

タイトル	Webプログラミングによる在庫管理のABC分析とPPM
著者	福永，厚； FUKUNAGA, Atsushi
引用	北海学園大学学園論集(181)： 33-43
発行日	2020-03-25

Web プログラミングによる 在庫管理の ABC 分析と PPM

福 永 厚

1. はじめに

経営科学と OR (Operations Research) で扱う問題の一つに在庫管理がある¹⁾。在庫とは、将来の需要に備えて商店や工場・倉庫などで、商品や原材料・部品などを一時的に保管することである。流通業、特に小売業の場合、在庫は、自店舗で販売するために、予め仕入れて店頭で陳列したり、倉庫に保管する商品のことである。非常に値段の高い商品や特殊な商品であれば、顧客から注文を受けてから仕入れる方式もあるが、通常は予め商品を仕入れておいてそれを販売する。

在庫がなければ顧客が商品を求めて店頭に来たときに販売することができず、販売機会を失うことになる。一方、在庫が過剰であれば、在庫を保管しておくことにコストがかかり、売れ残り在庫の一掃セールや処分によって、コストが膨らむことになる。従って、需要に合わせた適正な在庫量が必要となる。

筆者は、以前、Microsoft 社の Excel の VBA プログラミングによって、在庫管理のデータ分析を行った²⁾。Excel は広く使われているソフトであるが、コンピュータに Excel が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。そこで筆者は、特

別なソフトを必要せず、Web ブラウザがあればできるように、JavaScript プログラミングによる在庫管理のデータ分析を行った³⁾。JavaScript は、Web ページを作成する HTML (HyperText Markup Language) の中に記述するスクリプト言語で、JavaScript 単独で記述されるプログラムは、どの Web ブラウザ上でも実行できる便利な言語であったが、グラフィック機能が乏しかった。2014 年に改定された HTML バージョン 5 (以下、HTML5) には Canvas 要素が導入され、Canvas と JavaScript の連携によって、様々なグラフが描けるようになった⁴⁾⁻⁵⁾。

本稿では、以前行った在庫管理のデータ分析をさらに発展させて、Canvas と JavaScript を使って最適発注量のグラフ化を行うとともに、在庫管理の ABC 分析とパレート図作成の自動化を行う。さらに、パレート図のグラフ化と似ている為、PPM (プロダクト・ポートフォリオ・マネジメント) の分析とグラフ作成の自動化も行う。

以下、第 2 章では在庫管理と PPM の概要について、第 3 章では HTML5 の Canvas と JavaScript による最適発注量のグラフ化および ABC 分析と PPM のグラフ化を自動的に行うプログラムについて論述し、第 4 章

でまとめる。

2. 在庫管理とPPM

2.1 最適発注量

在庫管理にかかる費用としては様々な費用が考えられるが、代表的なものとしては、商品を発注する際にかかる手数料等の発注費用と、届いた商品の保管にかかる保管料等の在庫維持費用である。発注費用と在庫維持費用が1回の発注量に依存すると、1回の発注量を多くし発注回数を減らして発注費用を抑えると、仕入れた商品の在庫量が多くなり在庫維持費用が増える。一方、1回の発注量を少なくし在庫量を減らして在庫維持費用を抑えると、発注回数が多くなり発注費用が増える。発注費用と在庫維持費用はトレードオフの関係にあり、発注費用と在庫維持費用を合わせた総費用を考えて、総費用を最小にするような1回の発注量を決めなくてはならない。この発注量を最適発注量（EOQ：Economic Order Quantity）という。

ここでは、在庫管理の最も簡単な場合として、年間の需要は確定していて毎営業日ごとに同じ数だけ売れる場合を考える。発注費用は、等間隔での発注の回数に比例し、在庫維持費用も1個1年あたりの在庫にかかる費用に比例するとする。その場合の定式化から得られる最適発注量は以下の通りである。

年間需要量を D 個、1回の発注にかかる固定費用を A 円、商品を1個1年保管しておくのにかかる費用を B 円、1回の発注量を Q 、1年間の発注回数を N とする。

発注量 Q と発注回数 N には、 $Q = \frac{D}{N}$ 、 $N = \frac{D}{Q}$ の関係がある。発注費用は $\frac{AD}{Q} = AN$ 、在庫

維持費用は $\frac{BQ}{2} = \frac{BD}{2N}$ で計算され、発注費用と在庫維持費用を合わせた総費用は、 $\frac{AD}{Q} + \frac{BQ}{2} = AN + \frac{BD}{2N}$ で求められる。

総費用を1回の発注量 Q で1回微分して、最小となる Q を求める式が最適発注量（Economic Order Quantity）の公式となる。最適発注量 Q^* は、

$$Q^* = \sqrt{\frac{2AD}{B}}$$

で求められる。

2.2 ABC 分析

実際の在庫管理では、数多くの商品を扱うことになり、それぞれにきめ細かい在庫管理を行うことは非常に難しい。そこで、商品をAグループ、Bグループ、Cグループの3つに分類し在庫管理の重点度を変える¹⁾。売上に大きく貢献する商品群をAグループと呼び、品切れを起こさないよう在庫管理に最も力を入れる。Aグループに分類される商品群の目安としては、総売上高の50～70%くらいの売上を上げる商品群で、商品数は少ない。Cグループは、商品数は非常に多いが売上への貢献は少なく、総売上の10～20%くらいである。Cグループの在庫管理にはあまり手をかけず大雑把に行う。Bグループは、AグループとCグループの中間的な位置づけで、総売上の20～30%の売上で在庫管理も中間的な力の入れようである。

ABC分析では、扱う商品の売上合計を求めて、各商品の売上合計に対する売上比率を求め、売上比率の大きい順に商品を並べ替える。各グループの売上比率の上限値を設定し、Aグループの売上比率に入る商品を大き

い順に上限値を超えない範囲でぎりぎりを選んで、A グループに属する商品を決める。次に、A グループの商品を除いた商品の売上高の大きい順に B グループの売上比率の上限以内に入る商品を決める。A グループ、B グループに属する商品を除いた商品が C グループに分類される。

パレート図とは、売上比率の大きい順に並べられた商品を横軸に取り、縦軸には売上比率の累計を取った折れ線グラフである。

2.3 PPM

PPM は、ボストン・コンサルティング・グループやゼネラル・エレクトリック、ハーバード・ビジネス・スクールなどが協力して開発した手法である⁶⁾。全社的な観点から、各製品の位置づけを明らかにしてマーケティング戦略を考えていく手法である。縦軸にその製品の市場の成長率、横軸に製品の市場占有率をとった 2 次元マトリックスを作成する。市場成長率と市場占有率の高低により、4 つのセルに分類し、市場成長率と市場占有率がともに高い製品を「花形製品」、市場成長率が低い市場占有率が高い製品を「金の生る木製品」、市場成長率が高く市場占有率が低い製品を「問題児製品」、市場成長率も市場占有率もともに低い製品を「負け犬製品」と呼ぶ。

2.4 HTML5 の Canvas と JavaScript

HTML は、Web ページを記述する言語であり、W3C によって 2014 年に新しいバージョンである HTML バージョン 5 に改定された^{4), 5)}。

HTML5 では、新しく Canvas 要素が導入

され、JavaScript と連動することにより、画像やアニメーションの動的コンテンツが生成できるようになった。ただし、Canvas では、線や四角形を描いたり色を塗ることはできるが、描いた図を動かすことはできず、アニメーションを作りたい場合は一コマずつ図を書き直すという処理を行わなければならない。

Canvas 要素は、元々はアップル社が MacOS に導入した技術で、HTML5 に取り入れられ、現在では、Safari, Opera, Firefox のあるバージョン以降、対応している。Internet Explorer (IE) は、当初対応しておらず、IE6 以降、Canvas をエミュレートで対応していたが、最新のブラウザ Microsoft Edge では対応している。

Canvas を JavaScript で使うには、DOM (Document Object Model) によって、canvas 要素を指定し、操作を行う⁷⁾。HTML 文書中の canvas 要素に、

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

というように、“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。

canvas 要素で指定できる属性は、ボックス領域の幅と高さを表す width 属性と height 属性である。

本稿で用いる図形や文字を描く主な

Canvas 機能は以下のものである。

`strokeRect(x,y,w,h)...`(x,y) を左上端とする
幅 w, 高さ h の四角形を描く

`fillRect (x,y,w,h)...`(x,y) を左上端とする幅
w, 高さ h の塗りつぶし四角形を描く
`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。

`arc(x,y,r,0,2 $\pi$ ,anticlockwise)...`(x,y) を中心
とする半径 r の円を反時計回りに描く
`fill()`...塗りつぶす
`fillStyle...`図形の塗りつぶしの色を指定する

`rgba...`色を指定する場合に使用し, RGB
は赤, 緑, 青を 0~255 の数値で指定する。
A は透明度を表し, 0 (透明)~1 (不透明) の値で指定する。

JavaScript は, HTML の中に記述するスクリプト言語で, Java 言語に言語体系が似ているオブジェクト指向言語である。JavaScript は, クライアントコンピュータで動き, どのブラウザも対応している。HTML の `<script>~</script>` の中に記述する。

JavaScript の中で文字列や計算結果を表示する際には通常 `document.write` を用いる。しかし, 本稿のように Web ページ上でフォームタグを使ってデータを入力し, 同一ページに Canvas で描画する場合に, データ分析結果を `document.write` で表示すると, 新規に別ページが開いてしまい, そのページには分析結果のみが表示されてしまう。これ

を避けて最初と同じページに表示する為には, `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^{8),9)}。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=出力結果;`

によって, ID 名 "result" の部分に出力するのである。このような方法によって, データ分析結果が, 入力テキストボックスや Canvas と同じページに出力することができる。

3. Web プログラミングによるプログラムの実行結果

・最適発注量

在庫管理の例として, 年間需要量 $D=160$ 個, 1 回の発注にかかる発注費用 $A=80$ 円, 商品を 1 個 1 年保管しておくのにかかる費用 $B=100$ 円の場合に, Web プログラミングによる最適発注量と総費用の計算およびグラフ作成の結果が図 1 (Microsoft Edge のブラウザ画面) に示されている。

入力フォームに値を入力し, 「最適発注量の計算とグラフ表示」ボタンをクリックすると関数 `eqq()` が実行され, 下方に計算結果とグラフが表示される。最適発注量の計算では, 小数第 1 位で四捨五入して整数化している。グラフでは, 最適発注量が横軸の中央になるように位置付けられ, その 2 倍が横軸の

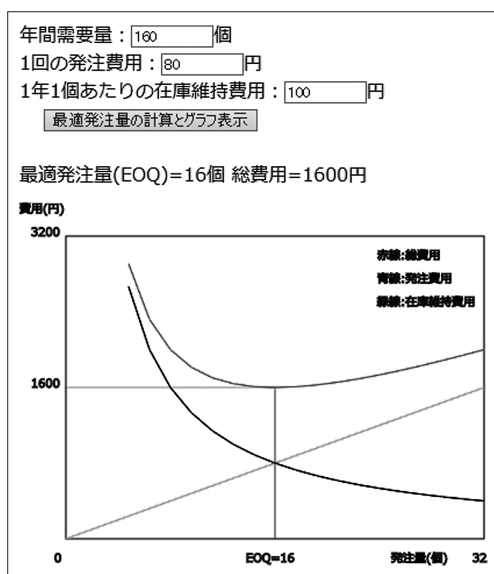


図1 最適発注量の計算 (Microsoft Edge によるブラウザ画面)

最大目盛となるようにしている。縦軸の費用においても、最適発注量のときの総費用の金額が縦軸の中央になり、最大目盛が2倍になるようにしている。それぞれの費用については、発注費用が青色、在庫維持費用が緑色、総費用が赤色で表示している。最適発注量の場合の総費用は、その位置まで水平線と垂直線によって示されている。他のいろいろな数値の場合にも正しく計算され、図が描画される。

図2には、このプログラムのソースが表示されている。

・ABC 分析

図3には、ABC 分析を行う為の入力フォームが配置された Microsoft Edge のブラウザ画面が表示されている。A グループと B グループの上限値と商品数を入力させている。商品名と売上高は10セット分が入力できる

```
<!DOCTYPE HTML>
<html lang="ja">
<head><title>最適発注量</title></head>
<body>

<form name="form1">
  年間需要量: <input type="text" name="demand" size="10">個<br>
  1回の発注費用: <input type="text" name="order" size="10">円<br>
  1年1個あたりの在庫維持費用: <input type="text" name="stock" size="10">円<br>

  <input type="button" value="最適発注量の計算とグラフ表示"
  onClick="eqo();"><br></form>
  <div id="result">
  </div>
  <canvas id="canvas" width="500" height="400"></canvas>
</body>

</p>

<script>
function eqo(){
  var d,a,b,eqo,teqo;

  d=Number(document.form1.demand.value);
  a=Number(document.form1.order.value);
  b=Number(document.form1.stock.value);
  eqo=Math.round(Math.sqrt(2*a*d/b));
  teqo=Math.round(a*d/eqo+0.5*b*eqo);

  var result=document.getElementById("result");
  result.innerHTML+=result.innerHTML+" 最適発注量(EOQ)="+teqo+"個 "+
  "総費用="+teqo+"円<br>";

  //グラフの作成
  var lx,ym,xmax,ymax,xi,yi,xh,yh,xf,yf,x1,y1,x2,y2;

  xm=eqo/10;
  xmax=eqo*10*xm;
  ymax=Math.ceil(2*teqo);

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

  //(xi,yi)・・・左上隅の座標、(xf,yf)・・・右下隅の座標、xh・・・横軸の幅、yh・・・縦軸の長さ

  //グラフ枠の描画
  ctx.rect(lx,ym,xh,yh);
  ctx.stroke();

  //最適発注量を示す線の描画
  ctx.strokeStyle="rgb(0,0,0)";
  ctx.lineWidth=2;
  ctx.beginPath();
  x1=xi+xh*eqo/xmax;
  y1=yf-yh*teqo/ymax
  ctx.moveTo(x1,yf);
  ctx.lineTo(x1,y1);
  ctx.closePath();
  ctx.stroke();
  ctx.moveTo(x1,y1);
  ctx.lineTo(xi,y1);
  ctx.closePath();
  ctx.stroke();

  //各見出し、キャプションの描画
  ctx.lineWidth=1;
  ctx.strokeText("0",xi-10,yf+20);
  ctx.strokeText("EOQ="+eqo,xi-25,yf+20);
  ctx.strokeText(2*eqo,2*x1-50,yf+20);
  ctx.strokeText(teqo,xi-30,y1);
  ctx.strokeText(2*teqo,xi-30,yi);
  ctx.strokeText("発注量(個)",x1+100,yf+20);
  ctx.strokeText("費用(円)",0,20);
  ctx.strokeText("赤線:総費用",1.4*x1,yi+20);
  ctx.strokeText("青線:発注費用",1.4*x1,yi+40);
  ctx.strokeText("緑線:在庫維持費用",1.4*x1,yi+60);

  //在庫維持費用の描画
  ctx.strokeStyle="rgb(0,255,0)";
  ctx.beginPath();
  ctx.moveTo(x1,yf);
  ctx.lineTo(xf,yf-yh*0.5*b*xmax/ymax);
  ctx.closePath();
  ctx.stroke();

  //発注費用の描画
  ctx.strokeStyle="rgb(0,0,255)";
  for(i=1;i<20;i++){
    x1=xi+i*xm*xh/xmax;
    y1=yf-yh*a*d/(i*xm)+0.5*b*i*xm/ymax;
    x2=x1+(i-1)*xm*xh/xmax;
    y2=yf-yh*a*d/((i-1)*xm)+0.5*b*(i-1)*xm/ymax;
    if((y1>0)&&(y2>0)){
      ctx.beginPath();
      ctx.moveTo(x1,y1);
      ctx.lineTo(x2,y2);
      ctx.closePath();
      ctx.stroke();
    }
  }

  //総費用の描画
  ctx.strokeStyle="rgb(255,0,0)";
  for(i=1;i<20;i++){
    x1=xi+i*xm*xh/xmax;
    y1=yf-yh*(a*d/(i*xm)+0.5*b*i*xm)/ymax;
    x2=x1+(i-1)*xm*xh/xmax;
    y2=yf-yh*(a*d/((i-1)*xm)+0.5*b*(i-1)*xm)/ymax;
    if((y1>0)&&(y2>0)){
      ctx.beginPath();
      ctx.moveTo(x1,y1);
      ctx.lineTo(x2,y2);
      ctx.closePath();
      ctx.stroke();
    }
  }
}

</script>
</body>
</html>
```

図2 最適発注量を計算するプログラムソース

ABC分析

各グループの上限値を%単位で入力してください。

グループA: %
 グループB: %

商品数を入力:
 10個までのデータを入力

	商品名	売上高
商品 1	商品A	30
商品 2	商品B	375
商品 3	商品C	90
商品 4	商品D	225
商品 5	商品E	600
商品 6	商品F	30
商品 7	商品G	150
商品 8		
商品 9		
商品 10		

ABC分析とパレート図の作成

図 3 ABC 分析の入力部分 (Microsoft Edge によるブラウザ画面)

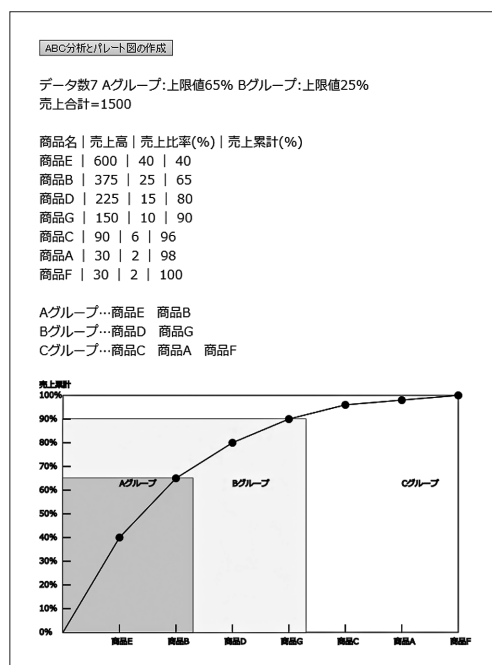


図 4 ABC 分析の計算結果とパレート図 (Microsoft Edge によるブラウザ画面)

ように配置されているが、10 個より少ない場合にも対応できる。10 個より多く入れる場合はプログラム上で入力フォームを増やさなければならないが、プログラムの変更は容易である。

図 4 には、プログラムの実行結果が表示されている。「ABC 分析とパレート図の作成」ボタンをクリックすると関数 abc() が実行される。商品をバブルソートアルゴリズム¹⁰⁾によって売上高の大きい順に並べ替え、売上高の合計を計算し、各商品の売上高の比率と累計を求めている。売上累計が A グループの上限値に最も近くて超えないところを探し、A グループの商品を決める。売上累計から A グループの売上比率を抜き去り、残った商品の売上累計が B グループの上限値に

最も近くて超えないところを探し、B グループの商品を決める。A, B グループ以外の残りの商品を C グループとする。

パレート図では、横軸に大きい順に並べられた商品、縦軸に売上累計を取り、商品の累計を小さな黒い円で表し、前後の円と直線で結んでいる。A グループに属する領域はピンク色で塗りつぶし、B グループに属する領域は黄色で塗りつぶしている。

他のいろいろな数値の場合にも正しく計算され、図が描画される。

図 5 には、このプログラムのソースが表示されている。

・PPM

PPM では、市場の年間成長率と市場占有率の入力が必要となるが、本稿では文献 6)

Web プログラミングによる在庫管理の ABC 分析と PPM (福永 厚)

```

<DOCTYPE HTML>
<html lang="ja">
<head><title>ABC分析</title></head>
<body>
<h1>ABC分析</h1>
<p>
各グループの上限値を%単位で入力してください。<br>
</p>
<form name="form1">
グループA:<input type="text" name="amax" size="10">%<br>
グループB:<input type="text" name="bmax" size="10">%<br>
</p>
商品数を入力:<input type="text" name="n" size="10"><br>
10番までのデータを入力<br>
商品名 売上高<br>
商品1 <input type="text" name="x0" size="20">
<input type="text" name="y0" size="20"><br>
商品2 <input type="text" name="x1" size="20">
<input type="text" name="y1" size="20"><br>
商品3 <input type="text" name="x2" size="20">
<input type="text" name="y2" size="20"><br>
商品4 <input type="text" name="x3" size="20">
<input type="text" name="y3" size="20"><br>
商品5 <input type="text" name="x4" size="20">
<input type="text" name="y4" size="20"><br>
商品6 <input type="text" name="x5" size="20">
<input type="text" name="y5" size="20"><br>
商品7 <input type="text" name="x6" size="20">
<input type="text" name="y6" size="20"><br>
商品8 <input type="text" name="x7" size="20">
<input type="text" name="y7" size="20"><br>
商品9 <input type="text" name="x8" size="20">
<input type="text" name="y8" size="20"><br>
商品10 <input type="text" name="x9" size="20">
<input type="text" name="y9" size="20"><br>
</p>
<input type="button" value="ABC分析とパレート図の作成" onClick="abc0;"><br>
</form>
<div id="result">
</div>
<canvas id="canvas" width="600" height="400"></canvas>
<br><br>
</p>
</script>
function abc0()
var n,amax,bmax;
var a,b,i,j;

n=Number(document.form1.n.value);
amax=parseFloat(document.form1.amax.value);
bmax=parseFloat(document.form1.bmax.value);

var result=document.getElementById("result");
result.innerHTML=result.innerHTML+"データ数 "+n+" Aグループ上限値 "+amax
+"% Bグループ上限値 "+bmax+"%<br>";

//d[i][j]...データ、i...データ番号(入力順) j...商品名 1:売上高 2:売上比率 3:売上累計

var d=new Array();
for(i=0;i<10;i++)
d[i]=new Array();

//フォームの商品名と売上高を2次元配列d[i][j]に代入

d[0][0]=document.form1.x0.value;
d[0][1]=Number(document.form1.y0.value);
d[1][0]=document.form1.x1.value;
d[2][0]=document.form1.x2.value;
d[2][1]=Number(document.form1.y2.value);
d[3][0]=document.form1.x3.value;
d[3][1]=Number(document.form1.y3.value);
d[4][0]=document.form1.x4.value;
d[4][1]=Number(document.form1.y4.value);
d[5][0]=document.form1.x5.value;
d[5][1]=Number(document.form1.y5.value);
d[6][0]=document.form1.x6.value;
d[6][1]=Number(document.form1.y6.value);
d[7][0]=document.form1.x7.value;
d[7][1]=Number(document.form1.y7.value);
d[8][0]=document.form1.x8.value;
d[8][1]=Number(document.form1.y8.value);
d[9][0]=document.form1.x9.value;
d[9][1]=Number(document.form1.y9.value);

//売上高の大きい順に並べ替えて、d[i][j]に格納

for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)
if(d[j-1][1]>d[i][1]){
a=d[j-1][1];
b=d[i-1][1];
d[i-1][1]=d[j][1];
d[j-1][1]=d[i][1];
d[i][0]=a;
d[j][0]=b;
}
}

//売上合計、売上比率、売上累計の計算
var sum,goukei,ruikei;
sum=0;
for(i=0;i<n;i++)
sum=sum+d[i][1];
result.innerHTML=result.innerHTML+"売上合計="+sum+"<br>";
for(i=0;i<n;i++)
d[i][2]=Math.round(d[i][1]/sum*10000)/10000;

ruikei=0;
for(i=0;i<n;i++)
ruikei=ruikei+d[i][2];
d[i][3]=Math.round(ruikei*10000)/10000;

result.innerHTML=result.innerHTML+"<br>商品名 | 売上高 | 売上比率(%) | 売上累計(%)<br>";
for(i=0;i<n;i++)
result.innerHTML=result.innerHTML+d[i][0]+" | "+d[i][1]+" | "+d[i][2]*100+" | "+d[i][3]*100+"<br>";

//グループ分け
var anum=0;
var bnum=0;
if(d[0][3]*100<amax){
for(i=0;i<n;i++){
if(d[i][3]*100<amax){
d[i][4]=0;
anum=anum+1;
}
else if(d[i][3]*100<=(amax+bmax)){
d[i][4]=1;
bnum=bnum+1;
}
else{
d[i][4]=2;
}
}
}
else{
d[0][4]=0;
anum=anum+1;
for(i=1;i<n;i++){
if(d[i][3]-d[0][3]*100<=bmax){
d[i][4]=1;
bnum=bnum+1;
}
else{
d[i][4]=2;
}
}
}

result.innerHTML=result.innerHTML+"<br>Aグループ...";
for(i=0;i<n;i++){
if(d[i][4]==0){
result.innerHTML=result.innerHTML+d[i][0]+" ";
}
}
result.innerHTML=result.innerHTML+"<br>Bグループ...";
for(i=0;i<n;i++){
if(d[i][4]==1){
result.innerHTML=result.innerHTML+d[i][0]+" ";
}
}
result.innerHTML=result.innerHTML+"<br>Cグループ...";
for(i=0;i<n;i++){
if(d[i][4]==2){
result.innerHTML=result.innerHTML+d[i][0]+" ";
}
}
result.innerHTML=result.innerHTML+"<br><br>";

//パレート図の作成
var c=document.getElementById("canvas");
var cnt=c.getContext("2d");

xi=30;
yi=20;
xi=500;
yi=300;
xf=xi+rh;
yf=yi+yh
xm=xi/2;
ym=yi/10;

//グラフ枠の描画
cnt.rect(xi,yh,yh);
cnt.stroke();

//Aグループ、Bグループの色分け
cnt.fillStyle="yellow";
cnt.fillRect(xi,yf-yh/100*(amax+bmax),(anum+bnum+0.3)*xm,yh/100*(amax+bmax));
cnt.fillRect(xi,yf-yh/100*(amax+bmax),(anum+bnum+0.3)*xm,yh/100*(amax+bmax));
cnt.strokeText("Bグループ",xi+(anum+bnum-1)*xm,yh/100*(amax+bmax)/2);
cnt.fillStyle="pink";
cnt.fillRect(xi,yf-yh/100*amax,(anum+0.3)*xm,yh/100*amax);
cnt.fillRect(xi,yf-yh/100*amax,(anum+0.3)*xm,yh/100*amax);
cnt.strokeText("Aグループ",xi+(anum-1)*xm,yh/100*(amax+bmax)/2);
cnt.strokeText("Cグループ",xi+(anum+yf-yh/100*(amax+bmax))/2);

//目盛りの描画
for(i=1;i<n+1;i++){
cnt.beginPath();
cnt.moveTo(xi+i*xm,yf);
cnt.lineTo(xi+i*xm,yf+10);
cnt.closePath();
cnt.stroke();
}

for(i=0;i<10;i++){
cnt.beginPath();
cnt.moveTo(xi,yf+i*ym);
cnt.lineTo(xi+10,yf+i*ym);
cnt.closePath();
cnt.stroke();
}

for(i=0;i<10;i++){
cnt.strokeText(10*i+"%",xi-30,yf-i*ym+3);
}

cnt.strokeText("売上累計",xi-30,10);
for(i=0;i<n;i++){
cnt.strokeText(d[i][0],xi+(i+1)*xm-10,yf+15);
}

//線と点のプロット
cnt.fillStyle="black";
ab=xi;
yb=yf;
for(i=0;i<n;i++){
x=xi+(i+1)*xm;
y=yf-yh*d[i][3];
cnt.beginPath();
cnt.arc(x,y,5,0,Math.PI*2,true);
cnt.closePath();
cnt.fill();
cnt.beginPath();
cnt.moveTo(x,yb);
cnt.lineTo(x,y);
cnt.closePath();
cnt.stroke();
}
}

</script>
</body>
</html>

```

図5 ABC分析の計算とパレート図作成のプログラムソース

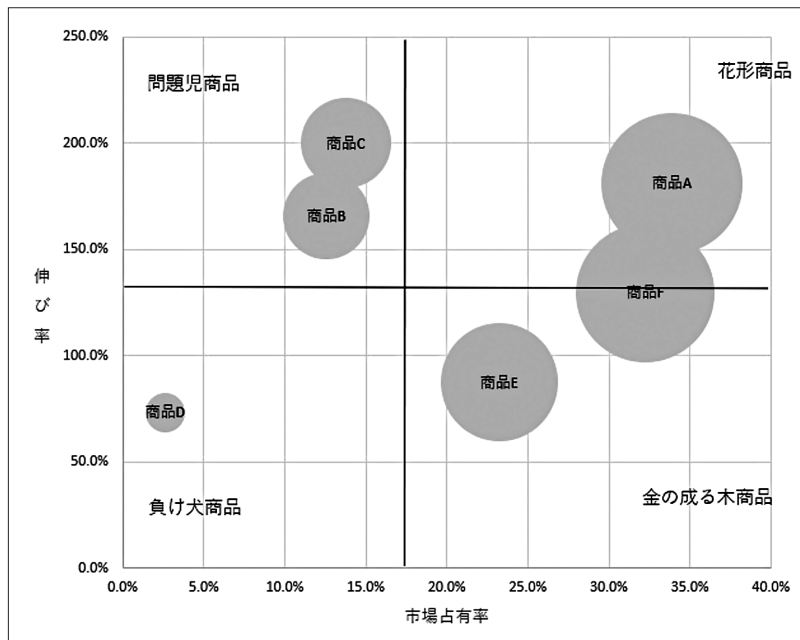


図6 PPMのバブルグラフ (Excel 2019 による)

と同様に入力された商品の過去と現在の売上比率，すなわち，

$$\text{伸び率} = \text{現在売上} \div \text{過去売上}$$

を年間成長率とする。また，市場占有率は入力された商品の現在売上の合計を求め，各商品の現在売上を合計によって割った値を用いる。

図6には，Excel 2019 のバブルグラフによる PPM 図が表示されている。横軸が市場占有率，縦軸が伸び率を表し，バブルの中心点がそれぞれの商品の値に対応し，バブルの大きさは現在売上の大きさと関連している。グラフ中央を横切る水平線は全体の伸び率を表し，垂直線は市場占有率の最大値の半分を表しており，これらはグラフ作成後に，図形描画機能により手動で追加したものである。この2本の線によってグラフが4つのセルに区

切られ，「花形商品」，「金の生る木商品」，「問題児商品」，「負け犬商品」に分類されている。セルの名称も横書きテキストボックス機能によって，追加したものである。

図7には，Web プログラミングによる入力画面が示されている。商品名，過去売上，現在売上のセットが10個配置されている。10セットより少なく入力しても良い。10セットより多く入力したい場合は，プログラムを変更しなければならないが，その作業は容易である。

図8には，「PPM図の作成」ボタンをクリックして関数 ppm() が実行された結果が示されている。バブルソートアルゴリズムによって，現在売上の小さい順に並べ替えられ，それぞれの商品の市場占有率と伸び率が表示されている。その下には，PPM図が描画され

PPM

商品数を入力:

10個までのデータを入力

	商品名	過去売上	現在売上
商品 1	商品A	7900	14300
商品 2	商品B	3200	5300
商品 3	商品C	2900	5800
商品 4	商品D	1500	1100
商品 5	商品E	11200	9800
商品 6	商品F	10500	13600
商品 7			
商品 8			
商品 9			
商品 10			

図 7 PPM の入力部分 (Microsoft Edge によるブラウザ画面)

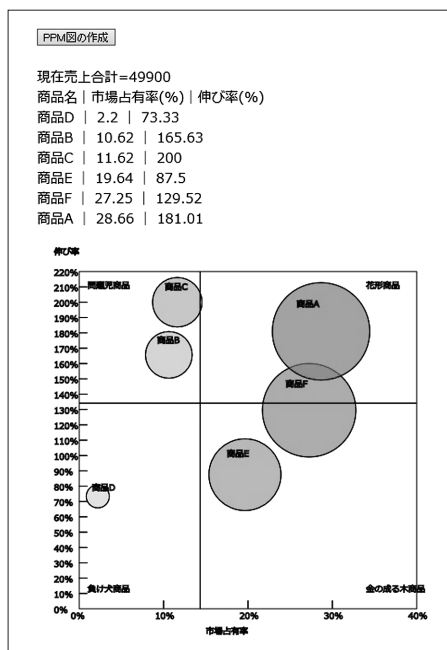


図 8 PPM 図 (Microsoft Edge によるブラウザ画面)

ている。Excel のバブルグラフは、バブルの大きさがどのように売上と関連しているかわからない為、現在売上の大きさに依存した大きさを適当に設定している。バブルの大きさも現在売上が大きくなるにつれて色が濃くなり、また、RGBA の A にあたる透明度も連動して上げている。4つのセルに分ける区切り線は自動的に引かれている。Excel のバブルグラフとほぼ同じ図が描かれていることがわかる。

他のいろいろな数値の場合にも正しく計算され、図が描画される。

図 9 には、このプログラムのソースが表示されている。

4. おわりに

本稿では、HTML5 の Canvas と

```

<!DOCTYPE HTML>
<html lang="ja">
<head><title>PPM</title></head>
<body>
<h1>PPM</h1>

<form name="form1">

<p>
商品数を入力:<input type="text" name="n" size="10"><br>
10個までのデータを入力<br>


| 商品名  | 過去売上                                    | 現在売上                                    |
|------|-----------------------------------------|-----------------------------------------|
| 商品1  | <input type="text" name="x0" size="20"> | <input type="text" name="y0" size="20"> |
| 商品2  | <input type="text" name="x1" size="20"> | <input type="text" name="y1" size="20"> |
| 商品3  | <input type="text" name="x2" size="20"> | <input type="text" name="y2" size="20"> |
| 商品4  | <input type="text" name="x3" size="20"> | <input type="text" name="y3" size="20"> |
| 商品5  | <input type="text" name="x4" size="20"> | <input type="text" name="y4" size="20"> |
| 商品6  | <input type="text" name="x5" size="20"> | <input type="text" name="y5" size="20"> |
| 商品7  | <input type="text" name="x6" size="20"> | <input type="text" name="y6" size="20"> |
| 商品8  | <input type="text" name="x7" size="20"> | <input type="text" name="y7" size="20"> |
| 商品9  | <input type="text" name="x8" size="20"> | <input type="text" name="y8" size="20"> |
| 商品10 | <input type="text" name="x9" size="20"> | <input type="text" name="y9" size="20"> |


<input type="button" value="PPM図の作成" onClick="ppm();"><br>
</form>
<div id="result">
</div>
<canvas id="canvas" width="500" height="500"></canvas>
<br><br>
</p>

<script>
function ppm(){
var n;
var a,b,c,i,j;

n=Number(document.form1.n.value);
var result=document.getElementById("result");

//d[i][j]...データ i...データ番号(入力順)
// j=0 商品名 1:過去売上 2:現在売上 3:市場占有率 4:伸び率

var d=new Array();
for(i=0;i<10;i++){
d[i]=new Array();
}

//フォームの商品名と過去売上、現在売上を2次元配列d[i][j]に代入

d[0][0]=document.form1.x0.value;
d[0][1]=Number(document.form1.y0.value);
d[0][2]=Number(document.form1.z0.value);
d[1][0]=document.form1.x1.value;
d[1][1]=Number(document.form1.y1.value);
d[1][2]=Number(document.form1.z1.value);
d[2][0]=document.form1.x2.value;
d[2][1]=Number(document.form1.y2.value);
d[2][2]=Number(document.form1.z2.value);
d[3][0]=document.form1.x3.value;
d[3][1]=Number(document.form1.y3.value);
d[3][2]=Number(document.form1.z3.value);
d[4][0]=document.form1.x4.value;
d[4][1]=Number(document.form1.y4.value);
d[4][2]=Number(document.form1.z4.value);
d[5][0]=document.form1.x5.value;
d[5][1]=Number(document.form1.y5.value);
d[5][2]=Number(document.form1.z5.value);
d[6][0]=document.form1.x6.value;
d[6][1]=Number(document.form1.y6.value);
d[6][2]=Number(document.form1.z6.value);
d[7][0]=document.form1.x7.value;
d[7][1]=Number(document.form1.y7.value);
d[7][2]=Number(document.form1.z7.value);
d[8][0]=document.form1.x8.value;
d[8][1]=Number(document.form1.y8.value);
d[8][2]=Number(document.form1.z8.value);
d[9][0]=document.form1.x9.value;
d[9][1]=Number(document.form1.y9.value);
d[9][2]=Number(document.form1.z9.value);

//現在売上小さい順に並べ替えて、d[i][j]に格納
for(j=0;j<n-1;j++){
for(i=j+1;i<n;i++){
if(d[i-1][2]>d[i][2]){
a=d[i-1][0];
b=d[i-1][1];
c=d[i-1][2];
d[i-1][0]=d[i][0];
d[i-1][1]=d[i][1];
d[i-1][2]=d[i][2];
d[i][0]=a;
d[i][1]=b;
d[i][2]=c;
}
}
}

//市場占有率、伸び率の計算
var sum=new Array();
for(i=0;i<2;i++){
sum[i]=0;
for(j=0;j<n;j++){
sum[i]=sum[i]+d[j][i+1];
}
}
result.innerHTML=result.innerHTML+"現在売上合計="+sum[1]+"<br>";

for(i=0;i<n;i++){
d[i][3]=Math.round(d[i][2]/sum[1]*10000)/100;
}

for(i=0;i<n;i++){
d[i][4]=Math.round(d[i][2]/d[i][1]*10000)/100;
}

result.innerHTML=result.innerHTML+"商品名 | 市場占有率(%) | 伸び率(%)<br>";
for(i=0;i<n;i++){
result.innerHTML=result.innerHTML+d[i][0]+" | "+d[i][3]+" | "+d[i][4]+"<br>";
}

//PPM図の作成
var c=document.getElementById("canvas");
var cnt=c.getContext("2d");

var ix,yh,xmax,ymax,xi,yi,xw,yf,yf,x,y;
var smax,gmax,hmax,vmax,r;

//市場占有率と伸び率の最大値
smax=d[0][3];
gmax=d[0][4];

for(i=1;i<n;i++){
if(d[i][3]>smax){smax=d[i][3];}
if(d[i][4]>gmax){gmax=d[i][4];}
}

hmax=Math.ceil(smax/10)+1;
vmax=Math.ceil(gmax/10)+2;

xi=50;
yi=50;
xw=400;
yw=400;
xf=xi+xw;
yf=yi+yw;
xh=xw/hmax;
yh=yw/vmax;

//グラフ枠の描画
cnt.rect(xi,yh,xw,yw);
cnt.stroke();

//目盛りの描画
for(i=1;i<hmax;i++){
cnt.beginPath();
cnt.moveTo(xi+ix,yf);
cnt.lineTo(xi+ix,yh+yf-10);
cnt.closePath();
cnt.stroke();
}

for(i=1;i<vmax;i++){
cnt.beginPath();
cnt.moveTo(xf,yf-iy);
cnt.lineTo(xf-10,yf-iy);
cnt.closePath();
cnt.stroke();
}

for(i=0;i<hmax;i++){
cnt.strokeText(10*i+"%",xi+ix-10,yf+12);
}

for(i=0;i<vmax;i++){
cnt.strokeText(10*i+"%",xi-30,yf-iy+5);
}

cnt.strokeText("伸び率",xi-30,yf-20);
cnt.strokeText("市場占有率",200,yf+30);

//円の描画
for(i=0;i<n;i++){
cnt.fillStyle="rgb("+[255,0,0.2+0.08*i]*"+)";
c.strokeStyle="black";
r=10+d[i][3]*xw/(hmax*10);
y=yf-d[i][4]*yw/(vmax*10);
r=10+d[i][3]*xw/(hmax*10)/8;
cnt.beginPath();
cnt.arc(xi,yf,0,Math.PI*2,true);
cnt.closePath();
cnt.stroke();
cnt.fill();
cnt.strokeText(d[i][0],xi-0.5*r,yf-0.5*r);
cnt.stroke();
}

//区切り線の描画
cnt.beginPath();
x=xi+0.5*d[i-1][3]*xw/(hmax*10);
cnt.moveTo(x,yf);
cnt.lineTo(x,yi);
cnt.closePath();
cnt.stroke();
cnt.beginPath();
y=yf-sum[1]/sum[0]*100*yw/(vmax*10);
cnt.moveTo(xf,y);
cnt.lineTo(xf,yi);
cnt.closePath();
cnt.stroke();

cnt.strokeText("負け犬商品",xi+10,yf-20);
cnt.strokeText("金の成る木商品",xf-60,yf-20);
cnt.strokeText("問題児商品",xi+10,yi+20);
cnt.strokeText("花形商品",xf-60,yi+20);
}

</script>
</body>
</html>

```

図 9 PPM 図作成プログラムソース

JavaScript を用いて、Web 上で在庫管理における最適発注量の計算とグラフ化、ABC 分析、さらには PPM 図の作成を自動的に行うプログラムを作成した。データを入力フォームに入力し、実行ボタンをクリックすると、適切に計算され、グラフが描かれることが確認された。

今後は、より現実的な様々な在庫管理の場合に対応できるようなプログラムを作成していく必要がある。

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:「経営科学のための VBA プログラミングによる在庫情報のデータ解析」, 北海学園大学経済論集第 49 巻第 2 号, pp.45-54, 2001 年
- 3) 福永 厚:「経営科学における情報リテラシーと Web プログラミングによる在庫情報のデータ解析」, 北海学園大学経済論集第 49 巻第 3 号, pp.81-91, 2001 年
- 4) 村山秀明:「HTML5 入門」, 工学社, 2012 年
- 5) スタジオ イー・スペース:「HTML5+CSS 標準テキスト」, 技術評論社, 2011 年
- 6) 飯島正樹, 大浜慶和, 伊藤和憲, 武藤明則:「経営情報シミュレーション」, 同文館出版, 2010 年
- 7) 高橋麻奈:「やさしい JavaScript の基本」, SB クリエイティブ, 2014 年
- 8) 伊藤静香:「3 日でマスター JavaScript」, ソシム, 2014 年
- 9) 河西朝雄:「ゼロからわかる JavaScript 超入門」, 技術評論社, 2010 年
- 10) 石田保輝, 宮崎修一:「アルゴリズム図鑑」, 翔泳社, 2017 年

