

タイトル	経営科学とOR のためのWeb プログラミングによる需要予測の時系列データ解析
著者	福永, 厚; FUKUNAGA, Atsushi
引用	北海学園大学学園論集(172): 15-29
発行日	2017-06-25

# 経営科学と OR のための Web プログラミング による需要予測の時系列データ解析

福 永 厚

## 1. はじめに

需要予測<sup>1)</sup>は、経営科学と OR (Operations Research) で扱うテーマの一つである。企業が立てる短期的、中長期的な計画の基礎となるのは需要予測であり、予測が外れれば、生産量や仕入量、資金、生産設備、人員等の計画に狂いが生じ、経営活動に大きな影響を与える。需要は、国際情勢や国内の社会情勢、政策、自然環境、市場の動向、新技術、業界、競合他社、新規参入等の様々な外部要因や企業内部の内部要因など様々な要因に依存しており、予測を行うことは難しい。

需要予測を行う方法には、需要の時間的変動パターンの規則性を分析する時系列分析と、需要とその決定的要因との間に存在する関係の法則性を分析する回帰分析がある。筆者は以前、Web 上で回帰分析を行うプログラムの作成を行った<sup>2)</sup>。本稿では時系列分析を取り扱う。

時系列分析は、時間の経過順に需要データを並べて、データの時間的変動に最も合うモデル式を算出し、そのモデル式から将来の需要を予測するものである。時系列分析は回帰分析と計算方法が類似しており、回帰式を求める統計的手法が使える。時系列分析は、

様々な計算を伴うので、コンピュータ上で統計分析用のパッケージソフトや時系列分析専用のソフトを使って行われる。表計算ソフト Microsoft Excel を使ってもある程度のことが行える。これらのソフトは、コンピュータにインストールされている必要があり、また、ローカルで動くソフトである。一方、Web ブラウザ上で時系列分析ができるようになれば、インターネット上のどこでも Web サービスとして時系列分析を行って需要予測ができるようになる。

本稿では、特別なソフトを必要とせずに、ブラウザ上でデータを入力するだけで自動的に基本レベルの時系列分析ができるプログラムと、需要の折れ線グラフを自動的に作成するプログラムも作成する。Web プログラミングには JavaScript 言語を使い、折れ線グラフ作成には HTML バージョン 5 (以下、HTML5) の Canvas を用いる。

以下、第 2 章では需要予測における時系列分析の概要について、第 3 章では JavaScript と HTML5 の Canvas によって、基本レベルの時系列分析と折れ線グラフの作成を行うプログラムについて解説し、さらに Excel との比較を行い、第 4 章でまとめる。

## 2. 需要予測における時系列分析

需要予測における時系列分析<sup>3)~5)</sup>では、需要として売上高を変数  $y$  とし、年、半期、四半期、月などの時間（等間隔とする）を変数  $x$  とする。売上高  $y$  は、時間の関数  $f(x)$  を使って、モデル式  $y=f(x)$  で表され、関数  $f(x)$  には様々な関数が考えられる。本稿では、 $r$  次多項式で表される関数（以下、 $r$  次関数）と、指数関数、対数関数、累乗関数を扱う。

### 2.1 $r$ 次関数

$r$  次関数は、

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_r x^r \quad (1)$$

というモデル式で表される。これは、回帰分析の  $r$  次回帰式と同じ式であるので、回帰分析の場合<sup>2)</sup>と同様の方法によって係数を求めることができる。

式(1)の係数  $a_0, a_1, \dots, a_r$  を求める為には、最小2乗法を使う<sup>6)~8)</sup>。実際に得られた  $n$  個の  $x$  と  $y$  のデータを  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  で表し（実測値）、実測値と式(1)によって理論的に計算された値（予測値）の差の2乗  $Q$  が、最小になるように係数  $a_0, a_1, \dots, a_r$  を求める。つまり、

$$Q = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2 - \dots - a_r x_i^r)^2$$

が、最小になるように係数を決める。具体的には、各係数  $a_j (j=0, \dots, r)$  が、 $\frac{\partial Q}{\partial a_j} = 0$  を満たすことを意味する。これは、 $a_0 \sim a_r$  の  $r+1$  個の未知数を求める

$$\begin{cases} \sum_{i=1}^n 1a_0 + \sum_{i=1}^n x_i a_1 + \dots + \sum_{i=1}^n x_i^r a_r = \sum_{i=1}^n y_i \\ \vdots \\ \sum_{i=1}^n x_i^r a_0 + \sum_{i=1}^n x_i^{r+1} a_1 + \dots + \sum_{i=1}^n x_i^{2r} a_r = \sum_{i=1}^n x_i^r y_i \end{cases} \quad (2)$$

という  $r+1$  元連立1次方程式で表される。

多元連立1次方程式を解くやり方には、効率的なアルゴリズムにする為に掃き出し法（Gaussの消去法）を用いる。

### 2.2 指数関数、対数関数、累乗関数

Excelの[近似曲線の追加]機能には、関数  $f(x)$  として、 $r$  次関数以外に指数関数、対数関数、累乗関数が利用できる。

指数関数は、

$$y = ae^{bx} \quad (3)$$

対数関数は、

$$y = \ln x + b \quad (4)$$

累乗関数は、

$$y = ax^b \quad (5)$$

という式で表される。ここで、 $e$  は自然対数の底、 $\ln$  は自然対数を表す。

指数関数の式(3)の両辺の自然対数を取ると、

$$\ln y = \ln a + bx$$

となり、 $Y = \ln y, X = x$  とおくと、 $Y = \ln a + bX$  という1次関数式になる。

対数関数の式(4)は、 $Y = y, X = \ln x$  とおくと、 $Y = aX + b$  という1次関数式になる。

累乗関数の式(5)の両辺の自然対数を取ると、

$\ln y = \ln a + b \ln x$  となり、 $Y = \ln y, X = \ln x$  とおくと、 $Y = \ln a + bX$  という1次関数式になる。

1次関数式になれば、2.1に示されたよう

に、容易に係数を求めることができる。

プログラム上は  $r$  次関数の計算に  $r = 1$  を代入すれば求められるが<sup>8</sup>、1 次式の求め方を露わに記述することにする。

$$Q = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2$$

を最小にする  $a_0$ ,  $a_1$  は、

$$\begin{cases} \sum_{i=1}^n 1 a_0 + \sum_{i=1}^n x_i a_1 = \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i a_0 + \sum_{i=1}^n x_i^2 a_1 = \sum_{i=1}^n x_i y_i \end{cases}$$

の 2 元連立 1 次方程式を解いて得られ、

$$a_1 = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sum_{i=1}^n x_i^2 - n \bar{x}^2} \quad (6)$$

$$a_0 = \bar{y} - a_1 \bar{x} \quad (7)$$

で求められる。ここで、 $\bar{x}$  は  $x$  の平均値、 $\bar{y}$  は  $y$  の平均値である。

## 2.3 決定係数

予測値と実測値がどのくらい合っているかを評価する指標として、決定係数  $R^2$  がある。

決定係数  $R^2$  は、

$$R^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^n (y_i - Y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (8)$$

で計算される。 $n$  はデータ数、 $y_i$  は実測値、 $\bar{y}$  は実測値の平均値、 $Y_i$  は予測値、 $\bar{Y}$  は予測値の平均値である。

$R^2$  は、 $0 \leq R^2 \leq 1$  を満たし、 $R^2$  が 1 に近いほど予測値と実測値は合っていると見える。

## 2.4 季節係数

時系列分析の場合、時系列データに様々な

変動要因が考えられる。長期的に上昇や下降等の一定方向を示す持続的な変動である長期変動、数年間から十数年くらいの周期で繰り返す上下変動である循環変動、1 年という定まった周期を持つ上下の変動である季節変動、原因がよくわからない偶然的な要因による不規則変動がある。変動がある場合、変動している生の需要データに対して、関数式によるモデルを立てて分析することは適切ではなく、変動を考慮しなければならない。

本稿では季節変動を扱う<sup>3)~5)</sup>。季節変動は、毎年季節ごとに特徴的なパターンを示す需要データである。変動している需要データに対して移動平均を取るなどして平滑化を行い、平滑化されたデータに対して、関数式によるモデルを立てて予測を行う。得られた予測値を季節係数から、元の需要データに変換する。平滑化にはいろいろな手法があるが、ここでは、以下のような単純移動平均を用いる。

平滑化されたデータ  $Y_i$  は、 $i$  月の需要データ  $y_i$  の前の 5 カ月と後の 6 カ月の需要データを含めて、平均を取った値として求められる。つまり、

$$Y_i = (y_{i-5} + y_{i-4} + \dots + y_{i-1} + y_i + y_{i+1} + y_{i+2} + \dots + y_{i+5} + y_{i+6}) / 12 \quad (9)$$

で計算される。この平滑化されたデータ  $Y_i$  は季節による変動がなくなり、このデータに対して関数式による分析を行う。得られた理論値は平滑化された値であるので、平滑化していない需要データに変換するには、季節係数  $Y_i / y_i$  で割る。ただし、季節係数は、同じ月の値で平均を取る。

## 2.5 Excelによる時系列分析

Excelにはデータに対して回帰分析に関する機能が備わっている。回帰分析に関する関数には、

SLOPE…線形単回帰直線の傾き

INTERCEPT…線形単回帰直線の切片

LINEST…線形重回帰式の係数や切片等

RSQ…1次関数式の場合の決定係数

がある。これらを時系列データに適用すれば、1次関数式と決定係数が求められる。

また、時間  $x$  と実測値  $y$  の散布図を作成し、プロットされたデータに対して [近似曲線の追加] を実行すると、直線や多項式、対数関数など様々な回帰式をグラフに加えることができ、回帰係数と決定係数も表示することができる。

## 3. Webプログラミングによる 需要予測の時系列データ解析

### 3.1 HTML5のCanvasとJavaScript

Webページ上でグラフを作成するには、HTML5から導入されたCanvas要素を用い、JavaScriptと連動することにより、線や円を描くことができる。Canvas要素は、Safari, Opera, Firefoxのあるバージョン以降、また、最新のブラウザInternet Edgeでは対応している。

JavaScriptでCanvas要素を使うには、DOM(Document Object Model)によって、Canvas要素を指定して操作を行う<sup>9)</sup>。HTML文書中のCanvas要素に、

```
<canvas id="canvas"></canvas>
```

というように、例えば"canvas"というID名をつけておく。そして、JavaScriptプログラ

ムの中で、

```
var c=document.getElementById("canvas");
のように、DOMのgetElementByIdメソッドを使って、ID名"canvas"の部分を参照する。
```

```
var cnt=c.getContext("2d");
により、コンテキスト名を指定し、平面図形を描く際の"2d"を指定している。
```

本稿で使う主なCanvas要素は以下のものである。

```
arc(x,y,r,0,2π,anticlockwise) …(x,y)
を中心とする半径rの円を描く
```

```
fill()…塗りつぶす
```

```
strokeStyle…図形の枠線の色を指定する
```

```
strokeText(t,x,y) …(x,y)から文字データtを描く
```

直線を描くには、beginPath()によってパスを開始し、moveTo(x,y)で(x,y)に移動し、lineTo(x',y')で(x',y')まで線を引き、closePath()でパスを閉じ、stroke()により線を描くという手順を経る。

JavaScriptの中で文字列や計算結果を表示するには、innerHTMLプロパティを用いて、HTMLの内容を書き換える方法を用いる<sup>10),11)</sup>。例えば、HTML文書内で、<div id="result"></div>のように、<div>要素にID名"result"をつけておき、JavaScriptプログラムの中で、

```
var result=document.getElementById("result");
のように、DOMのgetElementByIdメソッドを使って、ID名"result"の部分を参照し、result.innerHTML= result.innerHTML+出力結果;
```

として、ID名"result"の部分に出力結果を追

加していく。このような方法によって、データ解析結果が、入力テキストボックスや Canvas によるグラフと同じページに出力することができる。

### 3.2 時系列分析の計算結果と Excel との比較

本稿のプログラムでは、Web ブラウザ上で需要予測の時系列データを入力する場合、需要データ  $y$  は売上高であるので正の値に限定する。また、時間  $x$  にはグラフの横軸に表記する時間 (年, 期, 月など) を古い順に入力させているが、プログラムの中では一番古いデータを 1 とし、1, 2, 3, … というように連続して増やした簡単な数値として扱っている。データ数  $n$  は 10 個までと限定しているが、10 個より大きいデータ数を扱うようにプログラムを拡張することは容易である。

図 1 に時系列分析の場合の 10 個のデータの入力画面を Internet Edge ブラウザで表示している。入力には、HTML のフォームタグの入力テキストボックスを用いている<sup>12)</sup>。実測値の次の  $x$  からいくつ連続して予測するかは予測数と予測区間のラベルを入力させている。グラフの  $y$  軸の最大値と目盛り間隔も入力させている。

このプログラムでは、多項式の次数を任意の  $r$  次まで対応できるようにしているが、あまり次数が大きいと図が判別しにくくなることや、奇妙にふるまう曲線になったり、高次項が効かなくなったりする場合があるので、ここでは 4 次多項式までに限定している。

図 1 の”時系列分析”ボタンをクリックした結果が図 2 に示されている。

	時間(x)	需要(y)
No.0:	2007	94
No.1:	2008	105
No.2:	2009	135
No.3:	2010	148
No.4:	2011	196
No.5:	2012	249
No.6:	2013	290
No.7:	2014	374
No.8:	2015	412
No.9:	2016	466

図 1 需要データの入力画面 (Internet Edge によるブラウザ画面, データ数 10 個の場合)

1 次から 4 次までの多項式関数と指数関数, 対数関数, 累乗関数による分析の結果として, 各関数の係数と決定係数が表示上は四捨五入されて小数第 4 位まで表示されている (プログラム内部の計算では桁数を限定していない)。また, 予測区間の予測値も表示されている。HTML5 の Canvas によって, 4 次までの  $r$  次関数とその他の関数を別々のグラフとして描いている。図の中に実測値と各関数による予測値が, それぞれの関数を区別するために色と線のスタイルを変えて描かれている。

決定係数では, 4 次関数が最も 1 に近く実測値に合っているが, 予測区間では他の関

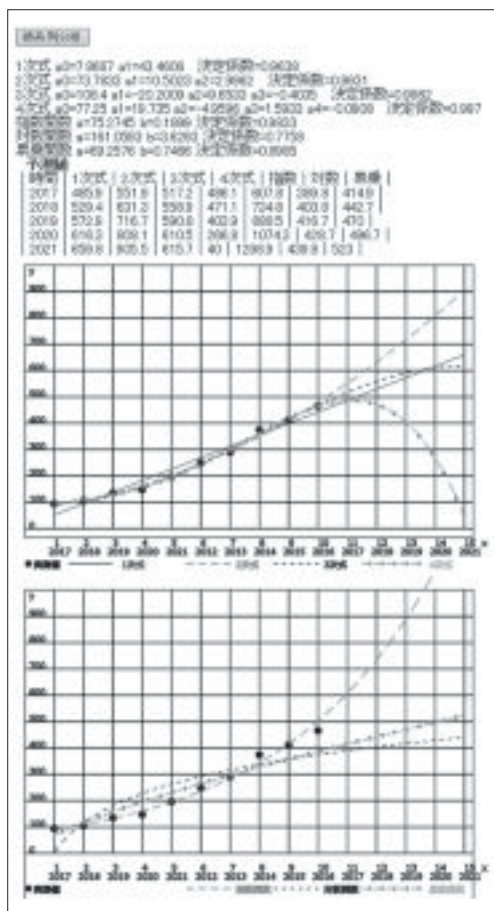


図2 時系列分析の結果 (Internet Edge によるブラウザ画面, データ数10個の場合)

数と大きく異なっている。また、指数関数も予測区間では他の関数と異なる傾向を示している。

図3には、Excelを使った場合のグラフと、[近似曲線の追加]機能を使用して求めた数式と決定係数が描かれている。本プログラムによる分析結果は、Excelの結果と一致している。

図4にはデータ数が8個で、減少傾向の需要データのブラウザ上の入力画面と、図5にはブラウザ上の分析結果が示されている。減

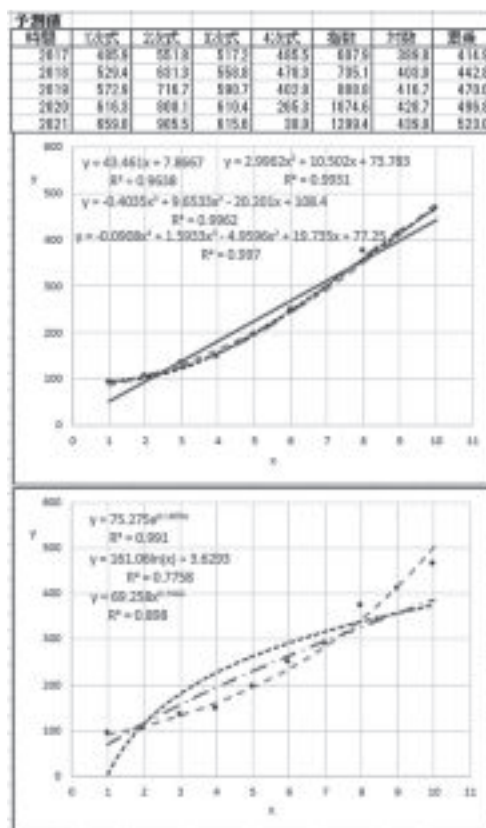


図3 時系列分析の結果 (Excelによる結果, データ数10個の場合)

少傾向の場合のグラフも適切に描かれていることが見て取れる。

図6にこのプログラムのソースが示されている。フォームタグの入力テキストボックスによりデータを入力させ、ボタンをクリックすると作成されたJavaScript関数 timeseries1 を実行するように作られている。関数 timeseries1 では、時系列分析と散布図の作成を行っている。フォームタグに入力されたデータは、データ数は変数 n に、時間変数 x のラベルは配列 dx[] に、時間変数 x は 1 からの連続データとして 2次元配列 d[i][0] に、需要変数 y のデータは 2次元配列 d[i]

データ数を入力:

データを古い順に連続して入力

No.	時間(x)	需要(y)
No.0:	第7期	1540
No.1:	第8期	1340
No.2:	第9期	1435
No.3:	第10期	1123
No.4:	第11期	1069
No.5:	第12期	1072
No.6:	第13期	960
No.7:	第14期	849
No.8:		
No.9:		

データに連続した予測数を入力:

予測する時間

第15期
第16期
第17期
第18期

y軸の最大値を入力:

y軸の目盛り間隔を入力:

図4 時系列分析のデータ入力 (Internet Edge によるブラウザ画面, データ数 8 個の場合)

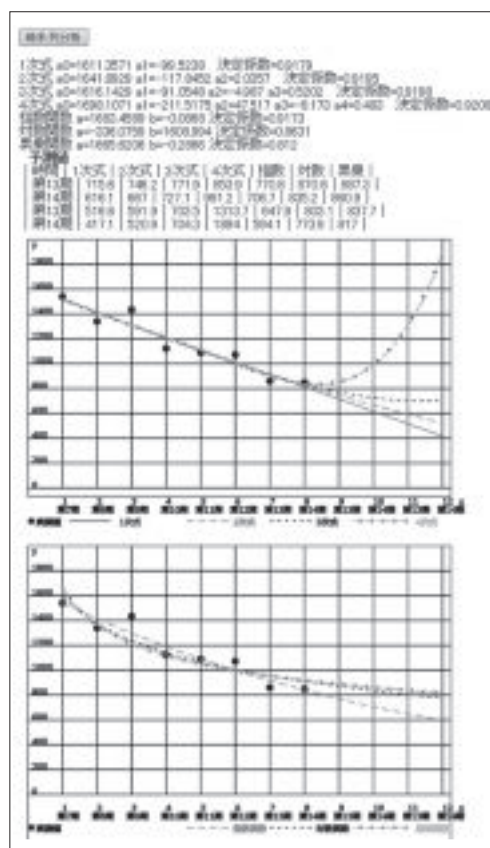


図5 時系列分析の結果 (Internet Edge によるブラウザ画面, データ数 8 個の場合)

[1]に格納されている。r 次関数の係数を求める為に、回帰分析の場合に作成した関数 gauss を使っている。r+1 元連立方程式を解くために作成した関数 gauss を使うと、式(1)の係数が求められ、この係数を使って式(1)により予測値を計算し、式(8)によって決定係数を求めている。

指数関数、対数関数、累乗関数では、式(3)~(5)にあるように、時間変数 x や需要変数 y の自然対数を取ると 1 次関数式に変換できる。1 次関数の係数を求める式(6)、(7)を用いて、関数式の係数 a, b が求められ、式(3)~(5)を使って予測値が計算される。

### 3.3 季節変動を含む場合の時系列分析

季節変動を含む場合のデータの入力画面を図7に示す。データ数は最大3年分の36個までとしているが、容易に拡張可能である。「周期期間」は、季節変動の場合月数の12となるが、季節変動以外の平滑化にも使えるように入力させるようにしている。「中心から古い方の片側期間を入力」とあるのは、式(9)の月 i を中心として、古い方のデータをいくつか用いるかを入力させている。時間は入力させずに、古いデータから、1, 2, ..., という数値を使っている。図8に平滑化されたデータ



```

<!DOCTYPE HTML>
<html lang="ja">
<head><title>様々な関数による時系列分析</title></head>
<body>
<form name="form1">
<p>データ数を入力:<input type="text" name="datasu" size="6"></p>
<p>データを古い順に連続して入力</p>
  需要(y)<br>
No.0:<input type="text" name="x0" size="10">
<input type="text" name="y0" size="10"><br>
No.1:<input type="text" name="x1" size="10">
<input type="text" name="y1" size="10"><br>
No.2:<input type="text" name="x2" size="10">
<input type="text" name="y2" size="10"><br>
No.3:<input type="text" name="x3" size="10">
<input type="text" name="y3" size="10"><br>
No.4:<input type="text" name="x4" size="10">
<input type="text" name="y4" size="10"><br>
No.5:<input type="text" name="x5" size="10">
<input type="text" name="y5" size="10"><br>
No.6:<input type="text" name="x6" size="10">
<input type="text" name="y6" size="10"><br>
No.7:<input type="text" name="x7" size="10">
<input type="text" name="y7" size="10"><br>
No.8:<input type="text" name="x8" size="10">
<input type="text" name="y8" size="10"><br>
No.9:<input type="text" name="x9" size="10">
<input type="text" name="y9" size="10"><br>
<p>データに連続した予測数を入力:<input type="text" name="yosokusuu" size="6"><br>
  予測する時間<br>
<input type="text" name="x10" size="10"><br>
<input type="text" name="x11" size="10"><br>
<input type="text" name="x12" size="10"><br>
<input type="text" name="x13" size="10"><br>
<input type="text" name="x14" size="10"><br>
  y軸の最大値を入力:<input type="text" name="ymax" size="6"><br>
  y軸の目盛り間隔を入力:<input type="text" name="yd" size="6">
</p>
<input type="button" value="時系列分析"
onClick="timeseries10;"><br>
</form>
<div id="result"></div>
<canvas id="canvas" width="600" height="1200"></canvas>
<br>
<script>
function timeseries10(){
var n,f,i,j,k,r,max,sum,sumx,sumy,sumyx,avex,avey,ymax,yv;
var deter,devy

//rmax…多項式の最大次数、n…測測値のデータセット数
rmax=4;
n=Number(document.form1.datasuu.value);

//予測期間
fc=yw=Number(document.form1.yosokusuu.value);

//縦軸の最大値と目盛り間隔
ymax=Number(document.form1.ymax.value);
yw=Number(document.form1.yd.value);

//dx[i]…xの実測値、d[i][0]…xの変換値、d[i][1]…yの実測値、
//m[i][0]…拡大係数行列、a[i][0]…回帰係数、
//lnd[i][0]…実測値の自然対数、

var dx=new Array();

var d=new Array();
for(i=0;i<n+fc;i++){
d[i]=new Array();
}

var lnd=new Array();
for(i=0;i<n+fc;i++){
lnd[i]=new Array();
}

var m=new Array();
for(i=0;i<=rmax+1;i++){
m[i]=new Array();
}

var a=new Array();
for(i=0;i<=rmax+1;i++){
a[i]=new Array();
}

//フォームの実測値を2次元配列d[i][j]に代入
dx[0]=document.form1.x0.value;
dx[1]=document.form1.x1.value;
dx[2]=document.form1.x2.value;
dx[3]=document.form1.x3.value;
dx[4]=document.form1.x4.value;
dx[5]=document.form1.x5.value;
dx[6]=document.form1.x6.value;
dx[7]=document.form1.x7.value;
dx[8]=document.form1.x8.value;
dx[9]=document.form1.x9.value;

d[0][1]=Number(document.form1.y0.value);
d[1][1]=Number(document.form1.y1.value);
d[2][1]=Number(document.form1.y2.value);
d[3][1]=Number(document.form1.y3.value);
d[4][1]=Number(document.form1.y4.value);
d[5][1]=Number(document.form1.y5.value);
d[6][1]=Number(document.form1.y6.value);
d[7][1]=Number(document.form1.y7.value);
d[8][1]=Number(document.form1.y8.value);
d[9][1]=Number(document.form1.y9.value);

//時間を整数値に変換
for(i=0;i<n+fc;i++){
d[i][0]=i+1;
}

dx[10]=document.form1.x10.value;
dx[11]=document.form1.x11.value;
dx[12]=document.form1.x12.value;
dx[13]=document.form1.x13.value;
dx[14]=document.form1.x14.value;

for(i=nj<n+fc;i++){
dx[i]=dx[i+n-10];
}

//実測値,yの平均値の計算
sumx=0;
sumy=0;
for(i=0;i<nj+i++){
sumx=sumx+dx[i][0];
sumy=sumy+d[i][1];
}
avex=sumx/n;
avey=sumy/n;

//実測値の偏差
sumy=0;
for(i=0;i<nj+i++){
sumy=sumy+Math.pow(d[i][1]-avey,2);
}
devy=sumy;

//n次多項式の拡大係数行列m[i][j]の作成
for(i=1;i<=rmax;i++){
for(j=0;j<=i-1;j++){
sum=0;
for(k=0;k<n+k+i){
sum=sum+Math.pow(d[k][0],i)*d[k][1];
}
m[i][j]=sum;
}
sum=0;
for(k=0;k<n+k+i){
sum=sum+Math.pow(d[k][0],i)*d[k][1];
}
m[i][i+1]=sum;
}

//+1元連立1次方程式の解法
gauss(m,r+1);

//結果の表示
var result=document.getElementById("result");
result.innerHTML=result.innerHTML+"n次式:";
for(i=0;i<=rmax;i++){
a[i][0]=m[i][i+1];
result.innerHTML=result.innerHTML+"a"+i+"="+Math.round(10000*a[i][0]/10000)+" ";
}

//決定係数の計算
sumyx=0;
for(i=0;i<nj+i++){
sum=a[0][i];
for(j=1;j<=rmax+i){
sum=sum+a[j][i]*Math.pow(d[i][0],j);
}
sumyx=sumyx+Math.pow(d[i][1]-sum,2);
}
deter=Math.round(10000*(1-sumyx/devy))/10000;
result.innerHTML=result.innerHTML+" 決定係数="+deter+"<br>";

//指数関数、対数関数、累乗関数
var sumx2,sumlx2,sumlnx,sumly,avelnx,avey;
var expa,expb,loga,logb,powa,powb,deterexp,deterlog,deterpow;
sumlnx=0;
sumly=0;
for(i=0;i<nj+i++){
lnd[i][0]=Math.log(d[i][0]);
lnd[i][1]=Math.log(d[i][1]);
sumlnx=sumlnx+lnd[i][0];
sumly=sumly+lnd[i][1];
}
avelnx=sumlnx/n;
avey=sumly/n;

//指数関数の係数の計算
sumx2=0;
sumyx=0;
for(i=0;i<nj+i++){
sumx2=sumx2+Math.pow(d[i][0],2);
sumyx=sumyx+d[i][0]*lnd[i][1];
}
expb=(sumyx-n*avelnx*avey)/(sumx2-n*avelnx*avelnx);
expa=Math.exp(avey-avelnx*expb);

sumyx=0;
for(i=0;i<nj+i++){
sumyx=sumyx+Math.pow(d[i][1]-expa*Math.exp(expb*d[i][0]),2);
}
deterexp=1-sumyx/devy;

//対数関数の係数の計算
sumlnx2=0;
sumy=0;
for(i=0;i<nj+i++){
sumlnx2=sumlnx2+Math.pow(lnd[i][0],2);
sumy=sumy+lnd[i][0]*d[i][1];
}
loga=(sumyx-n*avelnx*avey)/(sumlnx2-n*avelnx*avelnx);
logb=avey-avelnx*loga;

sumy=0;
for(i=0;i<nj+i++){
sumy=sumyx+Math.pow(d[i][1]-loga*Math.log(d[i][0])-logb,2);
}
deterlog=1-sumyx/devy;

//累乗関数の係数の計算
sumyx=0;
for(i=0;i<nj+i++){
sumyx=sumyx+lnd[i][0]*lnd[i][1];
}

```

## 経営科学と OR のための Web プログラミングによる需要予測の時系列データ解析 (福永 厚)

```

powb=(sumyx-n*aveInx*aveIny)/(sumInx2-n*aveInx*aveInx);
powa=Math.exp(aveIny-aveInx*powb);

sumyx=0;
for(i=0;cni+){
    sumyx=sumyx+Math.pow(d[i]-powa*Math.pow(d[i],0),powb),2);
}
deterpow=1-sumyx/devy;

result.innerHTML=result.innerHTML+
    "指数関数 a="+Math.round(10000*expa)/10000+" b="+Math.round(10000*expb)/10000+
    " 決定係数="+Math.round(10000*deterexp)/10000+"<br>"+
    "対数関数 a="+Math.round(10000*loga)/10000+" b="+Math.round(10000*logb)/10000+
    " 決定係数="+Math.round(10000*deterlog)/10000+"<br>"+
    "冪乗関数 a="+Math.round(10000*powa)/10000+" b="+Math.round(10000*powb)/10000+
    " 決定係数="+Math.round(10000*deterpow)/10000+"<br>";

//予測値
result.innerHTML=result.innerHTML+<br> 予測値<br><br>
    " | 時間 | 1次式 | 2次式 | 3次式 | 4次式 | "
    " +指数 | 対数 | 冪乗 | <br>";

for(i=0;cni+){
    result.innerHTML=result.innerHTML+ " | "+d[i]+ " | "
    +Math.round(10*a[1][1]*d[i][0]+a[0][1])/10+" | "
    +Math.round(10*(a[2][2]*Math.pow(d[i][0],2)+a[1][2]*d[i][0]+a[0][2])/10+" | "
    +Math.round(10*(a[3][3]*Math.pow(d[i][0],3)+a[2][3]*Math.pow(d[i][0],2)+a[1][3]*d[i][0]+a[0][3])/10+" | "
    +Math.round(10*(a[4][4]*Math.pow(d[i][0],4)+a[3][4]*Math.pow(d[i][0],3)+a[2][4]*Math.pow(d[i][0],2)
    +a[1][4]*d[i][0]+a[0][4])/10+" | "
    +Math.round(10*expa*Math.exp(d[i][0]))/10
    + " | "+Math.round(10*loga*Math.log(d[i][0])/logb)/10
    + " | "+Math.round(10*powa*Math.pow(d[i][0],powb))/10+" | <br>";
}

//散布図の作成
var d0min,d0max,d1min,d1max,xmax,xc;

//予測値を含めたx最大値と最小値
d0min=d[0][0];
d0max=d[n+fc-1][0];

//実測値の最大値と最小値
d1min=d[0][1];
d1max=d[0][1];

for(i=1;cni+){
    if(d[i][1]<d1min){
        d1min=d[i][1];
    }
    if(d[i][1]>d1max){
        d1max=d[i][1];
    }
}

//実測値の最大自盛りとスケール
xmax=d0max;

var c=document.getElementById("canvas");
var cnt=c.getContext("2d");

var x0,y0,xf,yf,w,h,ywm,xc;
x0=10;
y0=10;
w=500;
h=300;
xf=x0+w;
yf=y0+h;

for(i=0;cni+){
    cnt.beginPath();
    cnt.moveTo(x0+w*d[i][0]/xmax,y0);
    cnt.lineTo(x0+w*d[i][0]/xmax,yf);
    cnt.closePath();
    cnt.stroke();
    cnt.strokeText(d[i][0],x0+w*d[i][0]/xmax,yf+20);
    if(i<n+fc){cnt.strokeText(dx[i],x0+w*d[i][0]/xmax,yf+30);}
}
cnt.strokeText("x",x0+w*d[n+fc-1][0]/xmax+20,yf+20);

ywm=Math.ceil(ywm/yw);
for(i=0;cywn+){
    cnt.beginPath();
    cnt.moveTo(x0,yf-h*yw*i/ywm);
    cnt.lineTo(x0+xf-yf-h*yw*i/ywm);
    cnt.closePath();
    cnt.stroke();
    cnt.strokeText(yw*i*x0+5,yf-h*yw*i/ywm);
}
cnt.strokeText("y",x0+5,yf-h*yw*10/ywm+10);

cnt.rect(x0,yf,yf);
cnt.stroke();

for(i=0;cni+){
    cnt.beginPath();
    cnt.arc(x0+w*d[i][0]/xmax,yf-h*d[i][1]/ywm,4,0,Math.PI*2,false);
    cnt.fill();
    cnt.closePath();
    cnt.stroke();
}
cnt.arc(x0+4,yf+40,4,0,Math.PI*2,false);
cnt.fill();
cnt.strokeText("実測値",x0+10,yf+45);

//1次式の描画
cnt.beginPath();
cnt.moveTo(x0+w*d[0][0]/xmax,yf-h*a[0][1]+a[1][1]*d[0][0]/ywm);
cnt.strokeStyle="rgb(255,0,0)";
cnt.lineTo(x0+w*d[n+fc-1][0]/xmax,yf-h*a[0][1]+a[1][1]*d[n+fc-1][0]/ywm);
cnt.moveTo(x0+50,yf+40);
cnt.lineTo(x0+100,yf+40);
cnt.closePath();
cnt.stroke();
cnt.strokeText("1次式",x0+110,yf+45);

//2次式の描画
var ten=new Array();
var liner=new Array();

xx=(d[n+fc-1][0]-d[0][0])/100;
for(i=0;cni+){
    ten[i]=Math.floor(1000*(d[0][0]+xx*i))/1000;
}

for(i=0;j<100;i++){
    sum=a[0][2];
    for(l=1;l<2;j++){
        sum=sum+a[j][2]*Math.pow(ten[i],j);
    }
    liner[i]=sum;
}
cnt.strokeStyle="rgb(0,255,0)";
for(i=0;cni+){
    if(yf-h*liner[i]/ywm>y0)&&(yf-h*liner[i]/ywm<yf){
        cnt.beginPath();
        cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ywm);
        cnt.lineTo(x0+w*ten[i+2]/xmax,yf-h*liner[i+2]/ywm);
        cnt.closePath();
        cnt.stroke();
    }
}
for(i=0;j<10;i++){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax+150,yf+40);
    cnt.lineTo(x0+w*ten[i+2]/xmax+150,yf+40);
    cnt.closePath();
    cnt.stroke();
}
cnt.strokeText("2次式",x0+w*ten[10]/xmax+160,yf+45);

//3次式の描画
for(i=0;cni+){
    sum=a[0][3];
    for(l=1;l<3;j++){
        sum=sum+a[j][3]*Math.pow(ten[i],j);
    }
    liner[i]=sum;
}
cnt.strokeStyle="rgb(0,0,255)";
for(i=0;cni+){
    if(yf-h*liner[i]/ywm>y0)&&(yf-h*liner[i]/ywm<yf){
        cnt.beginPath();
        cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ywm);
        cnt.lineTo(x0+w*ten[i+1]/xmax,yf-h*liner[i+1]/ywm);
        cnt.closePath();
        cnt.stroke();
    }
}
for(i=0;j<12;i++){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax+250,yf+40);
    cnt.lineTo(x0+w*ten[i+1]/xmax+250,yf+40);
    cnt.closePath();
    cnt.stroke();
}
cnt.strokeText("3次式",x0+w*ten[10]/xmax+260,yf+45);

//4次式の描画
for(i=0;cni+){
    sum=a[0][4];
    for(l=1;l<4;j++){
        sum=sum+a[j][4]*Math.pow(ten[i],j);
    }
    liner[i]=sum;
}
cnt.strokeStyle="rgb(255,0,255)";
for(i=0;cni+){
    if((yf-h*liner[i]/ywm>y0)&&(yf-h*liner[i]/ywm<yf)){
        cnt.beginPath();
        cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ywm);
        cnt.lineTo(x0+w*ten[i+1]/xmax,yf-h*liner[i+1]/ywm);
        cnt.arc(x0+w*ten[i+2]/xmax,yf-h*liner[i+2]/ywm,2,0,Math.PI*2,false);
        cnt.fill();
        cnt.closePath();
        cnt.stroke();
    }
}
for(i=0;j<14;i++){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax+350,yf+40);
    cnt.lineTo(x0+w*ten[i+1]/xmax+350,yf+40);
    cnt.arc(x0+w*ten[i+2]/xmax+350,yf+40,2,0,Math.PI*2,false);
    cnt.fill();
    cnt.closePath();
    cnt.stroke();
}
cnt.strokeText("4次式",x0+w*ten[10]/xmax+380,yf+45);

//指数関数、対数関数、冪乗関数の別グラフ
y0=380;
yf=y0+h;

cnt.strokeStyle="rgb(0,0,0)";
for(i=0;cni+){
    cnt.beginPath();
    cnt.moveTo(x0+w*d[i][0]/xmax,y0);
    cnt.lineTo(x0+w*d[i][0]/xmax,yf);
    cnt.closePath();
    cnt.stroke();
}
cnt.strokeText(dx[i],x0+w*d[i][0]/xmax+20,yf+20);
}

for(i=0;cywn+){
    cnt.beginPath();
    cnt.moveTo(x0,yf-h*yw*i/ywm);
    cnt.lineTo(x0+xf-yf-h*yw*i/ywm);
    cnt.closePath();
    cnt.stroke();
}
cnt.strokeText("y",x0+5,yf-h*yw*10/ywm+10);

cnt.rect(x0,yf,yf);
cnt.stroke();

for(i=0;cni+){
    cnt.beginPath();
    cnt.arc(x0+w*d[i][0]/xmax,yf-h*d[i][1]/ywm,4,0,Math.PI*2,false);
    cnt.fill();
    cnt.closePath();
    cnt.stroke();
}
cnt.arc(x0+4,yf+40,4,0,Math.PI*2,false);
cnt.fill();
cnt.strokeText("実測値",x0+10,yf+45);

//1次式の描画
cnt.beginPath();
cnt.moveTo(x0+w*d[0][0]/xmax,yf-h*a[0][1]+a[1][1]*d[0][0]/ywm);
cnt.strokeStyle="rgb(255,0,0)";
cnt.lineTo(x0+w*d[n+fc-1][0]/xmax,yf-h*a[0][1]+a[1][1]*d[n+fc-1][0]/ywm);
cnt.moveTo(x0+50,yf+40);
cnt.lineTo(x0+100,yf+40);
cnt.closePath();
cnt.stroke();
cnt.strokeText("1次式",x0+110,yf+45);

//2次式の描画
var ten=new Array();
var liner=new Array();

xx=(d[n+fc-1][0]-d[0][0])/100;
for(i=0;cni+){
    ten[i]=Math.floor(1000*(d[0][0]+xx*i))/1000;
}

```

```

cnt.arc(x0+4,yf+40, 4, 0, Math.PI*2, false);
cnt.fill();
cnt.strokeText("実測値",x0+10,yf+45);

//指数関数の描画
for(i=0;i<=100;i++){
  liner[i]=expa*Math.exp(xp*b*ten[i]);
}
for(i=0;i<100;i+=3){
  if((yf-h*liner[i]/ymax>y0)&&(yf-h*liner[i]/ymax<yf)){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ymax);
    cnt.strokeStyle="rgb(0,255,0)";
    cnt.lineTo(x0+w*ten[i+2]/xmax,yf-h*liner[i+2]/ymax);
    cnt.closePath();
    cnt.stroke();
  }
}
for(i=0;i<100;i+=3){
  cnt.beginPath();
  cnt.moveTo(x0+w*ten[i]/xmax+150,yf+40);
  cnt.lineTo(x0+w*ten[i+2]/xmax+150,yf+40);
  cnt.closePath();
  cnt.stroke();
}
cnt.strokeText("指数関数",x0+w*ten[10]/xmax+160,yf+45);

//対数関数の描画
for(i=0;i<=100;i++){
  liner[i]=loga*Math.log(ten[i])+logb;
}
for(i=0;i<100;i+=2){
  if((yf-h*liner[i]/ymax>y0)&&(yf-h*liner[i]/ymax<yf)){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ymax);
    cnt.strokeStyle="rgb(0,255)";
    cnt.lineTo(x0+w*ten[i+1]/xmax,yf-h*liner[i+1]/ymax);
    cnt.closePath();
    cnt.stroke();
  }
}
for(i=0;i<120;i+=2){
  cnt.beginPath();
  cnt.moveTo(x0+w*ten[i]/xmax+250,yf+40);
  cnt.lineTo(x0+w*ten[i+1]/xmax+250,yf+40);
  cnt.closePath();
  cnt.stroke();
}
cnt.strokeText("対数関数",x0+w*ten[10]/xmax+260,yf+45);

//冪乗関数の描画
for(i=0;i<=100;i++){
  liner[i]=powa*Math.pow(ten[i],powb);
}
for(i=0;i<100;i+=3){
  if((yf-h*liner[i]/ymax>y0)&&(yf-h*liner[i]/ymax<yf)){
    cnt.beginPath();
    cnt.moveTo(x0+w*ten[i]/xmax,yf-h*liner[i]/ymax);
    cnt.strokeStyle="rgb(255,0,255)";
    cnt.lineTo(x0+w*ten[i+1]/xmax,yf-h*liner[i+1]/ymax);
    cnt.arc(x0+w*ten[i+2]/xmax,yf-h*liner[i+2]/ymax, 2, 0, Math.PI*2, false);
    cnt.fill();
    cnt.closePath();
    cnt.stroke();
  }
}
cnt.strokeText("冪乗関数",x0+w*ten[10]/xmax+380,yf+45);

}

//n元連立1次方程式を掃き出し法によって解く関数
function gauss(m,n){
  var i,j,k,mil,mjl;

  for(i=0;i<n;i++){
    mil=m[i][i];
    for(j=0;j<n+1;j++){
      m[i][j]=m[i][j]/mil;
    }
    for(j=i+1;j<n;j++){
      mj=m[j][i];
      for(k=i+k+1;k++){
        m[j][k]=m[j][k]-mj*m[i][k];
      }
    }
  }

  for(i=n-1;i>=0;i--){
    for(j=i-1;j>=0;j--){
      mj=m[j][i];
      for(k=i+k+1;k++){
        m[j][k]=m[j][k]-mj*m[i][k];
      }
    }
  }

  return m;
}
</script>
</body>
</html>

```

図6 時系列分析のプログラムソース



図7 季節変動がある場合の時系列分析のデータ入力 (Internet Edge によるブラウザ画面)

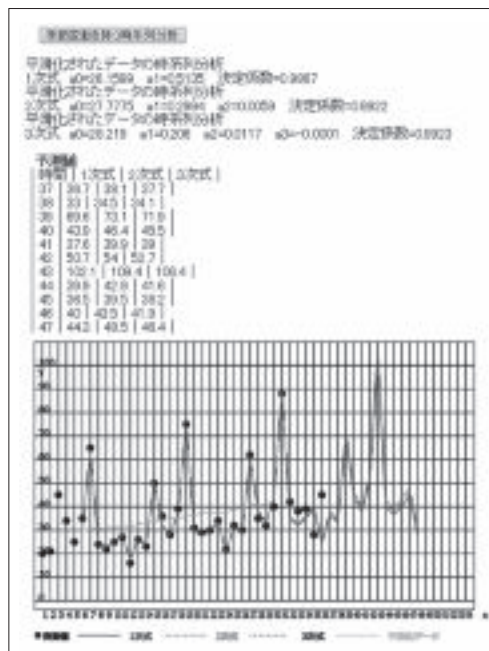


図8 季節変動がある場合の時系列分析の結果 (Internet Edge によるブラウザ画面)

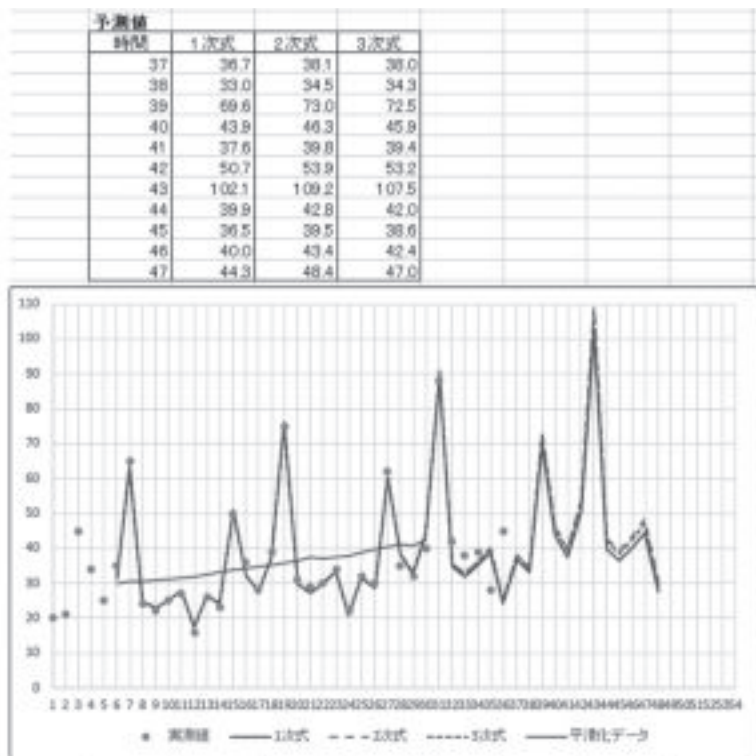


図9 季節変動がある場合の時系列分析の結果 (Excel による結果)

データ数を入力:

36個までの需要データ(y)を入力

No.0: 112	No.1: 123	No.2: 149
No.3: 137	No.4: 108	No.5: 116
No.6: 135	No.7: 168	No.8: 155
No.9: 121	No.10: 133	No.11: 160
No.12: 183	No.13: 166	No.14: 130
No.15: 137	No.16: 158	No.17: 190
No.18: 178	No.19: 145	No.20: 149
No.21: 165	No.22: 198	No.23: 187
No.24: 145	No.25:	No.26:
No.27:	No.28:	No.29:
No.30:	No.31:	No.32:
No.33:	No.34:	No.35:

周期期間を入力:

中心から古い方の片側期間を入力:

予測する期間を入力:

y軸の最大値を入力:

y軸の目盛り間隔を入力:

図10 周期期間 5 の場合の時系列分析のデータ入力 (Internet Edge によるブラウザ画面)

に対して時系列分析を行った結果, 各関数の係数と決定係数, 平滑化データの子測値を季節係数で割って求めた予測値およびグラフが示されている。グラフを見やすくするため, 用いた関数は 3 次関数までとしている。図 9 には, Excel を使った場合の結果が示されており, 両者は一致している。

図 10 には, データ数が 25 個, 周期期間が 5, 片側期間 2 の場合の入力画面, 図 11 に時系列分析結果とグラフが表示されている。季節変動ではなく周期期間 5 の場合にも, 適切に分析されてグラフが描かれている。

図 12 には, このプログラムソースが示されている。

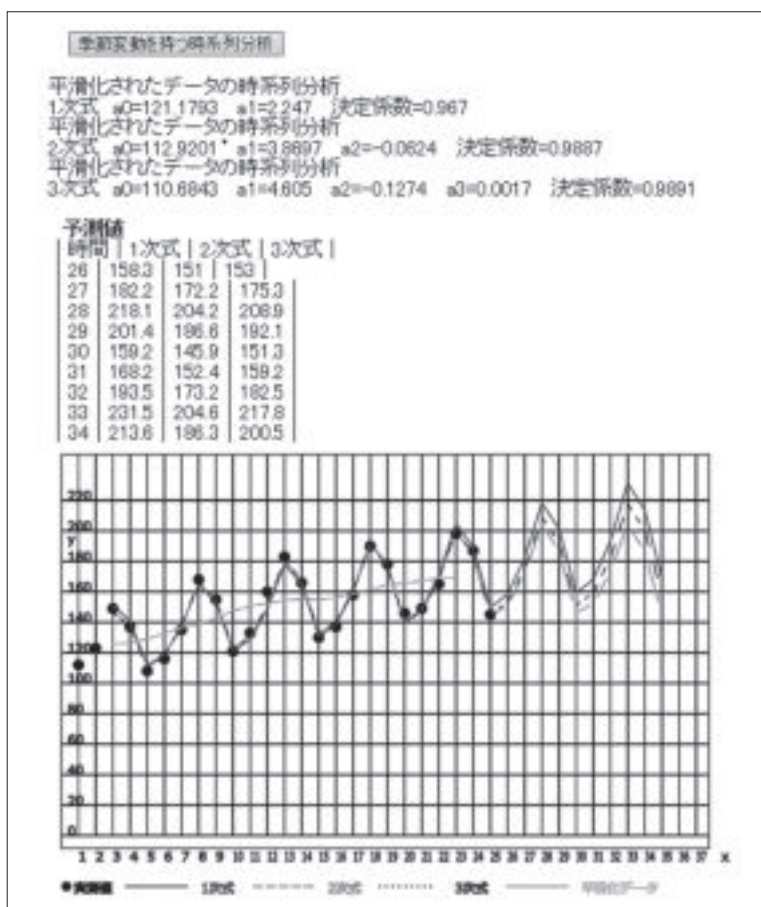


図 11 周期期間 5 の場合の時系列分析の結果 (Internet Edge によるブラウザ画面)

```

<!DOCTYPE HTML>
<html lang="ja">
<head><title>季節変動を持つ時系列分析</title></head>
<body>

<form name="form1">
<p>データ数を入力:<input type="text" name="datasuu" size="6"><br>
36個までの需要データ(y)を入力<br>
No.0: <input type="text" name="y0" size="8">
No.1: <input type="text" name="y1" size="8">
No.2: <input type="text" name="y2" size="8"><br>
No.3: <input type="text" name="y3" size="8">
No.4: <input type="text" name="y4" size="8">
No.5: <input type="text" name="y5" size="8"><br>
No.6: <input type="text" name="y6" size="8">
No.7: <input type="text" name="y7" size="8">
No.8: <input type="text" name="y8" size="8"><br>
No.9: <input type="text" name="y9" size="8">
No.10: <input type="text" name="y10" size="8">
No.11: <input type="text" name="y11" size="8"><br>
No.12: <input type="text" name="y12" size="8">
No.13: <input type="text" name="y13" size="8">
No.14: <input type="text" name="y14" size="8"><br>
No.15: <input type="text" name="y15" size="8">
No.16: <input type="text" name="y16" size="8">
No.17: <input type="text" name="y17" size="8"><br>
No.18: <input type="text" name="y18" size="8">
No.19: <input type="text" name="y19" size="8">
No.20: <input type="text" name="y20" size="8"><br>
No.21: <input type="text" name="y21" size="8">
No.22: <input type="text" name="y22" size="8">
No.23: <input type="text" name="y23" size="8"><br>
No.24: <input type="text" name="y24" size="8">
No.25: <input type="text" name="y25" size="8">
No.26: <input type="text" name="y26" size="8"><br>
No.27: <input type="text" name="y27" size="8">
No.28: <input type="text" name="y28" size="8">
No.29: <input type="text" name="y29" size="8"><br>
No.30: <input type="text" name="y30" size="8">
No.31: <input type="text" name="y31" size="8">
No.32: <input type="text" name="y32" size="8"><br>
No.33: <input type="text" name="y33" size="8">
No.34: <input type="text" name="y34" size="8">
No.35: <input type="text" name="y35" size="8"><br>

周期期間を入力:<input type="text" name="ter" size="6"><br>
中心から古い方の片側期間を入力:<input type="text" name="ter1"
size="6"><br>
予測する期間を入力:<input type="text" name="yosokusuu" size="6"><br>
y軸の最大値を入力:<input type="text" name="ymax" size="6"><br>
y軸の目盛り間隔を入力:<input type="text" name="yd" size="6">
</p>

<input type="button" value="季節変動を持つ時系列分析"
onClick="timeseries2();"><br><br>
</form>
<div id="result"></div>
<canvas id="canvas" width="600" height="1200"></canvas>
</body>
    
```

```

<script>
function timeseries2(){
var n,fc,i,j,k,r,rmax,sum,sumx,sumy,avex,avey;
var deter,devy,ter,ter1;
//rmax...多項式の最大次数、n...実測値のデータセット数
rmax=3;
n=Number(document.form1.datasuu.value);

//周期期間
ter=Number(document.form1.ter.value);
ter1=Number(document.form1.ter1.value);

//予測期間
fc=Number(document.form1.yosokusuu.value);

//y軸最大値と間隔
ymax=Number(document.form1.ymax.value);
yw=Number(document.form1.yd.value);

//dx[i]...xの実測値、d[i][0]...xの変換値、d[i][1]...yの実測値
//m[i]...拡大係数行列、a[i]...回帰係数

var d=new Array();
for(i=0;i<n+fc;i++){
d[i]=new Array();
}

var m=new Array();
for(i=0;i<=rmax+1;i++){
m[i]=new Array();
}

var a=new Array();
for(i=0;i<=rmax+1;i++){
a[i]=new Array();
}

//フォームの需要実測値を2次元配列d[i][j]に代入
d[0][1]=Number(document.form1.y0.value);
d[1][1]=Number(document.form1.y1.value);
d[2][1]=Number(document.form1.y2.value);
d[3][1]=Number(document.form1.y3.value);
d[4][1]=Number(document.form1.y4.value);
d[5][1]=Number(document.form1.y5.value);
d[6][1]=Number(document.form1.y6.value);
d[7][1]=Number(document.form1.y7.value);
d[8][1]=Number(document.form1.y8.value);
d[9][1]=Number(document.form1.y9.value);
d[10][1]=Number(document.form1.y10.value);
d[11][1]=Number(document.form1.y11.value);
d[12][1]=Number(document.form1.y12.value);
d[13][1]=Number(document.form1.y13.value);
d[14][1]=Number(document.form1.y14.value);
d[15][1]=Number(document.form1.y15.value);
d[16][1]=Number(document.form1.y16.value);
d[17][1]=Number(document.form1.y17.value);
d[18][1]=Number(document.form1.y18.value);
d[19][1]=Number(document.form1.y19.value);
d[20][1]=Number(document.form1.y20.value);
d[21][1]=Number(document.form1.y21.value);
d[22][1]=Number(document.form1.y22.value);
d[23][1]=Number(document.form1.y23.value);
d[24][1]=Number(document.form1.y24.value);
d[25][1]=Number(document.form1.y25.value);
d[26][1]=Number(document.form1.y26.value);
d[27][1]=Number(document.form1.y27.value);
d[28][1]=Number(document.form1.y28.value);
d[29][1]=Number(document.form1.y29.value);
d[30][1]=Number(document.form1.y30.value);
d[31][1]=Number(document.form1.y31.value);
d[32][1]=Number(document.form1.y32.value);
d[33][1]=Number(document.form1.y33.value);
d[34][1]=Number(document.form1.y34.value);
d[35][1]=Number(document.form1.y35.value);

//期間xを設定
for(i=0;i<n+fc;i++){
d[i][0]=i+1;
}

//移動平均
var dt=new Array();
for(i=0;i<n+fc-ter+1;i++){
dt[i]=new Array();
}
for(i=0;i<n-ter+fc+1;i++){
dt[i][0]=d[i+ter][0];
}

for(i=0;i<n-ter+1;i++){
sumy=0;
for(j=i+ter;j++){
sumy=sumy+d[j][1];
}
dt[i][1]=sumy/ter;
}

//季節係数の計算
var sea=new Array();
for(i=0;i<ter;i++){
sumy=0;
k=0;
for(j=i<n-ter+1;j+=ter){
sumy=sumy+dt[j][1]/dt[j+ter][1];
k=k+1;
}
sea[i]=sumy/k;
}

//実測値x,yの平均値の計算
sumx=0;
sumy=0;
for(i=0;i<n-ter+1;i++){
sumx=sumx+dt[i][0];
sumy=sumy+dt[i][1];
}
avex=sumx/(n-ter+1);
avey=sumy/(n-ter+1);

//実測値の偏差
sumy=0;
for(i=0;i<n-ter+1;i++){
sumy=sumy+Math.pow(dt[i][1]-avey,2);
}
devy=sumy;

//r次多項式の拡大係数行列m[i][j]の作成
for(r=1;r<=rmax;r++){
for(i=0;i<=r;i++){
for(j=0;j<=r-j++){
sum=0;
for(k=0;k<n-ter+1;k++){
sum=sum+Math.pow(dt[k][0],i+j);
}
m[i][j]=sum;
}
}
sum=0;
for(k=0;k<n-ter+1;k++){
sum=sum+Math.pow(dt[k][0],i)*dt[k][1];
}
m[i][r+1]=sum;
}

//r+1元連立1次方程式の解法
gauss(m,r+1);

//結果の表示
var result=document.getElementById("result");
result.innerHTML=result.innerHTML+"平滑化されたデータの時系列分析<br>";

result.innerHTML=result.innerHTML+r+"次式";
for(i=0;i<=r;i++){
a[i][r]=m[i][r+1];
result.innerHTML=result.innerHTML+"
a"+i+"="+Math.round(10000*a[i][r])/10000+" ";
}

//決定係数の計算
sumyx=0;
for(i=0;i<n-ter+1;i++){
sum=a[i][r];
for(j=1;j<=r;j++){
sum=sum+a[j][r]*Math.pow(dt[i][0],j);
}
}
k=i%ter;
sumyx=sumyx+Math.pow(dt[i][1]-sum,2);
}
deter=Math.round(10000*(1-sumyx/devy))/10000;
result.innerHTML=result.innerHTML+" 決定係数="+deter+"<br>";
result.innerHTML=result.innerHTML+"<br>";

//予測値
result.innerHTML=result.innerHTML+"<b> 予測値</b><br>"

```

```

+ " | 時間 | 1次式 | 2次式 | 3次式 | <br>";
for(i=n-ter1,j<n-ter+fc;i++){
j=i%ter;
result.innerHTML=result.innerHTML+" | "+dt[i][0]+" | "
+Math.round(10*(a[0][1]+a[1][1]*dt[i][0])/sea[j])/10+" | "
+Math.round(10*(a[2][2]*Math.pow(dt[i][0],2)+a[1][2]*dt[i][0]+a[0][2])/sea[j])/10
+" | "
+Math.round(10*(a[3][3]*Math.pow(dt[i][0],3)+a[2][3]*Math.pow(dt[i][0],2)
+a[1][3]*dt[i][0]+a[0][3])/sea[j])/10+" | <br>";
}

//散布図の作成
var d0min,d0max,d1min,d1max,xmin,xmax,ymin,ymax,xm,ym;

//予測を含めたx最大値
xmax=d[n+fc-1][0];

var c=document.getElementById("canvas");
var cnt=c.getContext("2d");

var x0,y0,xf,yf,w,h,xw,xx,yi,y1,xi,xl1;
x0=10;
y0=10;
w=500;
h=300;
xf=x0+w;
yf=y0+h;

xw=Math.floor(w/xmax);
for(i=0;i<n+fc;i++){
cnt.beginPath();
cnt.moveTo(x0+w*d[i][0]/xmax,y0);
cnt.lineTo(x0+w*d[i][0]/xmax,yf);
cnt.closePath();
cnt.stroke();
cnt.strokeText(d[i][0],x0+w*d[i][0]/xmax,yf+20,8);
}
cnt.strokeText("x",x0+w*d[n+fc-1][0]/xmax+20,yf+20);

ywn=Math.floor(ymax/yw);
for(i=0;i<ywn;i++){
cnt.beginPath();
cnt.moveTo(x0,yf-h*yw*i/ymax);
cnt.lineTo(x0+xf,yf-h*yw*i/ymax);
cnt.closePath();
cnt.stroke();
cnt.strokeText(yw*i,x0+5,yf-h*yw*i/ymax+10);
}

cnt.rect(x0,y0,xf,yf);
cnt.stroke();

for(i=0;i<n;j=i++){
cnt.beginPath();
cnt.arc(x0+w*d[i][0]/xmax,yf-h*d[i][1]/ymax,4,0,Math.PI*2,false);
cnt.fill();
cnt.closePath();
cnt.stroke();
}
cnt.arc(x0+4,yf+40,4,0,Math.PI*2,false);
cnt.fill();
cnt.strokeText("実測値",x0+10,yf+45);

//1次式の描画
cnt.strokeStyle="rgb(255,0,0)";
for(i=0;j<n-ter+fc;i++){
j=i%ter;
k=(i+1)%ter;
cnt.beginPath();
cnt.moveTo(x0+w*d[i][0]/xmax,yf-h*(a[0][1]+a[1][1]*dt[i][0])/sea[j]/ymax);
cnt.lineTo(x0+w*d[i+1][0]/xmax,yf-h*(a[0][1]+a[1][1]*dt[i+1][0])/sea[k]/ymax);
cnt.closePath();
cnt.stroke();
}
cnt.beginPath();
cnt.moveTo(x0+50,yf+40);
cnt.lineTo(x0+100,yf+40);
cnt.closePath();
cnt.stroke();
cnt.strokeText("1次式",x0+110,yf+45);

//2次式の描画
cnt.strokeStyle="rgb(0,255,0)";
for(i=0;j<n-ter+fc;i++){
j=i%ter;
k=(i+1)%ter;
xi=x0+w*d[i][0]/xmax;
yi=yf-h*(a[0][2]+a[1][2]*dt[i][0]+a[2][2]*Math.pow(dt[i][0],2))/sea[j]/ymax;
xi1=x0+w*d[i+1][0]/xmax;
yi1=yf-h*(a[0][2]+a[1][2]*dt[i+1][0]+a[2][2]*Math.pow(dt[i+1][0],2))/sea[k]/ymax;
if((yi>0)&&(yi<yf)){
cnt.beginPath();
cnt.moveTo(xi+(xi1-xi)/5,yi+(yi1-yi)/5);
cnt.lineTo(xi+(xi1-xi)*4/5,yi+(yi1-yi)*4/5);
cnt.closePath();
cnt.stroke();
}
}
for(i=0;j<5;j++){
cnt.beginPath();
cnt.moveTo(x0+150+10*i+1,yf+40);
cnt.lineTo(x0+150+10*i+9,yf+40);
cnt.closePath();
cnt.stroke();
}
cnt.strokeText("2次式",x0+210,yf+45);

//3次式の描画
cnt.strokeStyle="rgb(0,0,255)";
for(i=0;j<n-ter+fc;i++){
j=i%ter;
k=(i+1)%ter;
xi=x0+w*d[i][0]/xmax;
yi=yf-h*(a[0][3]+a[1][3]*dt[i][0]+a[2][3]*Math.pow(dt[i][0],2)
+a[3][3]*Math.pow(dt[i][0],3))/sea[j]/ymax;
xi1=x0+w*d[i+1][0]/xmax;
yi1=yf-h*(a[0][3]+a[1][3]*dt[i+1][0]+a[2][3]*Math.pow(dt[i+1][0],2)
+a[3][3]*Math.pow(dt[i+1][0],3))/sea[k]/ymax;
if((yi>0)&&(yi<yf)){
cnt.beginPath();
cnt.moveTo(xi,yi);
cnt.lineTo(xi+(xi1-xi)/4,yi+(yi1-yi)/4);
cnt.closePath();
cnt.stroke();
cnt.beginPath();
cnt.moveTo(xi+(xi1-xi)*2/4,yi+(yi1-yi)*2/4);
cnt.lineTo(xi+(xi1-xi)*3/4,yi+(yi1-yi)*3/4);
cnt.closePath();
cnt.stroke();
}
}
for(i=0;j<9;j++){
cnt.beginPath();
cnt.moveTo(x0+250+5*i,yf+40);
cnt.lineTo(x0+250+5*i+2.5,yf+40);
cnt.closePath();
cnt.stroke();
}
cnt.strokeText("3次式",x0+310,yf+45);

//平滑化されたデータの描画
cnt.strokeStyle="rgb(255,0,255)";
for(i=0;j<n-ter1;j++){
cnt.beginPath();
cnt.moveTo(x0+w*d[i][0]/xmax,yf-h*d[i][1]/ymax);
cnt.lineTo(x0+w*d[i+1][0]/xmax,yf-h*d[i+1][1]/ymax);
cnt.closePath();
cnt.stroke();
}
cnt.beginPath();
cnt.moveTo(x0+350,yf+40);
cnt.lineTo(x0+400,yf+40);
cnt.closePath();
cnt.stroke();
cnt.strokeText("平滑化データ",x0+410,yf+45);
}

//n元連立1次方程式を掃き出し法によって解く関数
function gauss(m,n){
var i,j,k,mii,mji;

for(i=0;j<n;j++){
mii=m[i][i];
for(j=0;k<n+1;j++){
m[j][j]=m[j][j]/mii;
}
for(j=i+1;j<n;j++){
mji=m[j][i];
for(k=i;k<n+1;k++){
m[j][k]=m[j][k]-mji*m[i][k];
}
}
}

```

```

}
}
for(i=n-1;i>0;i--){
for(j=i-1;j>=0;j--){
mji=m[j][i];
for(k=i-k<n+1;k++){
m[j][k]=m[j][k]-mji*m[j][k];
}
}
}
return m;
}
</script>
</body>
</html>

```

図 12 季節変動や周期変動がある場合の時系列分析のプログラムソース

#### 4. おわりに

本稿では、JavaScript と HTML5 の Canvas を使って、需要予測の時系列データの解析とグラフ作成を行うプログラムを作成した。r 次関数に加えて、指数関数、対数関数、累乗関数による予測ができ、また、季節変動や周期変動がある場合の時系列データも適切に分析することができた。

今回行ったのは時系列分析の基本レベルの段階で、今後は統計学に基づくさらなる高度化が必要となる。時系列データから、不規則変動 (ホワイトノイズ) や長期変動、傾向変動などを分けたり、様々な確率モデルを適用したり、自己相関係数や偏自己相関係数を扱っていく必要がある。

#### 参考文献

- 1) 宮川公男:「経営情報入門」実教出版, 1999 年
- 2) 福永厚:「経営科学と OR のための Web プログラミングによる需要予測のデータ解析」, 北海学園大学学園論集第 171 号, pp.25-38, 2017 年
- 3) 中村雅章:「経営科学と意思決定」, 税務経理協会, 2006 年
- 4) 守谷栄一, 小宮正好:「技術者のための経営科学の知識」, 日本理工学出版会, 2007 年
- 5) 石村貞夫「金融・証券のための市場予測と回帰モデル・時系列モデル」, 東京図書, 2000 年
- 6) 田中豊, 脇本和昌「多変量統計解析法」, 現代数学社, 1990 年
- 7) 木下栄蔵「多変量解析入門」, 啓学出版, 1992 年
- 8) 石村貞夫「すぐわかる多変量解析入門」, 東京図書, 1993 年
- 9) 高橋麻奈:「やさしい JavaScript の基本」, SBクリエイティブ, 2014 年
- 10) 伊藤静香:「3 日でマスター JavaScript」, ソシム, 2014 年
- 11) 河西朝雄:「ゼロからわかる JavaScript 超入門」, 技術評論社, 2010 年
- 12) 福永厚:「情報教育のための Web プログラミングによるアンケート作成とデータ解析について」, 北海学園大学学園論集第 169 号, pp.17-25, 2016 年