

SuperSQL による構造化データの二次元可視化

大多和俊介[†] 五嶋 研人[†] 遠山 元道[†]

[†] 慶應義塾大学 理工学部 情報工学科 〒223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: [†]{otawa,goto}@db.ics.keio.ac.jp, ^{††}toyama@ics.keio.ac.jp

あらまし SuperSQL とは SQL の拡張言語であり、関係データベースの出力結果を構造化することで、SQL では表現することのできない多彩なレイアウト表現で様々な形式の構造化文書の生成が可能となる。また、データを分析、またはデータの分析結果を他人と共有する際に、グラフなどのデータの可視化手法がよく用いられる。本研究では、SuperSQL と R 言語のパッケージである ggplot2 を用いることで、構造化されたデータから二次元グラフを Web ページ上に生成する機能の実装を提案する。これにより、ユーザーが容易にデータベース内の様々なデータの分析、共有をすることが可能となる。

キーワード SQL, SuperSQL, ggplot2, 可視化

1 はじめに

近年、データの量や種類は増加しており、データを適切に処理、分析しそれを他人に共有するということがよく行われている。その際、データをグラフや図などの手法で可視化することが求められている。可視化をすることで単なる数値などの羅列では分からなかったことが分かるようになり、データ利用者の意思決定を助けることができる。可視化には大きく分けて二次元可視化と三次元可視化の二つが存在するが、今回の研究では二次元可視化のみを対象とする。

SuperSQL とは、関係データベースの出力結果を構造化することによって多彩なレイアウト表現を可能とする SQL の拡張言語である。[1][2] 通常の SQL ではフラットな表しか作成することができないが、SuperSQL を用いることで様々な構造の表の作成が可能となる。また、HTML, PHP, PDF などの様々な出力形式を指定することができ、近年では Unity と組み合わせた研究も行われている。これらは通常と比べてはるかに少ないコード量で記述することができる。本研究では、この SuperSQL と R 言語のパッケージである ggplot2 を用いることで構造化したデータの可視化を実現した。

以下に本論文の構成を記述する。第 2 章では関連研究・技術について述べ、第 3 章では SuperSQL についての説明、第 4 章では SuperSQL を用いた二次元グラフの生成についての説明、第 5 章では評価、第 6 章ではまとめを記述する。

2 関連研究・技術

2.1 関連研究

2.1.1 情報可視化の手順

情報可視化は単にデータを目に見える形にし他者に共有するためだけに行うのではなく、データから新しい知見を得られるように処理するという側面もある。Ben Shneiderman はデータから情報を得るために必要な手順には、概観表示、ズーム、フィ

ルタリング、詳細表示の 4 つの段階がある述べている。[3] まずデータの全体像を俯瞰し、必要だと思われる部分のデータに注目し、必要ではない情報を排除し、詳細なデータを確認することにより、新しい知見を得ることができる。

2.1.2 二次元可視化と三次元可視化

可視化の手法には大きく分けて二次元可視化と三次元可視化の二つが存在する。伊藤貴之は三次元可視化にはいくつかの危険性があると述べている。[4]

三次元空間にデータをマッピングすることで立体感が生まれてしまう。この立体感によりデータを誤読してしまう可能性が生じる。また、空間内の手前のオブジェクトが奥のオブジェクトを隠してしまうことによってデータの見落としが発生してしまうリスクがある。他にも、遠近感によるオブジェクトの大きさや距離の誤読や、三次元空間に対するインタラクティブな操作の難解さ、可視化結果を他人と共有するために静止画にしまうと空間に対する操作ができず上記の問題を解決できないなど、三次元の可視化には多くの問題点がある。

それに対し二次元可視化は、平面上に可視化をするため上記の問題が起きず、よく用いられている。三次元可視化をする特別な理由や環境がない限りは二次元可視化の方が無難である。しかし、三次元可視化の問題点は現状の技術における問題であり、技術が進化することにより、三次元可視化がより普及することも考えられる。

2.2 関連技術

2.2.1 ggplot2

ggplot2 とは R 言語におけるパッケージの一つであり、綺麗なグラフを簡単に描くことが可能である。グラフィックス文法 (grammar of graphics) [5][6] と呼ばれる構成記述体系に基づいて実装されており、一貫した文法で記述しグラフを生成することができる。本研究ではこの ggplot2 を用いて二次元グラフを生成している。

ggplot2 におけるグラフ生成のための一般的な記法は次のようになる。

ggplot2 における記法

```
ggplot(data = <DATA>) +
  <GEOM 関数 >(
    mapping = aes(<MAPPINGS>),
    stat = <STAT>,
    position = <POSITION>
  ) +
  <COORDINATE 関数 > +
  <FACET 関数 > +
  <SCALE 関数 > +
  <THEME 関数 >
```

1 行目の ggplot 関数によってどのデータセットを用いるかを指定する。3 行目の mapping = aes(<MAPPINGS>) は、GEOM 関数内で各軸やオブジェクトの要素にどの変数を割り当てるかを指定するものである。

例: geom_point(mapping = aes(x = day, y = degree, color = place))

また、これは ggplot 関数内に書くこともできる。

4 行目の stat = <STAT> は、どのような統計変換をするかを指定するものであり、5 行目の position = <POSITION> は、グラフの位置調整をするためのものである。

グラフの生成において、1 から 3 行目は必須であるがそれ以外は必須ではない。また、それぞれの関数は複数回使用することができる。

2.2.2 Plotly

Plotly とはインタラクティブなグラフを生成するためのツールである。[7] R 言語や Python などの様々な言語に向けたライブラリが提供されている。本研究では、ggplot2 と組み合わせることによってインタラクティブなグラフを生成している。

2.2.3 Shiny

Shiny とは R 言語のパッケージであり、インタラクティブな Web ページを生成することができる。[8] R 言語のみで Web ページを生成することができ、比較的簡単にデータの分析結果を他人に共有することができる。また、CSS や JavaScript と組み合わせることでより拡張することができる。

3 SuperSQL とは

SuperSQL は関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語であり、慶應義塾大学遠山研究室で開発されている。そのクエリは SQL の SELECT 句を GENERATE <media> <TFE> の構文を持つ GENERATE 句で置き換えたものである。ここで <media> は出力媒体を示し、HTML, PDF, Mobile HTML5 などの指定ができる。また <TFE> はターゲットリストの拡張である Target Form Expression を表し、結合子、反復子などのレイアウト指定演算子を持つ一種の式である。

3.1 アーキテクチャ

図 1 に SuperSQL のアーキテクチャを示す。SuperSQL は大

きく分けて構文解析をする Parser、構造情報に基づいてデータを整形する DataConstructor、データからファイルを生成する CodeGenerator から成る。まず Parser が SuperSQL クエリを受け取り、通常の SQL クエリと構造情報に分け、SQL クエリを DBMS に渡すことで検索結果を受け取る。次に DataConstructor で受け取ったフラットな検索結果を構造情報に基づいて構造化する。最後に CodeGenerator が構造化データを受け取り、クエリで指定したメディアのファイルを生成する。本論文では、DataConstructor によって生成された構造化データを R 言語に渡すことによって二次元グラフを生成している。

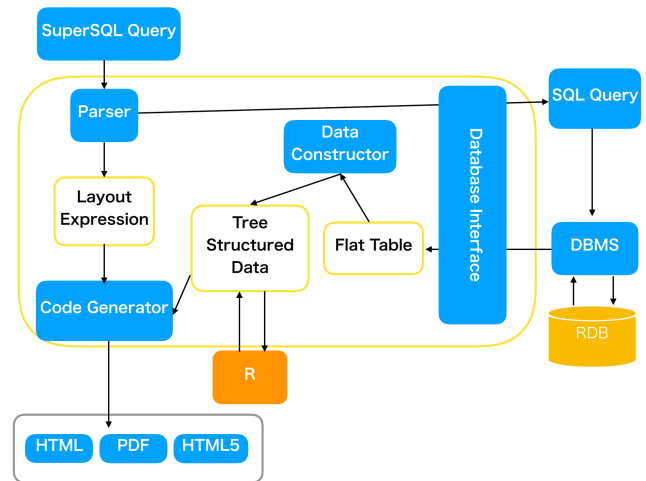


図 1 SuperSQL のアーキテクチャ

3.2 結合子

結合子はデータベースから得られたデータをどの方向 (次元) に結合するかを指定する演算子であり、以下の 3 種類がある。括弧内はクエリ中の演算子を示している。

- 水平結合子 (,)

データを横方向に結合して出力する。

例: name, age

name	age
------	-----

- 垂直結合子 (!)

データを縦方向に結合して出力する。

例: name! age

name
age

- 深度結合子 (%)

データを 3 次元方向へ結合して出力する。出力が HTML ならばリンクとなる。

例: name% age

name	→	age
------	---	-----

3.3 反 復 子

反復子は指定する方向にデータベースの値があるだけ繰り返して表示する。以下、その種類について述べる。

- 水平反復子 ([],)

データインスタンスがある限り、その属性のデータを横方向に繰り返し表示する。

例：[name],

name1	name2	...	nameN
-------	-------	-----	-------

- 垂直反復子 ([!])

データインスタンスがある限り、その属性のデータを縦方向に繰り返し表示する。

例：[name]!

name1
name2
...
nameN

3.4 複合反復子

反復子を組み合わせて使用する。ある方向 (次元) での反復の上限を指定し、これを超える場合に次に指定した方向に伸長する。

例：[name],3!

name 属性の値を横に 3 つごとに結合し、それを縦に伸長する。

name1	name2	name3
name4	name5	name6
name7	name8	name9
...	...	...

3.5 装 飾 子

SuperSQL では関係データベースより抽出された情報に、文字サイズ、文字スタイル、横幅、文字色、背景、高さ、位置などの情報を付加できる。これらは装飾演算子 (@) によって指定する。

< 属性名 >@{< 装飾指定 >}

装飾指定は“装飾子の名称 = その内容”として指定する。複数指定するときは各々を“,”で区切る。

[name@{width=100, color=red}]!

3.6 関 数

3.6.1 image 関数

image 関数を用いると画像を表示することが可能となる。引数には属性名、画像ファイルの存在するディレクトリにパスを指定する。

image(pict, “./pic”)

3.6.2 link 関数

link 関数は、FOREACH 句と同時に用いる。これらを用いることで深度結合子と同様にリンクを生成することができる。

link(name, “./index.ssql”, age)

4 二次元グラフの生成

この章では SuperSQL による二次元グラフ生成の概要や仕様、用例を説明する。

4.1 二次元グラフ生成の概要

SuperSQL は可視化するデータを関係データベースから取得し構造化する。その構造化されたデータを R 言語に送り、SuperSQL クエリでの指定に基づいて ggplot2 と Plotly を用いて二次元グラフを HTML として生成する。そして SuperSQL は生成されたグラフを受け取り、SuperSQL 自身が生成した HTML にグラフの HTML を埋め込むことで、グラフを含んだ Web ページを生成する。この一連の流れを図 2 に示す。

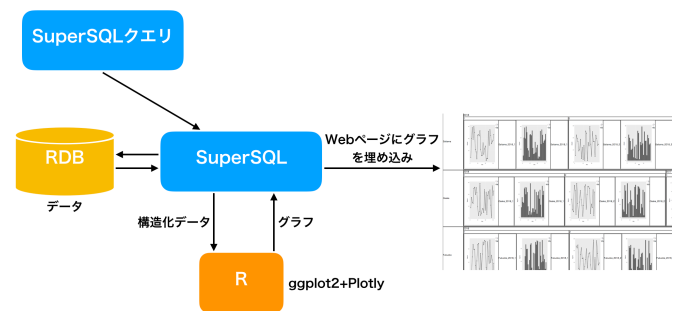


図 2 二次元グラフ生成の概要

4.2 二次元グラフ生成関数

二次元グラフの生成機能を SuperSQL 内の関数として実装した。グラフ生成の関数の一般系を以下に示す。

—— グラフ生成関数 ——

```
ggplot[x 軸, y 軸]@{<AESTHETIC> = 属性名,  
    <関数 1> = “引数 1, 引数 2, ...”,  
    <関数 2> = “引数 1, 引数 2, ...”,  
    ...  
}
```

ggplot[x 軸, y 軸]によってどの属性をグラフの x 軸, y 軸に割り当てるかを指定する。装飾子によって更にグラフの詳細を指定する。エステティック属性や ggplot における関数については後で説明する。エステティック属性や関数名、関数の引数は省略可能であり、記述する順序はグラフ生成に影響しない。

また、異なる二つのグラフを一つの座標系に重ねる時には次のように書く。

```
ggplot[x1, y1]@{...} + ggplot[x2, y2]@{...}
```

x1, y1, x2, y2 はそれぞれのグラフにおける x 軸, y 軸へ割り当てる属性である。これにより異なる属性から生成されたグラフを重ねることができる。

4.3 エステティック属性

エステティック属性とは、グラフのオブジェクトの視覚特性のことであり、色や形、大きさが含まれる。エステティック属性にデータの属性を割り当てることで、その属性の値によってオブジェクトの形を変えることができる。これにより、その属性でグルーピングし、それぞれのグループでグラフを生成することが可能となる。例えば、color = year とすることで year 属性の値ごとにグラフが色分けされて生成される。

まずエステティック属性を指定しない例を示す。

—— aesthetic 属性を指定しない場合 ——

```
ggplot[s.math, s.science]@{geom_point}
```

この例は、ある学生 30 人の数学のテストと理科のテストの点数の分布を散布図として表している。生成結果は図 3 のようになる。

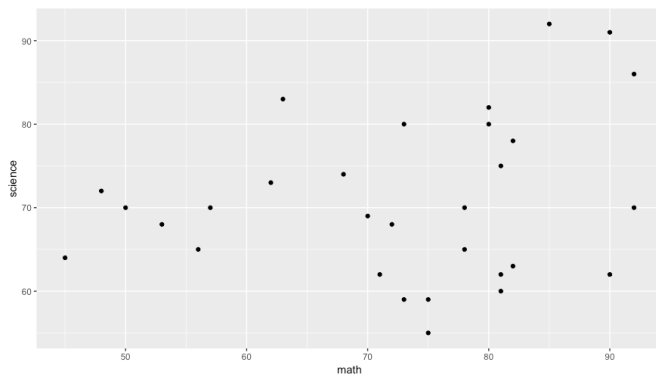


図 3 エステティック属性を指定しない場合

次にエステティック属性を指定した例を示す。

—— エステティック属性を指定した場合 ——

```
ggplot[s.math, s.science]@{color = s.class, geom_point}
```

この例は先の例と同様に、ある学生 30 人の数学のテストと理科のテストの点数の分布を散布図として表している。color というエステティック属性に class という属性を割り当てることで、オブジェクトの色に class 属性がマッピングされる。これにより、学生が所属しているクラスごとに色分けされたグラフが生成される。この例の生成結果は図 4 のようになる。

4.4 SuperSQL における ggplot2 の関数

4.2 節で示したように、装飾子では、グラフを生成するための様々な関数を用いることができ、それぞれの関数は引数をとることができる。

次に例を示す。

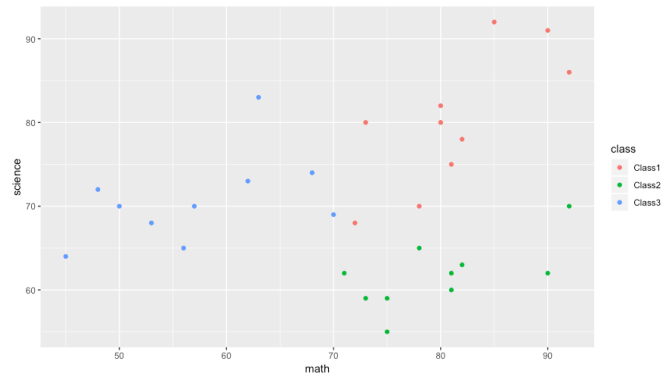


図 4 aesthetic 属性を指定した場合

—— SuperSQL における ggplot2 の関数 ——

```
ggplot[s.math, s.science]@{geom_point = "color = 'blue', size = 3"}
```

この例では装飾子として、geom_point という点をプロットする関数を指定して散布図を生成している。また、関数の引数として color や size を指定して、点の色や大きさを変更している。このように関数の引数はダブルクォーテーションで囲んで記述する。

また、関数の引数を指定しない場合は省略して次のように記述することができる。

—— 引数を省略した場合 ——

```
ggplot[s.math, s.science]@{geom_point}
```

4.5 グラフのサイズ指定

装飾子で width や height を指定することで、グラフ自体のサイズを指定することができる。

次に例を示す。

—— グラフのサイズ指定 ——

```
ggplot[s.math, s.science]@{geom_point, width = 500}
```

4.6 変換規則

本システムでは、SuperSQL クエリに記述された ggplot 関数を、R 言語における ggplot2 の記法に変換してから R 言語に実行させることによりグラフを生成している。ここではその変換規則について説明する。

—— 変換前のコード ——

```
ggplot[x 軸, y 軸]@{<AESTHETIC> = 属性名,  
  <関数 1> = "引数 1, 引数 2,  
  ...",  
  <関数 2> = "引数 1, 引数 2,  
  ...",  
  ...  
}
```

変換後のコード

```
ggplot(
  data = DATA,
  mapping = aes(x = x 軸, y = y 軸,
    <AESTHETIC> = 属性名)
) +
<関数 1>(引数 1, 引数 2, ...) +
<関数 2>(引数 1, 引数 2, ...) +
...
```

変換前のコードの 1 行目の `ggplot[x 軸, y 軸]` と装飾子の `<AESTHETIC> = 属性名` は、変換後のコードに示した `ggplot()` 関数に変換される。このとき `DATA` とは可視化するデータセットの名前であり自動的につけられる。3 行目の `mapping` によって、`x 軸`, `y 軸`, エステティックにどの属性をマッピングするかを指定している。

また、変換前に装飾子として書かれたそれぞれの関数は上記のように変換され、`ggplot()` 関数の後ろに+で連結される。

下にコードの変換の例を示す。

```
ggplot[s.math, s.science]@{color = s.class, geom_point =
  "size = 3", geom_line}
↓
ggplot(data = DATA, mapping = aes(x = math, y = science, color = class)) +
geom_point(size = 3) + geom_line()
```

4.7 用 例

二次元グラフの生成例を示す。例で使うデータベースは以下の通りである。

- `temperature(year, month, day, degree, prefecture)`
- `prefecture(id, name, region)`
- `region(id, name)`

`temperature` テーブルはある都道府県における年月日ごとの気温、`prefecture` テーブルは県の情報、`region` テーブルは地方の情報を持っている。

例 1

GENERATE HTML

```
[r.name!
  [p.name,
    [(desc)t.year,
      [(asc)t.month,
        ggplot[t.day,t.degree]@{geom_bar =
          "fill = 'blue' "}
      ],4!
    ],4!
  ]!
]!
FROM temperature t, prefecture p, region r
WHERE t.prefecture = p.id and p.region = r.id
```

上のクエリは、地域、都道府県、年、月ごとにグルーピングし、それぞれに対し `x 軸` に日付を、`y 軸` に気温を割り当てている。装飾子では青く塗られた棒グラフを生成するように指定している。5 から 7 行目で月名とグラフを横に結合したものを横に 4 回反復し、それを更に縦に反復して表示している。図 5 に生成結果を示す。

例 2

GENERATE HTML

```
[(desc)t.year!
  [(asc)t.month!
    ggplot[t.day, t.degree]@{color = p.name,
      geom_point = "size = 3", geom_line}
  ],4!
],1%
FROM temperature t, prefecture p
WHERE t.prefecture = p.id
```

上のクエリは、年、月ごとにグルーピングし、それぞれに対し `x 軸` に日付を、`y 軸` に気温を割り当てている。装飾子では、エステティック属性 `color` に `p.name` を指定することで県の名前ごとにグラフを生成している。また、サイズが 3 の点をプロットし、それらを線で結ぶ指定もしている。6 行目で複合反復子,4! を指定することで、グラフを横に 4 つ並べたものを縦に反復している。また、7 行目で複合反復子,1% を指定することで、年ごとに新しいページを生成するようにしている。図 6 に生成結果を示す。下のページ番号を選ぶことにより、異なる年のグラフを表示させることができる。

5 評 価

提案システムと `ggplot2` 単体、`Shiny` を表現力の観点から比較し評価する。

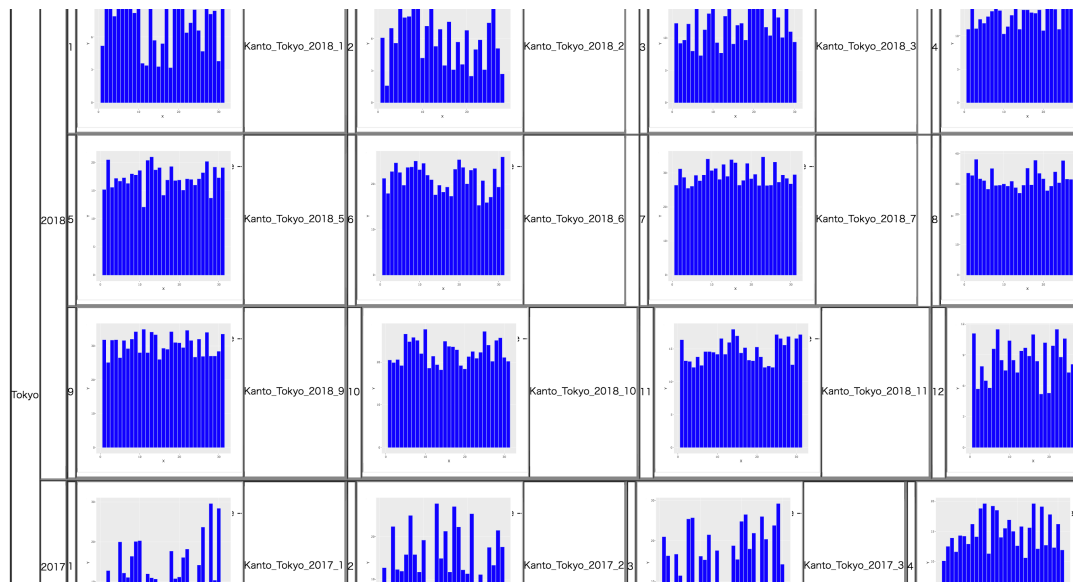


図 5 例 1 の生成結果

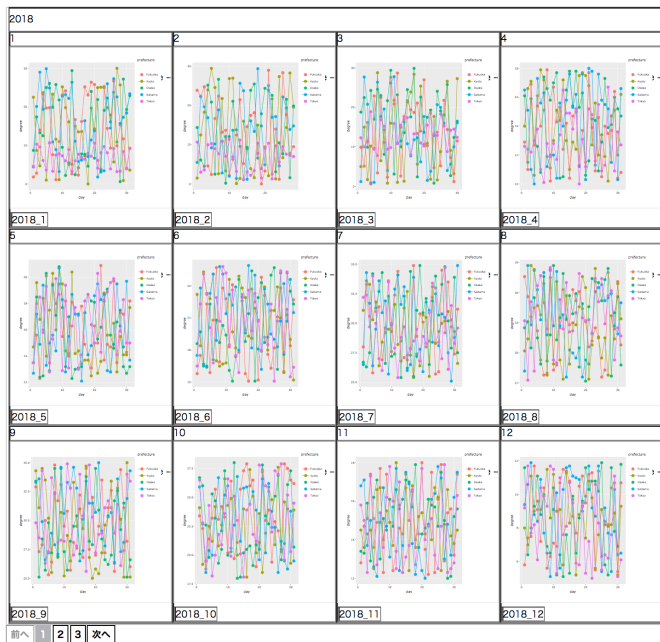


図 6 例 2 の生成結果

5.1 他システムとの比較

提案システムと他のシステムとの比較を表 1 に示す。提案システムは、構造化データの可視化、Web ページの生成の点で他システムより優れているが、インタラクティブな可視化の点で劣っている。以下で詳細を説明する。

表 1 各システムの比較

機能	提案手法	ggplot2	Shiny
構造的な可視化	○	△	×
Web ページの生成	○	×	○
グラフの画像ファイル出力	△	○	△
データベースとの接続	○	○	○
インタラクティブな可視化	△	×	○

5.2 ggplot2 との比較

まず、提案手法と ggplot2 との比較をする。表 1 に示した通り、提案手法も ggplot2 も構造化データの可視化を行うことができる。

ggplot2 にはファセットというプロットをある属性によって分割し、分割したグラフを並べて表示するというものがある。これにより、属性によってグルーピングされた複数のグラフを並べて表示することができる。例を図 7 に示す。

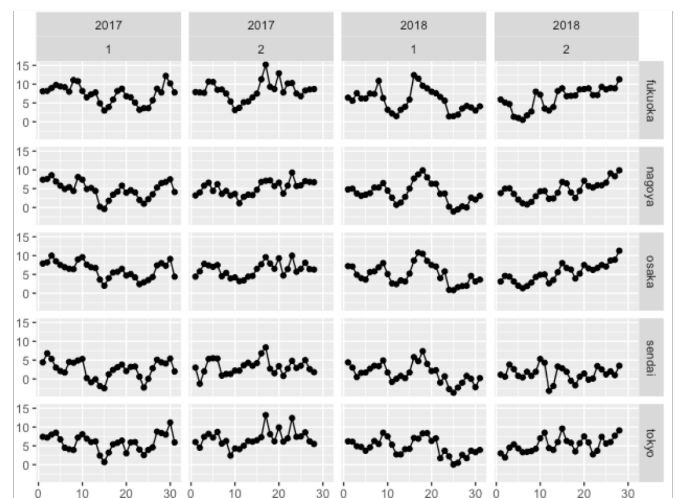


図 7 ファセット

このように ggplot2 はデータを構造化して可視化することができるが、一つのグループの中に複数のグラフを並べて表示することができない。それに対し、提案手法は一つのグループの中に複数のグラフを並べて表示できるなど、より自由なレイアウトでグラフを表示することが可能である。この点で提案手法は ggplot2 より優れている。

また、ggplot2 は画像を生成することしかできず、Web ページを生成することができない。そのため、Web 上でデータを共

有したい場合には自分で HTML を記述するなどをしなければならない。提案手法は自動的に Web ページを生成することができる。

ggplot2 が生成するグラフは静止画であるので、生成されたグラフに対して何かアクションを起こしてインタラクティブな操作をすることができない。それに対し提案手法は、ズームやパン、グラフの表示範囲の変更など簡単な操作なら行うことができるのでその点では優れている。

5.3 Shiny との比較

Shiny で生成される Web ページは例えば図 8 のようになる。Web ページ上で x 軸や y 軸に割り当てる属性を変更することができたり、他の変数を変更することができる。他にも Shiny によって生成した Web ページのグラフには様々な操作をすることが可能であり、インタラクティブである。この点で Shiny は提案手法より優れている。

また、Shiny はデータを構造化、グルーピングしそれぞれのグループでグラフを生成し並べることはできないので、その点では提案手法の方が優れている。

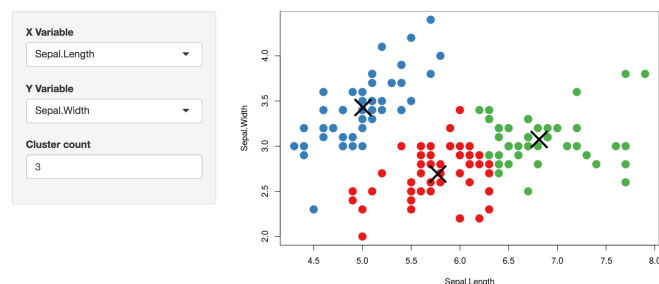


図 8 Shiny で生成した Web ページ

6 ま と め

6.1 結 論

本研究では SuperSQL を用いてデータベース内のデータを構造化し二次元可視化する機能を実装した。これにより、SuperSQL 自体の表現力が上がった。また、自由な構造でグラフを生成することができるので、データの分析や共有の際に有用であると考えられる。

6.2 今後の課題

現在のシステムでは、二次元グラフを HTML 上にしか生成することができない。他の様々なメディアに対してもグラフを出力できるようになればより有用性が増すと考えられる。また、R 言語の ggplot2 以外のパッケージを使用することができず、データに対する処理が十分に行えない場合があるのでそこを改善する必要がある。

他にも、SuperSQL を用いた研究は数多く行われており、これらの研究と本研究を組み合わせることでより有用なシステムになると考えられる。

- [1] SuperSQL: <http://ssql.db.ics.keio.ac.jp>
- [2] M. Toyama: “SuperSQL: An Extended SQL for Database Publishing and Presentation”, Proceedings of ACM SIGMOD’98 International Conference on Management of Data, pp. 584-586, 1998
- [3] B. Shneiderman. “The eyes have it: a task by data type taxonomy for information visualizations”, IEEE Symposium on Visual Languages, pp. 336-343, 1996
- [4] 伊藤貴之. “意思決定を助ける 情報可視化技術 —ビッグデータ・機械学習・VR/AR への応用—”, コロナ社, 2018
- [5] Wilkinson, L. “The Grammar of Graphics(2nd ed.)”, Statistics and Computing, New York: Springer, 2005
- [6] H.Wickham. “A layered grammar of graphics”, Journal of Computational and Graphical Statistics, 19(1):pp. 3-28, 2010
- [7] Plotly: <https://plot.ly/>
- [8] Shiny: <https://shiny.rstudio.com/>
- [9] ggplot2: <https://ggplot2.tidyverse.org/>
- [10] JRI: <https://www.rforge.net/JRI/>