}, {2014, 4.397}, {2015, 4.182}}
```

Out[66]//TeXForm=
```
\begin{array}{cc}
 1987 & 10.19 \\
 1988 & 9.890 \\
 1989 & 9.610 \\
 1990 & 9.260 \\
 1991 & 8.920 \\
 1992 & 8.300 \\
 1993 & 7.720 \\
 1994 & 7.160 \\
 1995 & 7.100 \\
 1996 & 7.270 \\
 1997 & 7.270 \\
 1998 & 7.270 \\
 1999 & 7.210 \\
 2000 & 7.220 \\
 2001 & 7.370 \\
 2002 & 7.230 \\
 2003 & 6.370 \\
 2004 & 5.680 \\
 2005 & 5.200 \\
 2006 & 5.120 \\
 2007 & 5.480 \\
 2008 & 5.500 \\
 2009 & 4.240 \\
 2010 & 3.940 \\
 2011 & 4.180 \\
 2012 & 4.150 \\
 2013 & 4.310 \\
 2014 & 4.400 \\
 2015 & 4.180 \\
\end{array}
```

In[68]:=
```
TradLogRet = 100 * Log[1 + TradArith[[All, 2]] / 100]
TradLogRet = MakeExact[TradLogRet];
```

Out[68]=
```
{9.70066, 9.42917, 9.18006, 8.85927, 8.54616, 7.97119, 7.43313,
 6.91582, 6.8598, 7.01953, 7.01953, 7.01953, 6.96388, 6.97227,
 7.11552, 6.97989, 6.17978, 5.52572, 5.07229, 4.99065, 5.33182,
 5.35742, 4.15009, 3.86197, 4.0925, 4.06817, 4.21683, 4.30308, 4.09692}
```

In[70]:=
```
(* find maximum feasible constant withdrawal amounts *)
(* the first withdrawal happens at the beginning,
so the total number of withdrawals is YearsOfWithdrawals+1  *)

YearsOfWithdrawals = 9;

Withdrawals[LogRet_] := Table[w /. {StartYear = t;
      Table[r[i] = LogRet[[i + StartYear - 1]], {i, 1, YearsOfWithdrawals}];
      FindRoot[x[YearsOfWithdrawals + 1], {w, 1400}, WorkingPrecision → 8]}[[1]],
  {t, 1, Length[LogRet] - YearsOfWithdrawals + 1}]
```

In[72]:=
```
ThreemWithdrawals = Withdrawals[ThreemLogRet]
```

Out[72]=
```
{1413.4708, 1518.7558, 1530.6356, 1479.5995, 1496.8193,
 1369.9200, 1357.8573, 1325.9243, 1462.5048, 1312.0471, 1347.8390,
 1291.1848, 1268.2429, 1368.9241, 1322.6019, 1300.7342, 1253.0651,
 1266.6701, 1256.3484, 1278.0816, 1299.1579, 1305.6506, 1336.4078}
```

In[73]:=
```
ThreetWithdrawals = Withdrawals[ThreetLogRet]
```

Out[73]=
```
{1384.1814, 1480.4344, 1498.8620, 1454.7302, 1473.5858,
 1365.0089, 1355.7553, 1308.8066, 1444.2123, 1302.2061, 1337.6818,
 1288.0314, 1256.6488, 1347.5391, 1298.6076, 1272.1637, 1238.7067,
 1242.2724, 1236.3930, 1263.2199, 1274.2631, 1266.2962, 1297.8427}
```

In[74]:=
```
TwomWithdrawals = Withdrawals[TwomLogRet]
```

Out[74]=
```
{1415.7049, 1521.8708, 1537.3550, 1476.0165, 1478.4263,
 1357.9935, 1349.7981, 1325.5785, 1462.1258, 1312.0332, 1351.7782,
 1296.4708, 1272.8869, 1378.8956, 1324.4516, 1298.6397, 1244.5286,
 1265.8883, 1256.7755, 1277.0774, 1300.3808, 1302.7419, 1308.7241}
```

In[75]:=
```
TwotWithdrawals = Withdrawals[TwotLogRet]
```

Out[75]=
```
{1383.4936, 1479.6284, 1502.9508, 1447.6151, 1449.4285,
 1350.3410, 1346.0376, 1306.3156, 1441.7192, 1301.0613, 1341.0305,
 1293.8207, 1260.7907, 1356.7431, 1298.0364, 1266.4356, 1226.9819,
 1238.4517, 1234.4573, 1260.1822, 1272.5997, 1258.2462, 1261.7055}
```

In[76]:=
```
FourmtWithdrawals = Withdrawals[FourmtLogRet]
```

Out[76]=
```
{1400.3592, 1501.6013, 1516.7050, 1468.0526, 1484.9734,
 1366.9481, 1356.4799, 1318.1937, 1454.2795, 1307.6067, 1343.5920,
 1290.2330, 1263.3216, 1359.9972, 1311.8998, 1287.6343, 1245.9673,
 1255.5858, 1247.4345, 1271.4380, 1288.0627, 1287.6155, 1316.8763}
```

In[77]:=
```
CashWithdrawals = Withdrawals[CashLogRet]
```

Out[77]=
```
{1283.0823, 1284.4587, 1274.9436, 1240.1387, 1207.0964,
 1196.5536, 1208.0575, 1225.4763, 1230.1408, 1212.2924, 1195.9293,
 1178.2169, 1163.9886, 1148.1380, 1114.7484, 1097.4257, 1099.4957,
 1106.6072, 1111.0155, 1094.7915, 1056.7226, 1016.5293, 1003.4902}
```

In[78]:=
```
Tradwithdrawals = Withdrawals[TradLogRet]
```

Out[78]=
```
{1445.8825, 1427.6449, 1409.2770, 1390.7492, 1373.2538, 1357.0663, 1346.2211,
 1340.9225, 1340.9911, 1340.4639, 1336.0001, 1329.2310, 1320.3619, 1310.0544,
 1296.2507, 1276.8233, 1255.2991, 1240.0221, 1230.2538, 1224.4188, 1218.3667}
```

In[79]:=
```
Totalwithdrawals = Withdrawals[TotalLogRet]
```

Out[79]=
```
{1372.1343, 1459.3553, 1483.4862, 1427.7357, 1418.6101,
 1330.6653, 1330.2048, 1299.9092, 1423.6763, 1299.1431, 1329.4850,
 1290.2100, 1257.4440, 1335.3936, 1274.4669, 1241.6631, 1209.1461,
 1220.4877, 1223.3265, 1249.8151, 1253.0352, 1223.4626, 1211.1071}
```

In[80]:=
```
Clear[wd];
(* For these constant withdrawal amounts, what is the corresponding
 constant crediting rate that would make them possible? *)
y[1] := 10 000 - wd; (* start of first year balance *)
y[t_] := Exp[r] * y[t - 1] - wd; (* balance at start of year t *)
(* assuming continuously-compounded returns *)
y[10]
Expand[%]
```

Out[83]=
$$e^r \left(e^r \left(e^r \left(e^r \left(e^r \left(e^r \left(e^r \left(e^r \left(10\,000 - wd\right) - wd\right) - wd\right) - wd\right) - wd\right) - wd\right) - wd\right) - wd\right) - wd\right) - wd$$

Out[84]=
$$10\,000\, e^{9\,r} - wd - e^r\, wd - e^{2\,r}\, wd - e^{3\,r}\, wd - e^{4\,r}\, wd - e^{5\,r}\, wd - e^{6\,r}\, wd - e^{7\,r}\, wd - e^{8\,r}\, wd - e^{9\,r}\, wd$$

In[85]:=
```
CorrespondingRate[withd_] := 100 * r /. {wd = withd;
     FindRoot[y[YearsOfWithdrawals + 1] == 0, {r, 4 / 100}]}[[1]]

(* check this should be equal to guaranteeTIAA *)
{CorrespondingRate[wTIAA], 100 * N[guaranteeTIAA]}
```

Out[86]=
```
{2.95588, 2.95588}
```

In[87]:=
```
(* Yes, that checks. *)
```

In[88]:=
```
CRThreem = Map[CorrespondingRate, ThreemWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[88]=
```
{8.32099, 10.2392, 10.452, 9.53276, 9.84439, 7.50892, 7.28187,
 6.67615, 9.22184, 6.41073, 7.09258, 6.00912, 5.56374, 7.49021, 6.61273,
 6.19335, 5.26686, 5.53306, 5.33123, 5.75523, 6.16298, 6.28793, 6.87577}
```

Out[89]=
```
{7.20277, 1.62715, 5.26686, 10.452}
```

In[90]:=
```
CRThreet = Map[CorrespondingRate, ThreetWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[90]= {7.77614, 9.5479, 9.88125, 9.07991, 9.42356, 7.41659, 7.24221,
6.34856, 8.88737, 6.22168, 6.89998, 5.94814, 5.33712, 7.0869, 6.15238,
5.64014, 4.98432, 5.05464, 4.93864, 5.46569, 5.681, 5.52576, 6.13763}

Out[91]= {6.81207, 1.58361, 4.93864, 9.88125}

In[92]:=
```
CRTwom = Map[CorrespondingRate, TwomWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[92]= {8.36234, 10.295, 10.572, 9.46772, 9.51147, 7.28444, 7.12965,
6.66956, 9.21493, 6.41047, 7.16709, 6.11118, 5.65422, 7.67725, 6.64805,
6.153, 5.09908, 5.5178, 5.3396, 5.73572, 6.18654, 6.23199, 6.34697}

Out[93]= {7.16462, 1.63039, 5.09908, 10.572}

In[94]:=
```
CRTwot = Map[CorrespondingRate, TwotWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[94]= {7.76328, 9.53328, 9.95498, 8.94973, 8.98293, 7.13992, 7.05847,
6.30071, 8.84164, 6.19965, 6.96355, 6.06004, 5.4182, 7.26085, 6.14137,
5.52848, 4.75236, 4.97929, 4.90038, 5.4063, 5.64863, 5.36841, 5.43609}

Out[95]= {6.72124, 1.58971, 4.75236, 9.95498}

In[96]:=
```
CRFourmt = Map[CorrespondingRate, FourmtWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[96]= {8.07773, 9.93066, 10.2024, 9.32292, 9.63018, 7.45307, 7.25589,
6.52846, 9.07167, 6.32552, 7.01213, 5.99072, 5.46768, 7.32222, 6.40791,
5.94045, 5.1274, 5.31629, 5.15626, 5.62601, 5.94874, 5.94009, 6.50325}

Out[97]= {7.02424, 1.60388, 5.1274, 10.2024}

In[98]:=
```
CRCash = Map[CorrespondingRate, CashWithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[98]= {5.85228, 5.87896, 5.69424, 5.01257, 4.3563, 4.14494, 4.37552, 4.72249,
4.81497, 4.46011, 4.13239, 3.77497, 3.48577, 3.16135, 2.46986, 2.1066,
2.15018, 2.29953, 2.39185, 2.05108, 1.24015, 0.365539, 0.0774807}

Out[99]= {3.43561, 1.66638, 0.0774807, 5.87896}

In[100]:=
```
CRTrad = Map[CorrespondingRate, Tradwithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[100]= {8.91798, 8.58281, 8.2433, 7.89878, 7.5715, 7.26695, 7.06195,
6.96151, 6.96281, 6.9528, 6.86802, 6.7392, 6.56993, 6.37251,
6.10693, 5.73078, 5.31067, 5.01027, 4.8172, 4.7015, 4.58118}

Out[101]= {6.62993, 1.26199, 4.58118, 8.91798}

In[102]:=
```
CRTotal = Map[CorrespondingRate, Totalwithdrawals]
{Mean[%], StandardDeviation[%], Min[%], Max[%]}
```

Out[102]=
```
{7.55049, 9.16438, 9.60323, 8.58449, 8.41606, 6.76652, 6.75775,
 6.17746, 8.50962, 6.1627, 6.74404, 5.99028, 5.3527, 6.85649, 5.68496,
 5.04263, 4.39728, 4.62338, 4.6798, 5.20305, 5.26627, 4.68251, 4.43645}
```

Out[103]=
```
{6.3762, 1.60203, 4.39728, 9.60323}
```

These are not the same as the average rate of return during the 9-year period because, for example, the first year's return affects all the withdrawals but the first, and so is quite important, while the last year's return only affect the last withdrawal, so it's less important.

In[104]:=
```
(* the average continuously-compounded returns over each 9-year period *)

AvgRetNineYrs[LogRet_] := (Clear[NineYearRetList, NineYearRet];
  NineYearRetList[1] = LogRet;
  NineYearRetList[t_] := NineYearRet[t] = RotateLeft[NineYearRetList[t - 1]];
  (* a recursive definition that remembers the values it has found *)
  NineYearRet[t_] := 100 * Mean[Take[NineYearRetList[t], 9]];
  Table[NineYearRet[t], {t, 1, Length[Withdrawals[LogRet]]}])

(* 3m *)
ThreemNineYrRets = AvgRetNineYrs[ThreemLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[106]=
```
{9.19556, 9.45556, 9.65667, 9.03889, 7.98556, 7.16, 7.14222,
 7.19222, 8.06444, 6.63111, 6.66889, 5.89222, 5.61778, 6.24778, 6.04889,
 6.10778, 6.20556, 6.41222, 5.33444, 6.08222, 5.61889, 5.58444, 5.88889}
```

Out[107]=
```
{6.92314, 1.34003}
```

In[108]:=
```
ThreetNineYrRets = AvgRetNineYrs[ThreetLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[109]=
```
{8.70222, 8.93222, 9.16556, 8.70444, 7.74444, 7.07444, 7.08333,
 6.84111, 7.79111, 6.36333, 6.32333, 5.66333, 5.38556, 5.87333, 5.68778,
 5.63889, 5.71889, 5.82778, 4.92556, 5.48889, 4.98778, 4.82556, 5.14778}
```

Out[110]=
```
{6.51725, 1.3816}
```

In[111]:=
```
TwomNineYrRets = AvgRetNineYrs[TwomLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[112]=
```
{9.16444, 9.4, 9.61889, 8.93333, 7.76111, 7.07444, 7.13,
 7.29778, 8.08222, 6.63, 6.70556, 5.91222, 5.65333, 6.49667, 6.03556,
 6.03222, 6.08889, 6.31778, 5.21333, 5.98, 5.55667, 5.44222, 5.53111}
```

Out[113]=
```
{6.87208, 1.34945}
```

In[114]:=
```
TwotNineYrRets = AvgRetNineYrs[TwotLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[115]= {8.60889, 8.80667, 9.06444, 8.54, 7.45, 6.96222, 7.06222,
6.92444, 7.78111, 6.33111, 6.32556, 5.66, 5.40111, 6.12333, 5.62889,
5.49444, 5.52444, 5.65, 4.73556, 5.30111, 4.84222, 4.57222, 4.64333}

Out[116]= {6.41014, 1.40768}

In[117]:=
```
FourmtNineYrRets = AvgRetNineYrs[FourmtLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[118]= {8.97111, 9.21556, 9.43222, 8.88111, 7.86111, 7.11667, 7.11667,
7.04222, 7.94333, 6.51, 6.51556, 5.79111, 5.51556, 6.09778, 5.88556,
5.89111, 5.97778, 6.14222, 5.14222, 5.80778, 5.32889, 5.23, 5.52667}

Out[119]= {6.73662, 1.3566}

In[120]:=
```
CashNineYrRets = AvgRetNineYrs[CashLogRet];
N[%]
{Mean[%], StandardDeviation[%]}
```

Out[121]= {5.34544, 5.31656, 5.19989, 4.83333, 4.50333, 4.53333, 4.56778, 4.43,
4.11889, 3.64444, 3.40778, 3.36111, 3.34111, 3.00778, 2.38222, 1.97667,
1.80111, 1.69444, 1.56333, 1.23667, 0.715556, 0.232222, 0.144444}

Out[122]= {3.1025, 1.6624}

In[123]:=
```
TradNineYrRets = AvgRetNineYrs[TradLogRet];
N[%]
Length[%%]
{Mean[%%], StandardDeviation[%%]}
```

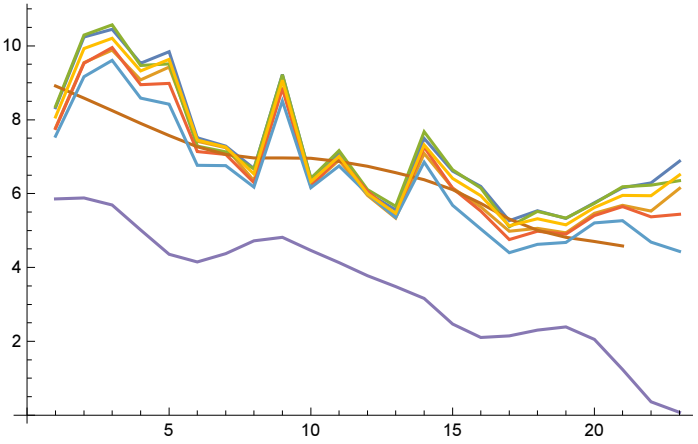
Out[124]= {8.32133, 8.02344, 7.75567, 7.51556, 7.30489, 7.13, 7.03489,
6.98444, 6.90267, 6.75444, 6.53811, 6.31267, 6.12511, 5.94667,
5.63311, 5.27156, 4.95078, 4.71622, 4.57078, 4.48533, 4.386}

Out[125]= 21

Out[126]= {6.31732, 1.22297}

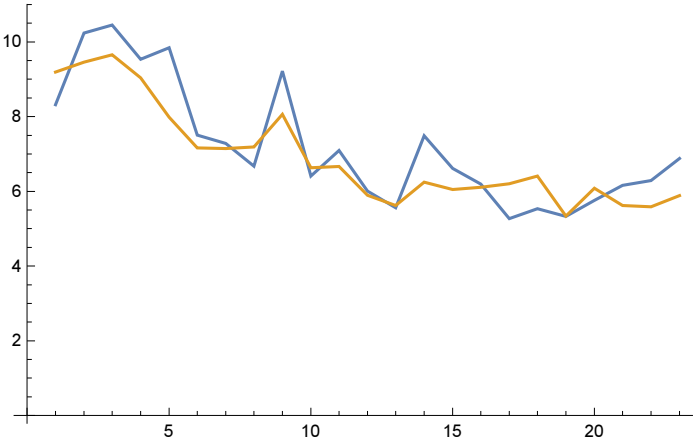
In[127]:=
```
ListPlot[{CRThreem, CRThreet, CRTwom,
  CRTwot, CRCash, CRTrad, CRTotal, CRFourmt}, Joined → True]
```

Out[127]=



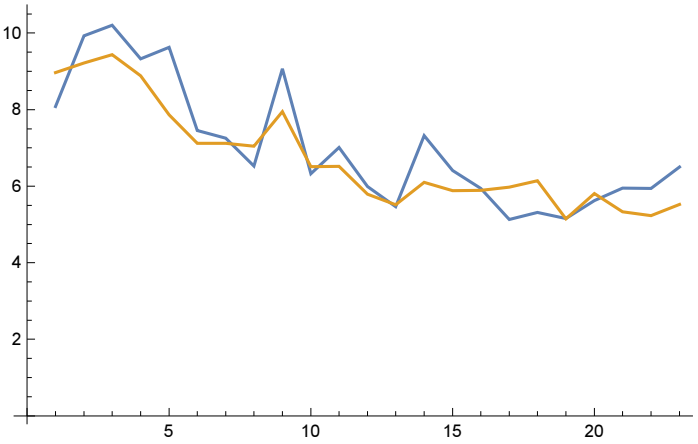
In[128]:=
```
ListPlot[{CRThreem, ThreemNineYrRets}, Joined → True, AxesOrigin → {0, 0}]
```

Out[128]=



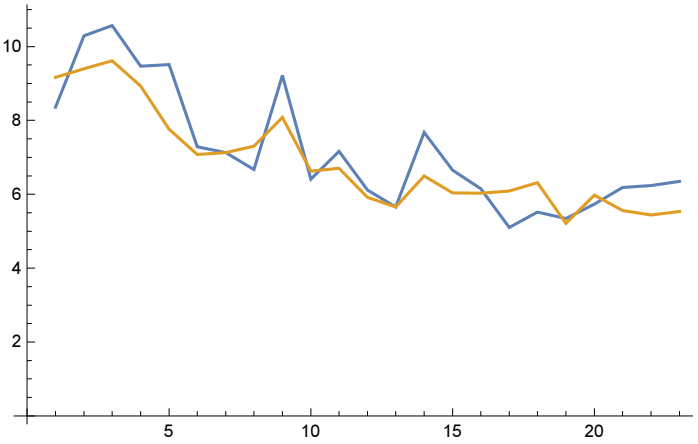
In[129]:=
```
ListPlot[{CRFourmt, FourmtNineYrRets}, Joined → True, AxesOrigin → {0, 0}]
```

Out[129]=



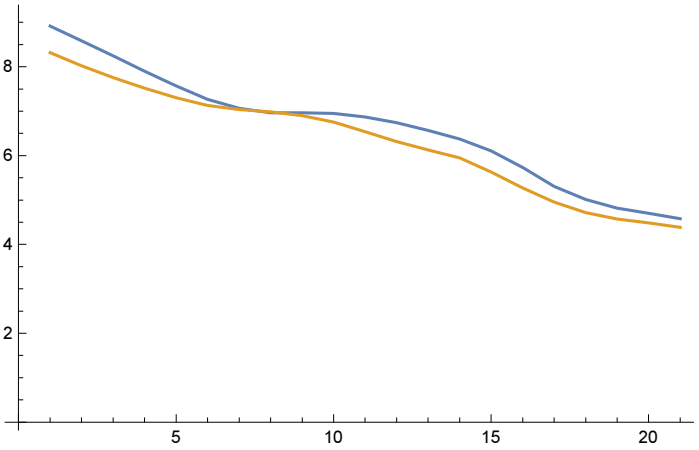
In[130]:=
```
ListPlot[{CRTwom, TwomNineYrRets}, Joined → True, AxesOrigin → {0, 0}]
```

Out[130]=



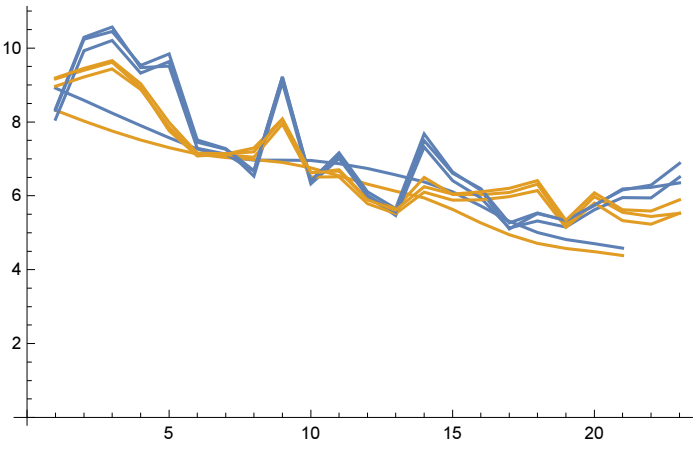
In[131]:=
```
ListPlot[{CRTrad, TradNineYrRets}, Joined → True, AxesOrigin → {0, 0}]
```

Out[131]=



In[132]:=
```
Show[%, %%, %%%, %%%%, PlotRange -> All]
```

Out[132]=



In[133]:=
```
Map[({Min[#], Max[#]}) &, {CRThreem, CRThreet,
   CRTwom, CRTwot, CRFourmt, CRCash, CRTotal, CRTrad}] // TableForm
```

Out[133]//TableForm=

5.26686	10.452
4.93864	9.88125
5.09908	10.572
4.75236	9.95498
5.1274	10.2024
0.0774807	5.87896
4.39728	9.60323
4.58118	8.91798

In[134]:=
```
NineYrStartingDates = Table[1986 + i, {i, 1, Length[Withdrawals[ThreemLogRet]]}]
NineYrStartingDatesTrad = Table[1986 + i, {i, 1, Length[Withdrawals[TradLogRet]]}]
```

Out[134]=
```
{1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997,
 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009}
```

Out[135]=
```
{1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996,
 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007}
```

In[136]:=
```
Export["Threem.dat", Transpose[{NineYrStartingDates - 1900, CRThreem}]];
Export["Threet.dat", Transpose[{NineYrStartingDates - 1900, CRThreet}]];
Export["Twom.dat", Transpose[{NineYrStartingDates - 1900, CRTwom}]];
Export["Twot.dat", Transpose[{NineYrStartingDates - 1900, CRTwot}]];
Export["Fourmt.dat", Transpose[{NineYrStartingDates - 1900, CRFourmt}]];
Export["Cash.dat", Transpose[{NineYrStartingDates - 1900, CRCash}]];
Export["Total.dat", Transpose[{NineYrStartingDates - 1900, CRTotal}]];
Export["Trad.dat", Transpose[{NineYrStartingDatesTrad - 1900, CRTrad}]];
```

In[144]:=
```
(* 1987 to 2017; has Cash but no Trad *)
Map[({Mean[#], StandardDeviation[#]}) &, {(*CRTrad,*)CRThreem,
   CRThreet, CRTwom, CRTwot, CRFourmt, CRCash, CRTotal}] // Transpose

N[Round[100 * %] / 100, 3] // TableForm // TeXForm
```

Out[144]=
```
{{7.20277, 6.81207, 7.16462, 6.72124, 7.02424, 3.43561, 6.3762},
 {1.62715, 1.58361, 1.63039, 1.58971, 1.60388, 1.66638, 1.60203}}
```

Out[145]//TeXForm=
```
\begin{array}{ccccccc}
 7.20 & 6.81 & 7.16 & 6.72 & 7.02 & 3.44 & 6.38 \\
 1.63 & 1.58 & 1.63 & 1.59 & 1.60 & 1.67 & 1.60 \\
\end{array}
```

(*
N.B.: Cash's standard deviation, at 1.67, is LARGER than any other instrument's!
This will not be true below with 1987--2015 data instead of 1987--2017 data.
*)

In[146]:=
```
choptotrad[list_] := Take[list, Length[CRTrad]]
(* 1987 to 2015; has Cash and Trad *)
SDsMeansChopped =
 Map[({StandardDeviation[choptotrad[#]], Mean[choptotrad[#]]}) &, {CRTrad,
    CRThreem, CRThreet, CRTwom, CRTwot, CRFourmt, CRCash, CRTotal}] // Transpose

N[Round[100 * {SDsMeansChopped[[2]], SDsMeansChopped[[1]]}] / 100, 3] // TableForm //
 TeXForm
```

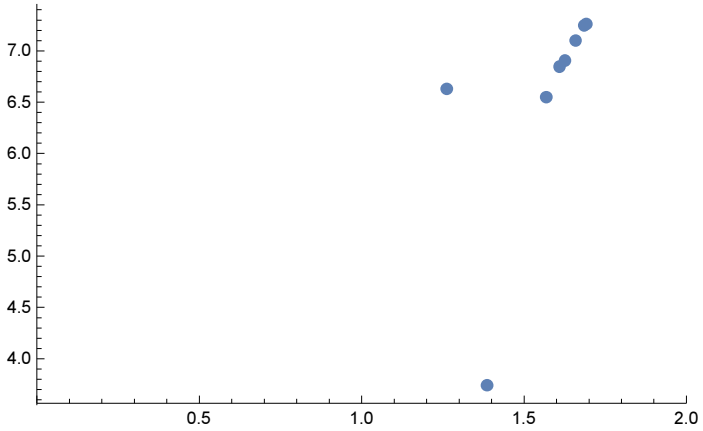
Out[147]=
```
{{1.26199, 1.6916, 1.62603, 1.68516, 1.60911, 1.65867, 1.386, 1.56849},
 {6.62993, 7.2619, 6.90543, 7.24796, 6.84686, 7.10068, 3.74172, 6.54922}}
```

Out[148]//TeXForm=
```
\begin{array}{cccccccc}
 6.63 & 7.26 & 6.91 & 7.25 & 6.85 & 7.10 & 3.74 & 6.55 \\
 1.26 & 1.69 & 1.63 & 1.69 & 1.61 & 1.66 & 1.39 & 1.57 \\
\end{array}
```

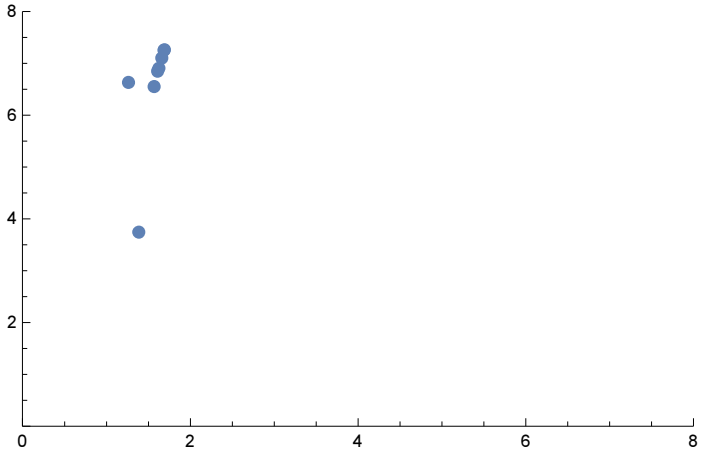
In[149]:=
```
ListPlot[Transpose[SDsMeansChopped], PlotRange → {{0, 2}, All}]
ListPlot[Transpose[SDsMeansChopped], PlotRange → {{0, 8}, {0, 8}}]

MapThread[Flatten[{#1, #2}] &,
    {Transpose[SDsMeansChopped], {"CRTrad", "CRThreem", "CRThreet", "CRTwom",
      "CRTwot", "CRFourmt", "CRCash", "CRTotal"}}] // Sort // TableForm // TeXForm
```

Out[149]=



Out[150]=



Out[151]//TeXForm=

```
\begin{array}{ccc}
 1.26199 & 6.62993 & \text{CRTrad} \\
 1.386 & 3.74172 & \text{CRCash} \\
 1.56849 & 6.54922 & \text{CRTotal} \\
 1.60911 & 6.84686 & \text{CRTwot} \\
 1.62603 & 6.90543 & \text{CRThreet} \\
 1.65867 & 7.10068 & \text{CRFourmt} \\
 1.68516 & 7.24796 & \text{CRTwom} \\
 1.6916 & 7.2619 & \text{CRThreem} \\
\end{array}
```

In[152]:=
```
MapThread[Flatten[{#1, #2}] &,
   {{"CRTrad", "CRThreem", "CRThreet", "CRTwom", "CRTwot", "CRFourmt",
     "CRCash", "CRTotal"}, Transpose[SDsMeansChopped]}] // TableForm // TeXForm
```

Out[152]//TeXForm=
```
\begin{array}{ccc}
 \text{CRTrad} & 1.26199 & 6.62993 \\
 \text{CRThreem} & 1.6916 & 7.2619 \\
 \text{CRThreet} & 1.62603 & 6.90543 \\
 \text{CRTwom} & 1.68516 & 7.24796 \\
 \text{CRTwot} & 1.60911 & 6.84686 \\
 \text{CRFourmt} & 1.65867 & 7.10068 \\
 \text{CRCash} & 1.386 & 3.74172 \\
 \text{CRTotal} & 1.56849 & 6.54922 \\
\end{array}
```

In[153]:=
```
Out[183]
```

Out[153]=
```
%183
```

In[154]:=
```
(*  I construct my own CDF function. *)
myCDF[CR_] := Module[{firstlist, secondlist}, (firstlist =
     Transpose[{Sort[CR], Table[(i - 1) / Length[CR], {i, 1, Length[CR]}]}];
   secondlist = Transpose[{Sort[CR], Table[i / Length[CR], {i, 1, Length[CR]}]}];
   Flatten[Transpose[{firstlist, secondlist}], 1])]
```

In[155]:=
```
chopCRThreem = choptotrad[CRThreem];
chopCRThreet = choptotrad[CRThreet];
chopCRTwom = choptotrad[CRTwom];
chopCRTwot = choptotrad[CRTwot];
chopCRFourmt = choptotrad[CRFourmt];
chopCRCash = choptotrad[CRCash];
chopCRTotal = choptotrad[CRTotal]
```

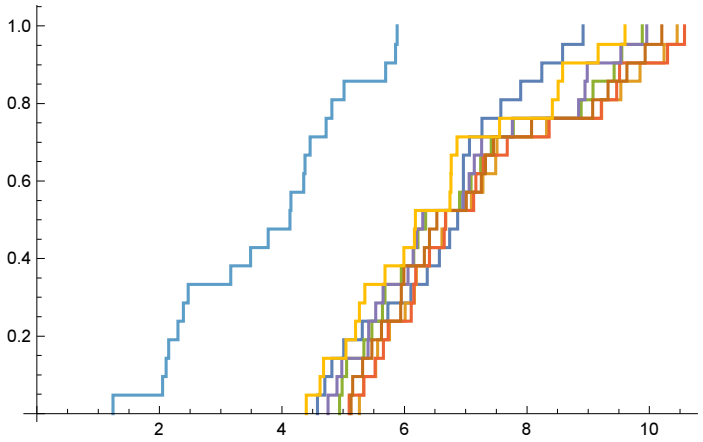
Out[161]=
```
{7.55049, 9.16438, 9.60323, 8.58449, 8.41606, 6.76652,
 6.75775, 6.17746, 8.50962, 6.1627, 6.74404, 5.99028, 5.3527, 6.85649,
 5.68496, 5.04263, 4.39728, 4.62338, 4.6798, 5.20305, 5.26627}
```

In[162]:=
```
MaxT = Apply[Max, {CRTrad, chopCRThreem, chopCRThreet,
   chopCRTwom, chopCRTwot, chopCRFourmt, chopCRCash, CRTotal}]
```

Out[162]=
```
10.572
```

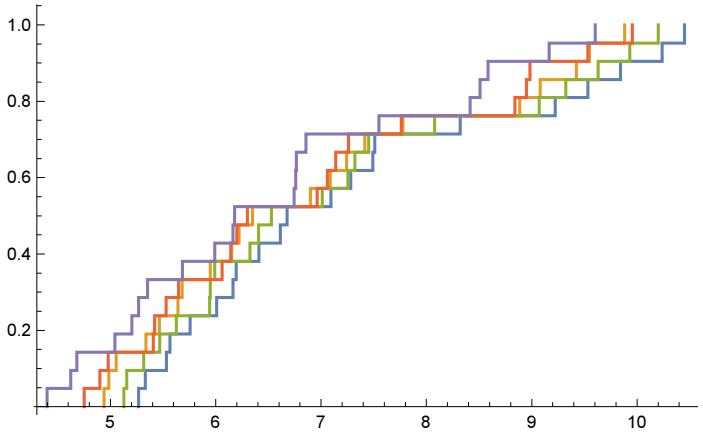
In[163]:=
```
ListPlot[{myCDF[CRTrad], myCDF[chopCRThreem], myCDF[chopCRThreet],
  myCDF[chopCRTwom], myCDF[chopCRTwot], myCDF[chopCRFourmt],
  myCDF[chopCRCash], myCDF[chopCRTotal]}, Joined → True]
(* Traditional (darker blue) is not 1DSDominated by
 anything.  Cash (lighter blue) is 1DSD by everything. *)
```

Out[163]=



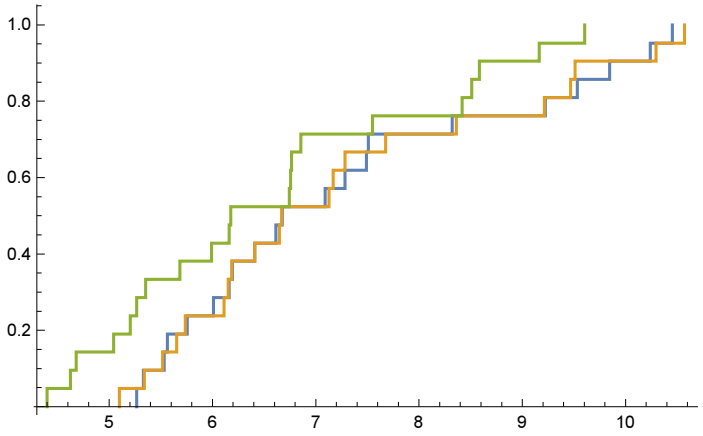
In[164]:=
```
ListPlot[{myCDF[chopCRThreem], myCDF[chopCRThreet],
  myCDF[chopCRFourmt], myCDF[chopCRTwot], myCDF[chopCRTotal]}, Joined → True]
(*3m (blue) 1DSDominates 3t, 4mt, 2t, and Total. Total (purple)
 is 1DSDominated by all of them, 3m, 3t, 4mt, and 2t.  *)
```

Out[164]=



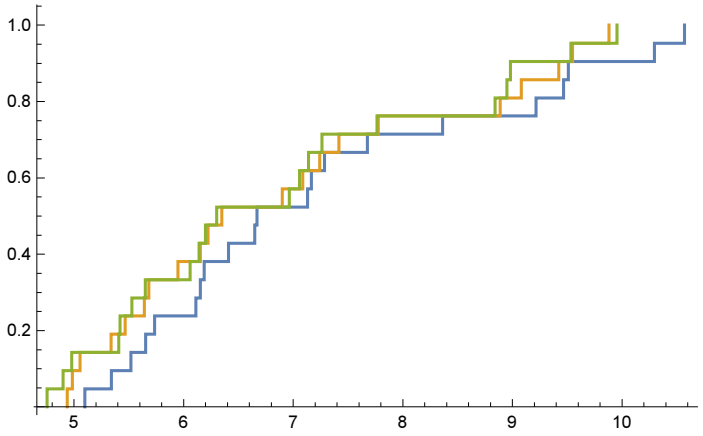
In[165]:=
```
ListPlot[{myCDF[chopCRThreem], myCDF[chopCRTwom], myCDF[chopCRTotal]},
 Joined → True] (*3m and 2m can't be ranked according
 to 1DSD.  We already knew that 3m 1DSDominates Total;
this shows that 2m also 1DSDominates Total. *)
```

Out[165]=



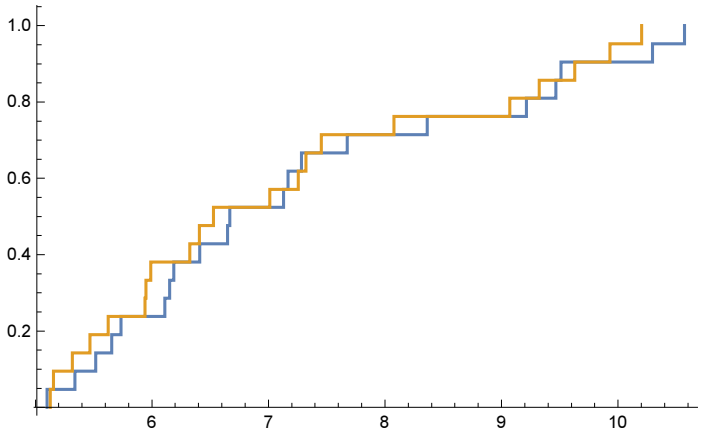
In[166]:=
```
ListPlot[{myCDF[chopCRTwom], myCDF[chopCRThreet], myCDF[chopCRTwot]},
 Joined → True] (*2m 1DSDominates 3t and 2t.  3
 t and 2t cannot be ranked according to 1DSD. *)
```

Out[166]=



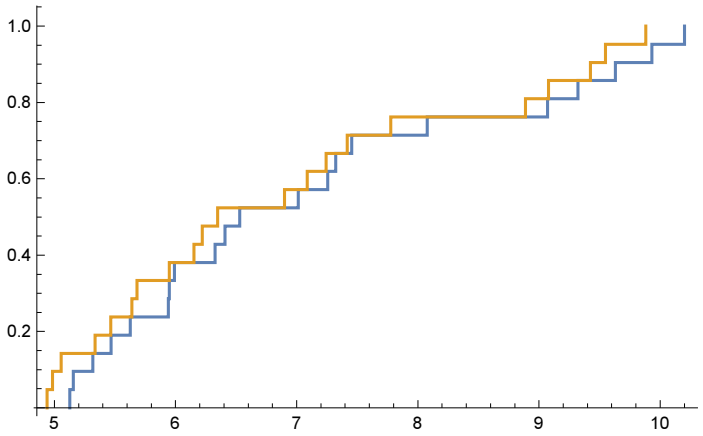
In[167]:=
```
ListPlot[{myCDF[chopCRTwom], myCDF[chopCRFourmt]}, Joined → True]
(*2m does not 1DSDominate 4mt (because of several observations). *)
```

Out[167]=



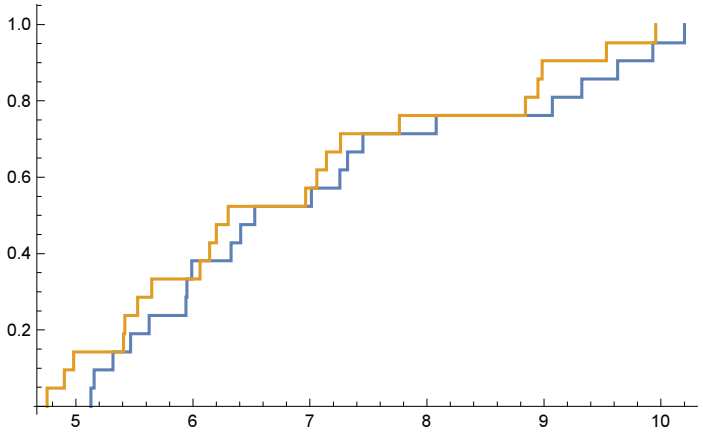
In[168]:=
```
ListPlot[{myCDF[chopCRFourmt], myCDF[chopCRThreet]}, Joined → True]
(* 4mt 1DSDominates 3t . *)
```

Out[168]=



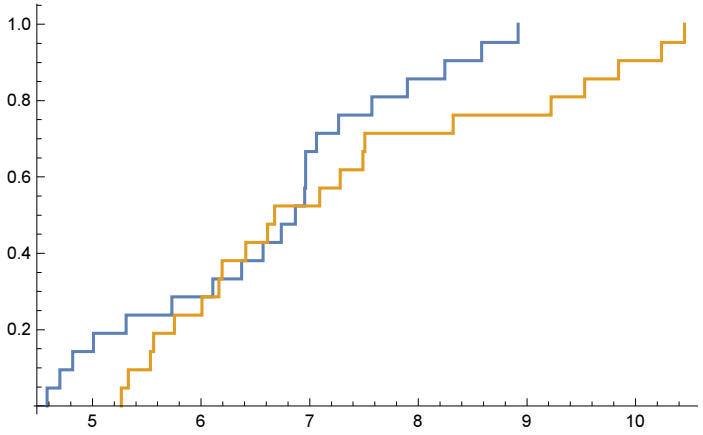
In[169]:=
```
ListPlot[{myCDF[chopCRFourmt], myCDF[chopCRTwot]}, Joined → True]
(* 4mt does not 1DSDominate 2t but only because of one data point. *)
```

Out[169]=

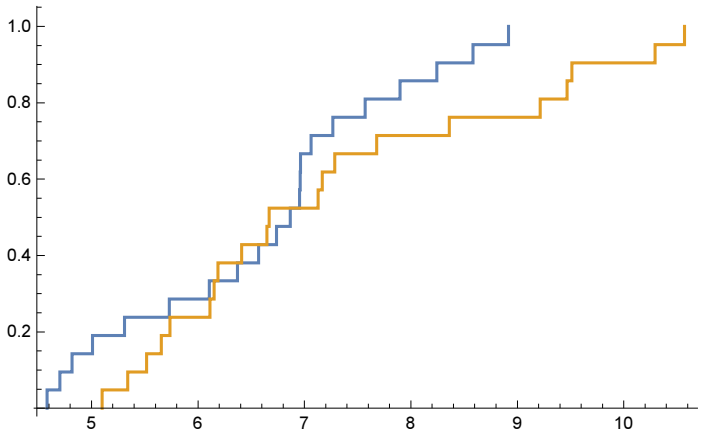


In[170]:=
```
ListPlot[{myCDF[CRTrad], myCDF[chopCRThreem]}, Joined → True]
ListPlot[{myCDF[CRTrad], myCDF[chopCRTwom]}, Joined → True]
```

Out[170]=

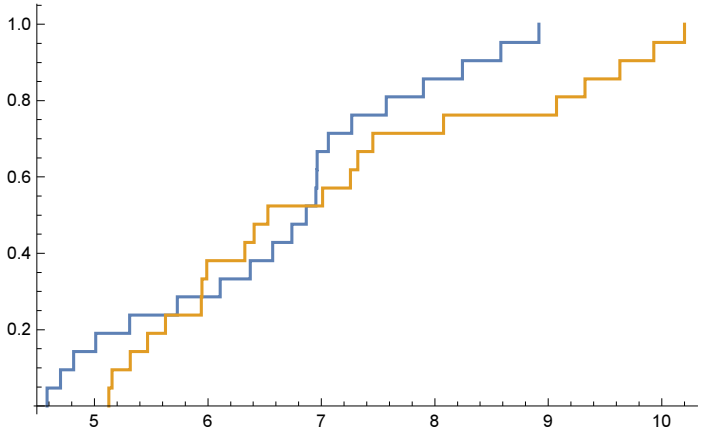


Out[171]=

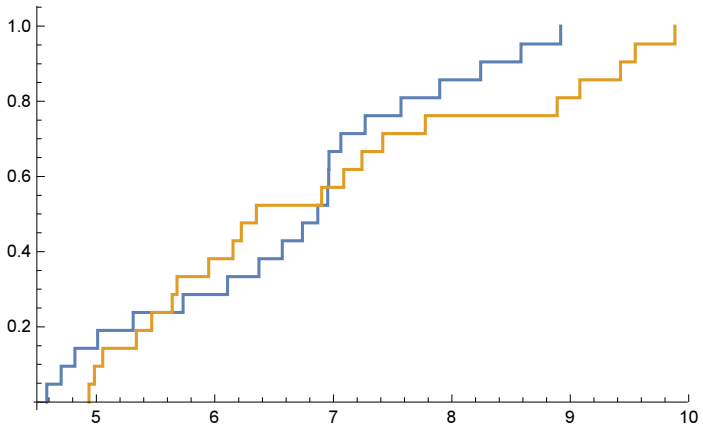


In[172]:=
```
ListPlot[{myCDF[CRTrad], myCDF[chopCRFourmt]}, Joined → True]
ListPlot[{myCDF[CRTrad], myCDF[chopCRThreet]}, Joined → True]
```

Out[172]=

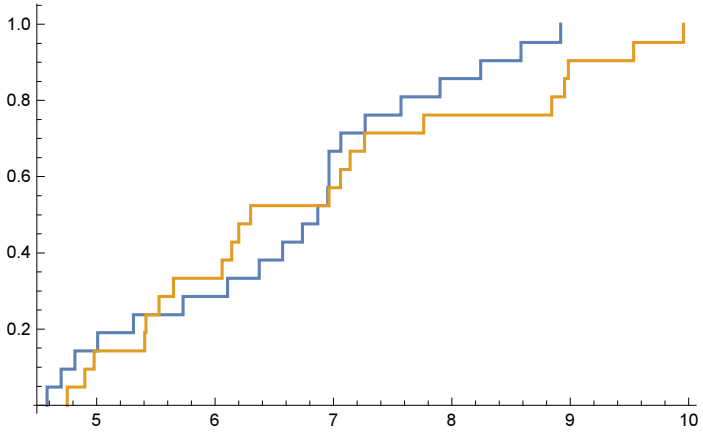


Out[173]=

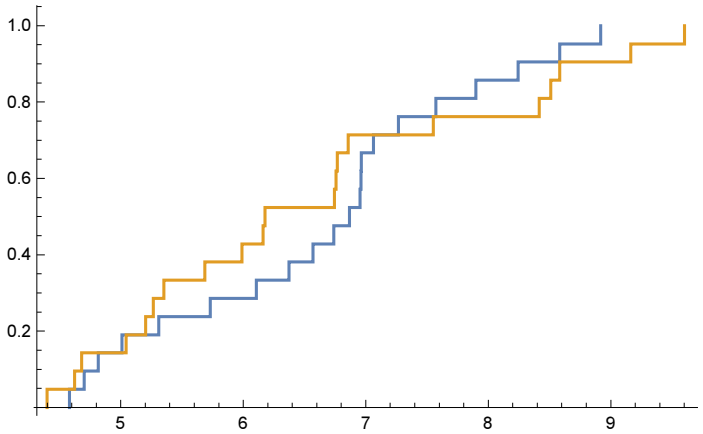


In[174]:=
```
ListPlot[{myCDF[CRTrad], myCDF[chopCRTwot]}, Joined → True]
ListPlot[{myCDF[CRTrad], myCDF[chopCRTotal]}, Joined → True]
```

Out[174]=



Out[175]=



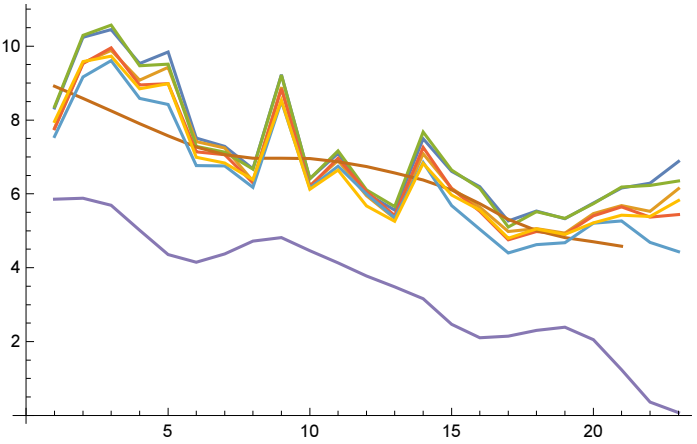
In[176]:=
```
Mixwithdrawals[alpha_?NumericQ, alternativeLogRet_] :=
  Withdrawals[SetPrecision[alpha * CashLogRet + (1 - alpha) * alternativeLogRet, 9]];
CRmix[alpha_?NumericQ, alternativeLogRet_] :=
  Map[CorrespondingRate, Mixwithdrawals[alpha, alternativeLogRet]];
```

In[178]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, ThreemLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, ThreemLogRet]], StandardDeviation[CRmix[%, ThreemLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
ListLinePlot[{CRThreem, CRThreet, CRTwom, CRTwot, CRCash, CRTrad, CRTotal,
  CRmix[thematchingalpha, ThreemLogRet]}] (* the yellow line is CRmix *)
```

Out[178]= 0.152871

Out[179]= {{6.62993, 1.55996}, {6.62993, 1.26199}}

Out[180]=



In[181]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, TwomLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, TwomLogRet]], StandardDeviation[CRmix[%, TwomLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
```

Out[181]= 0.144083

Out[182]= {{6.62993, 1.57301}, {6.62993, 1.26199}}

In[183]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, FourmtLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, FourmtLogRet]], StandardDeviation[CRmix[%, FourmtLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
```

Out[183]= 0.110456

Out[184]= {{6.62993, 1.56081}, {6.62993, 1.26199}}

In[185]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, ThreetLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, ThreetLogRet]], StandardDeviation[CRmix[%, ThreetLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
```

Out[185]= 0.0542268

Out[186]= {{6.62993, 1.56469}, {6.62993, 1.26199}}

In[187]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, TwotLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, TwotLogRet]], StandardDeviation[CRmix[%, TwotLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
```

Out[187]= 0.0279202

Out[188]= {{6.62993, 1.58147}, {6.62993, 1.26199}}

In[189]:=
```
thematchingalpha =
 alpha /. FindRoot[Mean[CRmix[alpha, TotalLogRet]] == Mean[CRTrad], {alpha, .2134}]
{{Mean[CRmix[%, TotalLogRet]], StandardDeviation[CRmix[%, TotalLogRet]]},
 {Mean[CRTrad], StandardDeviation[CRTrad]}}
```

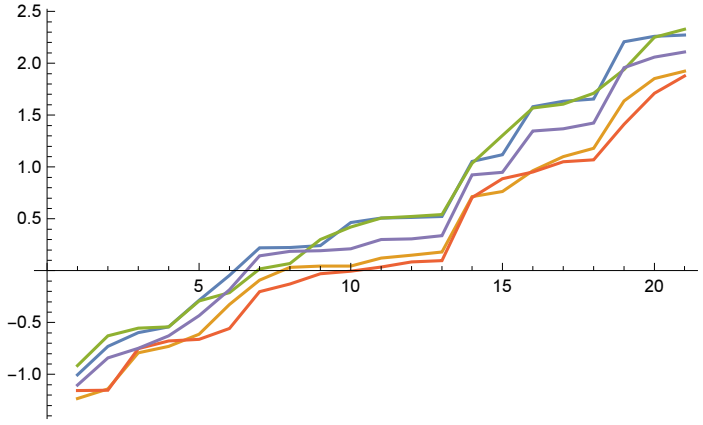
Out[189]= -0.0865881

Out[190]= {{6.62993, 1.62132}, {6.62993, 1.26199}}

(* Of course, the alpha for Total makes no sense because its return is below Traditional's. *)

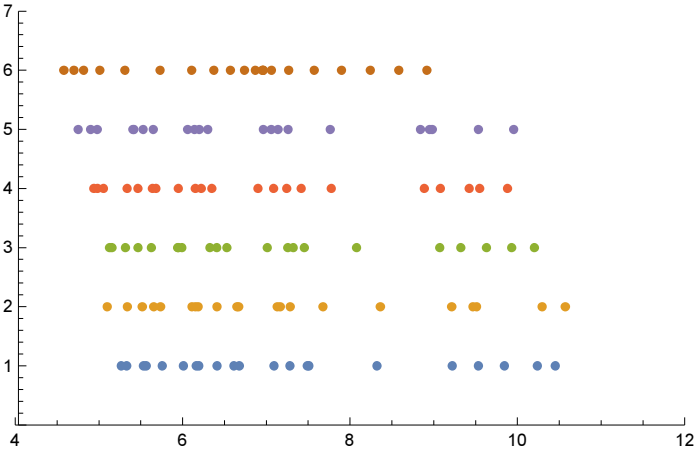
In[191]:=
```
ListLinePlot[Map[Sort, {chopCRThreem - CRTrad, chopCRThreet - CRTrad,
   chopCRTwom - CRTrad, chopCRTwot - CRTrad, chopCRFourmt - CRTrad}]]
```

Out[191]=



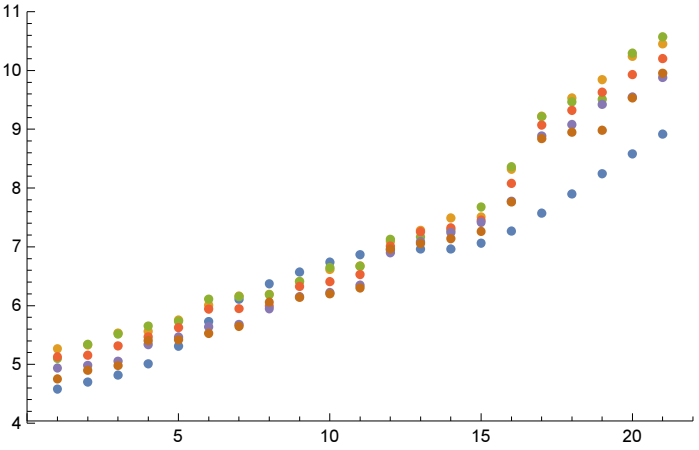
In[192]:=
```
prepMyPlot[series_, vert_] :=
 Transpose[{Sort[series], Table[vert, {i, 1, Length[series]}]}]
ListPlot[{prepMyPlot[chopCRThreem, 1], prepMyPlot[chopCRTwom, 2],
  prepMyPlot[chopCRFourmt, 3], prepMyPlot[chopCRThreet, 4],
  prepMyPlot[chopCRTwot, 5], prepMyPlot[CRTrad, 6]}, PlotRange → {{4, 12}, {0, 7}}]
```

Out[193]=



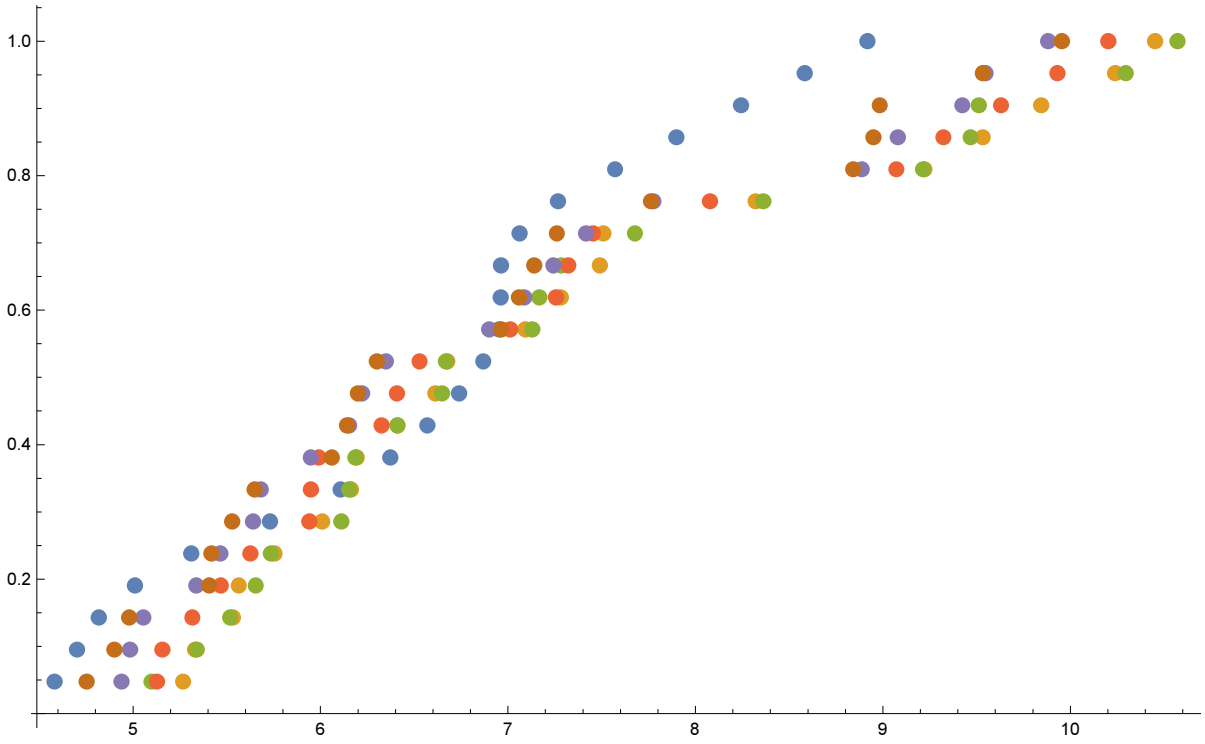
In[194]:=
```
prepMyPlot2[series_] := Transpose[{Range[Length[series]], Sort[series]}]
ListPlot[{prepMyPlot2[CRTrad], prepMyPlot2[chopCRThreem], prepMyPlot2[chopCRTwom],
  prepMyPlot2[chopCRFourmt], prepMyPlot2[chopCRThreet], prepMyPlot2[chopCRTwot]},
 PlotRange → {{0, Length[CRTrad] + 1}, {4, 11}}]
```

Out[195]=



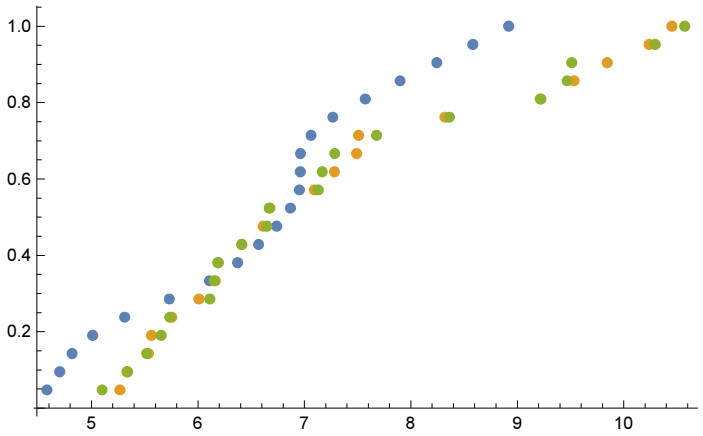
In[196]:=
```
prepMyPlot3[series_] :=
 Transpose[{Sort[series], Range[Length[series]] / Length[series]}]
ListPlot[{prepMyPlot3[CRTrad], prepMyPlot3[chopCRThreem], prepMyPlot3[chopCRTwom],
  prepMyPlot3[chopCRFourmt], prepMyPlot3[chopCRThreet], prepMyPlot3[chopCRTwot]}]
```

Out[197]=



In[198]:=
```
ListPlot[
 {prepMyPlot3[CRTrad], prepMyPlot3[chopCRThreem], prepMyPlot3[chopCRTwom]}]
```

Out[198]=



In[199]:=
```
(* Characterization of Stochastic Dominance for Discrete
Random Variable
Jean-Michel Courtault,Bertrand Crettez,Naïla Hayek
*)
f[x_Real /; MemberQ[chopCRThreem, x]] := 1 / Length[chopCRThreem]
f[x_Real /; Not[MemberQ[chopCRThreem, x]]] := 0
f[chopCRThreem[[3]]]
chopCRThreem // Sort
setOfXs = Sort[Join[chopCRThreem, CRTrad]]
f[i_Integer] := f[setOfXs[[i]]]
{f[4], f[5]}
capitalF[i_Integer] := Sum[f[j], {j, 1, i}]
capitalF[7]
```

Out[201]= $\frac{1}{21}$

Out[202]= {5.26686, 5.33123, 5.53306, 5.56374, 5.75523, 6.00912, 6.16298, 6.19335, 6.41073, 6.61273, 6.67615, 7.09258, 7.28187, 7.49021, 7.50892, 8.32099, 9.22184, 9.53276, 9.84439, 10.2392, 10.452}

Out[203]= {4.58118, 4.7015, 4.8172, 5.01027, 5.26686, 5.31067, 5.33123, 5.53306, 5.56374, 5.73078, 5.75523, 6.00912, 6.10693, 6.16298, 6.19335, 6.37251, 6.41073, 6.56993, 6.61273, 6.67615, 6.7392, 6.86802, 6.9528, 6.96151, 6.96281, 7.06195, 7.09258, 7.26695, 7.28187, 7.49021, 7.50892, 7.5715, 7.89878, 8.2433, 8.32099, 8.58281, 8.91798, 9.22184, 9.53276, 9.84439, 10.2392, 10.452}

Out[205]= $\left\{0, \frac{1}{21}\right\}$

Out[207]= $\frac{2}{21}$

In[208]:=
```
g[x_Real /; MemberQ[CRTrad, x]] := 1 / Length[CRTrad]
g[x_Real /; Not[MemberQ[CRTrad, x]]] := 0
g[i_Integer] := g[setOfXs[[i]]]
{g[4], g[5]}
capitalG[i_Integer] := Sum[g[j], {j, 1, i}]
capitalG[7]
```

Out[211]= $\left\{\frac{1}{21}, 0\right\}$

Out[213]= $\frac{5}{21}$

In[214]:=
```
{capitalF[1], capitalF[Length[setOfXs]], capitalG[1], capitalG[Length[setOfXs]]}
```

Out[214]= $\left\{0, 1, \frac{1}{21}, 1\right\}$

In[215]:=
```
capitalDsup1[i_Integer] := capitalG[i] - capitalF[i]
```

In[216]:=
```
(* the test of FOSD, their Proposition 1; it ought to fail *)
Table[capitalDsup1[i], {i, 1, Length[setOfXs]}]
Min[%] >= 0
```

Out[216]= $\left\{\frac{1}{21}, \frac{2}{21}, \frac{1}{7}, \frac{4}{21}, \frac{1}{7}, \frac{4}{21}, \frac{1}{7}, \frac{2}{21}, \frac{1}{21}, \frac{2}{21}, \frac{1}{21}, 0, \frac{1}{21}, 0, -\frac{1}{21}, 0, -\frac{1}{21}, 0, -\frac{1}{21}, -\frac{2}{21}, -\frac{1}{21}, 0, \frac{1}{21}, \frac{2}{21}, \frac{1}{7}, \frac{4}{21}, \frac{1}{7}, \frac{4}{21}, \frac{1}{7}, \frac{2}{21}, \frac{1}{21}, \frac{2}{21}, \frac{1}{7}, \frac{4}{21}, \frac{1}{7}, \frac{4}{21}, \frac{5}{21}, \frac{4}{21}, \frac{1}{7}, \frac{2}{21}, \frac{1}{21}, 0\right\}$

Out[217]= False

In[218]:=
```
capitalDsup2[i_Integer] :=
 Sum[capitalDsup1[j] * (setOfXs[[j + 1]] - setOfXs[[j]]), {j, 1, i}]
```

In[219]:=
```
(* the test of SOSD, their Proposition 2; I conjecture it should pass *)
Table[capitalDsup2[i], {i, 1, Length[setOfXs] - 1}]
Min[%] >= 0
```

Out[219]= {0.00572917, 0.0167489, 0.0443307, 0.0932039, 0.0994627, 0.10338, 0.132212, 0.135134, 0.143088, 0.145417, 0.157507, 0.157507, 0.160176, 0.160176, 0.151644, 0.151644, 0.144063, 0.144063, 0.141043, 0.135039, 0.128904, 0.128904, 0.129319, 0.129443, 0.143606, 0.149441, 0.17435, 0.177193, 0.206955, 0.208737, 0.211717, 0.242886, 0.292103, 0.306903, 0.344305, 0.408148, 0.480494, 0.539717, 0.584235, 0.621834, 0.631967}

Out[220]= True

In[221]:=
```
Clear[f, g, setOfXs, capitalF, capitalG, capitalDsup1, capitalDsup2]
doesFirstSeriesFoSoDominateSecond[firstCRseries_, secondCRseries_] :=
 Module[{f, g, setOfXs, capitalF, capitalG, capitalDsup1, capitalDsup2},
  f[x_Real /; MemberQ[firstCRseries, x]] := 1 / Length[firstCRseries];
  f[x_Real /; Not[MemberQ[firstCRseries, x]]] := 0;
  setOfXs = Sort[Join[firstCRseries, secondCRseries]];
  f[i_Integer] := f[setOfXs[[i]]];
  capitalF[i_Integer] := Sum[f[j], {j, 1, i}];
  g[x_Real /; MemberQ[secondCRseries, x]] := 1 / Length[secondCRseries];
  g[x_Real /; Not[MemberQ[secondCRseries, x]]] := 0;
  g[i_Integer] := g[setOfXs[[i]]];
  capitalG[i_Integer] := Sum[g[j], {j, 1, i}];
  capitalDsup1[i_Integer] := capitalG[i] - capitalF[i];
  capitalDsup2[i_Integer] :=
   Sum[capitalDsup1[j] * (setOfXs[[j + 1]] - setOfXs[[j]]), {j, 1, i}];
  {Min[Table[capitalDsup1[i], {i, 1, Length[setOfXs]}]] ≥ 0,
   Min[Table[capitalDsup2[i], {i, 1, Length[setOfXs] - 1}]] ≥ 0}]
```

In[223]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRTwom]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, CRTrad]
```

Out[223]= {False, False}

Out[224]= {True, True}

Out[225]= {True, True}

Out[226]= {True, True}

Out[227]= {False, True}

In[228]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, CRTrad]
```

Out[228]= {False, False}

Out[229]= {True, True}

Out[230]= {True, True}

Out[231]= {False, True}

In[232]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, CRTrad]
```

Out[232]= {True, True}

Out[233]= {False, True}

Out[234]= {False, True}

In[235]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRThreet, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRThreet, CRTrad]
```

Out[235]= {False, True}

Out[236]= {False, False}

In[237]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRTwot, CRTrad]
```

Out[237]= {False, False}

In[238]:=
```
(*doesFirstSeriesFoSoDominateSecond[chopCRThreem,chopCRThreem]*){}
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRTwom]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRThreem, CRTrad]
```

Out[238]= {}

Out[239]= {False, False}

Out[240]= {True, True}

Out[241]= {True, True}

Out[242]= {True, True}

Out[243]= {False, True}

In[244]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRThreem]
(*doesFirstSeriesFoSoDominateSecond[chopCRTwom,chopCRTwom]*){}
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRTwom, CRTrad]
```

Out[244]= {False, False}

Out[245]= {}

Out[246]= {False, False}

Out[247]= {True, True}

Out[248]= {True, True}

Out[249]= {False, True}

In[250]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRThreem]
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRTwom]
(*doesFirstSeriesFoSoDominateSecond[chopCRFourmt,chopCRFourmt]*){}
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRFourmt, CRTrad]
```

Out[250]= {False, False}

Out[251]= {False, False}

Out[252]= {}

Out[253]= {True, True}

Out[254]= {False, True}

Out[255]= {False, True}

In[256]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRThreet, chopCRThreem]
doesFirstSeriesFoSoDominateSecond[chopCRThreet, chopCRTwom]
doesFirstSeriesFoSoDominateSecond[chopCRThreet, chopCRFourmt]
(*doesFirstSeriesFoSoDominateSecond[chopCRThreet,chopCRThreet]*){}
doesFirstSeriesFoSoDominateSecond[chopCRThreet, chopCRTwot]
doesFirstSeriesFoSoDominateSecond[chopCRThreet, CRTrad]
```

Out[256]= {False, False}

Out[257]= {False, False}

Out[258]= {False, False}

Out[259]= {}

Out[260]= {False, True}

Out[261]= {False, False}

In[262]:=
```
doesFirstSeriesFoSoDominateSecond[chopCRTwot, chopCRThreem]
doesFirstSeriesFoSoDominateSecond[chopCRTwot, chopCRTwom]
doesFirstSeriesFoSoDominateSecond[chopCRTwot, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[chopCRTwot, chopCRThreet]
(*doesFirstSeriesFoSoDominateSecond[chopCRTwot,chopCRTwot]*){}
doesFirstSeriesFoSoDominateSecond[chopCRTwot, CRTrad]
```

Out[262]= {False, False}

Out[263]= {False, False}

Out[264]= {False, False}

Out[265]= {False, False}

Out[266]= {}

Out[267]= {False, False}

In[268]:=
```
doesFirstSeriesFoSoDominateSecond[CRTrad, chopCRThreem]
doesFirstSeriesFoSoDominateSecond[CRTrad, chopCRTwom]
doesFirstSeriesFoSoDominateSecond[CRTrad, chopCRFourmt]
doesFirstSeriesFoSoDominateSecond[CRTrad, chopCRThreet]
doesFirstSeriesFoSoDominateSecond[CRTrad, chopCRTwot]
(*doesFirstSeriesFoSoDominateSecond[CRTrad,CRTrad]*){}
```

Out[268]= {False, False}

Out[269]= {False, False}

Out[270]= {False, False}

Out[271]= {False, False}

Out[272]= {False, False}

Out[273]= {}

In[274]:=
```
Export["CDFTrad.dat", N[myCDF[CRTrad]]];
Export["CDF3m.dat", N[myCDF[chopCRThreem]]];
```