ドリトル 統計機能 (V3.3)

ドリトルは教育用に設計されたオブジェクト指向型のプログラミング言語です。本文書ではドリトルの統計機能について解説しています。

1.ドリトル

1.1 インストールとプログラムの実行

本文書の統計機能はV3.3以降のドリトルに含まれています。次のサイトからドリトルをダウンロードしてください。（オンライン版は未対応です）

- プログラミング言語「ドリトル」

ドリトルを起動後、編集画面にプログラム入力し、「実行！」ボタンでプログラムを実行してください。

1.2 基本的な使い方

ここでは、複数の文字列を表示するオブジェクトであるリストオブジェクトを例に、ドリトルによるプログラミングの流れを紹介します。ドリトルの詳しい機能はWebで公開されているマニュアル（インストール版V3をご覧ください。

1.2.1 オブジェクトを作る

ドリトルでは、オブジェクトを作り、それに命令を送る形でプログラムを動作させます。ドリトルには、数値オブジェクトや文字列オブジェクト、画面上でグラフィックスを用いて絵を書いたりするタートルオブジェクトなど、さまざまな種類のオブジェクトが標準で用意されています。

ここでは、複数の文字列を表示するオブジェクトであるリストオブジェクトを例に、ドリトルによるプログラミングの流れを紹介します。まず、オブジェクトを作るには、作りたいオブジェクトの名前に作るという命令を送ります。命令を送るには「！」の記号を使い、命令を送るオブジェクトを指定します。文の最後には「。」が必要です。

オブジェクトを作る
オブジェクト！作る。

リストオブジェクトを作成する場合は以下のようになります。実行すると画面にリストオブジェクトが表示されます。

```
リスト！作る。
```

1.2.2 オブジェクトに名前をつける

ドリトルでは、オブジェクトを指定して命令を送ります。先ほど作ったプログラムでは、生成したオブジェクトに名前を付けていないため、画面に表示されたオブジェクトに命令を送ることはできません。オブジェクトに名前を付けるには、『名前 = オブジェクト。』のように、「 = 」の左辺に名前を書き、右辺にオブジェクトを指定します。

オブジェクトに名前をつける
名前＝オブジェクト。

先ほどのリストオブジェクトに名前をつけたい場合は、以下のように記述します。

```
表示エリア＝リスト！作る。
```

実行すると、画面に変化はありませんが、作成したリストには「表示エリア」と言う名前がついています。この名前は変数とも呼れます。複数のオブジェクトに同じ名前をつけた場合には、その名前は最後のオブジェクトの名前になります。

1.2.3 オブジェクトに命令を送る

ドリトルでは、オブジェクトに命令を送ることでプログラムを実行します。命令を送るオブジェクトは「！」で指定し、その右側に送る命令を書きます。たとえば、以下のプログラムは、「表示エリア」に対して『120　書く』という命令を送っています。ここで［書く］は命令、「120」は命令とともに送られる値であり、パラメータ（引数）と呼ばれます。パラメータと命令の間には、空白を入れる必要があります。また、引数は文字列の場合はダブルクォーテーション(“)で囲む必要があります。実行すると、リストに120が書き込まれます。

```
表示エリア＝リスト！作る。
表示エリア！120　書く。
```

1.2.4 オブジェクトに続けて命令を送る

ドリトルのプログラムは上から順に１行ずつ実行されます。以下に、「表示エリア」に複数の命令を送る例を示します。

```
表示エリア＝リスト！作る。
表示エリア！120　書く。
表示エリア！150　書く。
```

上記のプログラムを実行すると、まず1行目が実行され、「表示エリア」という名前の付いたリストが作られます。次に、2行目が実行され、表示エリアと名付けられたリストオブジェクトに120という値が書き込まれます。さらに、3行目が実行され、リストオブジェクトの2行目に150という値が書き込まれます。

この例では、3行のプログラムを書きました。ドリトルには命令の実行結果に続けて命令をするカスケードという機能があり、以下のように書くことも可能です。

```
表示エリア＝リスト！作る。
```

```
表示エリア！120　書く　150　書く。
```

上の２つのプログラムは、どちらのプログラムを実行しても同じ結果が得られます。下の方のプログラムでは、「表示エリア」と名前の付いたリストオブジェクトに対して『120　書く』という命令を実行し、120を書き込んでいます。さらに、120を書き込んだ結果のオブジェクトに対して『150　書く』という命令を実行しています。

2.ドリトルの統計機能

本章では、ドリトルの統計機能について説明します。

- データ入力　　（データファイルの読み込み、配列でのデータ作成） … 2.1節、 2.2節
- データ処理　　（扱うデータの範囲を絞る、並び替え等） … 2.3節
- データ分析　　（相関係数や度数分布等） … 2.4節
- データの可視化（グラフや表の描画・表示） … 2.5節

2.1 テーブルオブジェクトの作成

統計機能ではテーブルオブジェクトにより、データを表のように扱うことができます。テーブルオブジェクトの持つ表形式のデータをテーブルと呼びます。テーブルオブジェクトは、外部データを読み込む方法と配列を使って作成する方法の2種類の方法で作成できます。

2.1.1 外部データの読み込みによる作成

次のようなテキストファイルからデータを読み込むことができます。Excelなどの表計算ソフトでは、保存時に「テキスト（タブ区切り）」*.txtを選びます。保存されたファイルの拡張子は「.txt」のままでも扱えますが、.tsvに変更しておくとタブ区切りであることが明確になります。

- 1行目は各列のフィールド名を書きます。フィールド名には漢字/かな/英数字などを使用できます。
- フィールド名には記号を含む名前は使えません。数字で始まるフィールド名は、先頭に「F」のような英字が付けられた形で読み込まれます。
- 1列目は主キーとなる重複のないデータを置きます。
- フィールド名やデータの間はタブまたはカンマで区切ります。ファイルの1行目がタブで区切られていた場合はタブで、そうでない場合はカンマで区切る形で読み込まれます。
- 標準の文字コードは環境に依存します(以後、環境標準文字コードと称します)。
    - Windows:Shift-JIS
    - Mac:UTF-8
    - Linux:UTF-8

タブとカンマ以外の区切り文字を使う場合は、テーブルオブジェクトの「区切り文字」メソッドで指定してください。環境標準文字コード以外の文字コードのファイルを読み込む場合は、「文字コード」メソッドでファイルで利用されている文字コード（UTF-8など）を指定してください。

ここではドリトルに標準で用意されている「data/school.tsv」というサンプルデータを読み込む例で解説します。 このデータは、西宮市の高等学校に通う生徒のデータの想定で作成されており、「出席番号」「通学手段」「住所」「読書冊数」「自宅までの距離」「年度」という6つのフィールドから構成されています。各データはタブで区切られています。

サンプルデータ(school.tsv)

次のプログラムでは□school.tsv□から、テーブルオブジェクトを作成し、「通学データ」という変数に代入しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る　表示。
```

外部データの読み込み
テーブルオブジェクト名=テーブル！”データファイル名“　ファイルから作る。

次のプログラムでは、区切り文字を「,」に、文字コードを「utf-8□に指定して□□school.csv□というファイルを読み込んでいます。

```
例題＝テーブル！"," 区切り文字 "utf-8" 文字コード□"data/school.csv" ファイルから作る。
```

区切り文字の指定
テーブルオブジェクト！”区切り文字“ 区切り文字。
文字コードの指定
テーブルオブジェクト！”文字コード“ 文字コード。

ファイル名が間違っている等の理由でファイルを読めない場合には、「ファイルから作る」メソッドは未定義オブジェクト□undef□を返します。未定義オブジェクトに「表示」などの命令を送ると「オブジェクトが作られていません」というエラーが表示されます。

次のようなエラーが表示された場合は、2行目で未定義オブジェクトに「表示」が送られていますので、1行目で通学データに未定義オブジェクトが代入されたことがわかります。1行目の「ファイルから作る」が正しく実行されてテーブルオブジェクトを返すのではなく、正しく実行されずに未定義オブジェクトを返していますので、その理由を検討してください。この例では□□data/school.tsv□が「school.tsv□と書かれていることが原因でした。

テーブルオブジェクトの生成後に、フィールド名を変更することができます。以下のプログラムは、”出席番号“ というフィールド名を ”学籍番号“ に変更する例です。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"出席番号" "学籍番号" フィールド名変更　表示。
```

フィールド名の変更
テーブルオブジェクト！”変更前フィールド名“ “変更後フィールド名” フィールド名変更。

2.1.2 プログラム中でデータを作る

テーブルオブジェクトをファイルからではなくプログラムの中で作ることができます。テーブルオブジェクトを生成する際に引数としてフィールド名を記述します。以下のプログラムでは最初に通学データという名前で6個のフィールドを持つテーブルオブジェクトを作成しています。次に、［追加］命令を使い、通学データに対して1レコードずつデータを追加しています。

```
通学データ＝テーブル！"出席番号" "通学手段" "住所" "読書冊数" "自宅までの距離" "年度" 作る。
通学データ! 1 "徒歩" "西宮市" 0.5 0.28 2015 追加。
通学データ! 2 "バス" "西宮市" 1 5.7 2015 追加。
通学データ! 3 "電車" "大阪市" 3 14.7 2016 追加。
```

```
通学データ! 4 "電車" "京都市" 2 37.1 2016　追加。
通学データ! 5 "徒歩" "西宮市" 0.5 1.2 2016　追加 表示。
```

2.2 テーブルオブジェクトの表示

テーブルを実行画面へ出力するには、［表示］命令を使います。テーブルは［表示］命令を使って出力できます。以下の例では、「通学データ」という名前のテーブルオブジェクトのテーブルを実行画面に新しく、テキストエリアとして表示します。データ中の欠損値は「NA」と表示されます。

```
通学データ＝テーブル"data/school.tsv" ファイルから作る。
通学データ！表示。
```

表示
テーブルオブジェクト！表示。

2.3 テーブルオブジェクトのデータ操作

データ分析を行うために必要なデータ処理を行います。テーブルオブジェクトに対して、フィールド（列）やレコード（行）の抽出や複数のテーブルの結合など、データベース操作と同様の処理を施すことができます。これらの機能を利用することで、抽出したレコードやフィールドのデータに計算を行ったり、グラフを描画したりすることができます。また、データを昇順、降順に並べ替えたり、重複するデータを削除したりする機能もあります。

2.3.1 選択：レコード（行）の抽出

テーブルから条件に該当するレコード（行）を取り出します。以下の例では、”住所“フィールドの値が「神戸市」のレコードを取り出し、表示しています。

```
通学データ＝テーブル"data/school.tsv" ファイルから作る。
結果＝通学データ！「住所=="神戸市"」選択。
結果！表示。
```

選択：レコード（行）の抽出
テーブルオブジェクト！「フィールド名==“キーワード”」　選択。

本機能を活用することで、指定したフィールドの特定の検索条件に該当するデータに対して平均値や合計値等を算出できます。以下の例は、”住所“フィールドが「西宮市」のレコードを対象に”自宅までの距離“フィールドの平均値を算出しています。

```
通学データ＝テーブル"data/school.tsv" ファイルから作る。
結果＝通学データ！「住所=="西宮市"」選択。
結果！"自宅までの距離"　平均値　表示。
```

2.3.2 射影：フィールド（列）の抽出

テーブルから条件に該当するフィールド（列）を取り出します。以下の例では、”住所“と”自宅までの距離“のフィールドを取り出し、表示しています。

```
通学データ＝テーブル"data/school.tsv" ファイルから作る。
結果＝通学データ！"住所"　"自宅までの距離"　射影。
```

結果！表示。

選択：レコード（行）の抽出
テーブルオブジェクト！”フィールド名“ “フィールド名”…　射影。

2.3.3 結合：テーブルオブジェクト同士の結合

2つのテーブルを結合したテーブルを作ります。２つのテーブルには、共通のフィールドが1個以上存在する必要があります。以下の例では「住所データ」と「読書データ」の２つのテーブルオブジェクトを結合し、「通学データ」というテーブルオブジェクトを作成しています。

```
住所データ＝テーブル！"出席番号" "通学手段" "住所" "自宅までの距離"　作る。
住所データ!1 "徒歩" "西宮市" 0.28　追加。
住所データ!2 "バス" "西宮市" 5.7　追加。
読書データ＝テーブル！"出席番号" "読書データ" "年度"　作る。
読書データ!1 0.5 2015　追加。
読書データ!2 1 2015　追加。
通学データ＝住所データ!(読書データ) 結合　表示。
```

選択：レコード（行）の抽出
テーブルオブジェクト！（結合するテーブルオブジェクト）　結合。

2.3.4 レコードの追加

テーブルに新たなレコードを追加します。以下の例では、ファイルから作成したテーブルに2件のレコードを追加し表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！73　"徒歩"　"西宮市"　0.5　1.2　2016　追加。
通学データ！74　"電車"　"神戸市"　2　10.2　2016　追加。
通学データ！表示。
```

追加：レコード（行）の挿入
テーブルオブジェクト！データ1　データ2…　追加。

2.3.5 レコードの並べ替え：昇順・降順

並べ替えの基準となるフィールドを指定し、昇順あるいは降順に並べ替えます。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　小さい順　表示。
通学データ！"自宅までの距離"　大きい順　表示。
```

レコードの並べ替え：昇順
テーブルオブジェクト！”フィールド名“　小さい順。
レコードの並べ替え：降順
テーブルオブジェクト！”フィールド名“　大きい順。

2.3.6 重複の削除

テーブル内の重複するレコードを取り除きます。この命令は引数を取りません。以下の例では、通学データから住所のフィールドのみを取り出し、データの重複を取り除いて表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"住所"　射影。
結果！重複なし　表示。
```

重複なし：重複の削除
テーブルオブジェクト！重複なし。

2.4 統計関数

作成したテーブルオブジェクトに対して統計処理を行えます。関数を実行すると、算出したデータを含むテーブルオブジェクトが返ります。

テーブルの中で、データが入っていない箇所（欠損値）は「NA□と表示されます。統計関数では欠損値を対象から除いて処理が行われます。

引数の数が足りない場合、関数は未定義オブジェクト□undef□を返します。合計値のような数値を処理する関数は、データ中に文字列などの数値以外のデータが存在する場合は未定義オブジェクト□undef□を返します。

処理結果は[配列にする]命令を使うことで、テーブルオブジェクトの値を配列として取得できます。また、[数にする]命令を使うことで、テーブルオブジェクトの値を数値として取得できます。[数にする]命令は複数の数値が結果としてある場合には一番目の要素を返します。

以下の例では、”自宅までの距離“の合計値を算出し、「結果の配列」という名前の配列として取得します。同様に、「結果の数値」という名前の変数に数値として取得しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離"　合計値。
結果の配列＝結果！配列にする。
結果の数値＝結果！数にする。
ラベル！(結果の配列) 作る。
ラベル！(結果の数値) 作る。
```

配列にする
テーブルオブジェクト！配列にする。
数にする
テーブルオブジェクト！数にする。

2.4.1 合計値

指定した1つのフィールドの合計値を算出します。以下の例では、”自宅までの距離“フィールドの合計値を算出した結果をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　合計値　表示。
```

合計値
テーブルオブジェクト！”フィールド名“　合計値。

2.4.2 平均値

指定した1つのフィールドの平均値を算出します。以下の例では、”自宅までの距離“フィールドの平均値を算出した結果をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 平均値 表示。
```

平均値
テーブルオブジェクト！”フィールド名“ 平均値。

2.4.3 中央値

指定した1つのフィールドの中央値を算出します。データが奇数個の場合は中央の値を返します。データが偶数個の場合は中央の2つの値の平均値を算出し返します。以下の例では、”自宅までの距離“フィールドの中央値をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 中央値 表示。
```

中央値
テーブルオブジェクト！”フィールド名“ 中央値。

2.4.4 最頻値

指定した1つのフィールドの最頻値を算出します。最頻値が複数ある場合はすべてを結果として返します。以下の例では、”自宅までの距離“フィールドの最頻値をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 最頻値 表示。
```

最頻値
テーブルオブジェクト！”フィールド名“ 最頻値。

2.4.5 最小値

指定した1つフィールドの最小値を算出します。以下の例では、”自宅までの距離“フィールドの最小値をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 最小値 表示。
```

最小値
テーブルオブジェクト！”フィールド名“ 最小値。

2.4.6 最大値

指定した1つのフィールドの最大値を算出します。以下の例では、”自宅までの距離“フィールドの最大値をテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 最大値 表示。
```

最大値
テーブルオブジェクト！”フィールド名“ 最大値。

2.4.7 四分位数

第1四分位数は指定した1つのフィールドのデータを昇順に並べ、二等分した際の前半の中央値を算出します。第3四分位数は指定した1つのフィールドのデータを昇順に並べ、二等分した際の後半の中央値を算出します。以下の例では、”自宅までの距離“フィールドの第1四分位数および第3四分位数をそれぞれテーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 第1四分位数 表示。
通学データ！"自宅までの距離" 第3四分位数 表示。
```

第1四分位数
テーブルオブジェクト！”フィールド名“ 第1四分位数。
第3四分位数
テーブルオブジェクト！”フィールド名“ 第3四分位数。

2.4.8 分散

指定した1つフィールドの分散を算出します。以下の例では、”自宅までの距離“フィールドの分散を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 分散 表示。
```

分散
テーブルオブジェクト！”フィールド名“ 分散。

2.4.9 不偏分散

指定した1つのフィールドの不偏分散を算出します。以下の例では、”自宅までの距離“フィールドの不偏分散を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 不偏分散 表示。
```

不偏偏差
テーブルオブジェクト！”フィールド名“ 不偏分散。

2.4.10 共分散

指定した2つのフィールド間の共分散を算出します。以下の例では、”自宅までの距離“フィールドと”読書冊数“フィールドの共分散を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
```

```
通学データ！"自宅までの距離"　"読書冊数"　共分散　表示。
```

共分散
テーブルオブジェクト！”フィールド名1“　”フィールド名2“　共分散。

2.4.11 不偏共分散

指定した2つのフィールド間の不偏共分散を算出します。以下の例では、”自宅までの距離“フィールドと”読書冊数“フィールドの不偏共分散を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　"読書冊数"　不偏共分散　表示。
```

不偏共分散
テーブルオブジェクト！”フィールド名1“　”フィールド名2“　不偏共分散。

2.4.12 偏差

　指定した1つのフィールドの偏差を算出します。以下の例では、”自宅までの距離“フィールドの偏差を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　偏差　表示。
```

偏差
テーブルオブジェクト！”フィールド名“　偏差。

2.4.13 標準偏差

指定した1つのフィールドの標準偏差を算出します。以下の例では、”自宅までの距離“フィールドの標準偏差を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　標準偏差　表示。
```

標準偏差
テーブルオブジェクト！”フィールド名“　標準偏差。

2.4.14 不偏標準偏差

指定した1つのフィールドの不偏標準偏差を算出します。以下の例では、”自宅までの距離“フィールドの不偏標準偏差を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離"　不偏標準偏差　表示。
```

不偏標準偏差
テーブルオブジェクト！”フィールド名“　不偏標準偏差。

2.4.15 相関係数

指定した1つのフィールドの相関係数を算出します。以下の例では、”自宅までの距離“フィールドと”読書冊数“フィールドの相関係数を算出し、テーブルオブジェクトの1列目に表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" "読書冊数" 相関係数 表示。
```

相関係数
テーブルオブジェクト！”フィールド名1“ ”フィールド名2“ 相関係数。

2.4.16 度数

指定した1つのフィールドの度数(件数)を算出します。以下の例では、”住所“フィールドの度数を算出し、結果を表示しています。［度数］命令で得られる結果は、1列目が指定したフィールド（この例では”住所“フィールド）の重複なしデータ、2列目が1列目に対応する度数から構成されるテーブルオブジェクトです。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"住所" 度数 表示。
```

度数
テーブルオブジェクト！”フィールド名“ 度数。

2.4.17 度数分布・度数分布表

指定した1つのフィールドの度数分布を返します。級数（区画）を算出し、級数ごとに度数を算出します。級数はスタージェスの公式を用いて算出されます。以下の例では、”自宅までの距離“フィールドの度数分布を算出し表示しています。［度数分布］命令で得られる結果は、1列目が階級、2列目が度数から構成されるテーブルオブジェクトです。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 度数分布 表示。
```

相対度数も算出する場合は［度数分布表］命令を使用し、相対度数を算出しない場合は［度数分布］命令を使用します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 度数分布表 表示。
```

級数を指定することもできます。フィールド名に続けて、2つめの引数として指定してください。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" 5 度数分布 表示。
```

度数分布
テーブルオブジェクト！”フィールド名“　度数分布。
度数分布表
テーブルオブジェクト！”フィールド名“　度数分布表。
度数分布：級数の指定
テーブルオブジェクト！”フィールド名“　級数　度数分布。

2.4.18 クロス集計・クロス集計表

指定した2つのフィールド間のクロス集計を行った結果を返します。合計を算出せずに集計する場合は［クロス集計］命令をします。以下の例では、”通学手段“フィールドと”住所“のクロス集計を算出し、表示しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"通学手段"　"住所"　クロス集計　表示。
```

合計も含め集計したい場合は［クロス集計表］命令を使用します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"通学手段"　"住所"　クロス集計表　表示。
```

クロス集計
テーブルオブジェクト！”フィールド名1“　”フィールド名2“　クロス集計。
クロス集計表
テーブルオブジェクト！”フィールド名1“　”フィールド名2“　クロス集計表。

2.5 グラフ描画機能

棒グラフや円グラフなど、テーブルオブジェクトから様々なグラフオブジェクトを生成できます。グラフオブジェクトは、[縦軸タイトル]と[横軸タイトル]命令などで必要な設定を行った後、[描画]命令で画面に表示します。

2.5.1 棒グラフ

指定したテーブルオブジェクトの1列目のデータを横軸、2列目以降のデータを縦軸とした棒グラフを描画します。以下の例では、各通学手段の利用人数を棒グラフで描画するために、［度数］命令で通学手段ごとの度数を算出し、「結果」という変数にテーブルオブジェクトとして代入しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段"　度数。
結果！棒グラフ　描画。
```

そして「結果」のテーブルオブジェクトから棒グラフを作成し、画面に描画しています。

複数系列の棒グラフを描画したいときは、3列目以降にデータを追加します。以下の例では、住所別の通学手段の棒グラフを描画します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段" "住所" クロス集計。
結果！棒グラフ 描画。
```

まず、住所ごとの通学手段の利用人数（度数）を求めるために、［クロス集計］命令を使って“通学手段”と”住所“のクロス集計結果を算出します。ここでは、1列目が横軸に利用する”通学手段“フィールドの重複なしデータ、2列目以降が住所別の通学手段の度数から構成されるテーブルオブジェクトが得られ、「結果」と名付けられたテーブルオブジェクトに代入しています。そして最後に、得られた「結果」から棒グラフを描画しています。

フィールド名を指定すると、指定したフィールドの棒グラフのみ描画できます。以下の例では、”西宮市“と”京都市“の通学手段に関する棒グラフを描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段" "住所" クロス集計。
結果！"西宮市" "京都市" 棒グラフ 描画。
```

棒グラフの描画
テーブルオブジェクト！ 棒グラフ 描画。
テーブルオブジェクト！”フィールド名“ 棒グラフ 描画。

2.5.2 積み上げ棒グラフ

指定したテーブルオブジェクトの1列目のデータを横軸、2列目以降のデータを縦軸とした積み上げ棒グラフを描画します。以下の例では、年度ごとの人数の増減を通学手段別に積み上げ縦グラフで描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段" "年度" クロス集計。
結果！積み上げ棒グラフ 描画。
```

まず、年度ごとの通学手段の利用人数（度数）を求めるために、［クロス集計］命令を使って”通学手段“と”年度“のクロス集計結果を算出します。ここでは、1列目が横軸に利用する”通学手段“フィールドの重複なしデータ、2列目以降が年度別の通学手段の度数から構成されるテーブルオブジェクトが得られ、「結果」と名付けられたテーブルオブジェクトに代入しています。

最後に、得られた「結果」から積み上げ棒グラフを描画しています。

また、フィールド名を指定すると、指定したフィールドの積み上げ棒グラフのみ描画できます。以下の例では、”2015“年度の積み上げ棒グラフのみを描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段" "年度" クロス集計。
結果！"2015"積み上げ棒グラフ 描画。
```

積み上げ棒グラフの描画
テーブルオブジェクト！ 積み上げ棒グラフ 描画。
テーブルオブジェクト！”フィールド名“ 積み上げ棒グラフ 描画。

2.5.3 ヒストグラム

指定した1つのフィールドのデータからヒストグラムを描画します。引数としてテーブルオブジェクトの持つフィールド名を1つ指定します。以下の例では、”自宅までの距離“からヒストグラムを描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" ヒストグラム 描画。
```

ヒストグラム
テーブルオブジェクト！”フィールド名“ ヒストグラム 描画。

2.5.4 散布図

指定した第1引数を横軸、第2引数を縦軸とした散布図を描画します。引数としてテーブルオブジェクトの持つフィールド名を2つ指定します。以下の例では、”自宅までの距離“と”読書冊数“から散布図を描画しています。また、命令を追加することで線形近似直線を表示したり、グリッド線を消したりすることができます。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" "読書冊数" 散布図 描画。
```

散布図
テーブルオブジェクト！”フィールド名1“ ”フィールド名2“ 散布図 描画。
線形近似直線の描画
テーブルオブジェクト！”フィールド名1“ ”フィールド名2“ 散布図 線形近似 描画。
グリッドの描画
テーブルオブジェクト！”フィールド名1“ ”フィールド名2“ 散布図 グリッド線なし 描画。

2.5.5 円グラフ

指定した1つのフィールドデータから円グラフを描画します。引数としてテーブルオブジェクトの持つフィールド名を1つ指定します。以下の例では、2015年度の通学手段の人数割合を円グラフで描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段" "年度" クロス集計。
```

```
結果！"2015"　円グラフ　描画。
```

まず、2015年度の通学手段の人数（度数）を求めるために、［クロス集計］命令を使って“通学手段”と”年度“のクロス集計結果を算出します。ここでは、1列目が横軸に利用する”通学手段“フィールドの重複なしデータ、2列目以降が年度別の通学手段の度数から構成されるテーブルオブジェクトが得られ、「結果」と名付けられたテーブルオブジェクトに代入しています。

最後に、「結果」と名付けられたテーブルオブジェクトの2列目のフィールド名”2015“を引数として指定し、円グラフを描画しています。

円グラフ
テーブルオブジェクト！”フィールド名“　円グラフ　描画。

2.5.6 帯グラフ

指定したテーブルオブジェクトの1列目のデータを横軸、2列目以降のデータを縦軸とした帯グラフを描画します。以下の例では、年度ごとの人数の増減を通学手段別に帯グラフで描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段"　"年度"　クロス集計　表示。
結果！帯グラフ　描画。
```

まず、年度ごとの通学手段の利用人数（度数）を求めるために、［クロス集計］命令を使って“通学手段”と”年度“のクロス集計結果を算出します。ここでは、1列目が横軸に利用する”通学手段“フィールドの重複なしデータ、2列目以降が年度別の通学手段の度数から構成されるテーブルオブジェクトが得られ、「結果」と名付けられたテーブルオブジェクトに代入しています。

最後に、得られた「結果」から帯グラフを描画しています。

また、フィールド名を指定すると、指定したフィールドの帯グラフのみ描画できます。以下の例では、”2015“年度の帯グラフのみを描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果=通学データ！"通学手段"　"年度"　クロス集計　表示。
結果！"2015"帯グラフ　描画。
```

帯グラフの描画
テーブルオブジェクト！　帯グラフ　描画。
テーブルオブジェクト！”フィールド名“　帯グラフ　描画。

2.5.7 箱ひげ図

指定したテーブルオブジェクトの1列目のデータを横軸、2列目のデータを縦軸とした箱ひげ図を描画します。以下の例では、”通学手段“と”自宅までの距離“に関する箱ひげ図を描画しています。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ!"通学手段""自宅までの距離" 箱ひげ図 描画。
```

箱ひげ図の描画
テーブルオブジェクト！”フィールド名1“ “フィールド名2” 箱ひげ図 描画。

2.6 グラフオブジェクトの操作

描画するグラフに対して様々な設定を行えます。

2.6.1 縦軸間隔

グラフの縦軸のメモリ間隔を設定します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
通学データ！"自宅までの距離" "読書冊数" 散布図 1 縦軸間隔 描画。
```

縦軸間隔：グラフの縦軸のメモリ間隔を変更する
グラフ！(縦軸のメモリ間隔) 縦軸間隔。

2.6.2 横軸タイトル

グラフの横軸のタイトルを設定します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離" "読書冊数" 散布図。
結果！"通学距離" 横軸タイトル 描画。
```

横軸タイトル変更：グラフの横軸のタイトルを設定する
グラフ！”タイトル“ 横軸タイトル。

2.6.3 縦軸タイトル

グラフの縦軸のタイトルを設定します（現在は縦書き文字に対応していないため、（）などの記号は表示が崩れます）。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離" "読書冊数" 散布図。
結果！"一月の読書冊数" 縦軸タイトル 描画。
```

縦軸タイトル：グラフの縦軸のタイトルを設定する
グラフ！”タイトル“ 縦軸タイトル。

2.6.4 位置の指定

グラフを描画するときの原点(縦軸と横軸の交点)の位置を絶対座標で指定できます（円グラフの場合は中心が原点となります）。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離" "読書冊数" 散布図 100 -100 位置 描画。
```

位置：原点を絶対座標で移動させる
グラフ！(横方向の位置) (縦方向の位置) 位置。

2.6.5 移動する

グラフを描画するときの原点(縦軸と横軸の交点)の位置を相対座標で指定します（円グラフの場合は中心が原点となります）。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離" "読書冊数" 散布図 100 -100 移動する 描画。
```

移動する：原点を相対座標で移動させる
グラフ！(横方向の位置) (縦方向の位置) 移動する。

2.6.6 描画

グラフオブジェクトからグラフを画面に出力します。

```
通学データ＝テーブル□"data/school.tsv" ファイルから作る。
結果＝通学データ！"自宅までの距離" "読書冊数" 散布図 描画。
```

描画：画面にグラフを出力する
グラフオブジェクト！描画。