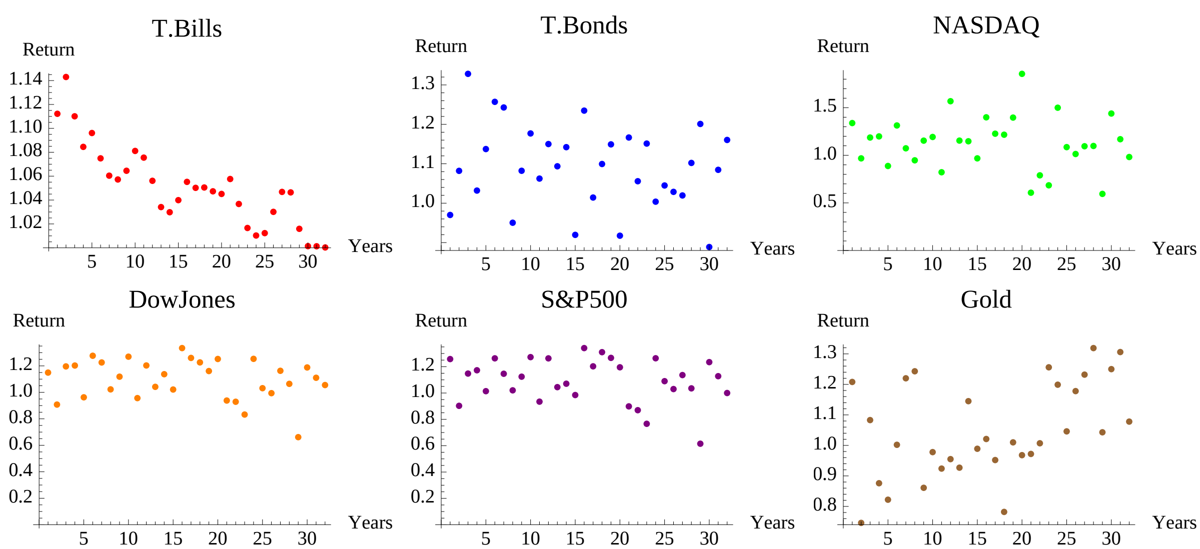
Portfolio Optimization

By Mary Catherine Roberts, Avery St. Dizier, and Joshua Vaughan


```
(*Annual return data for each investment option in percents*)

tbills = {11.22, 14.30, 11.01, 8.45, 9.61, 7.49, 6.04, 5.72, 6.45,
   8.11, 7.55, 5.61, 3.41, 2.98, 3.99, 5.52, 5.02, 5.05, 4.73, 4.51,
   5.76, 3.67, 1.66, 1.03, 1.23, 3.01, 4.68, 4.64, 1.59, .14, .13, .03};

tbonds = {-2.99, 8.20, 32.81, 3.20, 13.73, 25.71, 24.28, -4.96, 8.22, 17.69,
   6.24, 15.00, 9.36, 14.21, -8.04, 23.48, 1.43, 9.94, 14.92, -8.25, 16.66,
   5.57, 15.12, .38, 4.49, 2.87, 1.96, 10.21, 20.10, -11.12, 8.46, 16.04};

nasdaq = {33.88, -3.21, 18.67, 19.87, -11.22, 31.36, 7.36, -5.26, 15.41, 19.26,
   -17.80, 56.84, 15.45, 14.75, -3.20, 39.92, 22.71, 21.64, 39.63, 85.59, -39.29,
   -21.05, -31.53, 50.01, 8.59, 1.37, 9.52, 9.81, -40.54, 43.89, 16.91, -1.80};

dowjones = {14.93, -9.23, 19.61, 20.27, -3.74, 27.66, 22.58, 2.26, 11.85, 26.96,
   -4.34, 20.32, 4.17, 13.72, 2.14, 33.45, 26.01, 22.64, 16.10, 25.22, -6.18,
   -7.10, -16.76, 25.32, 3.15, -0.61, 16.29, 6.43, -33.84, 18.82, 11.02, 5.53};

sp500 = {25.77, -9.73, 14.76, 17.27, 1.40, 26.33, 14.62, 2.03, 12.40, 27.25,
   -6.56, 26.31, 4.46, 7.06, -1.54, 34.11, 20.26, 31.01, 26.67, 19.53, -10.14,
   -13.04, -23.37, 26.38, 8.99, 3.00, 13.62, 3.53, -38.49, 23.45, 12.78, 0.00};

gold = {20.8, -25.4, 8.3, -12.4, -17.8, 0.2, 22.0, 24.3, -13.9,
   -2.2, -7.6, -4.5, -7.3, 14.5, -1.1, 2.1, -4.8, -21.8, 1.0, -3.2,
   -2.8, 0.7, 25.6, 19.9, 4.6, 17.8, 23.2, 31.9, 4.3, 25.0, 30.6, 7.8};

(*Collective list of all data sets*)
data = {tbills, tbonds, nasdaq, dowjones, sp500, gold};

(*Converting the percents to positive decimals*)
For[i = 1, i ≤ Length[data], i++,
  For[j = 1, j ≤ Length[data[[i]]], j++,
   If[data[[i]][[j]] ≥ 0, data[[i]][[j]] = data[[i]][[j]] / 100 + 1,
    data[[i]][[j]] = 1 + data[[i]][[j]] / 100]]];
```

```
(*Graphs of the returns of each
 investment option relative to years after 1980*)
options = {"T.Bills", "T.Bonds", "NASDAQ", "DowJones", "S&P500", "Gold"};
For[i = 1; plots = {};
 colors = {Red, Blue, Green, Orange, Purple, Brown}, i ≤ Length[data], i++,
 plots = Append[plots,
   ListPlot[data[[i]], PlotStyle → colors[[i]], AxesLabel → {"Years"
, "Return"}, PlotLabel → options[[i]]]]]
Grid[Partition[plots, 3]]

(*Constructing a chart of the data by year*)
For[dates = {}; i = 1, i ≤ 32, i++, dates = Append[dates, (1979 + i)]];
Grid[Join[{Join[{""}, options]}, Join[{dates}, data] // Transpose],
 Frame → All, Background → {{Gray, None}, {LightGray, None}}]

(*Constructing a second chart with the mean, variance and standard deviation*)
Grid[Transpose[Join[{{"", "Mean", "Variance", "Standard Deviation"}},
   Transpose[Join[{options}, {Mean[data // Transpose],
      Variance[data // Transpose], StandardDeviation[data // Transpose]}]]]],
 Frame → All, Background → {{Gray, None}, {LightGray, None}}]
```



	T.Bills	T.Bonds	NASDAQ	DowJones	S&P500	Gold
1980	1.1122	0.9701	1.3388	1.1493	1.2577	1.208
1981	1.143	1.082	0.9679	0.9077	0.9027	0.746
1982	1.1101	1.3281	1.1867	1.1961	1.1476	1.083
1983	1.0845	1.032	1.1987	1.2027	1.1727	0.876
1984	1.0961	1.1373	0.8878	0.9626	1.014	0.822
1985	1.0749	1.2571	1.3136	1.2766	1.2633	1.002
1986	1.0604	1.2428	1.0736	1.2258	1.1462	1.22
1987	1.0572	0.9504	0.9474	1.0226	1.0203	1.243
1988	1.0645	1.0822	1.1541	1.1185	1.124	0.861
1989	1.0811	1.1769	1.1926	1.2696	1.2725	0.978
1990	1.0755	1.0624	0.822	0.9566	0.9344	0.924
1991	1.0561	1.15	1.5684	1.2032	1.2631	0.955
1992	1.0341	1.0936	1.1545	1.0417	1.0446	0.927
1993	1.0298	1.1421	1.1475	1.1372	1.0706	1.145
1994	1.0399	0.9196	0.968	1.0214	0.9846	0.989
1995	1.0552	1.2348	1.3992	1.3345	1.3411	1.021
1996	1.0502	1.0143	1.2271	1.2601	1.2026	0.952
1997	1.0505	1.0994	1.2164	1.2264	1.3101	0.782
1998	1.0473	1.1492	1.3963	1.161	1.2667	1.01
1999	1.0451	0.9175	1.8559	1.2522	1.1953	0.968
2000	1.0576	1.1666	0.6071	0.9382	0.8986	0.972
2001	1.0367	1.0557	0.7895	0.929	0.8696	1.007
2002	1.0166	1.1512	0.6847	0.8324	0.7663	1.256
2003	1.0103	1.0038	1.5001	1.2532	1.2638	1.199
2004	1.0123	1.0449	1.0859	1.0315	1.0899	1.046
2005	1.0301	1.0287	1.0137	0.9939	1.03	1.178
2006	1.0468	1.0196	1.0952	1.1629	1.1362	1.232
2007	1.0464	1.1021	1.0981	1.0643	1.0353	1.319
2008	1.0159	1.201	0.5946	0.6616	0.6151	1.043
2009	1.0014	0.8888	1.4389	1.1882	1.2345	1.25
2010	1.0013	1.0846	1.1691	1.1102	1.1278	1.306
2011	1.0003	1.1604	0.982	1.0553	1.	1.078

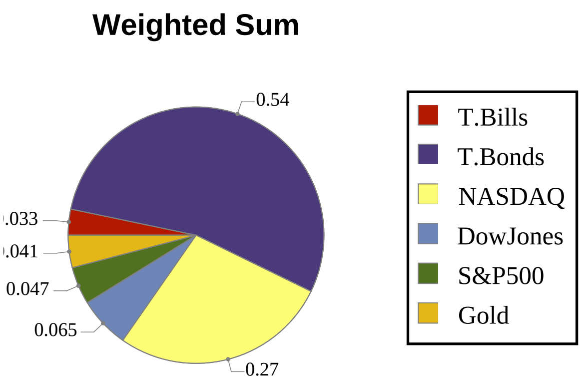
	T.Bills	T.Bonds	NASDAQ	DowJones	S&P500	Gold
Mean	1.05136	1.09216	1.12736	1.09833	1.09379	1.04994
Variance	0.00116085	0.0109041	0.0761365	0.0225648	0.0280729	0.0243316
Standard Deviation	0.0340712	0.104423	0.275928	0.150216	0.16755	0.155986

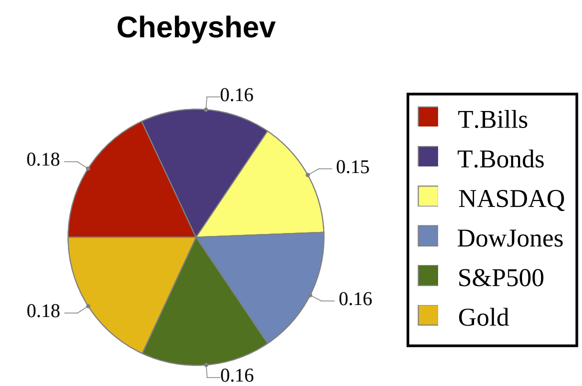
```
(*Definitions of our objective functions, goals,
weights, and other variables we will need*)

(*The historical mean return value of each investment option*)
meanReturn = Mean[Transpose[data]];

(*The covariant matrix of our data with entry aij =
 Covariance[option i, option j] and aii = Variance[option i]*)
covariants = Covariance[Transpose[data]];

(*A list of the variables representing the
 percent each variable will have in our allocation*)
variables = Map[Subscript[x, #] &, options];

(*A list of our objective functions*)
f = {variables.meanReturn, variables.covariants.variables} // Simplify;

(*A list of our goals for each objective*)
g = {1.1, 0};

(*A list of our weights for each objective*)
w = {.5, .5};

(*Our standard constraints: xi≥0 for every x and that ∑xi=1*)
constraints = Total[variables] == 1 && Apply[And, Thread[Greater[variables, 0]]];

(*The Weighted Sum of Deviations function is derived by taking a weighted
 sum of the deviations of each objective function from its goal*)
weightedSum = Sum[Abs[(f[[i]] - g[[i]])] w[[i]], {i, 1, Length[f]}] // Simplify;

(*The Chebyshev function is derived by finding the
 maximum deviation of any objective function from its goal*)
chebyshev = Max[Table[Abs[(f[[l]] - g[[l]])] w[[l]], {l, 1, Length[f]}]] // Simplify;
```

```
(*This section is our calculations using the two different methods*)

(*
                                    Weighted Sum of Deviations:
  We combine our two objective functions into a weighted average of
   their deviations from the goals using weights derived from our risk
   preferences. We minimize this function using the original constraints.
*)
results2 = Quiet[FindMinimum[{weightedSum, constraints}, variables]];

(*
                                    Chebyshev:
 We create a new function to minimize by finding
  the maximum deviation of any objective funtion from its
  respective goal and attempting to minimize just this deviation.
*)
results3 = Quiet[FindMinimum[{chebyshev, constraints}, variables]];


(*Constructing a chart summarizing the
 results of the calculations using the two methods*)
For[results = {results2, results3}; resultsD = Table[{}, {Length[results]}];
 j = 1, j ≤ Length[resultsD], j++,
 For[i = 1, i ≤ Length[variables], i++,
  resultsD[[j]] = Append[resultsD[[j]],
     PaddedForm[(variables[[i]] /. results[[j]][[2]][[i]]) * 100, {4, 2}]];]]
For[j = 1, j ≤ Length[resultsD], j++,
 For[i = 1, i ≤ Length[resultsD[[j]]], i++,
   resultsD[[j]][[i]] = StringJoin[ToString[resultsD[[j]][[i]]], "%"]];]
Grid[Join[{Join[{""}, options, {"...", "Return", "Variance"}]},
   {Join[{"Weighted Sum"}, resultsD[[1]], {"..."}, f /. results[[1]][[2]]]},
   {Join[{"Chebyshev"}, resultsD[[2]], {"..."}, f /. results[[2]][[2]]]}] //
  Transpose, Frame → All, Background → {{Gray, None}, {LightGray, None}}]
```

	Weighted Sum	Chebyshev
T.Bills	3.27%	18.02%
T.Bonds	53.96%	16.42%
NASDAQ	27.48%	14.94%
DowJones	6.47%	16.17%
S&P500	4.73%	16.36%
Gold	4.09%	18.09%
...	...	...
Return	1.09925	1.08369
Variance	0.00955972	0.00853878

```
(*Pie Charts of each allocation*)
For[name = {"Weighted Sum", "Chebyshev"};
 picharts = Table[{}, {Length[results]}]; i = 1, i ≤ Length[results], i++,
 picharts[[i]] = PieChart[variables /. results[[i]][[2]],
   ChartStyle → 61, PerformanceGoal → "Accuracy", ChartLegends → options,
   ChartLayout → "Stacked", PlotLabel → Style[name[[i]], "Title", FontSize → 14],
   LabelingFunction → (Placed[NumberForm[#, 2], "RadialCallout"] &)]]
Grid[Partition[picharts, 1]]
```

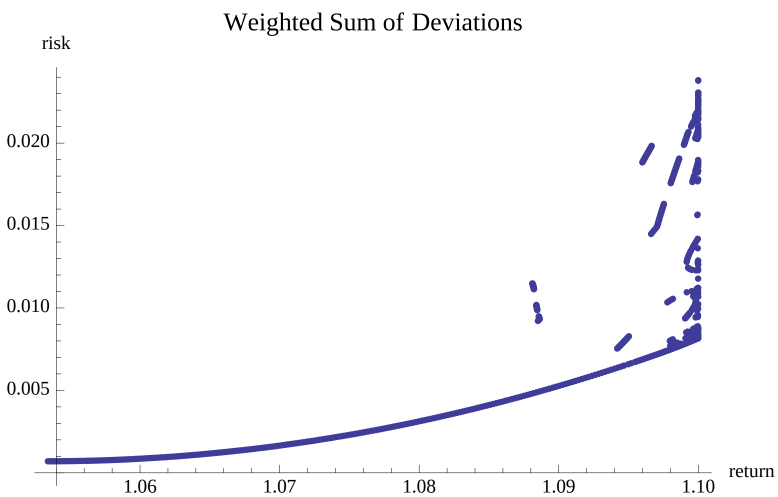


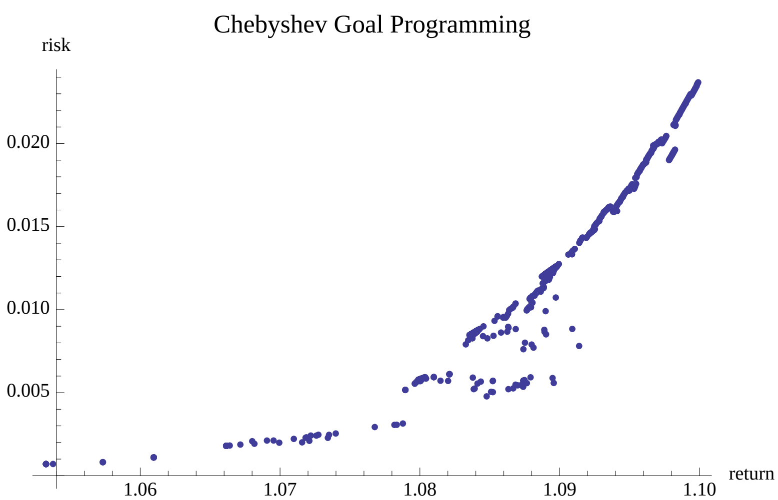


```
(*Single function versions*)
f1results = Quiet[FindMinimum[{Abs[f[[1]] - g[[1]]], constraints}, variables]];
f2results = FindMinimum[{f[[2]], constraints}, variables];

For[fresults = {f1results, f2results};
 fresultsD = Table[{}, {Length[fresults]}]; j = 1, j ≤ Length[fresultsD], j++,
 For[i = 1, i ≤ Length[variables], i++,
  fresultsD[[j]] = Append[fresultsD[[j]],
     PaddedForm[(variables[[i]] /. fresults[[j]][[2]][[i]]) * 100, {2, 2}]];]]
For[j = 1, j ≤ Length[fresultsD], j++,
 For[i = 1, i ≤ Length[fresultsD[[j]]], i++,
   fresultsD[[j]][[i]] = StringJoin[ToString[fresultsD[[j]][[i]]], "%"]];]
Grid[Join[{Join[{""}, options, {"...", "Return", "Variance"}]},
   {Join[{"min f1(x) - r*"}, fresultsD[[1]], {"..."}, f /. fresults[[1]][[2]]]},
   {Join[{"min f2(x)"}, fresultsD[[2]], {"..."}, f /. fresults[[2]][[2]]]}] //
  Transpose, Frame → All, Background → {{Gray, None}, {LightGray, None}}]
```

	min $f_1(x) - r^*$	min $f_2(x)$
T.Bills	12.00%	84.00%
T.Bonds	12.00%	3.20%
NASDAQ	39.00%	0.75%
DowJones	13.00%	0.28%
S&P500	12.00%	0.22%
Gold	12.00%	12.00%
...	...	...
Return	1.09669	1.05327
Variance	0.0198787	0.00069933

```
(*This cell plots the Pareto fronts for
 Weighted Average and Chebyshev by varying weights*)
(*IT IS NOT EVALUATABLE.*)
experiment2[w1_] :=
  Module[{},
   w = {w1, 1 - w1};
   weightedSum = Sum[Abs[(f[[i]] - g[[i]])] w[[i]], {i, 1, Length[f]}] // Simplify;
   results2 = Quiet[FindMinimum[{weightedSum, constraints}, variables]];
   f /. results2[[2]]];

experiment3[w1_] :=
  Module[{},
   w = {w1, 1 - w1};
   chebyshev =
    Max[Table[Abs[(f[[l]] - g[[l]])] w[[l]], {l, 1, Length[f]}]] // Simplify;
   Quiet[results3 = FindMinimum[{chebyshev, constraints}, variables]];
   f /. results3[[2]]];
ListPlot[Table[experiment2[i], {i, .001, .999, .001}], PlotRange → All,
 PlotLabel → "Weighted Sum of Deviations", AxesLabel → {"return", "risk"}]
ListPlot[Table[experiment3[i], {i, .001, .999, .001}], PlotRange → All,
 PlotLabel → "Chebyshev Goal Programming", AxesLabel → {"return", "risk"}]
```


Weighted Sum of Deviations
risk
0.020
0.015
0.010
0.005
1.06
1.07
1.08
1.09
1.10
return


Chebyshev Goal Programming
risk
0.020
0.015
0.010
0.005
1.06
1.07
1.08
1.09
1.10
return