

# CLI::LaTeX::Table のインストールと最初の使い方

連絡先: bin4tsv@gmail.com

2018 年 6 月 7 日 (木) - 第 1 版  
2018 年 6 月 9 日 (土) - 第 2 版  
2018 年 6 月 12 日 (火) - 第 3 版

この文書は、CPAN にアップロードされたモジュール CLI::LaTeX::Table について、インストール方法などを解説する。他のマニュアルに書き切れなかったことで、本モジュールのインストールの方法と試す方法について、作成者の意図がきちんと伝わるようにこの文書で説明をする。ただし本モジュールの作成者のモジュール配布の経験は半年以下であり、間に合わせの知識でこの文書を作成しており、もっと良い方法はある得る。実際の利用場面において、よく読むべき箇所は 1 ページにも満たないし、残りは本書の太字の部分のをさっと目で追うだけになることも多いかもしれないが、この文書の作成意図に従い、記すべきことはできるだけ網羅したため 10 ページを超えている。

## 目次

|       |   |    |
|-------|---|----|
| 1     | 前提: 想定する計算機環境および本文書の作成方針                        | 2  |
| 2     | インストールの方法                                       | 2  |
| 2.1   | インストール前の注意点                                     | 2  |
| 2.2   | インストールの手順 (4 通りから選ぶ)                            | 3  |
| 2.2.1 | 方法 1: ディレクトリ scripts 下のプログラムに手動でパスを通す場合         | 4  |
| 2.2.2 | 方法 2: Module::Starter の Build でインストールする場合       | 5  |
| 2.2.3 | 方法 3: cpanm でインストールする場合                         | 6  |
| 2.2.4 | 方法 4: cpan でインストールする場合                          | 6  |
| 2.3   | cpan または cpanm をインストールする方法                      | 6  |
| 2.4   | CLI::LaTeX::Table のアンインストールの方法                  | 6  |
| 2.5   | 依存する他のモジュールのインストール                              | 6  |
| 3     | 各コマンドを使い始める                                     | 6  |
| 3.1   | saikoro 一様乱数                                    | 7  |
| 3.2   | transpose 転置行列                                  | 7  |
| 3.3   | csel 列選択  | 8  |
| 3.4   | latextable L <sup>A</sup> T <sub>E</sub> X 用製表機 | 9  |
| 3.5   | csv2tsv CSV 形式→TSV 形式                           | 9  |
| 4     | その他   | 10 |
| 4.1   | 注意点となりうること (Perl に多少の経験がある人が気にするようなこと)          | 10 |
| 4.2   | 本モジュール CLI::LaTeX::Table の仕組み                   | 11 |

## 1 前提: 想定する計算機環境および本文書の作成方針

Perl 5.14 以降<sup>\*1</sup>が使える環境であればどんな環境でも使えるように、本モジュール CLI::LaTeX::Table は作られている。すなわち、大学、政府および国際機関、インフラ系企業、金融機関などで、システム更新が近年なされていない場合でも、計算機環境が先進的でもそうでなくても、動くように試みられている。そのため、Perl 5 の知識があれば、意図しない動作をしない場合に迅速に対処できる程度に、プログラムは簡素に作られている。(5.14 でも新しすぎる場合は、手でプログラムを 5.1(Perl5.1 は 1994 年公開) と書き換えてエラーを起こす箇所をコメントアウトするなどの方法が考えられる。)

Unix でも Linux でも Windows でも、Mac OS X でも使えるが、本文書作成者は、Mac OS X 上の Unix 環境及びクラウドサービスの AWS (Amazon Web Service) の EC2 の Linux 端末で検証を行っている。bash と zsh のシェルで検証をしているが、それ以外の csh などのシェルを使う場合は、読者が下記の文書の意図を読み取って独自調査をしてインストール及び利用をする必要がある。

本書は、「全く知識が無くてもすぐ使えるマニュアル」にはなっていない。「初心者であってもキーワードを元にインターネットで調べたり試行錯誤を何度か繰り返せば為になる意味の分かる手順書」かつ「いくらか知識のある人がさっと読んで要領の理解しやすい手順書」になっていることを意図して書かれている。

## 2 インストールの方法

下記のインストールの方法のどれかが完了したら、次の節の「各コマンド使い始める」を参照のこと。

### 2.1 インストール前の注意点

#### インストール方法の選択:

本モジュールのインストールの方法は、以下のどれか 1 つであり、利用者の状況に応じて選択する。<sup>\*2</sup>

4. cpan を使う方法
3. cpanm を使う方法
2. Module::Starter の Build を使う方法
1. ディレクトリ scripts 下のプログラムにパスを通す方法

上記 4 通りのインストールの方法を決めるには、表 1 による比較が参考になるであろう。**cpan** がインストールされていない場合それをインストールするには通常 root 権限 (管理者権限) を必要とするし、そのインストールには数分間画面を注視するような作業を要する。近年、**cpanm** の利用も多いと思われるが、それが未インストールの場合は cpanm のインストール作業が数分間必要である。またこれら適切に使いこなすに

<sup>\*1</sup> Perl のバージョンの確認方法は `perl -v` を実行することである。5.14 は 2011 年に公開。2018 年の最新版は 5.26 である。

<sup>\*2</sup> どれを採用するのが最適であるかについては、利用者の計算機環境での権限、本モジュールを迅速にインストールしたいかどうか、計算機環境の変更を最小限にしたいかどうか、本モジュールを試しに使うかもしくは恒久的に使うか、利用者の外部ライブラリの管理ポリシーなどに依存する。

は\*3、様々な知識の習得があらかじめ必要と考えられる。Module::Starter の **Build** を使う方法とパスを通す方法は、利用者が metacpan のサイトから本モジュールをダウンロードして、`tar xzvf` または `tar -xzvf` もしくは `tar zxvf` などで解凍する作業を伴う。

インストール直後に記録/記憶すべきこと:

今後のインストール更新やアンインストールに備えて、どの方法でインストールをしたか、どの権限 (root かその他のユーザーか) でインストールしたか、どこのディレクトリにプログラムをインストールしたか (手動設定をした場合) は、必要な場合に思い出せるようにする必要がある。また、インストール時に見つけた不具合、もしくは誤操作をしそうに思われた箇所は、対処のために記録することが望ましい。

2.2 インストールの手順 (4 通りから選ぶ)

この節で下記は 主に Unix/Linux 環境を仮定する。しかし他の OS であっても十分に参考になるであろう。

本モジュール CLI::LaTeX::Table をインストールする方法 4 個の比較

| 本 module の install 法→    | 4. cpan コマンドの方法                                     | 3. cpanm コマンドの方法          | 2. Build の方法                     | 1. 手作業展開とパス設定        |
|--------------------------|---|---------------------------|----------------------------------|----------------------|
| 作業前に必要な環境:               | cpan install 済みの環境                                  | cpanm intall 済みの環境        | Module::Starter 事前 install       | Perl のみ :-)          |
| その環境の準備の手間:              | cpan の初期設定に数分間                                      | cpanm の install           | おそらく cpan や cpanm に頼る            | 皆無 :-)               |
| 管理者 (root) 権限の必要性:       | 必要がある場合が多い  | install 先 directory による   | install 先 directory による          | 不要 :-)               |
| 本 module の install の手間:  | cpan module 名 で一発 :-)                               | cpanm module 名 で一発 :-)    | download, 解凍, コマンド数回             | 左に加え、PATH 設定作業       |
| ありがちな追加の手間:              | cpan 設定の理解と右の (#)                                   | cpanm の install 法の選択と (#) | Module::Starter の install (#)    | コマンド格納 directory の検討 |
| install 先 directory の指定法 | cpan 起動して o conf など                                 | cpanm -l foo              | Build install --install_base foo | 手作業; mkdir など        |
| アンインストールの方法:             | やや煩雑  | cpanm -U (実験的)            | 書換/書き加えのファイルの追跡                  | PATH 設定の書き加えの除去など    |
| 計算機環境の破損の危険:             | root 権限で目で追いきれない操作があるのでありうる。ただし、この 8 年間実例を聞いたことがない。 |                           |                                  | install 作業時にまず起きない   |
| 既存のコマンド上書きの危険:           | あり得る  | あり得る                      | あり得る                             | 手作業で確認するので起きにくい      |
| 既存コマンドと名前衝突:             | あり得る  | あり得る                      | あり得る                             | 手で名前を書き換えて回避可能       |

表 1 この表の情報は確実とは限らない。最後の 3 行は誇張の可能性もある。文字幅を減らすため、インストール、ディレクトリは英単語で表記した。CPAN モジュールインストールの方法として「標準的」な方法は上記 4 通りの方法の内左側の 3 個であるが、環境設定をしたことがない人が試しに使うには、一見手間がかかっても、最も右の方法が結局は容易で確実であるように思われる。

\*3 たとえば次の方法を **cpan** 及び **cpanm** について知ることが望ましい: 計算機環境に不具合を起こさないかテストする方法、ユーザー権限のみあれば実行可能なローカル環境にインストールする方法、アンインストールの方法。

## 2.2.1 方法 1: ディレクトリ scripts 下のプログラムに手動でパスを通す場合

## ”手作業”で本モジュールをインストールする方法: (全行程の概要)

1. metacpan などのサイトから **ダウンロード**。(Download と書かれた項目をブラウザ上で探してクリックする。)
2. 解凍する。ファイル名 \*.tar.gz を対象に tar xzvf または tar -xzvf を実行することで解凍は可能。
3. 解凍先のディレクトリに入って (cd コマンド)、scripts 直下のファイルを確認 (ls コマンド):
  - これからコマンドとして使うファイル名なので、後の必要に応じてすぐ把握可能とするため。
  - (必要に応じ) init.sh 以外の各ファイルのパーミッションが実行可能に設定されていることを確認。<sup>\*4</sup>
  - 他の実行可能ファイルと名前が衝突することがないかどうかを後で必要に応じ確認可能とするため。<sup>\*5</sup>
4. scripts 直下の全ファイルの新たな格納先のディレクトリを決める (例: ~/bin や /home/you/bin4tsv )。
  - そのコピー先のディレクトリがまだ無い場合は、mkdir コマンドで作成。
  - そのコピー先に既に他のファイルが存在する場合は、scripts 下のファイルと名前が衝突しないかを確認する。衝突する場合は、scripts 直下のファイル名を適宜変更する (mv -i コマンド)。
5. パス **PATH** の設定: (格納されたディレクトリにあるファイルが今後必要ときにすぐ実行出来るようにする。)
  - ここで前記で決めた格納先のディレクトリを以下、somewhere とする。
  - Windows の場合は、システム環境変数 の Path の値を編集し、somewhere を書き足す。<sup>\*6</sup>
  - Unix/Linux の OS の場合: 使っているシェルに応じて、Bash ならば ~/.bash\_profile に、Zsh ならば ~/.zshrc に、次の行を (ログイン時に毎回実行されるように) 書き加える: <sup>\*7</sup>

```
source somewhere/init.sh # ← 直接実行ではなく、あるファイルに挿入。
# init.sh はそれを格納するディレクトリを PATH に追加する働きがある。
```
  - 一時的に PATH を設定する場合: Bash または Zsh の場合、次を実行。<sup>\*8</sup>: PATH=\$PATH:somewhere  
もしくは、init.sh を source コマンドにより実行する (source init.sh など)。

## 格納先のディレクトリの決め方に関して

- ~/bin4tsv と決めてしまう方法がある。(～ はホームディレクトリを意味する。)
- echo \$PATH | tr : "\n" で検討する方法もある。自分で以前独自に作ったディレクトリがあればそこに追加することもあり。
- man hier のコマンドを実行して、Unix のファイルシステムレイアウトを知っておくと、上記のコマンドの実行結果で現れたディレクトリ名の意味が分かるであろう。なお、Build の方法で root 権限でインストールすると、本モジュールの提供するプログラムファイルは /usr/local/bin に配置される。

<sup>\*3</sup> ls -l コマンドの出力において、空白区切りの 1 列目で r,w,x のなどの文字の配列を確認する。r は読み取り可能、w は書込可能、x は実行可能を意味する。rwxrwxrwx や r-xr-xr-x のような文字列において、これら 9 文字は 3 文字ごとに左から、ユーザーの権限、グループの権限、その他の人の権限を意味する。この 9 文字の前後に、ディレクトリであるか否か、リンクファイルであるか否かを表す文字が付加される場合がある。

<sup>\*4</sup> which -a プログラム名 で検証可能。後の作業で既存ファイルへの上書き回避と、パス上の実行優先順位の検討が可能。

<sup>\*5</sup> セミコロン (;) で somewhere を後ろに追加する。システム環境変数の設定ができるようにするには、Windows 7 の場合は、「システムのプロパティ」を起動して「詳細設定」のメニューを選び「環境変数」のボタンを押す。(なお、システム環境変数の設定の方法は、インターネットで検索すれば容易に見えてくる。)

<sup>\*6</sup> 使っているシェルを確かめる場合は echo \$0 (ドル・ゼロ) を実行する。

<sup>\*7</sup> = の前後に空白文字を含めてはいけない。

ある環境で\$PATH に記載されたディレクトリが持つファイルの個数

|     |                     |
|-----|---------------------|
| 451 | /usr/local/bin      |
| 976 | /usr/bin            |
| 36  | /bin                |
| 246 | /usr/sbin           |
| 61  | /sbin               |
| 424 | /Library/TeX/texbin |
| 126 | /opt/X11/bin        |

```
表 2 for i in `echo $PATH | tr ":" "\n" | tail -7`; do echo `ls $i | wc -l`; echo $i `; done | column -t
| awk '{print $1 "\t" $2 }' | latextable -3 1
```

- なお、ある環境にてパスで指定されたディレクトリの中に含まれているファイルの個数の例は表 2 のようになった。
- プログラムの優先順位の問題を考えるには、下記を実行。  
解凍したファイルの scripts ディレクトリ下で `which -a $(ls)` または `which -a `ls`` このコマンドは、次の上書きを防ぐ問題の簡易な検査になっている。
- 既存のディレクトリに書き込む場合に上書きを防ぐためには、下記のコマンドを解凍したファイル直下で実行して、scripts 下にあるファイルと、パスが既に通っているディレクトリにあるファイルと名前が一致しないか、確認する。

```
grep -f <(ls scripts) <(find -L $(echo $PATH | tr ":" "\n")) または
grep -f <(ls scripts | perl -pe 'chomp;$_="^\$_\n"') <(find -L $(echo
$PATH | tr ":" "\n"))
```

#### PATH 上の優先順位に関して

- 本モジュール CLI::LaTeX::Table をインストールする目的は、これが提供するコマンドが動くようにすることであるから、そのコマンド (プログラムファイル) を格納したディレクトリ somewhere は既存のパスに記載されたディレクトリよりも優先順位を低くする理由はない。従って、`$PATH=$PATH:somewhere` とするより `$PATH=somewhere:$PATH` とするのが妥当のように思われるので、init.sh ではそのようにしてある。
- しかし、\$PATH 中の順序がきちんと管理されているように見えることは、管理上重要なことでもあるので、実際の動作上に差し支えなければ、`$PATH=$PATH:somewhere` と書き換えても良い。

#### 2.2.2 方法 2: Module::Starter の Build でインストールする場合

1. metacpan などのサイトから CLI::LaTeX::Table をダウンロード。(Download と書かれた項目をブラウザ上で探してクリック)。
2. 解凍する。ファイル名 \*.tar.gz を対象に `tar xzvf` または `tar -xzf` を実行することで解凍は可能。
3. 解凍先のディレクトリに入って (cd コマンドを使う)、下記を順に実行。  
(1) `perl Build.PL` # Module::Starter が未インストールの場合はエラーとなる。

- (2) `./Build`
- (3) `./Build test`
- (4) `./Build install`

### 2.2.3 方法 3: cpanm でインストールする場合

`cpanm CLI::LaTeX::Table` で完了する。cpanm の設定によってはローカル (`~/perl5/bin/` など) にインストールされる。もしくは `root` の権限を使って `cpanm -S CLI::LaTeX::Table` もしくは `sudo cpanm CLI::LaTeX::Table` でインストール。`Module::Starter` のインストールが必要である場合は、それをあらかじめインストール。

### 2.2.4 方法 4: cpan でインストールする場合

`cpan CLI::LaTeX::Table` で完了する。もしくは `sudo cpan CLI::LaTeX::Table` で完了する。`Module::Starter` のインストールが必要である場合は、それをあらかじめインストール。

## 2.3 cpan または cpanm をインストールする方法

利用している環境 (特に OS) により方法は異なるので、自分で調べることが望まれる。AWS(Amazon Web Service) の EC2 で Linux を使っている場合は、`yum install cpan` または `sum install perl-CPAN` をまず実行する。`root` 権限が必要な場合は、このコマンドの先頭に `sudo` を付加する。そして `cpan` コマンドの実行をして、適宜質問に答えながら進める。Mac の場合は `brew` がインストールされていれば `brew install cpanminus` で `cpanm` コマンドが使えるようにする。もしも、`cpan` がインストールされていて使える状態の場合に、`cpanm` をインストールする場合は `cpan App::cpanminus` を実行する。

## 2.4 CLI::LaTeX::Table のアンインストールの方法

`cpan`, `cpanm` もしくは `Build` の方法を使った場合には、それぞれの方法のアンインストールの方法が利用できる。(本モジュールの作成者は現状、このやり方についてここでは触れない。) 手作業でプログラムにパスを通す方法でインストールをした場合は、上記の作業で行ったことを元に戻す。

## 2.5 依存する他のモジュールのインストール

本モジュール `CLI::LaTeX::Table` は `csv2tsv` というプログラムがあり、このプログラムはコアモジュールではない CPAN モジュール `Text::CSV_XS` に依存する。これがインストールされていない場合は、`cpanm` または `cpan` でインストールする。(Text::CSV\_XS は既に広く使われているので、この方法で問題が起きることは考えにくい。)

## 3 各コマンドを使い始める

提供されている全てのプログラムについて、第一行目は `#!/usr/bin/perl` で始まる文字列で書かれている。これは、シェバングもしくはシバングと呼ばれ、たとえば `latexable` というコマンドについて `perl latexable` のように `perl` を付加して実行しなくても、単に `latexable` と書くだけで同じ動作をす

る。ただし、下記の条件を必要とする。

- そのプログラムが実行可能であること。  
(そうでないならパーミッションを変更。`chmod +x filename` を実行。)
- そのプログラムを格納しているディレクトリにパスが通っていること。
- 実行可能な perl が `/usr/bin/perl` にあること。  
(他の場所にあるなら、シェバングをその場所書き換える必要がある。)
- Unix/Linux などの OS であること。シェバングを認識しない OS であれば、コマンドを実行したい場合は `perl filename` と律儀に書く必要がある。

### 3.1 saikoro 一様乱数

`saikoro` は一様乱数を出力する。コマンドラインで下記を順次実行する。

```
saikoro → 乱数が表示される。サイコロのように 1 から 6 までの整数が出力される。
saikoro > /dev/null → 標準エラー出力のみが表示される。
saikoro -g 5x3 → 5 行 3 列の乱数が表示される。
saikoro -g 10x5 -y 91..100 → 91 から 100 の範囲の整数の一様乱数が、10 行 5 列の乱数で出力。
saikoro -s 123 → 乱数シードの設定。再現性が確保される (繰り返し実行しても同じ結果になる)。
saikoro -. 3 → 小数点以下 3 桁まで表示する。連続一様乱数になる。
saikoro --help → ヘルプマニュアルの表示
saikoro --help opt → ヘルプの内、オプションスイッチのみを表示。
saikoro --help en → 日本語では無く、英語のヘルプを表示。

saikoro -/ , -g 5x5 → 横方向の区切り文字の変更。CSV 形式として使える。
saikoro -g12x12 -y 1.5e5 → 科学的表記法 (1.5e5) も使える。
saikoro -g12x12 -y 1.5e5 | less -x7 → タブ間隔は less で表示中でも -x N enter で変更可能。
tabs -6 → タブの表示間隔を 6 文字に変更する。元に戻す場合は、単に tabs を実行。
```

### 3.2 transpose 転置行列

`transpose` は標準入力から TSV 形式 (タブ文字区切り形式) で行列を読み取り、その転置行列 (縦方向と横方向を逆転した行列) を標準出力に出力する。入力は数値で無くても良い。

```
transpose <( saikoro -s123 ) → <( .. ) はプロセス置換である。
saikoro -s123 -g3x5 | transpose → saikoro -s123 -g3x5 と比較せよ。
saikoro -s56 -g 7x8 | transpose
saikoro -/, -g3x5 | transpose -/ ";" → トリッキーな使い方なので、汎用性はあまり無い。
```

### 3.3 csel 列選択

`csel` は簡単な列処理を AWK 言語や `cut` コマンドよりも簡単に実行出来るように考えて作られた。標準出力に出力する。入力数値で無くても良い。

```
saikoro -q -y23 -g4x5 -s67 > tmp01 ← 一時的なファイル tmp01 にデータを書き込む。
csel -p4,2 tmp01 ← 4 列目と 2 列目のみを出力。上記のと比較せよ。
csel -p4,2 < tmp01 ← リダイレクション ( < )
< tmp01 csel -p4,2 ← 似た様なコマンドの末尾のみを変更したい場合、便利。
csel -d 2,4 tmp01 ← 2 列目と 4 列目の出力を抑制。
csel -d 2..4 tmp01 ← 2 列目から 4 列目までの出力を抑制。
csel -d -1 tmp01 ← 最右列の出力を抑制
csel -h -1 tmp01 ← 最右列を最も左 (先頭 head) に出力。
csel -t -2,3 tmp01 ← 2,3 列目を最も右 (末尾 tail) に移動。
csel -t -2,3 tmp01 | csel -~ -t 2,3 ← 列の並びを元に戻す。何か処理を挟む場合に便利。
```

`latextable` コマンドによりこのような表が容易に作成出来る。罫線の有無は選択可能。  
Excel でも Google のワークシートでもコピペして L<sup>A</sup>T<sub>E</sub>X 用の表が作成出来る。

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 9 | 1 | 1 | 3 | 6 | 8 | 6 | 1 | 7 | 3 | 9 | 1 | 8 | 1 | 8 | 3 | 9 | 1 | 9 | 5 | 2 | 8 | 3 | 8 | 7 | 1 | 7 | 8 |
| 7 | 9 | 0 | 9 | 6 | 9 | 2 | 5 | 8 | 6 | 8 | 6 | 1 | 5 | 9 | 4 | 1 | 9 | 3 | 1 | 6 | 3 | 5 | 2 | 1 | 6 | 2 | 3 | 1 | 6 |
| 9 | 5 | 2 | 4 | 3 | 3 | 9 | 4 | 7 | 9 | 3 | 6 | 8 | 6 | 3 | 0 | 1 | 5 | 2 | 5 | 0 | 3 | 9 | 9 | 1 | 5 | 5 | 8 | 2 | 6 |
| 2 | 4 | 1 | 3 | 7 | 2 | 7 | 5 | 8 | 3 | 2 | 1 | 2 | 9 | 8 | 0 | 2 | 0 | 0 | 5 | 3 | 5 | 6 | 2 | 9 | 8 | 7 | 3 | 5 | 0 |
| 8 | 7 | 4 | 9 | 4 | 1 | 4 | 9 | 4 | 7 | 9 | 0 | 9 | 1 | 7 | 7 | 7 | 2 | 5 | 1 | 0 | 9 | 1 | 9 | 4 | 5 | 9 | 4 | 0 | 2 |
| 6 | 3 | 2 | 2 | 7 | 2 | 9 | 8 | 1 | 0 | 2 | 9 | 7 | 3 | 6 | 9 | 7 | 7 | 8 | 8 | 5 | 9 | 6 | 5 | 9 | 9 | 2 | 7 | 4 | 0 |
| 2 | 0 | 8 | 4 | 8 | 9 | 7 | 3 | 4 | 0 | 1 | 8 | 3 | 0 | 2 | 8 | 9 | 1 | 3 | 3 | 3 | 7 | 3 | 5 | 5 | 9 | 7 | 8 | 2 | 2 |
| 5 | 0 | 7 | 6 | 8 | 5 | 6 | 0 | 5 | 0 | 6 | 7 | 8 | 1 | 1 | 1 | 7 | 4 | 3 | 9 | 6 | 4 | 6 | 9 | 7 | 4 | 9 | 7 | 4 | 2 |
| 6 | 4 | 6 | 2 | 9 | 2 | 6 | 5 | 7 | 0 | 5 | 7 | 5 | 4 | 4 | 8 | 0 | 8 | 1 | 7 | 7 | 2 | 6 | 9 | 1 | 6 | 3 | 7 | 9 | 4 |
| 5 | 8 | 3 | 9 | 1 | 3 | 4 | 1 | 3 | 0 | 7 | 7 | 4 | 2 | 6 | 4 | 1 | 0 | 8 | 8 | 6 | 1 | 8 | 9 | 9 | 0 | 6 | 8 | 4 | 7 |
| 5 | 1 | 3 | 7 | 8 | 7 | 8 | 7 | 7 | 3 | 2 | 1 | 5 | 9 | 0 | 9 | 7 | 2 | 7 | 7 | 5 | 4 | 6 | 0 | 0 | 2 | 5 | 9 | 5 | 5 |
| 4 | 8 | 2 | 5 | 2 | 8 | 4 | 6 | 6 | 9 | 9 | 2 | 4 | 9 | 0 | 8 | 8 | 2 | 0 | 3 | 5 | 7 | 6 | 7 | 0 | 6 | 0 | 7 | 3 | 9 |
| 4 | 7 | 3 | 8 | 1 | 5 | 2 | 4 | 9 | 5 | 8 | 6 | 9 | 8 | 2 | 5 | 5 | 2 | 9 | 6 | 8 | 3 | 0 | 6 | 8 | 1 | 8 | 0 | 6 | 0 |
| 0 | 5 | 8 | 8 | 3 | 9 | 1 | 4 | 4 | 2 | 9 | 8 | 1 | 9 | 3 | 5 | 1 | 9 | 6 | 4 | 9 | 7 | 9 | 5 | 1 | 3 | 8 | 7 | 2 | 0 |
| 7 | 5 | 8 | 1 | 7 | 1 | 7 | 4 | 8 | 2 | 7 | 1 | 4 | 5 | 3 | 4 | 4 | 2 | 2 | 1 | 5 | 4 | 2 | 6 | 7 | 3 | 1 | 9 | 9 | 1 |
| 7 | 1 | 0 | 2 | 0 | 5 | 8 | 0 | 8 | 3 | 4 | 8 | 8 | 9 | 9 | 6 | 3 | 8 | 8 | 1 | 6 | 8 | 4 | 3 | 7 | 4 | 0 | 1 | 0 | 8 |
| 3 | 1 | 4 | 5 | 6 | 1 | 8 | 4 | 4 | 3 | 0 | 1 | 2 | 7 | 2 | 6 | 5 | 4 | 3 | 2 | 0 | 0 | 0 | 8 | 3 | 7 | 6 | 9 | 2 | 3 |
| 9 | 6 | 5 | 8 | 7 | 2 | 2 | 0 | 5 | 2 | 7 | 8 | 5 | 2 | 6 | 3 | 4 | 2 | 7 | 0 | 1 | 7 | 8 | 9 | 4 | 5 | 9 | 9 | 4 | 2 |
| 2 | 0 | 7 | 6 | 2 | 6 | 3 | 9 | 8 | 5 | 1 | 8 | 2 | 5 | 4 | 1 | 3 | 2 | 4 | 9 | 6 | 4 | 3 | 4 | 5 | 7 | 6 | 4 | 6 | 1 |
| 2 | 2 | 6 | 5 | 7 | 1 | 6 | 4 | 7 | 1 | 8 | 5 | 9 | 5 | 1 | 9 | 7 | 2 | 2 | 9 | 8 | 9 | 7 | 7 | 0 | 8 | 6 | 5 | 4 | 8 |
| 5 | 9 | 8 | 9 | 9 | 4 | 6 | 9 | 0 | 7 | 8 | 6 | 3 | 8 | 0 | 2 | 0 | 8 | 2 | 6 | 0 | 6 | 1 | 1 | 7 | 0 | 4 | 4 | 3 | 9 |
| 5 | 1 | 3 | 7 | 5 | 0 | 3 | 7 | 8 | 2 | 5 | 7 | 8 | 0 | 2 | 8 | 9 | 1 | 0 | 0 | 2 | 7 | 3 | 3 | 8 | 8 | 1 | 8 | 0 | 8 |
| 4 | 4 | 5 | 4 | 6 | 5 | 6 | 5 | 3 | 2 | 8 | 0 | 6 | 9 | 6 | 6 | 3 | 1 | 5 | 4 | 0 | 3 | 2 | 5 | 5 | 3 | 4 | 4 | 4 | 8 |
| 7 | 0 | 0 | 7 | 2 | 3 | 9 | 0 | 5 | 5 | 8 | 1 | 1 | 8 | 8 | 0 | 2 | 2 | 3 | 5 | 1 | 2 | 6 | 2 | 9 | 0 | 0 | 1 | 2 | 8 |
| 0 | 1 | 5 | 0 | 1 | 7 | 1 | 2 | 7 | 1 | 2 | 7 | 5 | 9 | 7 | 9 | 1 | 6 | 9 | 5 | 3 | 0 | 6 | 4 | 7 | 6 | 0 | 7 | 6 | 6 |
| 7 | 1 | 2 | 9 | 7 | 6 | 8 | 8 | 6 | 2 | 4 | 0 | 8 | 9 | 8 | 4 | 6 | 1 | 1 | 9 | 7 | 5 | 4 | 5 | 3 | 5 | 2 | 3 | 1 | 5 |
| 7 | 1 | 6 | 1 | 1 | 0 | 0 | 5 | 4 | 1 | 5 | 2 | 0 | 9 | 1 | 0 | 9 | 3 | 4 | 0 | 2 | 6 | 3 | 6 | 2 | 1 | 5 | 9 | 4 | 3 |
| 3 | 3 | 1 | 7 | 4 | 0 | 9 | 6 | 0 | 7 | 3 | 0 | 0 | 4 | 7 | 9 | 0 | 8 | 6 | 3 | 5 | 6 | 8 | 1 | 4 | 3 | 0 | 5 | 8 | 1 |
| 8 | 5 | 4 | 4 | 4 | 9 | 1 | 2 | 3 | 8 | 0 | 5 | 1 | 6 | 4 | 9 | 6 | 7 | 6 | 8 | 9 | 4 | 3 | 7 | 6 | 9 | 0 | 4 | 6 | 6 |
| 1 | 0 | 8 | 9 | 4 | 1 | 1 | 0 | 6 | 3 | 6 | 5 | 6 | 0 | 3 | 6 | 7 | 2 | 1 | 7 | 9 | 5 | 8 | 7 | 4 | 1 | 7 | 0 | 1 | 4 |

表3 `saikoro -s123 -g 30x30 -y0..9 | latextable -m0.6 -\# . -1 ==`



### 3.4 latextable L<sup>A</sup>T<sub>E</sub>X 用製表機

`latextable` は、エクセルや各種 SQL ソフトの出力結果から L<sup>A</sup>T<sub>E</sub>X の表 (table 環境) への使い易く高機能な変換器として作られた。様々な記号を含む文字列を L<sup>A</sup>T<sub>E</sub>X に使えるようにすぐ変換する目的に使うこともできる。本モジュール CLI::LaTeX::Table の中心となる機能である。

```
saikoro -g12x12 | latextable ← LATEX を知っているなら何が起きたか分かるはず。
latextable ← この 1 つの単語を端末で入力して (Enter を押下する)。
    そしてキーボードで何かを入力し最後に Ctrl+D を押下。
    すると \begin{table} と \end{table} で囲まれた文字列が生成される。
    要領がつかめたら、LATEX の .tex 形式のファイルに記入して動作を確かめる。
    エクセルや各種表計算ソフトからコピーをしてペーストをして、いろいろ試すと良い。
latextable --help opt ← どのようなスイッチオプションが使えるか確かめる。
latextable -s ← 入力の余計な空白を取り去る。(BigQuery をブラウザ経由で使う場合に便利。)
latextable -_ ← 半角と全角の空白、全角ハイフンの存在を分かり安く表示するようにする。
latextable -3 コンマ区切り 1 始まりの列番号 ← 数値を 3 桁毎に区切って右寄せする。
latextable -j ← 日本語の半角カナを半角の幅で表示するようにする。
latextable -m0.7 -x40mm ← 大きな表を 0.7 倍の大きさにし、センタリングをする時に使う。
latextable -C1 ← 任意の (UTF8 の) 文字列を、LATEX 用に表とは関係無く簡潔に出力。
latextable -Cmn ← table 環境の caption で複数行にまたがる SQL 文を挿入する時便利。
```

### 3.5 csv2tsv CSV 形式→TSV 形式

`csv2tsv` は CSV 形式 (RFC4180) を TSV 形式に変換する。このプログラムは コアモジュールでは無い CPAN モジュール Text::CSV\_XS に依存している。(似た様な実装は容易に他の人も思いつくので、プログラム名が衝突する可能性は、本モジュールの他の 4 個のプログラムに比べ、最も高いと考えられる。)

```
csv2tsv foo.csv > foo.tsv ← CSV 形式の foo.csv を読取り TSV 形式に変換。
csv2tsv もしくは cat | csv2tsv ← 手で CSV 形式で入力する。
csv2tsv -t '(TAB)' ← CSV 形式のあるレコードがタブ文字を含む場合、何に変換するか指定。
csv2tsv -n '(ENTER)' ← C あるレコードが改行文字を含む場合、何に変換するか指定。
csv2tsv -2 ← 出力の区切りが 2 行の改行文字になる。レコード中の改行の様子を簡易に調べる。
csv2tsv -~ ← 逆変換、つまり、TSV 形式のデータを CSV 形式に変換する。
csv2tsv --help ← ヘルプ
csv2tsv --help opt ← オプションのヘルプのみ見る。opt は opti, optio, option, options でも良い。
```

## 4 その他

### 4.1 注意点となりうること (Perl に多少の経験がある人が気にするようなこと)

- 本モジュールが通常の CPAN にあるモジュールと異なる点: 本モジュールは、単数または複数の Perl 言語で書かれたプログラムファイルが単独で働くように作られている。各プログラムの中に、オンラインマニュアルヘルプも含まれているが、それらの文書は必ずしも POD 形式に従って書かれている訳ではない。通常のモジュールと異なり、ライブラリとなることを意図して作られていないので、本モジュールに含まれる \*.pm ファイルを必要としているプログラムファイルは存在しない。
- 当該モジュールが別のモジュールを必要とするか:
  - ー 本モジュールのインストール時 — Build の方法を使う場合は、Module::Starter を必要とするかも知れない。(本モジュールをダウンロードした場合に含まれていると考えられるが、場合による。) cpan がインストールされている状態で新たに cpanm をインストールする場合は、様々な方法があるが、cpan App::cpanminus を実行する方法もある。(この場合はつまり App::cpanminus に依存する。)
  - ー プログラムの実行時 — 各プログラムファイルは、内部では use 文によりいくつかのモジュールに依存している。できるだけコアモジュールに依存するので、CPAN から新たなインストールを必要としないようにしてある。ただし List::MoreUtils と Text::CSV\_XS と PerlIO::gzip などが必要により使うかもしれない。<sup>\*9</sup>
- 利用者の計算機環境に影響を与えないか:
  1. インストール時に既存の実行可能なプログラムとのファイル名の衝突 — 本モジュールのプログラムファイルの名前と別のプログラムのファイルが衝突<sup>\*10</sup>した場合に、実行時にどちらが実行されるかの問題、および、インストール時に上書きされる問題がある。cpan または cpanm の設定などに依存するが、/usr/local/bin/ に同名のファイルがあっても上書きがされないように、インストールするディレクトリを別の場所 (ローカルディレクトリなど) に変更することはできる。<sup>\*11</sup> 本モジュールのインストール方法に記載した cpan, cpanm, Module::Starter を使わない方法でインストールする場合には、パスの設定 (環境変数 \$PATH への書き込み) が通常は (特にその計算機に再びログインしてから本モジュールをすぐ使える状態にする場合は) 必要であるが、その場合でもそのパスに記載された複数のディレクトリの優先順位を検討する必要が発生しうる。<sup>\*12</sup>
  2. 実行時の汚染チェック: 一般に、Perl インタプリタの起動時に -T または -t のスイッチを与えることで汚染チェックが可能であるが、本モジュールにおいてそのような設定はまだ一部しか与えていない。

<sup>\*9</sup> 任意のモジュールがどの Perl のバージョンからコアモジュールとされているか知る方法: たとえば List::Util が Perl 5.7 からコアモジュールとなっていることを知るには、次を実行する。

```
perl -MModule::CoreList -e 'print Module::CoreList->first_release(q[List::Util]);'
```

もしくは端末上で `corelist List::Util` を実行。

<sup>\*10</sup> ファイル名の衝突とは「一致」と同じ意味であるが、この一致は望ましいものではないので、あえて衝突と表記する。なお、ファイル名とは、パス名と混同されがちだがここでは異なる。パス名とは、ディレクトリ名が階層的にファイル名の前に/を区切り文字にして連結した文字列である。

<sup>\*11</sup> ただし、その場合は cpan, cpanm, Module::Starter についての単なる利用者であること以上の知識が必要とされる。

<sup>\*12</sup> `echo $PATH | tr : "\n"` や `which -a コマンド名` のようなコマンドを使って検討することになるであろう。

3. 本モジュールにテストモジュールは付属しているか: 未実装である。<sup>\*13</sup> 実行結果が正しいかどうかの検証には、下記の方法により確かめることが望ましい。
- (a) 単純な例を入力して、動作を理解し、意図した通りに動作するか確認すること。
  - (b) 別のプログラムをうまく利用して異なる方法で結果をチェックをすること。
  - (c) Perl で書かれている本モジュールのプログラムファイルを閲覧して、意図しない動作が起こりにくいことを確認すること。(閲覧して、プログラムが十分にシンプルに書かれていることに疑問がある場合は、連絡して欲しい。たとえば、同じような動作を 2 箇所以上で実装していたりすると、バグの原因になりやすい。)

## 4.2 本モジュール CLI::LaTeX::Table の仕組み

- どのバージョンの Perl に依存するか:
 

現状 5.14 以前に合わせている。(2018 年 4 月現在の Perl の最新バージョンは 5.26 である。) 世界のさまざまな機関で利用可能となるよう、2011 年以降の Perl であれば十分かもしれないという理由がある。技術的な要請として、乱数シードを設定する `srand` の動作が 5.14 以降で変更となったためでもある。<sup>\*14</sup>それでも、出来るだけ古い計算機環境でも利用可能とすべく、5.1 でも動くように、`case`, `say`, `state` などの利用は出来るだけ避けるか避けようとしている。同様の理由で `splice` でのリファレンスの利用もしていない。
- 本モジュールに含まれるプログラムが どのモジュールをよくインポートしているか:
  - `strict`
  - `warnings` → Perl 5.6 からコアモジュール。プラグマである。
  - `Getopt::Std` → オプションスイッチのオプションとそのパラメータを取得するため、および、`--help` でオンラインマニュアルの出力をするため。Perl 5 の最初からコアモジュールである。
  - `FindBin` → 実行中のプログラムの名前を `$Script` で参照するため。Perl 5.3.7 からコアモジュール。
  - `Term::ANSIColor` → 出力に ANSI シーケンスカラーで色を付けるため。Perl 5.6 からコアモジュール。
  - `List::Util` → `min` や `max`、`sum`、`sum0` 関数を使う為。Perl 5.7.3 からコアモジュール。
- 本モジュールのインストールに必要なモジュール:
 

`Module::Starter` の CPAN モジュールの事前インストールが、`CLI::LaTeX::Table` を `cpan`, `cpanm` もしくは `Build` の方法でインストールする際に、必要である。

<sup>\*13</sup> 通常 Perl のテストはライブラリ (\*.pm 形式のファイル) に対して行う。`Test::More` などのモジュールはそうように設計されているように思われる (異なるかも知れないが、少なくともチュートリアルではそのようだ)。Perl のスクリプトに対して、汎用性の高いテストの方法を現在検討中である。

<sup>\*14</sup> `perldoc -f srand` で参照が可能。

## 5 参考文献

- 続・初めての Perl, Randal L.Schwartz, biran d foy, Tom Phoenix 著 (12 章 Perl ディストリビューションを作るには, 21 章 CPAN への投稿)
- PERL HACKS プロが教えるテクニック& ツール 101 選, chromatic, Damian Conway, Cutris ”Ovid” Poe 著 (4 章 モジュールの操作)