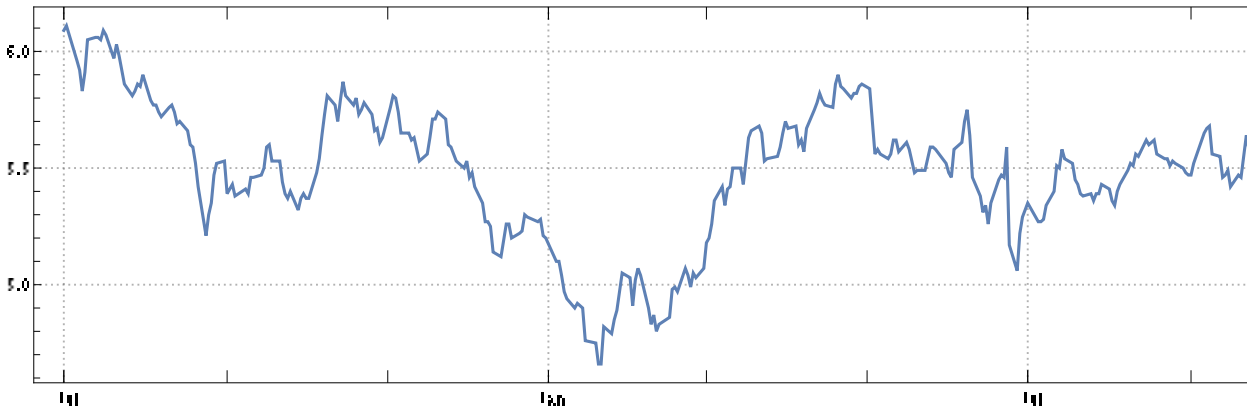
```
Import[
 "https://raw.githubusercontent.com/antononcube/MathematicaForPrediction/master/
   MonadicProgramming/MonadicQuantileRegression.m"]
ts2 = FinancialData["EUROX", {{2015, 7, 1}, {2017, 1, 1}, "Day"}];
DateListPlot[ts2, PlotTheme → "Detailed", AspectRatio → 1 / 4, ImageSize → 800]
ts2 = QRMonUnit[ts2] ⟹ QRMonTakeData;
bFuncs = Prepend[
   Flatten[Table[{Sin[b + h x], Cos[b + h x]}, {h, 1, 100, 1}, {b, 0, 1, 0.5}]], 1];
Length[bFuncs]
(*601*)
AbsoluteTiming[
 qrObj2 = QRMonUnit[ts2] ⟹ QRMonRescale ⟹ QRMonQuantileRegressionFit[bFuncs, 0.5] ⟹
    QRMonSetRegressionFunctionsPlotOptions[PlotStyle → Red] ⟹
    QRMonPlot[PlotTheme → "Detailed", AspectRatio → 1 / 4, ImageSize → Large];]
(*{24.2521,Null}*)
qFunc2 = (qrObj2 ⟹ QRMonTakeRegressionFunctions)[0.5][t];
terms = Cases[qFunc2, (f_?NumberQ * c_) :↦ {f, c}];
TakeLargestBy[terms, Abs@*First, 6]
ListPlot[terms[[All, 1]], PlotRange → All, Filling → Axis, PlotTheme → "Scientific"]
(*{{15.1087,Sin[0.+t]},{6.00884,Cos[1.+3 t]},{1.95579,Sin[0.+5 t]},
 {0.875732,Cos[1.+9 t]},{0.549675,Sin[0.+11 t]},{0.311493,Cos[1.+21 t]}}*)

largestTerms = TakeLargestBy[terms, First, 7]
(*{{15.1087,Sin[0.+t]},{6.00884,Cos[1.+3 t]},
 {1.95579,Sin[0.+5 t]},{0.875732,Cos[1.+9 t]},{0.549675,Sin[0.+11 t]},
 {0.311493,Cos[1.+21 t]},{0.218136,Sin[0.+18 t]}}*)
terms = SortBy[terms, -Abs[#[[1]]] &];
spans = {Span[1, 3], Span[4, 8], Span[9, 50]};
res = MapThread[
   Function[{terms, span}, QRMonUnit[ts2] ⟹ QRMonRescale[Axes → {True, False}] ⟹
     QRMonQuantileRegressionFit[Prepend[terms[[All, 2]], 1], 0.5] ⟹
     QRMonSetRegressionFunctionsPlotOptions[PlotStyle → Red] ⟹
     QRMonPlot[PlotTheme → "Detailed", AspectRatio → 1 / 4,
      ImageSize → Large, PlotLabel → Row[{"span: ", span}]] ⟹
     QRMonTakeRegressionFunctions], {terms[[#]] & /@ spans, spans}];
Block[{data = ts2, rData = qrObj2 ⟹ QRMonTakeData, lines, rFunc},
 rFunc = Rescale[x, {0, 1}, MinMax[data[[All, 1]]]];
 lines = Outer[{rFunc /. x → #2, #1[#2]} &, res[[All, 1]], rData[[All, 1]]];
 Show[{DateListPlot[data, PlotStyle → {GrayLevel[0.3]}, PlotTheme → "Detailed"],
   ListLinePlot[lines, PlotRange → All, PlotStyle → {Thickness[0.004]},
    PlotLegends → Map[Row[{"terms: ", #}] &, spans]]},
  ImageSize → 800, AspectRatio → 1 / 4]]
GridTableForm[Map[{#, Simplify[
     terms[[#]][[All, 2]] /. t → Rescale[t, MinMax[ts2[[All, 1]]], {0, 1}]]} &,
  spans], TableHeadings → {"span", "terms"}]
```

» Importing from GitHub: MathematicaForPredictionUtilities.m

» Importing from GitHub: MosaicPlot.m

» Importing from GitHub: CrossTabulate.m

» Importing from GitHub: QuantileRegression.m



» GetData: Cannot find data.

» QRMonBind: Failure when applying: QRMonTakeData

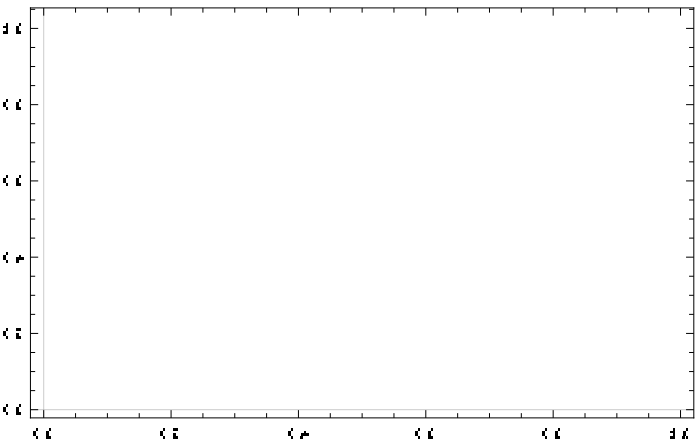
Out[] 601

Out[] {0.000044, Null}

... TakeLargestBy : The rank 6 is not an integer between 1 and 0.

In[] TakeLargestBy[{}, Abs@*First, 6]

Out[]



... TakeLargestBy : The rank 7 is not an integer between 1 and 0.

In[] TakeLargestBy[{}, First, 7]

... Part: Cannot take positions 1 through 3 in {}.

... Part: Cannot take positions 4 through 8 in {}.

... Part: Cannot take positions 9 through 50 in {}.

... General: Further output of Part::take will be suppressed during this calculation.

... Part: Part 2 of {} does not exist.

Part: The expression {}[[1 ;; 3]] cannot be used as a part specification.

Part: Part 2 of {} does not exist.

Part: The expression {}[[4 ;; 8]] cannot be used as a part specification.

Part: Part 2 of {} does not exist.

General: Further output of Part::partw will be suppressed during this calculation.

Part: The expression {}[[9 ;; 50]] cannot be used as a part specification.

General: Further output of Part::pkspec1 will be suppressed during this calculation.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Part: Part specification {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]] is longer than depth of object.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Outer: Heads Symbol and Part at positions 3 and 2 are expected to be the same.

DateListPlot: $QRMonFailure is not a valid dataset or list of datasets.

ListLinePlot: Outer[{rFunc /.
x → ##2, ##1[##2]} &, {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]], Symbol[]] is not a list of numbers or pairs of numbers.

ListLinePlot: Outer[{rFunc /.
x → ##2, ##1[##2]} &, {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]], Symbol[]] is not a list of numbers or pairs of numbers.

ListLinePlot: Outer[{rFunc /.
x → ##2, ##1[##2]} &, {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]], Symbol[]] is not a list of numbers or pairs of numbers.

General: Further output of ListLinePlot::lpn will be suppressed during this calculation.

Show: Could not combine the graphics objects in
Show[{DateListPlot[$QRMonFailure, PlotStyle → {■}, PlotTheme → Detailed], ListLinePlot[Outer[{rFunc /.
Rule[≪2≫], ##1[Slot[≪1≫]]} &, {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]],
Symbol[]], PlotRange → All, PlotStyle → {Thickness[0.004]}, PlotLegends → {terms:
Span[≪2≫], terms: Span[≪2≫], terms:
Span[≪2≫]}]}, ImageSize → 800, AspectRatio → $\frac{1}{4}$].

Outer: Heads Symbol and Part at positions 3 and 2 are expected to be the same.

Show: Could not combine the graphics objects in
Show[{DateListPlot[$QRMonFailure, PlotStyle → {■}, PlotTheme → Detailed], ListLinePlot[Outer[{rFunc /.
Rule[≪2≫], ##1[Slot[≪1≫]]} &, {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]],
Symbol[]], PlotRange → All, PlotStyle → {Thickness[0.004]}, PlotLegends → {terms:
Span[≪2≫], terms: Span[≪2≫], terms:
Span[≪2≫]}]}, ImageSize → 800, AspectRatio → $\frac{1}{4}$].

```
Show[{DateListPlot[$QRMonFailure, PlotStyle → {■}, PlotTheme → Detailed],
  ListLinePlot[Outer[{rFunc /. x → #2, #1[#2]} &,
    {$QRMonFailure, $QRMonFailure, $QRMonFailure}[[All, 1]], Symbol[]],
   PlotRange → All, PlotStyle → {Thickness[0.004]},
   PlotLegends → {terms: 1 ;; 3, terms: 4 ;; 8, terms: 9 ;; 50}]},
 ImageSize → 800, AspectRatio → 1/4]
```

Part: Cannot take positions 1 through 3 in {}.

Part: Part 2 of {} does not exist.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Divide: Infinite expression $\frac{1}{0}$ encountered.

Power: Infinite expression $\frac{1}{0}$ encountered.

Infinity: Indeterminate expression ComplexInfinity + ComplexInfinity encountered.

Part: Cannot take positions 4 through 8 in {}.

Part: Part 2 of {} does not exist.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Divide: Infinite expression $\frac{1}{0}$ encountered.

Power: Infinite expression $\frac{1}{0}$ encountered.

Infinity: Indeterminate expression ComplexInfinity + ComplexInfinity encountered.

Part: Cannot take positions 9 through 50 in {}.

General: Further output of Part::take will be suppressed during this calculation.

Part: Part 2 of {} does not exist.

General: Further output of Part::partw will be suppressed during this calculation.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

General: Further output of Symbol::argx will be suppressed during this calculation.

Divide: Infinite expression $\frac{1}{0}$ encountered.

General: Further output of Divide::infy will be suppressed during this calculation.

Power: Infinite expression $\frac{1}{0}$ encountered.

General: Further output of Power::infy will be suppressed during this calculation.

Infinity: Indeterminate expression ComplexInfinity + ComplexInfinity encountered.

General: Further output of Infinity::indet will be suppressed during this calculation.

#	span	terms
1	1 ;; 3	{}⟦1 ;; 3⟧⟦All, 2⟧
2	4 ;; 8	{}⟦4 ;; 8⟧⟦All, 2⟧
3	9 ;; 50	{}⟦9 ;; 50⟧⟦All, 2⟧