Time series analysis with Quantile Regression

Anton Antonov
MathematicaVsR project at GitHub
October, 2016

Introduction

This document (*Mathematica* notebook) is made for the *Mathematica*-part of the MathematicaVsR project “Time series analysis with Quantile Regression”.

The main goal of this document is to demonstrate how to do in *Mathematica*:

1. getting weather data (or other time series data),
2. fitting Quantile Regression (QR) curves to time series data, and
3. using QR to find outliers and conditional distributions.

Get weather data

Assume we want to obtain temperature time series data for Atlanta, Georgia, USA for the time interval from 2011.04.01 to 2016.03.31 .

We can download that weather data in the following way.

First we find weather stations identifiers in Atlanta, GA:

In[65]:=
```
Dataset@Transpose[{WeatherData[{{"Atlanta", "GA"}, 12}],
   WeatherData[{{"Atlanta", "GA"}, 12}, "StationDistance"]}]
```

Out[65]=

AN146	7.65881 km
C7927	8.08444 km
C3112	8.59898 km
KFTY	9.27331 km
D0284	9.45112 km
AR372	11.1662 km
C4296	13.3839 km
C2468	13.9523 km
KATL	14.0248 km
D4391	14.7875 km
AN142	14.8467 km
KPDK	16.8239 km

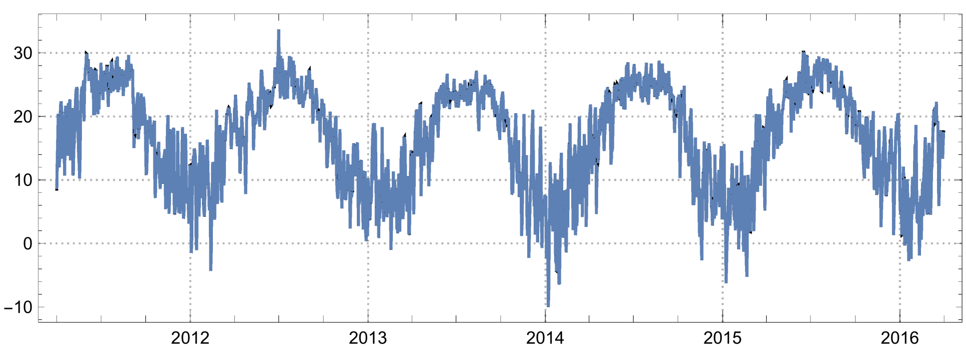
Because in the R-part of the project we used “KATL” we will use it here too.

In[66]:=
```
location = "KATL"; (*{"Atlanta","GA"}*)
{startDate, endDate} = {{2011, 4, 1}, {2016, 3, 31}};
tempData = WeatherData[location, "MeanTemperature", {startDate, endDate, "Day"}]
```

Out[67]=
```
TimeSeries[ Time: 01 Apr 2011 to 31 Mar 2016
            Data points: 1826 ]
```

In[68]:=
```
DateListPlot[tempData, PlotRange → All,
 AspectRatio → 1 / 3, PlotTheme -> "Detailed", ImageSize → 500]
```

Out[68]=



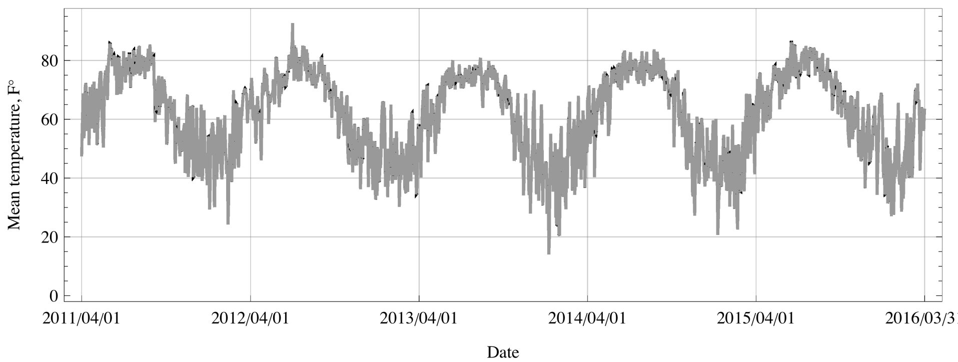
Convert to Fahrenheit in order to get results similar to those in the R-part.

In[69]:=
```
tempDataArray = tempData["Path"];
tempDataArray⟦All, 2⟧ =
  UnitConvert[Quantity[tempDataArray⟦All, 2⟧, "DegreesCelsius"],
    "DegreesFahrenheit"] /. Quantity[v_, _] ⧴ v;
```

Here we are going to plot the time series data array and re-use the obtained plot below. (Not necessary, but convenient and makes the plotting commands shorter.)

In[71]:=
```
dateTicks =
  AbsoluteTime /@ Union[Append[DateRange[{2011, 4, 1}, {2016, 3, 31}, "Month"][[
      1 ;; -1 ;; 12]], {2016, 3, 31}]];
grDLP = ListLinePlot[tempDataArray, PlotRange → All, AspectRatio → 1 / 3,
  PlotTheme → "Scientific", FrameLabel → {"Date",
    "Mean temperature, F°"},
  PlotStyle → GrayLevel[0.6],
  GridLines → {dateTicks, Automatic}, FrameTicks → {{Automatic, Automatic},
    {Map[{AbsoluteTime[#], DateString[#, {"Year", "/", "Month", "/", "Day"}]} &,
      dateTicks], None}}, ImageSize → 500]
```

Out[72]=



Fitting Quantile regression curves and finding outliers

This command loads the package [1] with QR implementations:

In[73]:=
```
Import[
 "https://raw.githubusercontent.com/antononcube/MathematicaForPrediction/master/
   QuantileRegression.m"]
```

How to use the function `QuantileRegression` from that package is explained in [2].

First we choose quantiles:

In[74]:=
```
qs = {0.02, 0.1, 0.25, 0.5, 0.75, 0.9, 0.98}
```

Out[74]=
```
{0.02, 0.1, 0.25, 0.5, 0.75, 0.9, 0.98}
```

Then we find the QR curves – called regression quantiles – at these quantiles:

In[75]:=
```
AbsoluteTiming[
 qFuncs = QuantileRegression[N@tempDataArray, 30, qs,
    Method → {LinearProgramming, Method → "CLP", Tolerance → 10^-8.}];
]
```

Out[75]=
```
{1.47838, Null}
```

At this point finding the outliers is simple – we just pick the points (dates) with temperatures higher than the 0.98 regression quantile (multiplied by some factor close to 1, like 1.005.)

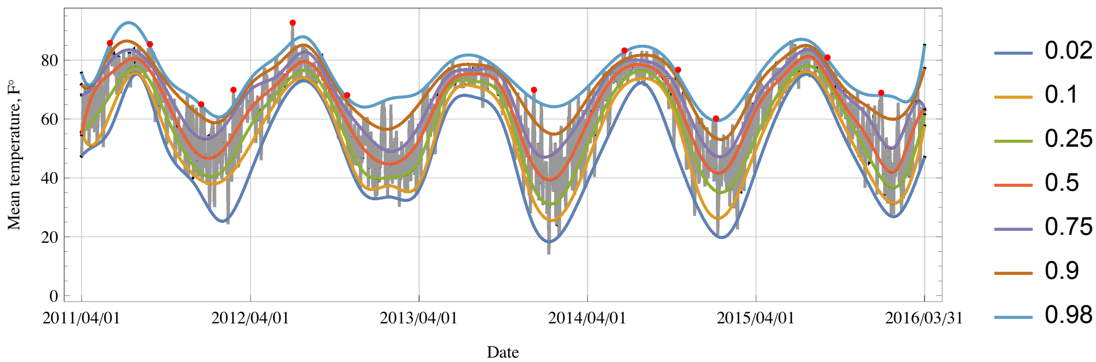
In[76]:=
```
outlierInds = Select[Range[Length[tempDataArray]],
   tempDataArray[[#, 2]] > 1.005 qFuncs[[-1]][tempDataArray[[#, 1]]] &]
```

Out[76]=
```
{62, 149, 260, 330, 458, 576, 981, 1177, 1293, 1375, 1617, 1732}
```

Plot time series data, regression quantiles, and outliers:

In[77]:=
```
Show[{
   grDLP,
   Plot[Evaluate[Through[qFuncs[x]]],
    {x, Min[tempDataArray[[All, 1]]], Max[tempDataArray[[All, 1]]]},
    PerformanceGoal → "Speed", PlotPoints → 130, PlotLegends → qs],
   ListPlot[tempDataArray[[outlierInds]], PlotStyle → {Red, PointSize[0.007]}]},
  ImageSize → 500]
```

Out[77]=



(The identified outliers are given with red points.)

Reconstruction of PDF and CDF at a given point

CDF re-construction function definitions

In[78]:=
```
Clear[CDFEstimate]
CDFEstimate[qs_, qFuncs_, t0_] :=
  Interpolation[Transpose[{Through[qFuncs[t0]], qs}], InterpolationOrder → 1];
```

Using the CDF function obtained with `CDFEstimate` we can find the PDF function by differentiation.

Plot definition

In[80]:=
```
Clear[CDFPDFPlot]
CDFPDFPlot[t0_?NumberQ, qCDFInt_InterpolatingFunction,
   qs : {_?NumericQ ..}, opts : OptionsPattern[]] :=
  Block[{},
   Plot[{qCDFInt[x], qCDFInt'[x]},
    {x, qCDFInt["Domain"]⟦1, 1⟧, qCDFInt["Domain"]⟦1, 2⟧}, PlotRange → {0, 1},
    Axes -> False, Frame → True, PlotLabel → "Estimated CDF and PDF for " <>
      DateString[t0, {"Year", ".", "Month", ".", "Day"}], opts]
  ];
```

QR with for lots of quantiles

Consider the quantiles:

In[82]:=
```
qs = Join[{0.02}, FindDivisions[{0, 1}, 20]⟦2 ;; -2⟧, {0.98}] // N
```

Out[82]=
```
{0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45,
 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.98}
```

In[83]:=
```
AbsoluteTiming[
 qFuncs = QuantileRegression[N@tempDataArray, 25, qs,
    Method → {LinearProgramming, Method → "CLP"}, InterpolationOrder → 3];
]
```

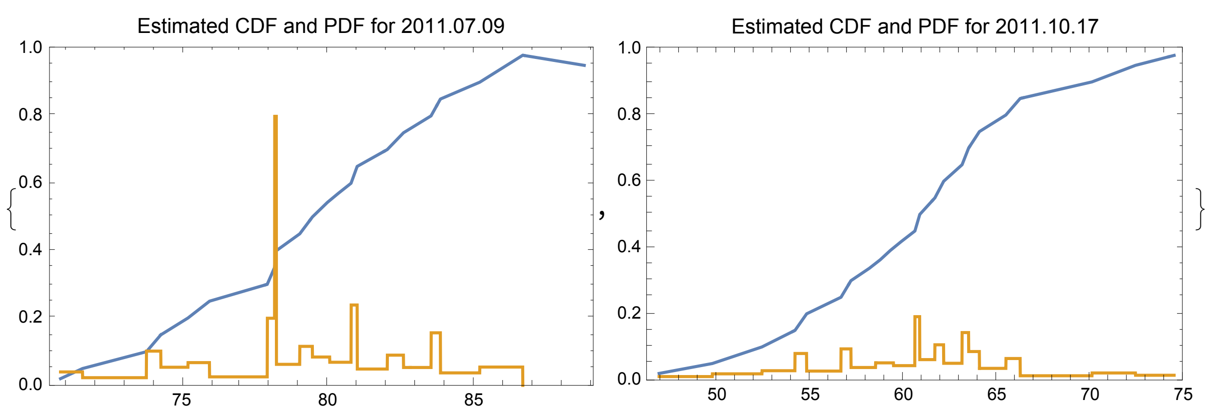
Out[83]=
```
{3.22185, Null}
```

CDF and PDF re-construction

At this point we are ready to do the reconstruction of CDF and PDF for selected dates and plot them.

In[84]:=
```
Map[CDFPDFPlot[#, CDFEstimate[qs, qFuncs, #], qs, ImageSize → 300] &,
 tempDataArray⟦{100, 200}, 1⟧]
```

Out[84]=



References

[1] Anton Antonov, Quantile regression Mathematica package, (2014), MathematicaForPrediction at GitHub, package QuantileRegression.m .

[2] Anton Antonov, “Quantile regression with B-splines”, (2014), MathematicaForPrediction at WordPress.