

Google Visualisation API を R で利用する： *googleVis*-0.2.14 パッケージ入門

Markus Gesmann*, Diego de Castillo†
2012 年 2 月 5 日版

日本語訳 荒木 孝治‡

2012 年 4 月 13 日

要旨

googleVis パッケージは、R と Google Visualisation API の間のインターフェースを提供する。Google Visualisation API は、ウェブページに埋め込むことができるインタラクティブチャートを提供する。このチャートで最も有名なものは、TED の講演において Hans Rosling が広めたモーションチャートだろう。

googleVis パッケージの機能によって、R のデータフレームとして保存されているデータを Google にアップロードしないで、Google Visualisation API で視覚化することができる。*googleVis* の関数の出力には、データおよび Google がホストする JavaScript 関数を参照する html コードが含まれる。

googleVis は、出力をローカルに表示するために R の内蔵 HTTP サーバを利用する。フラッシュを利用できるブラウザと、インターネットへの接続が必要である。

* markus.gesmann@gmail.com

† decastillo@gmail.com

‡ 翻訳に関する意見等は arakit@kansai-u.ac.jp まで。

目次

1	はじめに	3
1.1	動機	3
1.2	Google Visualisation API	3
2	<i>googleVis</i> パッケージ	5
2.1	インストール	6
2.2	<i>googleVis</i> パッケージの利用	7
2.3	モーションチャートの例	8
2.4	<i>gvis</i> オブジェクトをローカルに表示	13
2.5	<i>gvisMerge</i> によるチャートの結合	14
2.6	オプションの設定	14
2.6.1	チャートエディタ	15
2.6.2	列名におけるアポストロフィの取り扱い	16
3	<i>googleVis</i> をウェブサイト埋め込む	17
3.1	既にあるサイトに <i>gvis</i> オブジェクトを統合する	17
3.2	<i>googleVis</i> 出力を WordPress で利用する	18
3.3	<i>googleVis</i> の出力を Google サイト, Blogger 等で利用する	19
3.4	<i>googleVis</i> をウェブサイトに動的に埋め込む	20
3.4.1	<i>googleVis</i> を <i>R.rsp</i> で利用する	20
3.4.2	<i>RApache</i> と <i>brew</i> で <i>googleVis</i> を利用する	20
4	プレゼンテーションで <i>googleVis</i> を用いる	22
5	R を超えて	22
5.1	Registering to catch events	22
6	連絡	23
6.1	コラボレーション	23
6.2	引用	24
6.3	トレーニングとコンサルティング	24

1 はじめに

1.1 動機

ますます多くのデータが利用できる環境が整いつつあるが、データに関する物語と洞察は今なおしばしば欠如している。私たちはデータのジャングルの中で道に迷い、木を見て森を見ようと苦闘している。

それゆえに、データを生き返らせ、データを利用者にぴったりと合った様々な方法で分析し、様々な視点から観察し、語る価値のある物語（外れ値、トレンド、自明のものであっても）を見つけるための新しいツールが必要とされている。

2006年、Hans Rosling は過去50年間の世界の社会的・経済的発展について感動的な講演をTEDで行った[Ros06]。それは多くの聴衆の考え方や認識に疑問を呈した。Roslingは、結論に達するために詳細なデータ分析を使った。話を視覚化するために、彼とGapminder[Fou10b]のチームは、モーションチャートというバブルチャートのアニメーション版を開発した(Figure 1参照)。

Roslingのプレゼンテーションは、インタラクティブチャートについての考えと利用を世界に広めた。その結果の1つとして、Gapminderの背後にあったソフトウェアがGoogleによって買収され、1年後にGoogle Visualisation API[Inc11]にモーションチャートとして組み込まれた。

近年、データジャーナリズムが成長している。Guardian(英国)とtaz.de(Die Tageszeitung, ドイツ)のデータブログは、より広い観衆にデータ分析とデータ視覚化を提示した。

2010年、計算ファイナンスと金融工学に関するR/Rmetricsワークショップで、Sebastián Pérez Saaibi[Saa10]は、Googleのモーションチャートを使ってRの出力をR.rspパッケージ[Ben12]と一緒に視覚化するというアイデアを提案した。

これらのトークに刺激を受け、また、データ分析者と他者との対話を促進するためにインタラクティブな視覚化を行いたいという願望から、我々はgoogleVisパッケージ[GdC12],[GdC11]の開発を開始した。

もちろん、他に、Many Eyes[RtICsg10]、Open Flash Chart(Flash)[JG10]、OpenLayers(JavaScript)[Fou10c]、Processing(Java)[FR10]、simile(AJAX)[DKM10]、FLARE(ActionScript)[Lab10]といった多くの視覚化ツールキットが既に存在する。

1.2 Google Visualisation API

Google Visualisation API[Inc11]を用いると、インタラクティブチャートを作成して、それをGoogleドキュメント、表計算、ウェブページの一部とすることができる。本稿では、APIを利用する方法に焦点を合わせる。

Google Public Data Explorer[Inc10b]は、インタラクティブチャートの利用法、および、それがどのようにしてデータ分析の手助けとなるかということを示すよい例である。ここでの図のほとんどはブラウザ内で描画される。

作図データは、HTMLファイルに埋め込まれるか、ダイナミックに読み込まれるかのいずれかである。Google Visualisation APIの鍵は、データがDataTable[Inc10c]の中に構築されるということである。なぜなら、それはRJSONIOパッケージ[Lan12]の機能を利用して、RのデータフレームをJSON[JSO06]オブジェクト(これがDataTableの基礎になる)に変換するからである。

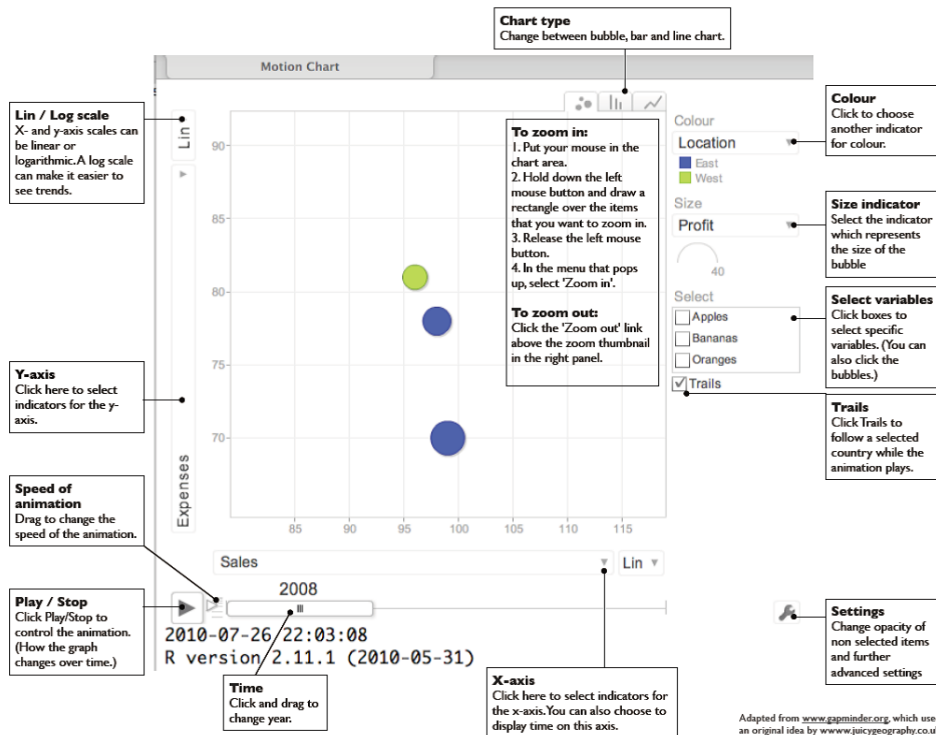


図 1: Google モーションチャートの概要. `plot(gvisMotionChart(Fruits, idvar='Fruit', timevar='Year'))` の出力のスクリーンショット.

例として, Google の visualisation gallery [Inc10a] にあるモーションチャートの html コードを見る. これは, 図 1 に示す図を生成する.

```

1 <html>
2 <head>
3 <script type="text/javascript"
4 src="http://www.google.com/jsapi">
5 </script>
6 <script type="text/javascript">
7 google.load('visualization', '1',
8 {'packages':['motionchart']});
9 google.setOnLoadCallback(drawChart);
10 function drawChart() {
11 var data=new google.visualization.DataTable();
12 data.addColumn('string', 'Fruit');
13 data.addColumn('date', 'Date');
14 data.addColumn('number', 'Sales');
15 data.addColumn('number', 'Expenses');
16 data.addColumn('string', 'Location');
17 data.addRows([
18 ['Apples',new Date(1988,0,1),1000,300,'East'],
19 ['Oranges',new Date(1988,0,1),1150,200,'West'],
20 ['Bananas',new Date(1988,0,1),300,250,'West'],
21 ['Apples',new Date(1989,6,1),1200,400,'East'],
22 ['Oranges',new Date(1989,6,1),750,150,'West'],
23 ['Bananas',new Date(1989,6,1),788,617,'West']

```

```

24 ]);
25 var chart=new google.visualization.MotionChart(
26 document.getElementById('chart_div'));
27 chart.draw(data, {width: 600, height:300});
28 }
29 </script>
30 </head>
31 <body>
32 <div id="chart_div"
33 style="width:600px; height:300px;">
34 </div>
35 </body>
36 </html>

```

コードとデータはブラウザの中で処理・描画され、サーバーには送られない¹.
上記の html コードは、次に示す 5 つの部分から構成される².

- Google の AJAX (第 4 行) と Visualisation API (第 7 行~8 行) の参照
- 視覚化に利用するデータを DataTable に変換 (第 11 行~24 行)
- チャートを作成するためのインスタンスの呼び出し (第 25 行~26 行)
- オプションを付けた描画メソッドの呼び出し. ここで示したオプションは、描画領域の *width* と *height* (第 27 行)
- ページにチャートを追加するための HTML の `<div>` 要素 (第 32 行~34 行)

この原則は、Google Visualisation API の大部分のインタラクティブチャートに当てはまる (図 2 の例を参照).

なお、API を利用する前に、Google Visualization API のサービス利用規約 [Inc12] を読んでおくべきである.

2 *googleVis* パッケージ

googleVis パッケージは、R と Google Visualisation API の間のインターフェースを提供する. パッケージの関数は、Google Visualisation API を用いて R データフレームに保存されているデータの視覚化を可能とする.

パッケージ (バージョン 0.2.15) は、モーションチャート (Motion Chart)、注釈付タイムライン (Annotated Time Line)、地理マップ (Geo Map)、地図 (Map)、地理チャート (Geo Chart)、強度マップ (Intensity Map)、表 (Table)、ゲージ (Gauge)、木マップ (Treemap)、さらに、折れ線グラフ (Line Chart)、横棒グラフ (Bar Chart)、バブルチャート (Bubble Chart)、縦棒グラフ (Column Chart)、面グラフ (Area Chart)、段階状面グラフ (Stepped Area)、複合グラフ (Combo Chart)、散布図 (Scatter Chart)、ロウソクチャート (Candlestick Chart)、円グラフ (Pie Chart)、組織図 (Org Chart) へのインターフェースを提供する. 例については、図 2 を参照. 利用可能なチャートの全てのリストが、プロジェクトのサイト³にある.

googleVis の関数の出力は、データと、Google がホストする JavaScript の関数への参照を含む HTML コードである. 出力を見るには、インターネットに接続されたブラウザが必要であり、モーショ

¹ http://code.google.com/apis/visualization/documentation/gallery/motionchart.html#Data_Policy

² 詳細については、http://code.google.com/apis/chart/interactive/docs/adding_charts.html を参照.

³ <http://code.google.com/p/google-motion-charts-with-r/wiki/GadgetExamples>

ンチャート, 地理マップ, 注釈付タイムラインには, さらにフラッシュが必要である. 実際のチャートは, ブラウザで描画される.

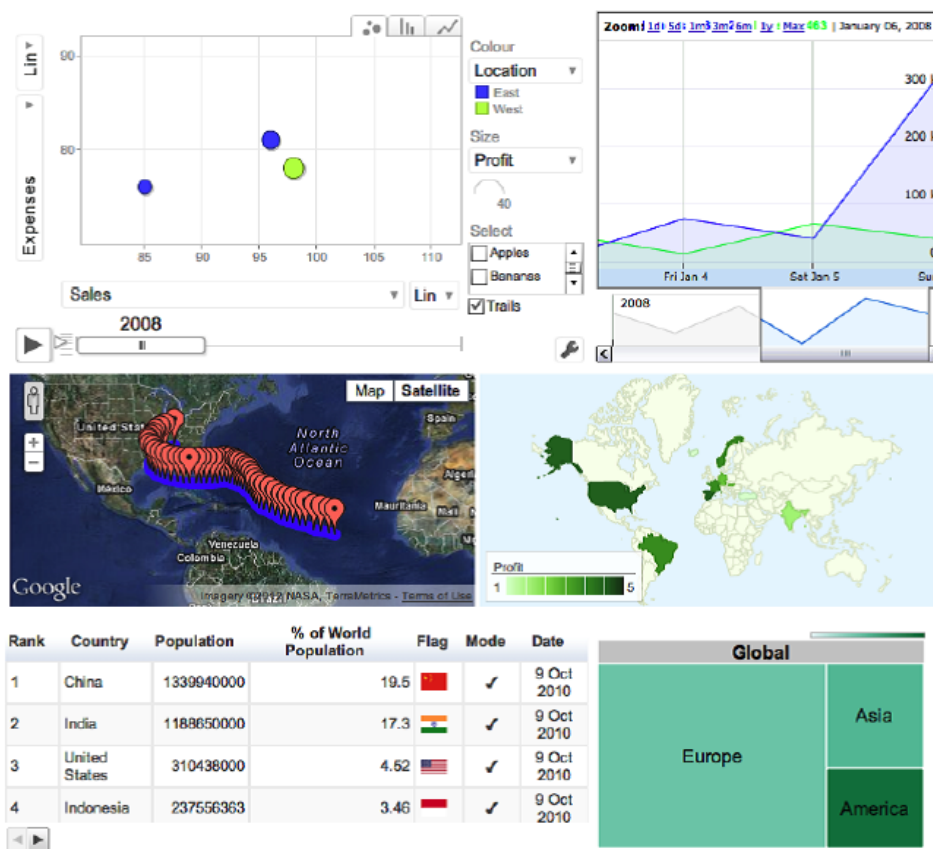


図 2: demo(googleVis) の出力のスクリーンショット. 上の左から時計回りに, gvisMotionChart, gvisAnnotatedTimeLine, gvisGeoMap, gvisTreeMap, gvisTable, gvisMap による.

2.1 インストール

googleVis のインストールは, CRAN より通常の方法で実行できる. 例えば.

```
R> install.packages('googleVis')
```

コマンド `library(googleVis)` が次のメッセージを表示したとき, インストールは成功したと判断できる.

```
R> library(googleVis)
```

```
Welcome to googleVis version 0.2.15
Please read the Google API Terms of Use
before you use the package:
http://code.google.com/apis/terms/index.html
```

Type `?googleVis` to access the overall documentation and

vignette('googleVis') for the package vignette.
You can execute a demo of the package via: demo(googleVis)

More information is available on the googleVis project web-site:
<http://code.google.com/p/google-motion-charts-with-r/>

Contact: <rvisualisation@gmail.com>

To suppress the this message use:
suppressPackageStartupMessages(library(googleVis))

2.2 googleVis パッケージの利用

googleVis パッケージ内の各関数は、ヘルプページにおいて詳細に文書化されている。本稿では、パッケージの原理のみを説明する。

例として、モーションチャートを作る方法を図 1 に示す。これは他の API に関しても同様である。さらなる例が、googleVis パッケージのデモ⁴にある。図 2 も参照のこと。

視覚化関数のデザインは、かなり共通している。名前は、'gvis'+ チャートの種類である。だから、モーションチャートの場合、次のようになる。

```
gvisMotionChart(data, idvar='id', timevar='date', options=list(), chartid)
```

ここで、data は入力データフレームであり、idvar と timevar では id 変数の列名とプロットする時間変数を指定する。表示のオプションは options リストで指定するが、これについては 14 ページで詳しく説明する。オプションとデータの要求事項は Google Visualisation API に準拠しており、ヘルプページで説明している。ヘルプを参照するには次のようにする。

```
R> help('gvisMotionChart')
```

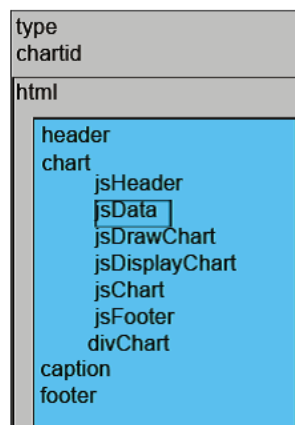


図 3: gvis リストオブジェクトの構造図。

⁴ 利用可能なデモのリストを表示するには、<http://code.google.com/p/google-motion-charts-with-r/wiki/GadgetExamples> を用いる。

引数 `chartid` により、出力チャートのチャート id を手動でセットすることができる。複数のチャートを 1 つのページに入れるには、ユニークなチャート id が必要である。

`googleVis` の関数の出力は、チャートタイプ、チャート id、html コード（サブリストとして、ヘッダ、チャート、キャプション、フッタを含む）といったリストのリスト（入れ子になったリスト）である（図 3 参照）。

このコンセプトの背後にある考え方は、特定のパーツ（例えばチャート）を抽出すると同時に、ユーザが完全なウェブページを得ることができるようにするということである。視覚化の出力を他のサイトに送ったり、それを `rsp` のページ（20 ページ参照）や `RApache`（20 ページ参照）、`Google` ガジェットに埋め込んだりする場合、特に有益である。

`googleVis` の関数の出力は、クラス `'gvis'` と `'list'` である。ジェネリックな `print` 関数（`print.gvis`）と `plot` 関数（`plot.gvis`）が、オブジェクトの取り扱いを容易にするためにある。コンセプトを例示するために、`Fruits` データセットを用いてモーションチャートを作成してみる。

2.3 モーションチャートの例

`Google` モーションチャート API のドキュメントによると、少なくとも 4 列を持つデータセットが必要である。1 つはプロットしたい変数、もう 1 つは時間の変数、そして少なくとも 2 つの数値変数である。さらに数値や文字の列があってもよい。

例として `Fruits` データセットを利用する。

```
R> data(Fruits)
R> Fruits
  Fruit Year Location Sales Expenses Profit      Date
1 Apples 2008     West   98        78    20 2008-12-31
2 Apples 2009     West  111        79    32 2009-12-31
3 Apples 2010     West   89        76    13 2010-12-31
4 Oranges 2008     East   96        81    15 2008-12-31
5 Bananas 2008     East   85        76     9 2008-12-31
6 Oranges 2009     East   93        80    13 2009-12-31
7 Bananas 2009     East   94        78    16 2009-12-31
8 Oranges 2010     East   98        91     7 2010-12-31
9 Bananas 2010     East   81        71    10 2010-12-31
```

`id` として列 `"Fruit"` を、時間の変数として `"Year"` を利用することにする。`"Year"` の代わりに `"Date"` を用いてもよい。

```
R> M <- gvisMotionChart(Fruits, idvar="Fruit", timevar="Year")
```

`gvisMotionChart` の出力の構造は、先に述べたようにリストのリストとなっている。

```
R> str(M)
List of 3
 $ type   : chr "MotionChart"
 $ chartid: chr "MotionChartID2dba6a651a62"
 $ html   :List of 4
 ..$ header : chr "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0
 ..$ chart  : Named chr [1:7] "<!-- MotionChart generated in R 2.
 .. ..- attr(*, "names")= chr [1:7] "jsHeader" "jsData" "jsDrawCh
 ..$ caption: chr "<div><span>Data: Fruits &#8226; Chart ID: <a h
 ..$ footer : chr "\n<!-- htmlFooter -->\n<span> \nR Under develo
```



```
- attr(*, "class")= chr [1:2] "gvis" "list"
```

リストの最初の2つの項目は、使われたチャートタイプ (`type`) とチャート id (`chartid`) に関する情報を含む⁵。

```
R> M$type
[1] "MotionChart"
R> M$chartid
[1] "MotionChartID2dba6a651a62"
```

html 出力は、`header` (ヘッダ)、`chart` (チャート)、`caption` (キャプション)、`footer` (フッタ) のリストである。これは、ユーザがページの特定の部分だけを抽出したり、完全な html ページを作ったりするのを可能とする。html ページのヘッダには、基本的な html とフォーマットタグだけがある。

```
R> print(M, tag='header')
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>MotionChartID2dba6a651a62</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <style type="text/css">
    body {
      color: #444444;
      font-family: Arial,Helvetica,sans-serif;
      font-size: 75%;
    }
    a {
      color: #4D87C7;
      text-decoration: none;
    }
  </style>
</head>
<body>
```

上記で、フォーマットされたスクリーン出力を得るために、`M$html$header` の代わりに `'header'` タグ (`tag`) を持つ `print` 命令を使った。これは、`cat(M$html$header)` の出力と同じである。

実際の Google visualisation のコードでは、html リストのチャート項目の名前付きの文字ベクトルとしてデータが保存されている。チャートは、JavaScript と HTML 文から構成される。JavaScript 関数がチャート id によりユニークに名付けられていることに注意。このコンセプトにより、ユーザがチャートのコードの全てまたは特定の部分だけを得ることができる。詳細については、`print.gvis` のヘルプページを参照。

```
R> names(M$html$chart)
[1] "jsHeader"      "jsData"        "jsDrawChart"   "jsDisplayChart"
[5] "jsChart"       "jsFooter"      "divChart"
```

チャートの内容全体を表示するには、次のようにする。

```
R> print(M, tag='chart') ## または cat(M$html$chart)
```

⁵ (訳注) 出力の一部は省略されている。

```
<!-- MotionChart generated in R 2.15.0 by googleVis 0.2.15 package -->
<!-- Sun Mar 4 12:21:23 2012 -->
```

```
<!-- jsHeader -->
```

```
<script type="text/javascript" src="http://www.google.com/jsapi">
```

```
</script>
```

```
<script type="text/javascript">
```

```
// jsData
```

```
function gvisDataMotionChartID2dba6a651a62()
```

```
{
```

```
  var data = new google.visualization.DataTable();
```

```
  var datajson =
```

```
[
```

```
[
```

```
  "Apples",
```

```
    2008,
```

```
  "West",
```

```
    98,
```

```
    78,
```

```
    20,
```

```
  "2008-12-31"
```

```
],
```

```
[
```

```
  "Apples",
```

```
    2009,
```

```
  "West",
```

```
    111,
```

```
    79,
```

```
    32,
```

```
  "2009-12-31"
```

```
],
```

```
[
```

```
  "Apples",
```

```
    2008,
```

```
  "West",
```

```
    98,
```

```
    78,
```

```
    20,
```

```
  "2008-12-31"
```

```
],
```

```
[
```

```
  "Apples",
```

```
    2009,
```

```
  "West",
```

```
    111,
```

```
    79,
```

```
    32,
```

```
  "2009-12-31"
```

```
],
```

```
[
  "Apples",
    2010,
  "West",
    89,
    76,
    13,
  "2010-12-31"
],
[
  "Oranges",
    2008,
  "East",
    96,
    81,
    15,
  "2008-12-31"
],
[
  "Bananas",
    2008,
  "East",
    85,
    76,
    9,
  "2008-12-31"
],
[
  "Oranges",
    2009,
  "East",
    93,
    80,
    13,
  "2009-12-31"
],
[
  "Bananas",
    2009,
  "East",
    94,
    78,
    16,
  "2009-12-31"
],
[
  "Oranges",
    2010,
  "East",
    98,
    91,
    7,
```

```

"2010-12-31"
],
[
  "Bananas",
    2010,
"East",
    81,
    71,
    10,
"2010-12-31"
]
];
data.addColumn('string','Fruit');
data.addColumn('number','Year');
data.addColumn('string','Location');
data.addColumn('number','Sales');
data.addColumn('number','Expenses');
data.addColumn('number','Profit');
data.addColumn('string','Date');
data.addRow(datajson);
return(data);
}

// jsDrawChart
function drawChartMotionChartID2dba6a651a62() {
  var data = gvisDataMotionChartID2dba6a651a62();
  var options = {};
options["width"] = 600;
options["height"] = 500;

  var chart = new google.visualization.MotionChart(
    document.getElementById('MotionChartID2dba6a651a62')
  );
  chart.draw(data,options);

}

// jsDisplayChart
function displayChartMotionChartID2dba6a651a62()
{
  google.load("visualization", "1", { packages:["motionchart"] });
  google.setOnLoadCallback(drawChartMotionChartID2dba6a651a62);
}

// jsChart
displayChartMotionChartID2dba6a651a62()

<!-- jsFooter -->
<!-->
</script>

```

```
<!-- divChart -->

<div id="MotionChartID2dba6a651a62"
      style="width: 600px; height: 500px;">
</div>
```

同様に、チャートの特定の要素，例えば‘jsChart’にアクセスするには次のようにする。

```
R> cat(M$html$chart['jsChart']) # または print(M, 'jsChart')
```

```
// jsChart
displayChartMotionChartID2dba6a651a62()
```

基本的なチャートのキャプションと html フッタは，html リストの最終項にある（下記では出力を省略している）。

```
R> print(M, tag='caption')
```

```
<div><span>Data: Fruits &#8226; Chart ID: <a href="Chart_MotionCha
```

```
R> print(M, tag='footer')
```

```
<!-- htmlFooter -->
<span>
R Under development (unstable) (2012-02-03 r58258) &#8226; <a href
&#8226; <a href="http://code.google.com/apis/terms/index.html">Goo
</span></div>
</body>
</html>
```

2.4 gvis オブジェクトをローカルに表示

ページをローカルに表示するには，次を入力する。

```
R> plot(M) # ファイル名を返すが見えない
```

gvis オブジェクトの plot メソッドは，オブジェクトのタイプとチャート id の情報を利用して一時フォルダの中に HTML ファイルを作成する。そして，R の HTTP ヘルプウェブサーバー（通常 <http://127.0.0.1>）のもとにローカルに出力を表示する。

チャートのキャプションは，コピー&ペーストに対して，チャート id を通してチャートコードへのリンクを提供する。

R の tempdir() コマンドは，セッション単位での一時ディレクトリ（ここにファイルが書き込まれる）のパスを示す。例えば次のようにして，file 引数を与えた print コマンドによってローカル html ファイルにチャートを書き込むことができる。

```
R> print(M, file="mygoogleVisChart.html")
```

セキュリティの設定のため、ローカルファイルとしてロードされたとき、フラッシュのチャート⁶が機能せず、ウェブサーバ経由で表示することを要求されることがある。この問題は、フラッシュのセキュリティの設定を変更することにより解決できる。Tony Breyal は、stackoverflow.com で次の解決法をポストしている。

1. http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html へ行く。
2. ドロップボックスの‘編集’をクリックし、‘追加’を選択
3. ‘フォルダを参照’をクリック
4. html ファイルを保存したフォルダを選択
5. OK をクリック

これにより、*googleVis* の html ファイルを開き、うまく表示することができる。

2.5 gvisMerge によるチャートの結合

関数 *gvisMerge* は 2 つの *gvis* オブジェクトを引数として取り、これらを 1 つのページに統合する。HTML テーブルの中で、2 つのチャートを縦 (*horizontal*) または横 (*vertival*) に並べることができる。

gvisMerge の出力も *gvis* オブジェクトである。これにより、*gvisMerge* を繰り返して適用することができ、チャートの複雑なレイアウトを実現することを可能としている。次の例では、地図と表を上下に並べてから、右にモーションチャートを配置している (図 4)。

```
R> G <- gvisGeoChart(Exports, "Country", "Profit",
+                   options=list(width=200, height=100))
R> T <- gvisTable(Exports,
+                options=list(width=200, height=270))
R> M <- gvisMotionChart(Fruits, "Fruit", "Year",
+                       options=list(width=400, height=370))
R> GT <- gvisMerge(G,T, horizontal=FALSE)
R> GTM <- gvisMerge(GT, M, horizontal=TRUE,
+                  tableOptions="bgcolor=\"#CCCCCC\" cellspacing=10")
R> plot(GTM)
```

2.6 オプションの設定

googleVis オブジェクトのいろいろなオプション⁷を設定することは、最初は面倒かもしれない。オプションは Google Visualisation API のそれに倣っており、引数 *option* を通じて指定することができる。次の例では、折れ線グラフを作成して、様々なオプションを設定している。

```
R> df <- data.frame(country=c("US", "GB", "BR"),
+                  val1=c(1,3,4), val2=c(23,12,32))
R> Line <- gvisLineChart(df, xvar="country", yvar=c("val1","val2"),
+                        options=list(
+                          title="Hello World",
```

⁶ 現在、フラッシュを必要とするチャートは、モーションチャート、地図、注釈付きタイムラインである。

⁷ オプションはチャートによって異なるので、個々の *googleVis* 関数のそれらについては、ヘルプファイルを参照してほしい。

```

+         titleTextStyle="{color:'red',
+         fontName:'Courier',
+         fontSize:16}",
+         backgroundColor="#D3D3D3",
+         vAxis="{gridlines:{color:'red', count:3}}",
+         hAxis="{title:'Country', titleTextStyle:{color:'blue'}}",
+         titleTextStyle:{color:'blue'}}",
+         series="[{color:'green', targetAxisIndex: 0},
+         {color:'orange',targetAxisIndex:1}]",
+         vAxes="[{title:'val1'}, {title:'val2'}]",
+         legend="bottom",
+         curveType="function",
+         width=500,
+         height=300
+     ))
R> plot(Line)

```

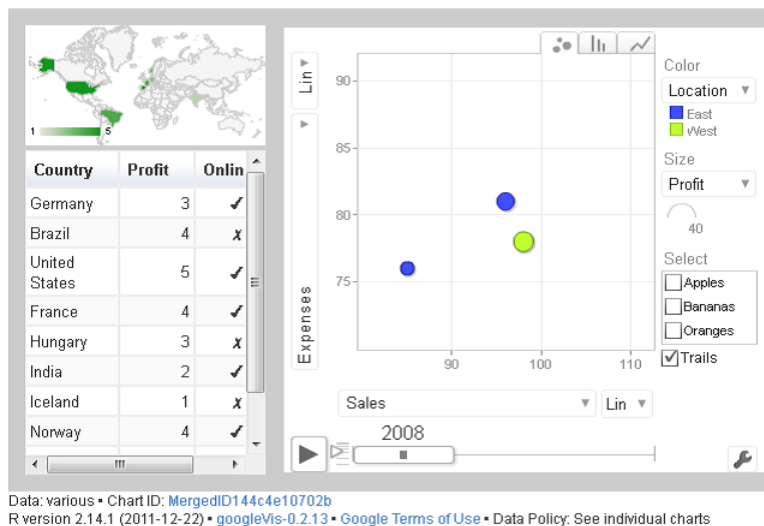


図 4: gvisMerge で結合された 3 つのチャート.

上記の例からわかるように、簡単なオプションは、名前=値（例えば、width=500）の形で設定できる。下位要素を持つような複雑なオプションは、中括弧の中にリストされ、例えば、2 つの軸を定義するためのアレイには角括弧 [] を用いる。

2.6.1 チャートエディタ

gvis.editor は、全てのチャートのための特別なオプションを設定できるのものである。これは編集ボタンをページに加えるので、ユーザがその場でチャートを編集・変更・カスタマイズすることができる。これに関しては、次の例と図 6 を参照。例における gvis.editor のリスト項目の中身は、ブラウザ

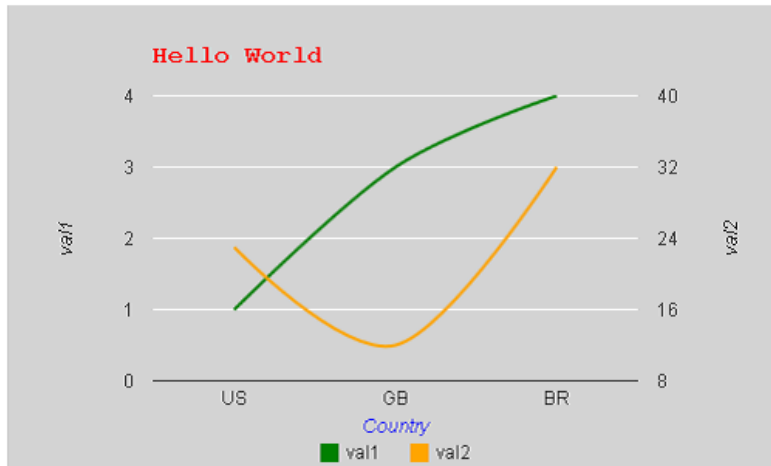


図 5: 様々なオプションを設定した折れ線グラフ.

ボタンのラベル ('Edit me!') を指定している。⁸

```
R> Editor <- gvisLineChart(df, options=list(gvis.editor='Edit me!'))
R> plot(Editor)
```

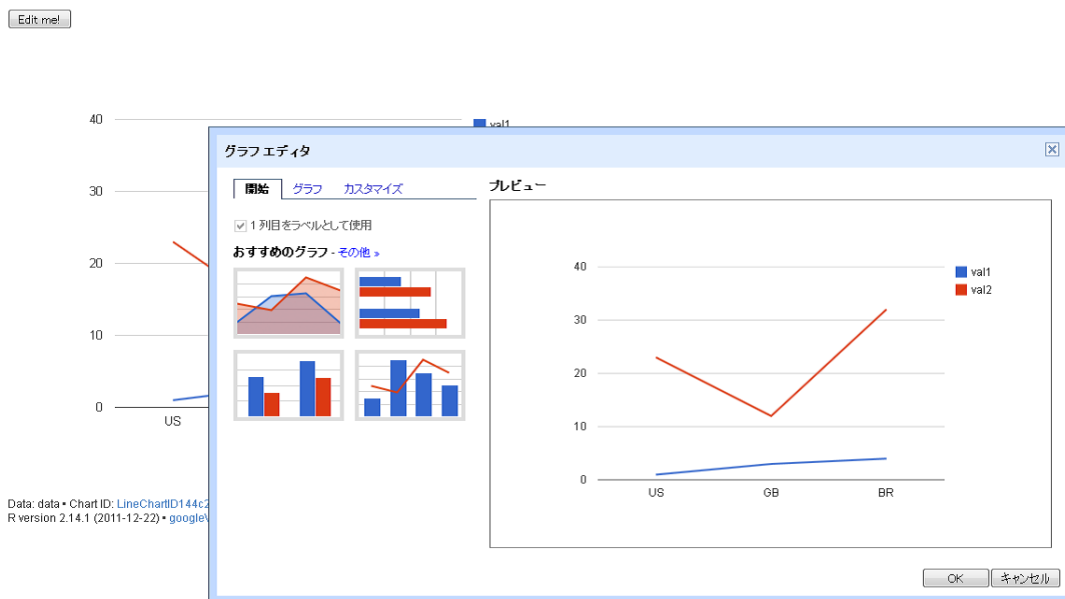


図 6: options = list (gvis.editor = 'Edit me!') としたときの *googleVis* チャートの例.

2.6.2 列名におけるアポストロフィの取り扱い

googleVis パッケージは、データフレームを JSON オブジェクトに変換する。結果として得られる JSON の表の列名は、単一引用符で括られる (4 ページのコード例の 12 – 16 行目参照)。よって、入力

⁸ Google Visualization API Reference (<http://code.google.com/apis/chart/interactive/docs/reference.html#google.visualization.charteditor>) も参照。

データフレームの列名にアポストロフィがある場合、2重のバックスラッシュをつける必要がある。簡単な例を次に示し、その出力を図7に示す。

```
R> df <- data.frame("Year"=c(2009,2010), "Lloyd\\'s"=c(86.1, 93.3),
+                  "Munich Re\\'s R/I"=c(95.3, 100.5),
+                  check.names=FALSE)
R> df

Year Lloyd\\'s Munich Re\\'s R/I
1 2009 86.1 95.3
2 2010 93.3 100.5
R> CR <- gvisColumnChart(df, options=list(vAxis='{baseline:0}',
+                                       title="Combined Ratio %",
+                                       legend="{position:'bottom'}"))
R> plot(CR)
```

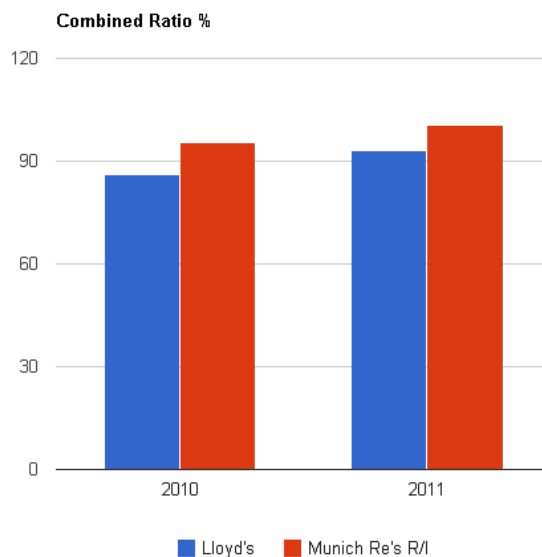


図7: 列名にアポストロフィが入っている場合のデータの可視化。

3 *googleVis* をウェブサイト に埋め込む

3.1 既にあるサイトに *gvis* オブジェクトを統合する

既にウェブページがあり、*googleVis* 関数の、例えば *gvisMotionChart* の出力を組み込みたいとする。この場合、*gvisMotionChart* からのチャートの出力だけを必要とする

```
R> print(M, 'chart') ## または cat(M$html$chart)
```

により出力し、存在する html ページに R コンソールからの出力をコピー&ペーストしてもよい。また、次のようにして直接ファイルに内容を書き込んで処理してもよい。

```
R> print(M, 'chart', file='myfilename')
```

3.2 *googleVis* 出力を WordPress で利用する

WordPress は、ウェブサイトやブログを作成するための人気があるウェブソフトウェアである。ここでは、*googleVis* の出力を WordPress で利用する方法を簡単に説明する。

デフォルトでは、WordPress は JavaScript のコード（よって、*googleVis* の出力）がページに挿入されるのを許さない。しかし、プラグインを追加することにより WordPress の機能を拡張することができる。

JavaScript コードを WordPress に埋め込む方法の 1 つとして、“custom fields shortcode” プラグイン⁹を利用することがある。プラグインによって、*googleVis* コード用のカスタムメイドのフィールドを作成し、それを自分の記事の中で参照することができる。

例えば、R でモーションチャートを作成したとしよう：

```
R> M <- gvisMotionChart(Fruits, "Fruit", "Year", options=list(width=400, height=370))
```

チャートのこのコードを、

```
R> print(M, 'chart')
```

により表示し、これをコピーして WordPress のカスタムフィールドの `value` のテキスト領域にペーストする（例えば、Fruits というインスタンス名で）。記事にモーションチャートを含むには、ポストに `[cf]Fruits[/cf]` を付け加える（例については、図 8 参照）。

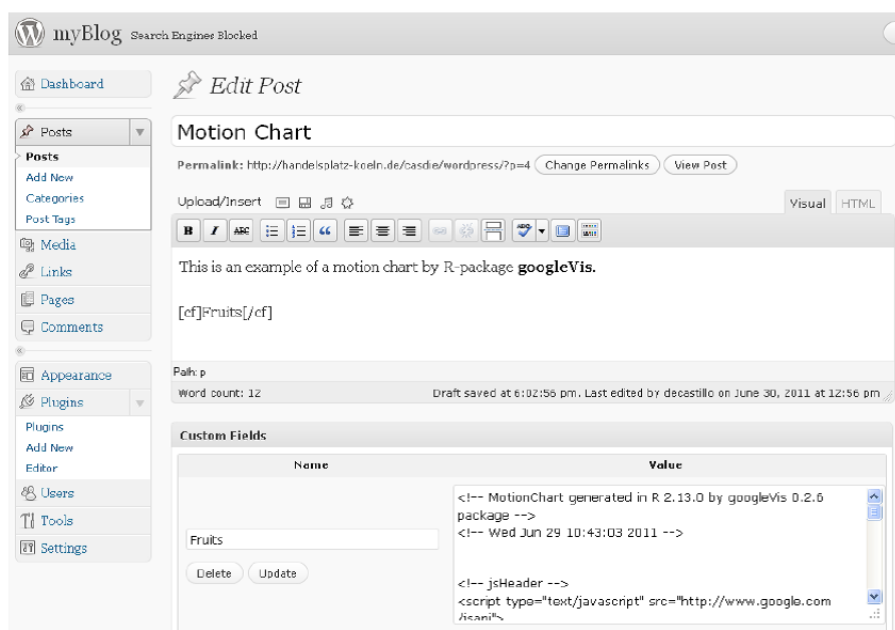


図 8: *googleVis* の出力を WordPress のブログエントリーに入れる。

⁹ <http://wordpress.org/extend/plugins/custom-fields-shortcode>

3.3 *googleVis* の出力を Google サイト, Blogger 等で利用する.

Google チャートを, 他の Google 製品 (例えば, Google サイト, Blogger, Google Code wiki) で使うことができる. しかし, ほとんどの場合, チャートは Google ガジェットとして埋め込む必要がある. Google ガジェットは XML で書かれ, HTML と Javascript 要素を持つことができる. Google ガジェットの技術を利用して書かれた 'Hello World' プログラムの Wikipedia からの例を示す.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
<ModulePrefs title="simple hello world example" />
<Content type="html">
<![CDATA[
Hello, world!
]]>
</Content>
</Module>
```

googleVis パッケージには関数 `createGoogleGadget` がある. これは `gvis` オブジェクトを引数として取り, XML ガジェットファイルにラップする. モーションチャートの例を次に示す.

```
R> M <- gvisMotionChart(Fruits, "Fruit", "Year")
R> G <- createGoogleGadget(M)
R> cat(G, file="myGadget.xml")
```

ガジェットを使うためには, 例えば Google Docs を使用して, ファイル `myGadget.xml` をオンラインにホストしなければならない. ガジェットに対する URL が `http://example.com/myGadget.xml` だとすると, メニューによりガジェットを Google Site に埋め込むことができる:

- メニューを用いて Google Site に:
"Insert" -> "More gadgets ..." -> "Add gadget URL",
- `wiki:gadget` タグを用いて Google Code wiki に:
`<wiki:gadget url="http://example.com/gadget.xml" />`,
- デザインタブ (ガジェットサイト参照) を用いて Blogger に:

しかし, blogger post に *googleVis* の出力を含めたい場合は, 状況は異なる. この場合, ガジェットは機能しない. post にチャートを直接コピー&ペーストするか (マイクロソフトのインターネットエクスプローラでは機能しない), 間接的なアプローチを使うかである. 後者は, *googleVis* のチャートを別のページ (例えばパブリックな Dropbox フォルダに) にホストし, `<iframe>` タグを用いて, 例えば,

```
<iframe width="100%" height="400px" frameborder="0"
src="http://example.com/mygoogleVisChart.html">
</iframe>
```

のように埋め込むことを意味する.

例に関しては, ブログのエントリ <http://lamages.blogspot.com/2011/09/including-googleVis-output-into-blogger.html> を参照.

3.4 *googleVis* をウェブサイト動的に埋め込む

本節では、*googleVis* 関数を動的にウェブサイトに埋め込む例を示す。R のパッケージ *R.rsp* [Ben12] と *brew* [Hor11a] を用いて R のスニペットを HTML コードに統合するには、2つの選択肢がある。*R.rsp* パッケージはそれ自身の内蔵ウェブサーバを同梱しているが、*brew* は RApache モジュール [Hor11b] がインストールされた Apache HTTP サーバ [Fou10a] を必要とする。現在、RApache モジュールは、UNIX/Linux と Mac OS X 上でしか動作しないことに注意。

3.4.1 *googleVis* を *R.rsp* で利用する

R.rsp パッケージにより、R コードを HTML コードに統合することができる。R コードは *R.rsp* ウェブサーバでフィルターにかけられ、ランタイムに実行される。rsp ページにモーションチャートを埋め込む例を次に示す。

```
<html>
<body>
<% library(googleVis)
M <- gvisMotionChart(Fruits, idvar="Fruit", timevar="Year") %>
<%= M$html$chart %>
</body>
</html>
```

<%...%> 内にある R コードは *R.rsp* HTTP サーバで読まれるときに実行されるが、R の出力は表示されない。: HTML コードに R の出力を埋め込むためには、等号を加えなければならない<%=...%>。これば、cat 文として機能する。

googleVis パッケージの中に例がある。これは、次の R のコマンドにより表示することができる。

```
R> library(R.rsp)
R> browseRsp()
R> # 後は、開かれたブラウザの googleVis のリンクをたどっていく
```

実際の rsp ファイルは *googleVis* パッケージのディレクトリにあり、そのファイルを見つけるには次のコマンドを利用する。

```
R> file.path(system.file("rsp", package = "googleVis"), "index.rsp")
```

詳細は、*R.rsp* パッケージの文書を参照のこと。

3.4.2 *RApache* と *brew* で *googleVis* を利用する

RApache は、R と Apache HTTP サーバを使用してウェブアプリケーションの開発をサポートする。*RApache* モジュールは、Apache ウェブサーバに R インタプリタを埋め込む。しかし、R と HTML コードを統合するのでパーサーが必要となり、そのために、R のパッケージ *brew* の出番となる。

ブラウザで開かれると、HTTP リポジトリの専用の *brew* フォルダにあるファイルは *brew* により実行される。R コードは *RApache* で実行され、出力はサイトに埋め込まれる。よって、アプローチは *R.rsp* と似ているが、2つの仕事に分割されるという違いがある。これは、R が別々のウインドウで動作する必要がないという利点がある。

RApache のインストール手順の詳細については、プロジェクトのサイトで取得可能である (<http://rapache.net/manual.html>)、Mac OS X 特有の注意に関しては、<http://-worldofcraft>。

blogspot.com/2010/08/installing-rapache-on-mac-os-x-snow.html を参照.

RApache のマニュアルに従うと, *RApache* をインストール後, Apache の設定が必要となる. おそらく, `apache2.conf` ファイルまたは `httpd.conf` ファイルに次の行を追加する必要があるだろう (/etc/httpd. Mac OS X に関しては /private/etc/apache2/httpd.conf で見つけることができるだろう):

```
LoadModule R_module \url{/usr/lib/apache2/modules/mod_R.so}
## On Mac OS X more likely to be:
## LoadModule R_module libexec/apache2/mod_R.so
ROutputErrors
RSourceOnStartup "/var/www/rapache/R/startup.R"
## On Mac OS X the www folder is often equivalent to:
## /Library/WebServer/Documents/
```

1 行目は, Apache ウェブサーバがスタートしたときに R モジュールをロードし, 2 行目は, エラー処理を行う. `startup.R` ファイルは初期設定 (例えばライブラリとグローバル変数) に適している:

```
## Ensure the packages are installed so that mod_R
## has access to them, e.g. not in your home folder
library{googleVis}
library{lattice}
library{Cairo}
MyGlobalVar <- 42
```

RApache が機能しているかどうかを調べるには, `http://localhost/RApacheInfo` を開いて自分のシステムの詳細を見る. 例が, *RApache* のサイト `http://biostat.mc.vanderbilt.edu/rapache/files/RApacheInfo.html` にある.

次に行うべき事は, *brew* パッケージを通常の方法でインストールすることである:

```
R> install.packages('brew')
```

この後, 特定のフォルダ内のファイルを *brew* でパースするよう Apache に教える必要がある. 再び, `apache2.conf` または `httpd.conf` を編集して, `RHandler` と *brew* の機能との接続を追加する:

```
<Directory /var/www/rapache/brew>
## On Mac OS more likely to be something like:
## <Directory /Library/WebServer/Documents/rapache/brew>
SetHandler r-script
RHandler brew::brew
</Directory>
```

これで終わりである. HTTP デーモンを再開し, *brew* ディレクトリにファイル (例えば次に示すものを含むファイル) を置くことができ, `http://localhost/rapache/brew/filename` を通してそれらにアクセスすることができる:

```
<html>
<body>
<h1>Fruits</h1>
<% library(googleVis)
M <- gvisMotionChart(Fruits, idvar="Fruit", timevar="Year") %>
<%= M$html$chart %>
</body>
```

</html>

brew の構文は *rsp* と非常に似ている。詳細については、*RApache* モジュールと *brew* パッケージの文書を参照。 *brew* ファイルの簡単な例が、*googleVis* パッケージに 2 つある。また、次の R のコマンドにより、フォルダのパスを示すことができる：

```
R> system.file("brew", package = "googleVis")
```

4 プレゼンテーションで *googleVis* を用いる

Google Visualisation API はウェブページ用に設計されているので、*googleVis* の出力を MS PowerPoint¹⁰ や Google Docs, OpenOffice の Impress, Apple Keynot のような伝統的なプレゼンテーションソフトに埋め込むことが難しかったり不可能であったりすることは不思議ではない。

最も簡単な方法は、ウェブページへのリンクによりスライドにスクリーンショットを含むことである。しかし、このアプローチは、発表者がトークの間に、アプリケーションを切り替える必要がある。これは楽しみでもあるが、そうではないこともしばしばある。

別の方法として、ウェブページ自体としてプレゼンテーションを構築することもできる。JavaScript ライブラリの *deck.js* [Tro11], Caleb Troughton による HTML プレゼンテーションのための jQuery フレームワークは、例えば YouTube ビデオや *googleVis* の出力といったものとスライド方式とを結合するエレガントな方法を提供する。 *mages's* ブログに *googleVis* の例がある：<http://lamages.blogspot.com/2011/11/interactive-presentations.html>

Getting started with *googleVis*

図 9: *deck.js* を利用した *googleVis* プレゼンテーション:
<http://dl.dropbox.com/u/7586336/blogger/deck.js/googleVis/index.html>

5 R を超えて

本項では、R の通常のコーディングを越えたいくぶん実験的な考えを示す。

5.1 Registering to catch events

Google visualisations はイベントを作り出し、それを受けるための¹¹、次の 2 つの JavaScript のメソッドを公開している：

- `google.visualization.events.trigger()` がイベントを fire する

¹⁰ MS-Windows の Microsoft PowerPoint に関しては、プラグイン *liveweb* が、ウェブページを PowerPoint スライドに挿入して、スライドショーの間、ページをリアルタイムに更新する機能を提供している。

¹¹ <http://code.google.com/apis/chart/interactive/docs/reference.html#addlistener>

- `google.visualization.events.addListener()` がイベントを listen する

Google ドキュメンテーションにある selection イベントを受け取るための registering の例を 1 つ示す.

```
var table = new google.visualization.Table(document.getElementById('table_div'));
table.draw(data, options);
google.visualization.events.addListener(table, 'select', selectHandler);
function selectHandler() {
  alert('A table row was selected');
}
```

‘addListener’ メソッドの ‘select’ イベントの特殊な場合のみを扱う. このイベントは大部分の可視化に利用でき, ユーザとの対話処理 (例えば, ユーザによる選択のクリック) に役に立つ.

‘addListener’ メソッドは, パラメータ `gvis.listener.jscode` (まだ文書化されていない) によるオプションによって `gvis` オブジェクトに埋め込むことができる JavaScript コードを期待する. 例をいくつか示す:

Wikipedia の項目を調べる:

```
R> jscode <- "window.open('http://en.wikipedia.org/wiki/'
+ + data.getValue(chart.getSelection()[0].row,0)); "
R> J1 <- gvisGeoMap(Exports, locationvar='Country', numvar='Profit',
+ options=list(dataMode="regions", gvis.listener.jscode=jscode))
R> plot(J1)
```

同様に, 他のチャート, 例えば組織図または折れ線グラフでこのコードを利用することができる:

```
R> plot(gvisOrgChart(Regions, options=list(gvis.listener.jscode=jscode)))
R> plot(gvisLineChart(Regions[,c(1,3)], options=list(gvis.listener.jscode=jscode)))
```

次のより進んだ例では, テーブルで選択された値が, メッセージボックスに表示される:

```
R> jscode <- "
+ var sel = chart.getSelection();
+ var row = sel[0].row;
+ var text = data.getValue(row,1);
+ alert(text);
+ "
R> J2 <- gvisTable(Population, options=list(gvis.listener.jscode=jscode))
R> plot(J2)
```

詳細については, `demo(EventListener)` と Google Visualisation API リファレンスを参照のこと.

6 連絡

6.1 コラボレーション

パッケージの開発は進行中であり, また, Google Visualisation API の他の多くの機能については触れていない. 新しいバージョンに関する情報を受けたいならば, 気軽に電子メールを送ってほしい. また, フィードバック, アイデア, 提案があったり, コラボレートしたりしたい場合もそうである. アドレスは `rvisualisation@gmail.com`.

6.2 引用

あなたの仕事や出版において R および *googleVis* を利用する場合、これらの引用情報を記載してほしい。その際の形式は、次で表示される情報を参照。

```
R> citation("googleVis")
R> citation()
```

6.3 トレーニングとコンサルティング

特定目的のトレーニングまたはコンサルティングを検討したいとき、私たちに連絡してほしい：
rvisualisation@gmail.com.

参考文献

- [Ben12] Henrik Bengtsson. R.rsp: R server pages. <http://CRAN.R-project.org/package=R.rsp>, 2012. R package version 0.7.1.
- [DKM10] MacKenzie Smith (MIT Libraries) David Karger (MIT CSAIL). Simile: Semantic Interoperability of Metadata and Information in unLike Environments. <http://simile.mit.edu/>, 2010.
- [Fou10a] Apache Foundation. Apache HTTP Server 2.2. <http://httpd.apache.org>, 2010.
- [Fou10b] Gapminder Foundation. Gapminder. <http://www.gapminder.org>, 2010.
- [Fou10c] Open Source Geospatial Foundation. Openlayers: Free maps for the web. <http://www.openlayers.org/>, 2010.
- [FR10] Ben Fry and Casey Reas. Processing an open source programming language and environment to create images, animations, and interactions. <http://processing.org/>, 2010.
- [GdC11] Markus Gesmann and Diego de Castillo. "Using the Google Visualisation API with R. *The R Journal*,3(2):40-44, December 2011.
- [GdC12] Markus Gesmann and Diego de Castillo. googleVis: Using the Google Visualisation API with R. <http://code.google.com/p/google-motion-charts-with-r/>, 2012. R package version 0.2.15.
- [Hor11a] Jeffrey Horner. brew: Templating framework for report generation. <http://CRAN.R-project.org/package=brew>, 2011. R package version 1.0-6.
- [Hor11b] Jeffrey Horner. RApache: Web application development with R and Apache. <http://www.rapache.net/>, 2011.
- [Inc10a] Google Inc. Google Motion Chart API. <http://code.google.com/apis/visualization/documentation/gallery/motionchart.html>, 2010.
- [Inc10b] Google Inc. Google Public Data Explorer. <http://www.google.com/publicdata/home>, 2010.
- [Inc10c] Google Inc. Google Visualisation Reference. <http://code.google.com/apis/visualization/documentation/reference.html>, 2010.
- [Inc11] Google Inc. Google Visualization API. <http://code.google.com/apis/visualization/documentation/gallery.html>, 2011.

- [Inc12] Google Inc. Google API Terms of Service. <http://code.google.com/apis/terms/index.html>, 2012.
- [JG10] George Neusse John Glazebrook, Guenther Harrasser. Open ash chart. <http://teethgrinder.co.uk/open-flash-chart/>, 2010.
- [JSO06] JSON.org. JSON. <http://www.json.org/>, 2006. RFC 4627 application/json.
- [Lab10] UC Berkeley Visualization Lab. flare: Data visualisation for the web. <http://flare.prefuse.org>, 2010.
- [Lan12] Duncan Temple Lang. RJSONIO: Serialize R objects to JSON, JavaScript Object Notation. <http://www.omegahat.org/RJSONIO/>, 2012. R package version 0.98-0.
- [Ros06] Hans Rosling. TED Talk: Hans Rosling shows the best stats you've ever seen. http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html, 2006.
- [RtICsg10] IBM Research and the IBM Cognos software group. Many eyes. http://services.alphaworks.ibm.com/manyeyes/page/Create_a_Visualization.html, 2010.
- [Saa10] Sebasti'an Pe'rez Saaibi. *R/RMETRICS Generator Tool for Google Motion Charts*. <https://www.rmetrics.org/>, 2010. Meielisalp, Lake Thune Switzerland, June 27 - July 1, 2010.
- [Tro11] Caleb Troughton. *deck.js: Modern HTML Presentations*, 2011. jQuery framework for creating HTML presentations.