Common Trend and Seasonality

Application to real time series - BOR1 daily observations
(deterministic and non-deterministic components modelled together)

Initial settings

▪ packages read-in

```
<< Statistics`LinearRegression`
<< Statistics`HypothesisTests`
<< "timeseri\\timeseri.m"
<< "timeseri\\userfunc.m"
```

▪ system settings

```
SetDirectory["d:\\math\\Analyza CR\\seasonality"];
SetOptions[ListPlot, PlotJoined → True, PlotRange → All, DisplayFunction → Identity];
SetOptions[Plot, PlotRange → All, DisplayFunction → Identity];
Off[General::spell1];
```

▪ data

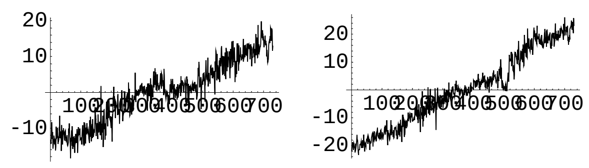
load and assign

```
ReadList["bor1c.dat", Number, RecordLists → True];
data = Transpose[%];
{y1, y2, y3} = data;
nF = 30;
n = Length[y1]
```

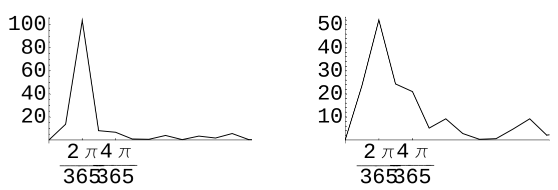
```
730
```

preliminary analysis (visual, spectral, correlation)

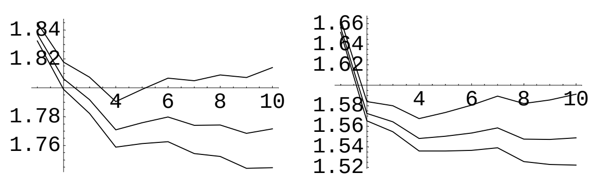
```
GraphicsArray[ListPlot /@ {y1, y2}] // fShow;
```




```
fDeterministicsRemoval[#, {0, 1}, {}] & /@ {y1, y2};
fSpectrumPlot[%, {365}, PlotRange → {{0, 2 π / 60}, All}];
fGreatestPeriods[%%, 5]
fABHQIC[%%%, 10];
```




```
{{730., 365., 243.333, 182.5, 8.58824}, {730., 365., 243.333, 182.5, 121.667}}
```



Modelling original data

■ deterministic variables

```
line1 = Table[t, {t, n}];
seas1 = N[fTrigVar[{365, 365 / 2}, n]];
```

■ Modelling separately

▫ initials

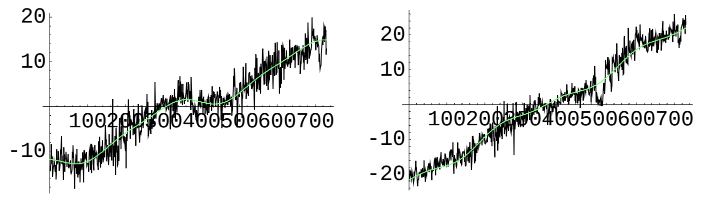
```
{mo1, mo2} = {y1, y2};
n = Length[mo1]
```

```
730
```

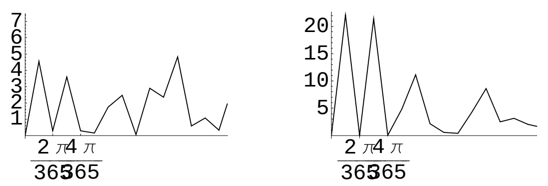
▫ diagnostics

```
mDesignM = Table[Flatten[{1, line1⟦t⟧, seas1⟦t⟧}], {t, 1, n}];
mo1r = mo1 - Flatten[mDesignM.fΦ[1, n - nF, mDesignM, mo1]];
mDesignM = Table[Flatten[{1, line1⟦t⟧, seas1⟦t⟧}], {t, 1, n}];
mo2r = mo2 - Flatten[mDesignM.fΦ[1, n - nF, mDesignM, mo2]];
```

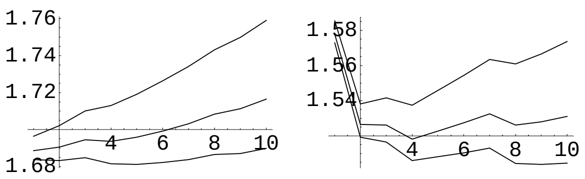
```
GraphicsArray[{Show[ListPlot[mo1], ListPlot[mo1 - mo1r, PlotStyle -> RGBColor[0.5, 1, 0.5]]],
    Show[ListPlot[mo2], ListPlot[mo2 - mo2r, PlotStyle -> RGBColor[0.5, 1, 0.5]]]}] // fShow;
```



```
fSpectrumPlot[{mo1r, mo2r}, {365}, PlotRange → {{0, 2 π / 50}, All}];
fGreatestPeriods[{mo1r, mo2r}, 5]
fABHQIC[{mo1r, mo2r}, 10];
```



```
{{730., 66.3636, 36.5, 13.5185, 8.58824}, {730., 243.333, 121.667, 66.3636, 29.2}}
```



▫ forecast

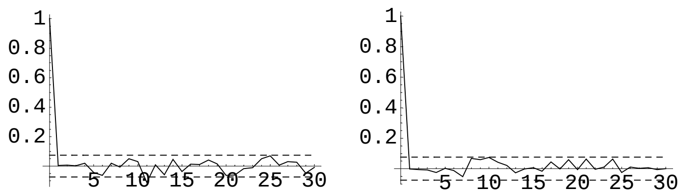
```
p = Max[p1 = {1, 2, 3, 4}, p2 = {1, 2, 3, 4}];
delayed1 = Table[mo1⟦t - p1⟧, {t, p + 1, n}];
delayed2 = Table[mo2⟦t - p2⟧, {t, p + 1, n}];
{mo1, mo2, lin1, sea1} = Drop[#, p] & /@ {mo1, mo2, line1, seas1};
n = Length[Transpose[%]]
```

```
726
```

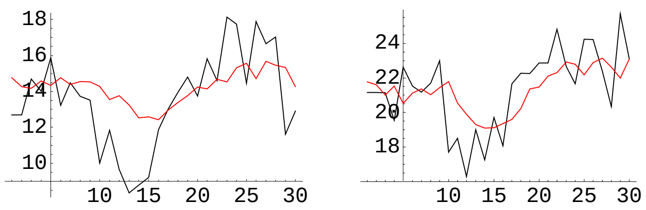
```
mDesignM = Table[Flatten[{1, lin1⟦t⟧, sea1⟦t⟧, delayed1⟦t⟧}], {t, 1, n}];
mo1m = fUnNestIfOneCol[mDesignM.fΦ[1, n, mDesignM, mo1]];
mo1f = fUnNestIfOneCol[f1stepForecast[1, n - nF + 1, n, mDesignM, mo1] ];
```

```
mDesignM = Table[Flatten[{1, lin1⟦t⟧, sea1⟦t⟧, delayed2⟦t⟧}], {t, 1, n}];
mo2m = fUnNestIfOneCol[mDesignM.fΦ[1, n, mDesignM, mo2]];
mo2f = fUnNestIfOneCol[f1stepForecast[1, n - nF + 1, n, mDesignM, mo2] ];
```

```
fCorPlot[{mo1, mo2} - {mo1m, mo2m}, 30];
```



```
GraphicsArray[{Show[ListPlot[Take[mo1, -nF]], ListPlot[mo1f, PlotStyle -> RGBColor[1, 0, 0]]],
    Show[ListPlot[Take[mo2, -nF]], ListPlot[mo2f, PlotStyle -> RGBColor[1, 0, 0]]]}] // fShow;
```




```
{y1f, y2f} = {mo1f, mo2f};
```

▫ **forecast errors**

```
{eSEPjy1, eSEPjy2} = (Take[#, -nF] & /@ {y1, y2}) - {y1f, y2f};
```

```
Dot[#, #] & /@ {eSEPjy1, eSEPjy2}
```

```
{155.036, 96.0569}
```

▪ Modelling by VAR

▫ **initials**

▫ **diagnostics**

▫ **forecast**

▫ **forecast errors**

```
{eVARjy1, eVARjy2} = (Take[#, -nF] & /@ {y1, y2}) - {y1f, y2f};
```

```
Dot[#, #] & /@ {eVARjy1, eVARjy2}
```

```
{156.068, 93.4208}
```

Trend-transformation

▪ initials

```
ty12 = Transpose[{y1, y2}];
n = Length[ty12]
dty12 = Drop[ty12, 1] - Drop[ty12, -1];
```

```
730
```

```
p = Max[p1 = {1, 2, 3, 4}];
z = {
   Table[dty12⟦t - 1⟧, {t, p + 1, n - nF}],
   Table[Flatten[{1, seas1⟦t⟧, dty12⟦(t - 1) - Drop[p1, -1]⟧}], {t, p + 1, n - nF}],
   Table[Flatten[{ty12⟦t - p⟧}], {t, p + 1, n - nF}]
  };
```

```
n = Length[ty12] - p - nF
```

```
696
```

```
M = Table[Sum[Outer[Times, z⟦i, t⟧, z⟦j, t⟧], {t, 1, n}] / n, {i, 1, 3}, {j, 1, 3}];
S = Table[M⟦i, j⟧ - M⟦i, 2⟧.Inverse[M⟦2, 2⟧].M⟦2, j⟧, {i, 1, 3}, {j, 1, 3}];
```

▪ Cointegration relations

eigenvectors corresponding to decreasingly ordered eigenvalues

```
{eigval, eigvec} = Eigensystem[{S⟦3, 1⟧.Inverse[S⟦1, 1⟧].S⟦1, 3⟧, S⟦3, 3⟧}];
Ordering[eigval, All, Greater];
{eigval, eigvec} = #⟦%⟧ & /@ {eigval, eigvec}
```

```
{{0.134888, 0.000929257}, {{-0.855007, 0.518617}, {-0.443686, -0.896182}}}
```

Likelihood ratio (trace, λ-max) test statistic (to be compared with the 3rd case critical values for johansen's test)

```
fLRtrace[λ_, r_] := -n * Sum[Log[1 - λ⟦i⟧], {i, r + 1, 2}];
fLRλmax[λ_, r_] := -n * Log[1 - λ⟦r + 1⟧];
```

```
fLRtrace[eigval, #] & /@ {0, 1}
fLRλmax[eigval, #] & /@ {0, 1}
```

```
{101.495, 0.647064}
```

```
{100.848, 0.647064}
```

initialization of trend-transformation matrix

```
mTransfT = {eigvec⟦1⟧};
```

■ Common trends

```
{eigval, eigvec} = Eigensystem[{S⟦1, 3⟧.Inverse[S⟦3, 3⟧].S⟦3, 1⟧, S⟦1, 1⟧}];
Ordering[eigval, All, Greater];
{eigval, eigvec} = #⟦%⟧ & /@ {eigval, eigvec}
```

```
{{0.134888, 0.000929257}, {{-0.814746, 0.579818}, {0.499794, 0.866144}}}
```

trend-transformation matrix

```
mTransfT = Prepend[mTransfT, eigvec⟦2⟧];
```

■ Forward transformation

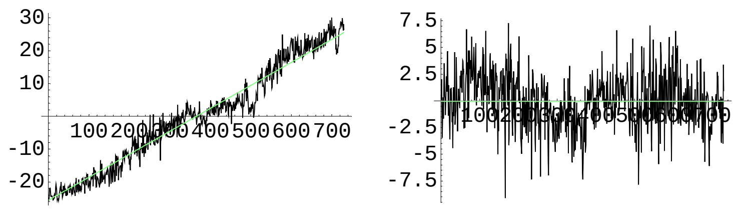
```
{u1, u2} = mTransfT.{y1, y2};
```

■ Constant and trend extraction

```
n = Length[y1]
mDesignM = Table[Flatten[{1, line1⟦t⟧, seas1⟦t⟧}], {t, n}];
u1mCT = Flatten[mDesignM⟦All, {1, 2}⟧.fΦ[1, n - nF, mDesignM, u1]⟦{1, 2}⟧];
mDesignM = Table[Flatten[{1, seas1⟦t⟧}], {t, n}];
u2mC = Flatten[mDesignM⟦All, {1}⟧.fΦ[1, n - nF, mDesignM, u2]⟦{1}⟧];
```

```
730
```

```
GraphicsArray[{Show[ListPlot[u1], ListPlot[u1mCT, PlotStyle -> RGBColor[0.5, 1, 0.5]]],
    Show[ListPlot[u2], ListPlot[u2mC, PlotStyle -> RGBColor[0.5, 1, 0.5]]]}] // fShow;
```



■ Modelling separately

▫ initials

▫ diagnostics

▫ forecast

■ Backward transformations and forecast errors

```
{y1f, y2f} = Inverse[mTransfT].{u1f, u2f};
```

```
{eTjy1, eTjy2} = (Take[#, -nF] & /@ {y1, y2}) - {y1f, y2f};
Dot[#, #] & /@ {eTjy1, eTjy2}
```

```
{134.095, 94.7376}
```

Season-transformation (of trend-transformed series)

■ initials

```
u1r = u1 - u1mCT; u2r = u2 - u2mC;
tu12 = Transpose[{u1r, u2r}];
n = Length[tu12]
dtu12 = Drop[tu12, 1] - Drop[tu12, -1];
```

```
730
```

```
p = Max[p1 = {1, 2, 3, 4}];
z = {
   Table[dtu12⟦t - 1⟧, {t, p + 1, n - nF}],
   Table[Flatten[{dtu12⟦(t - 1) - Drop[p1, -1]⟧}], {t, p + 1, n - nF}],
   Table[Flatten[{seas1⟦t⟧}], {t, p + 1, n - nF}]
  };
```

```
n = Length[tu12] - p - nF
```

```
696
```

```
M = Table[Sum[Outer[Times, z⟦i, t⟧, z⟦j, t⟧], {t, 1, n}] / n, {i, 1, 3}, {j, 1, 3}];
S = Table[M⟦i, j⟧ - M⟦i, 2⟧.Inverse[M⟦2, 2⟧].M⟦2, j⟧, {i, 1, 3}, {j, 1, 3}];
```

■ Cointegration relations

eigenvectors corresponding to increasingly ordered eigenvalues

```
{eigval, eigvec} = Eigensystem[ Inverse[S⟦1, 1⟧].S⟦1, 3⟧.Inverse[S⟦3, 3⟧].S⟦3, 1⟧ ];
Ordering[eigval, All, Less];
{eigval, eigvec} = #⟦%⟧ & /@ {eigval, eigvec}
```

```
{{0.000325787, 0.00121306}, {{0.439992, 0.898002}, {0.935795, -0.352545}}}
```

Likelihood ratio test statistic for deterministic seasonality test with 90% quantile of $\chi^2(\, r\,[2\,(4-1)+r]\,)$ distribution

```
fLRseas[λ_, r_] := -(n - p) * Sum[Log[1 - λ⟦i⟧], {i, 1, r}];
```

```
{fLRseas[eigval, #], Quantile[ChiSquareDistribution[# * (2 * (4 - 1) + #)], 0.90]} & /@ {1, 2}
```

```
{{0.225481, 12.017}, {1.06543, 23.5418}}
```

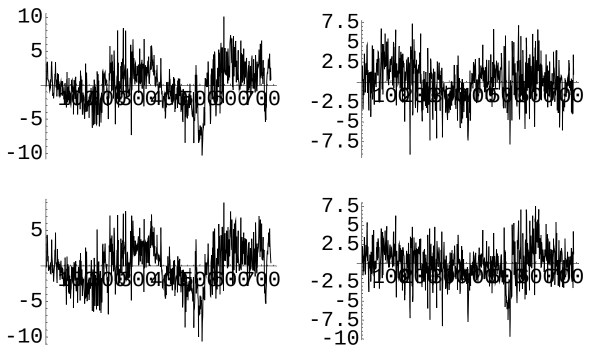
```
mTransfS = Reverse[eigvec];
```

■ Forward transformation

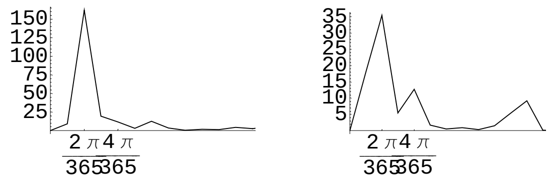
```
{v1, v2} = mTransfS.{u1r, u2r};
```

□ visual check and spectral diagnostic

```
GraphicsArray[{ListPlot /@ {u1r, u2r}, ListPlot /@ {v1, v2}}] // fShow;
```

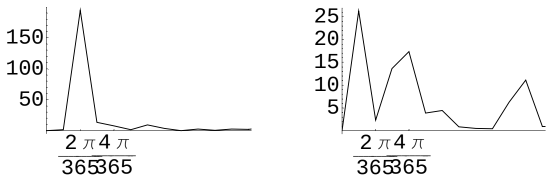


```
fSpectrumPlot[{u1r, u2r}, {365}, PlotRange → {{0, 2 π / 60}, All}];
fGreatestPeriods[{u1r, u2r}, 5]
```



```
{{730., 365., 243.333, 182.5, 121.667}, {730., 365., 182.5, 66.3636, 8.58824}}
```

```
fSpectrumPlot[{v1, v2}, {365}, PlotRange → {{0, 2 π / 60}, All}];
fGreatestPeriods[{v1, v2}, 5]
```



```
{{365., 243.333, 182.5, 121.667, 19.7297}, {730., 243.333, 182.5, 66.3636, 8.58824}}
```

■ Modelling separately

▫ initials

▫ diagnostics

▫ forecast

■ Backward transformations and forecast errors

```
{u1f, u2f} = Take[#, -nF] & /@ {u1mCT, u2mC} + Inverse[mTransfS].{v1f, v2f};
{y1f, y2f} = Inverse[mTransfT].{u1f, u2f};

{eTSjy1, eTSjy2} = (Take[#, -nF] & /@ {y1, y2}) - {y1f, y2f};
Dot[#, #] & /@ {eTSjy1, eTSjy2}

{127.863, 101.692}
```

Season-transformation (of original trend-free series)

■ Constant and trend extraction

■ initials

■ Cointegration relations

eigenvectors corresponding to increasingly ordered eigenvalues

```
{eigval, eigvec} = Eigensystem[ Inverse[S⟦1, 1⟧].S⟦1, 3⟧.Inverse[S⟦3, 3⟧].S⟦3, 1⟧ ];
Ordering[eigval, All, Less];
{eigval, eigvec} = #⟦%⟧ & /@ {eigval, eigvec}

{{0.000325426, 0.00121291}, {{-0.543472, 0.839428}, {0.774544, 0.63252}}}
```

Likelihood ratio test statistic for deterministic seasonality test with 90% quantile of $\chi^2(\, r\,[2\,(4-1)+r]\,)$ distribution

```
fLRseas[λ_, r_] := -(n - p) * Sum[Log[1 - λ⟦i⟧], {i, 1, r}];

{fLRseas[eigval, #], Quantile[ChiSquareDistribution[# * (2 * (4 - 1) + #)], 0.90]} & /@ {1, 2}

{{0.225231, 12.017}, {1.06507, 23.5418}}

mTransfS = Reverse[eigvec];
```

■ Forward transformation

```
{w1, w2} = mTransfS.{y1r, y2r};
```

▫ visual check and spectral diagnostic

■ Modelling separately

▫ initials

▫ diagnostics

▫ forecast

■ Backward transformations and forecast errors

```
{y1f, y2f} = Take[#, -nF] & /@ {y1mCT, y2mCT} + Inverse[mTransfS].{w1f, w2f};
```

```
{eSjy1, eSjy2} = (Take[#, -nF] & /@ {y1, y2}) - {y1f, y2f};
Dot[#, #] & /@ {eSjy1, eSjy2}
```

```
{128.447, 101.421}
```

Comparing

■ hit parade

forecast errors of the rival models

```
{{eSEPjy1, eSEPjy2}, {eVARjy1, eVARjy2}, {eTjy1, eTjy2}, {eTSjy1, eTSjy2}, {eSjy1, eSjy2}};
```

hitparade by simply comparing MSPE (model entering position in the previous list, corresponding MSPE)

```
fMeanXPredictionErrorHitparade[%, Dot[#, #] &]
```

```
{{3, 4, 5, 2, 1}, {7.62776, 7.65182, 7.66227, 8.31629, 8.36976}}
```

Diebold-Mariano hitparade at 5% significance level (equally precise models are grouped into brackets)

```
fDieboldMarianoHitparade[%%, 1, 0.05, Dot[#, #] &]
```

```
{{3, 4, 5}, {1, 2}}
```