AMS-512 Capital Markets and Portfolio Theory

Final Project

Spring 2020 (Online Course, COVID-19 Isolation)

Robert J. Frey, Research Professor
Stony Brook University, Applied Mathematics and Statistics

frey@ams.sunysb.edu
http://www.ams.sunysb.edu/~frey

Set-Up

Packages

```
In[•]:= Get[FileNameJoin[{NotebookDirectory[], "QuadraticProgramming.m"}]]
        Get[FileNameJoin[{NotebookDirectory[], "MeanCovMissingMLE.m"}]]
        Get[FileNameJoin[{NotebookDirectory[], "FactorFitMLE.m"}]]
```

Calendar Functions

```
In[•]:= xEndOfMonth[{y_, m_, d_}] := If[m == 12, {y, m, 31}, First@DayPlus[{y, m + 1, 1}, -1]];
```

```
In[•]:= xExpandCalendar[vxCalendar_, vxSeries_] := Sort[
          Join[
           vxSeries,
           Transpose[{#, Array[Missing[] &, Length[#]]}] &[
            Complement[vxCalendar, First /@ vxSeries]
           ]
          ]
         ];
```

Candidate Dataset

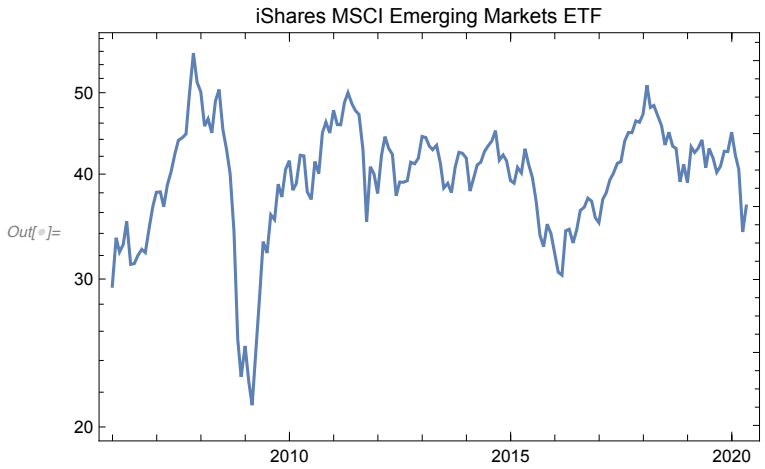
```
In[•]:= vsTickers =
         {"EEM", "EFA", "EWJ", "ICF", "IEF", "IEV", "IVV", "RWR", "SHY", "TIP", "TLT", "VTI"}
Out[•]= {EEM, EFA, EWJ, ICF, IEF, IEV, IVV, RWR, SHY, TIP, TLT, VTI}

In[•]:= vsNames = FinancialData[#, "Name"] & /@ vsTickers;
Out[•]= $Aborted

In[•]:= TableForm[{vsTickers, vsNames}^T, TableHeadings → {Range[Length[vsTickers]]}]
Out[•]//TableForm=
```

1	EEM	iShares MSCI Emerging Markets ETF
2	EFA	iShares MSCI EAFE ETF
3	EWJ	iShares MSCI Japan ETF
4	ICF	ICF International Inc
5	IEF	iShares 7-10 Year Treasury
6	IEV	iShares Europe ETF
7	IVV	iShares Core S&P 500 ETF
8	RWR	SPDR Dow Jones REIT ETF
9	SHY	iShares 1-3 Year Treasury
10	TIP	iShares TIPS
11	TLT	iShares 20 Year Treasury
12	VTI	Vanguard Total Stock Market ETF

```
In[•]:= mnDateRange = {{2005, 12, 15}, {2020, 04, 30}};

In[•]:= buffer = FinancialData[#, "Close", mnDateRange, Method → "Legacy"] & /@ vsTickers;

In[•]:= Dimensions /@ buffer
Out[•]= {{3618, 2}, {3615, 2}, {3604, 2}, {3416, 2}, {3599, 2}, {3606, 2},
          {3604, 2}, {3604, 2}, {3600, 2}, {3598, 2}, {3600, 2}, {3604, 2}}

In[•]:= And @@ (FreeQ[#, _Missing] & /@ buffer)
Out[•]= True

In[•]:= monthly = (Last /@ Split[#, (#1[[1, 2]] == #2[[1, 2]]) &]) & /@ buffer;

In[•]:= Dimensions /@ monthly
Out[•]= {{173, 2}, {173, 2}, {173, 2}, {164, 2}, {173, 2}, {173, 2},
          {173, 2}, {173, 2}, {173, 2}, {173, 2}, {173, 2}, {173, 2}}
```

In[•]:=
```
DateListLogPlot[monthly[[1]], PlotLabel → vsNames[[1]]]
```

Out[•]=



In[•]:=
```
vmxReturns = {xEndOfMonth /@ Rest[First /@ #], (Rest[#]/Most[#] - 1 &)[Last /@ #]}ᵀ & /@ monthly;
```

In[•]:=
```
Dimensions /@ vmxReturns
```

Out[•]=
```
{{172, 2}, {172, 2}, {172, 2}, {163, 2}, {172, 2}, {172, 2},
 {172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2}}
```

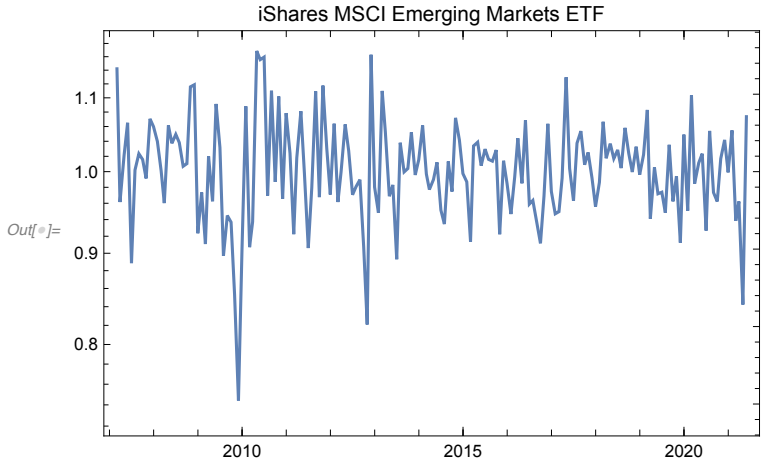
In[•]:=
```
vxCalendar = Union[Flatten[#[[All, 1]] & /@ vmxReturns, 1]];
```

In[•]:=
```
Length@vxCalendar
```

Out[•]=
```
172
```

In[•]:=
```
DateListLogPlot[1 + vmxReturns[[1]], PlotLabel → vsNames[[1]]]
```

Out[•]=



In[•]:=
```
vmxRepaired = xExpandCalendar[vxCalendar, #] & /@ vmxReturns;
```

In[•]:=
```
Dimensions /@ vmxRepaired
```

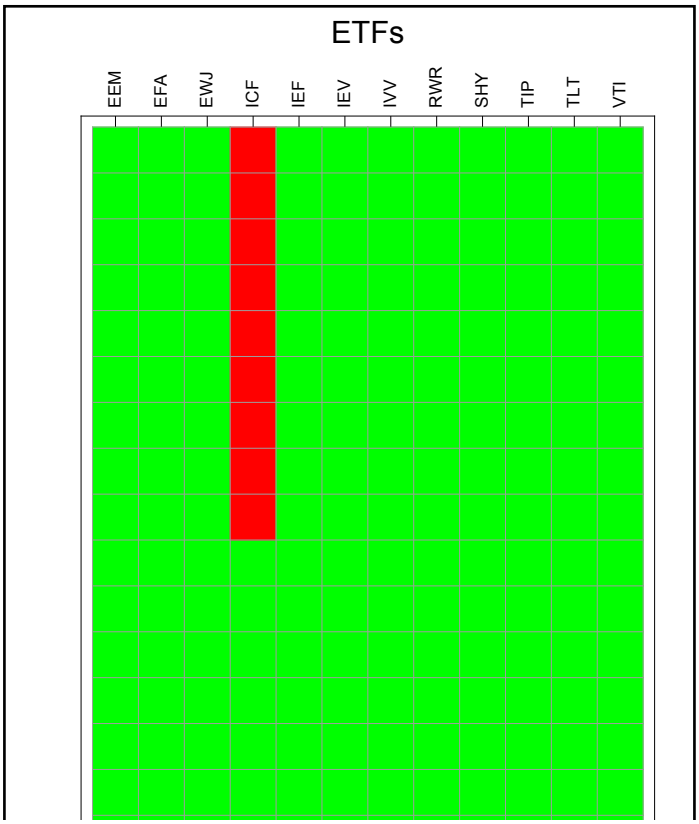
Out[•]=
```
{{172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2},
 {172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2}, {172, 2}}
```

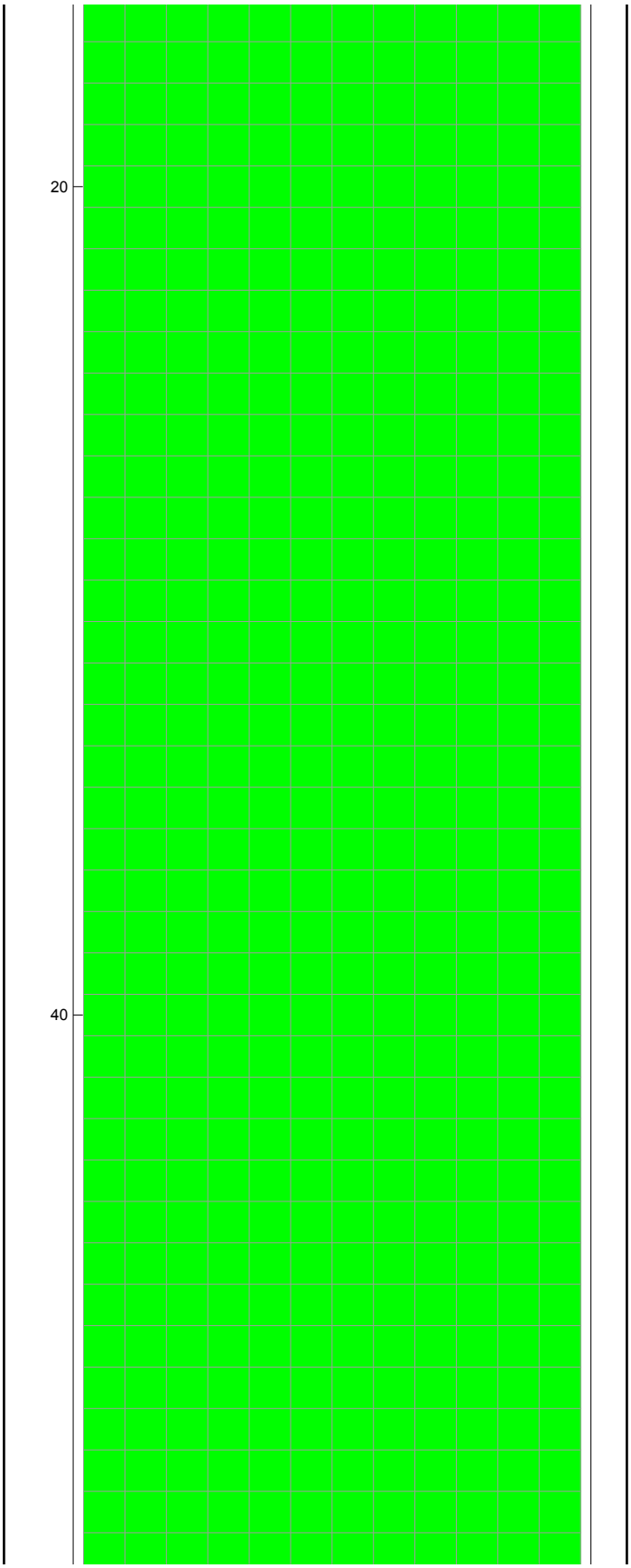
```
In[•]:= dbReturns = {
          vxCalendar,
          vsTickers,
          Transpose[#[[All, 2]] & /@ vmxRepaired]
        };

In[•]:= Export[FileNameJoin[{NotebookDirectory[], "dbReturns.m"}], dbReturns]
```

Missing Data Analysis

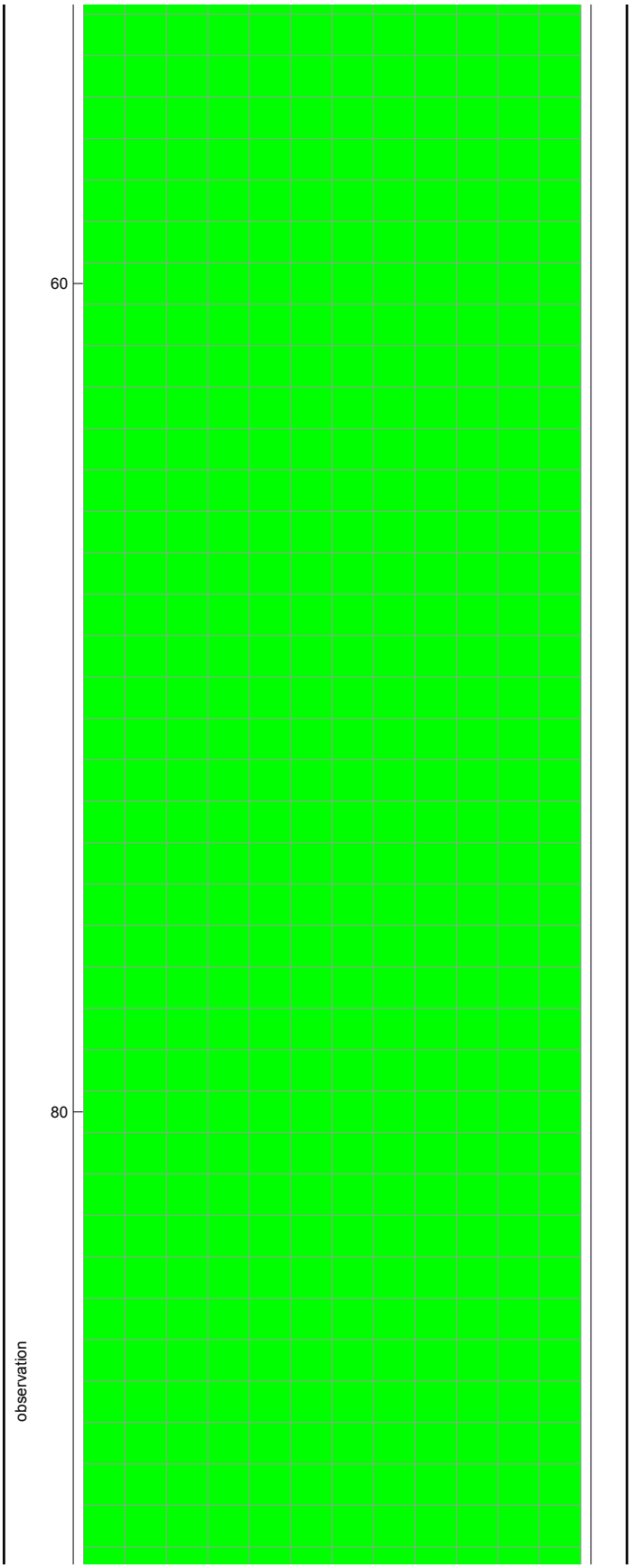
Missing Data Pattern

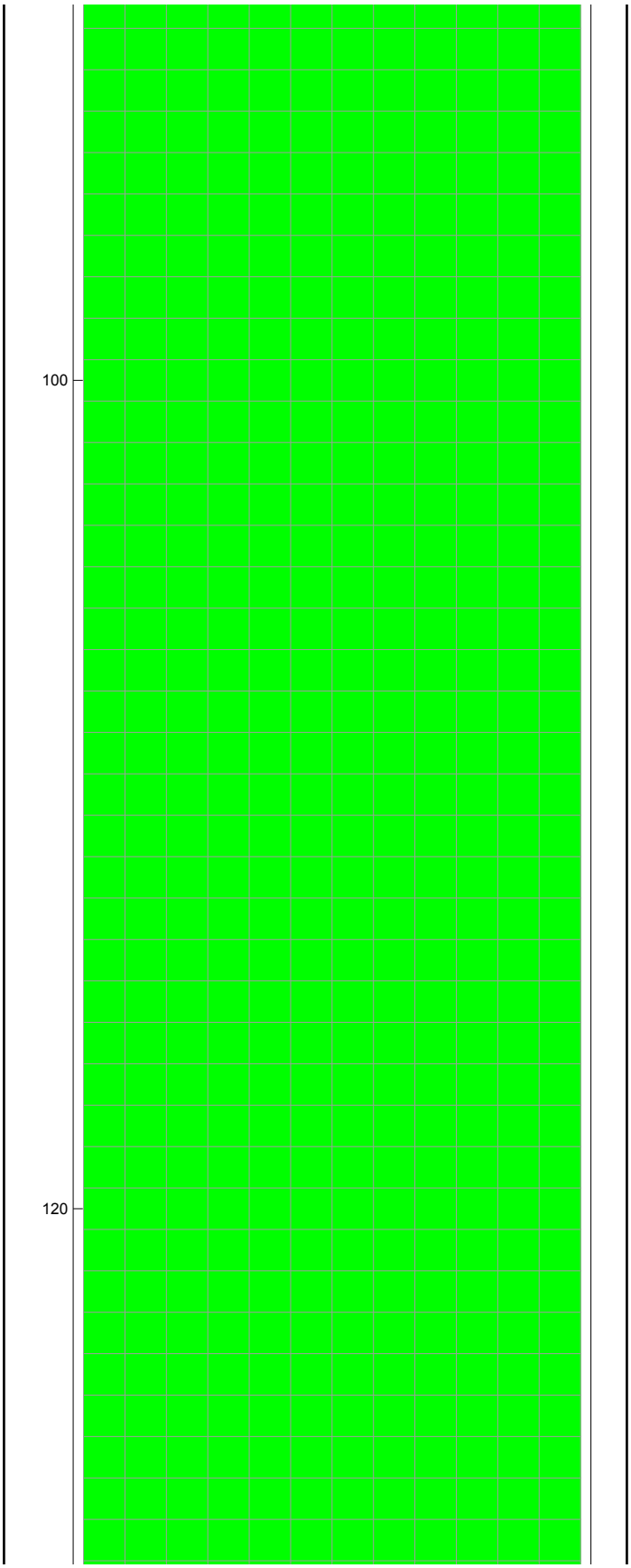
```
In[•]:= Framed@ArrayPlot[
          Map[Boole@FreeQ[#, _Missing] &, dbReturns[[3]], {2}],
          ImageSize → 350,
          Mesh -> True,
          ColorFunction → (If[# == 1, Green, Red] &),
          Frame → True ,
          FrameLabel → {{"variable", Style["ETFs", FontSize → 16]}, {"observation", ""}},
          FrameTicks → {{Rest@Range[0, Length[dbReturns[[1]]], 20], None},
            {None, {Range[Length[#]], #}ᵀ &[Rotate[#, 90 Degree] & /@ dbReturns[[2]]]}}
        ]
```

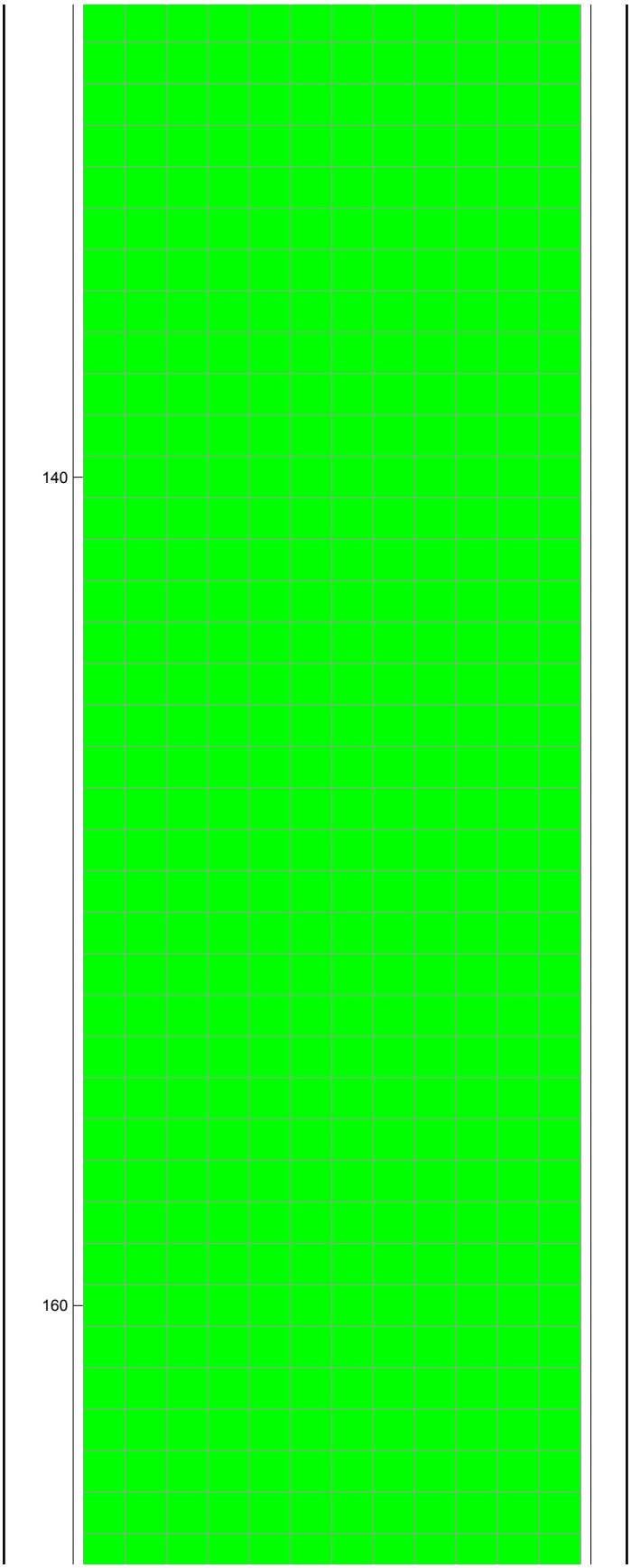


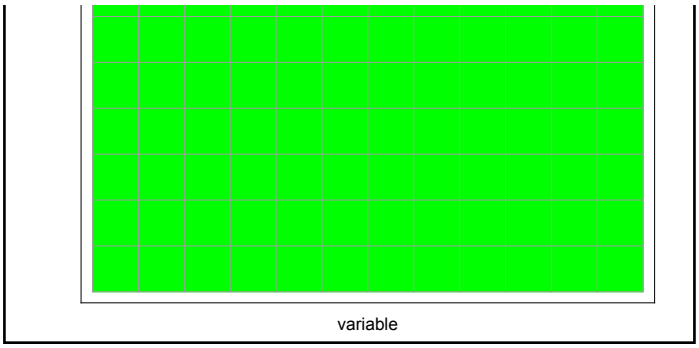

20
40

Out[•]=




100
120


140
160



Mean and Covariance Estimation

In[•]:=
```
? xMeanCovMissingMLE
```

Out[•]=

Symbol

{{Mean, Cov}, {Completed, Cross}, LLHistory, RunID} =
xMeanCovMissingMLE[dbReferenceReturns, opts] – MLE estimate of mean and cov with missing data.
Uses XEM–algorithm (extrapolation, expectation, minimization).

dbReferenceReturns – db of reference returns; missing values indicated by Missing[].
Option names are strings:
"InitialEstimate" – "Diagonal" (default), "CompleteObs", and "ImputedMean" or {Mean, Cov}.
If a string, then defines how the intial estimate is produced;
otherwise, intial starting values for the mean vector and coveariance matrix.
"ToleranceGoal" – Rate of change in log likelihood, criterion for EM termination, default 10^–6.
"MaxIterations" – Max number of iterations, criterion for EM termination, default 100.
"Extrapolation" – True to use the X–Step (default); False to use standard EM algorithm.
"CheckPoint" – Frequency at which state sufficient for restart saved; 0 (default) for no check–pointing.
"Verbose" – True to print info on each iteration, default False.

Mean – Mean vector
Cov – Covariance matrix
Completed – Data with missing values replaced by expectations based on regression estimates.
Cross – Matrix for second order correction of cross products involving missing data.
LLHistory – History of log likelihood at each EM iteration.
RunID – Unique string representing the environment and time stamp of the run.

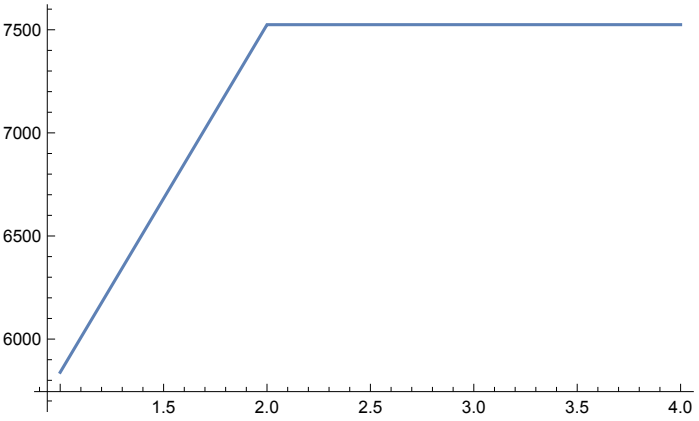
In[•]:=
```
{{vnMean, mnCovariance}, {mnCompleted, mnCross}, vnLogLik, sRunID} =
  xMeanCovMissingMLE[dbReturns];
```

In[•]:=
```
sRunID
```

Out[•]=
```
robertjfrey_MacOSX-x86-64_0_20200430_20200504025716
```

In[•]:=
```
ListPlot[vnLogLik, Joined → True, PlotRange → All]
```

Out[•]=



In[•]:=
```
TableForm[{vsTickers, vsNames, vnMean, Mean[Last /@ #] & /@ vmxReturns}ᵀ,
 TableHeadings → {Automatic, {"Tickers", "Names", "Imputed μ", "Observed μ"}}
]
```

Out[•]//TableForm=

	Tickers	Names	Imputed μ	Observed μ
1	EEM	iShares MSCI Emerging Markets ETF	0.00341402	0.00341402
2	EFA	iShares MSCI EAFE ETF	0.00104696	0.00104696
3	EWJ	iShares MSCI Japan ETF	0.000720955	0.000720955
4	ICF	ICF International Inc	0.0152929	0.0149707
5	IEF	iShares 7-10 Year Treasury	0.00232153	0.00232153
6	IEV	iShares Europe ETF	0.00111804	0.00111804
7	IVV	iShares Core S&P 500 ETF	0.00588997	0.00588997
8	RWR	SPDR Dow Jones REIT ETF	0.00327475	0.00327475
9	SHY	iShares 1-3 Year Treasury	0.000465615	0.000465615
10	TIP	iShares TIPS	0.00110197	0.00110197
11	TLT	iShares 20 Year Treasury	0.00420936	0.00420936
12	VTI	Vanguard Total Stock Market ETF	0.00603939	0.00603939

In[•]:=
```
TableForm[{vsTickers, vsNames, √Diagonal[mnCovariance],
   StandardDeviation[Last /@ #] & /@ vmxReturns}ᵀ,
 TableHeadings → {None, {"Tickers", "Names", "Imputed σ", "Observed σ"}}
]
```

Out[•]//TableForm=

Tickers	Names	Imputed σ	Observed σ
EEM	iShares MSCI Emerging Markets ETF	0.0648916	0.0650811
EFA	iShares MSCI EAFE ETF	0.0509951	0.051144
EWJ	iShares MSCI Japan ETF	0.04377	0.0438978
ICF	ICF International Inc	0.0918093	0.0922028
IEF	iShares 7-10 Year Treasury	0.0178814	0.0179336
IEV	iShares Europe ETF	0.0556086	0.0557709
IVV	iShares Core S&P 500 ETF	0.0431052	0.043231
RWR	SPDR Dow Jones REIT ETF	0.0682853	0.0684847
SHY	iShares 1-3 Year Treasury	0.00359401	0.0036045
TIP	iShares TIPS	0.0166922	0.016741
TLT	iShares 20 Year Treasury	0.0387789	0.0388922
VTI	Vanguard Total Stock Market ETF	0.0447602	0.0448909

Efficient Portfolios and Frontier with Correlation Denoising

Denoising the Correlation Matrix

Restate Covariance Using Denoised Correlation

Factor Model (Denoised Covariance)

Portfolio Optimization

Mean-Variance Efficient Frontier (Using Factor Model Covariance)

Student *t* Fit of Returns of Efficient Portfolios

Compute Expected Shortfall (CVaR) at 99% Confidence Level

Recommended approach is to write a function

```
xComputeCVaR[ret_, port_, conf_]
```

where ret is the return matrix, port the portfolio allocations and conf the confidence level.

This function then computes the portfolio returns, fits a Student *t* distribution to them, and estimates a CVaR. In the form

```
xComputeCVaR[dbReturns[[3]], #, 0.99]&
```

it can be mapped over the portfolios in the efficient frontier to produce a vector of CVaRs.

Plot Mean-Variance-CVaR Frontier