

# Plotly を活用した低速データ表示システムの開発

鶴 岡 弘<sup>\*†</sup>

## Development of a Low-Sampling Data Plotting System Utilizing Plotly

Hiroshi TSURUOKA<sup>\*†</sup>

### はじめに

地震研究所では、東京都から「東京都伊豆諸島火山観測データによる地殻変動に関する研究」という受託研究を受け、年に4回都庁データ検討会に出席するとともにその報告書資料の作成業務を行なっている。報告書資料は地震活動に関するものの他、傾斜計、水温、水位、気圧、雨量などの1時間サンプリングのデータ（通称、低速データ）に関する資料を作成している。図1に2017年11月1ヶ月分の三宅島における観測データの資料を示す。この観測記録の図示にあるように1枚の資料の中に多くの種類の低速データがプロットされている。これは受託研究において複数の低速データの重ね合わせ図の作成が指示されていることによる。描画において非常に工夫がなされているが、多量の低速データの時刻歴の特徴の把握と資料の枚数節約に重点がおかれているため、各低速データの相対値の比較の図示に限定されているなどの課題があった。なお、この点については個別の低速データを選択してプロットすれば解決するが受託研究の中では要求されてこなかった。ただ、新型コロナウイルス対策のため、このデータ検討会もZoom等を用いたオンライン会議に移行し、これまでは作成した資料のみを用いてのデータ検討が主であったが、電子的な資料（Zoomによる共有）による検討が一般的となった。上記で述べた課題を解決するためには、個別の低速データの図示を含めた資料作成での解決も考えられたが、今回は本報告が示したWebを用いたデータ可視化プラットフォームが簡単に利用できるようになった背景などもあり、Plotly（Plotly ウェブサイト）を活用した低速データ表示システムの開発を行なった。

### Plotly とは

JavaScriptのデータ可視化ツールとしては、D3.js, Google Charts, Highcharts や Recharts 等、今回採用したPlotly以外にも多く存在している。また、JavaScript以外にもRやPython言語等による強力なツールも存在する。ここでシステム開発に用いたPlotlyのJavaScriptバージョンでは、対応しているグラフの種類が豊富であること、画像ファイルに出力が可能であることや可視化結果をインタラクティブに操作可能である。また、PlotlyにはJavaScriptの他、R, Python, MATLABなどの言語のバージョンもあり、精力的に開発が進められている。今回は、色々な選択肢があるなかにおいて、高機能で綺麗なグラフを簡単に作成できる、このツールを採用して開発を行なった。なお、このようなプラットフォームはすでに述べたようにPlotlyだけではなく世の中には数多く存在しており、この採用がベストでない可能性があることは注意されたい。

### 低速データフォーマットについて

表1に三宅島阿古観測点の傾斜計の東西成分データの2017年11月の例を示した。データは11行のデータを説明するヘッダ、データ開始を意味する空行および実際のデータからなる。また、実際のデータは1行に1列で構成される。1時間に1サンプルのデータなので、地震波形データ（例えば、100Hz）等と比較すると圧倒的にデータ量が少ないため、Plotlyのようなプラットフォームでの描画に適切なデータとなっている。なお、データが欠測する場合もあり、このようなデータについても表示できるように対応を行なった。

### 低速データ表示例

図2に今回の表示システムによる2022年1月の三宅島観測点における低速データの表示例を示した。図2においては、Plotlyの標準機能であるauto scaleでの図示である。

2022年11月4日受付, 2022年12月20日受理.

<sup>†</sup> tsuru@eri.u-tokyo.ac.jp

\* 東京大学地震研究所地震火山情報センター

\* Earthquake and Volcano Information Center, Earthquake Research Institute, the University of Tokyo

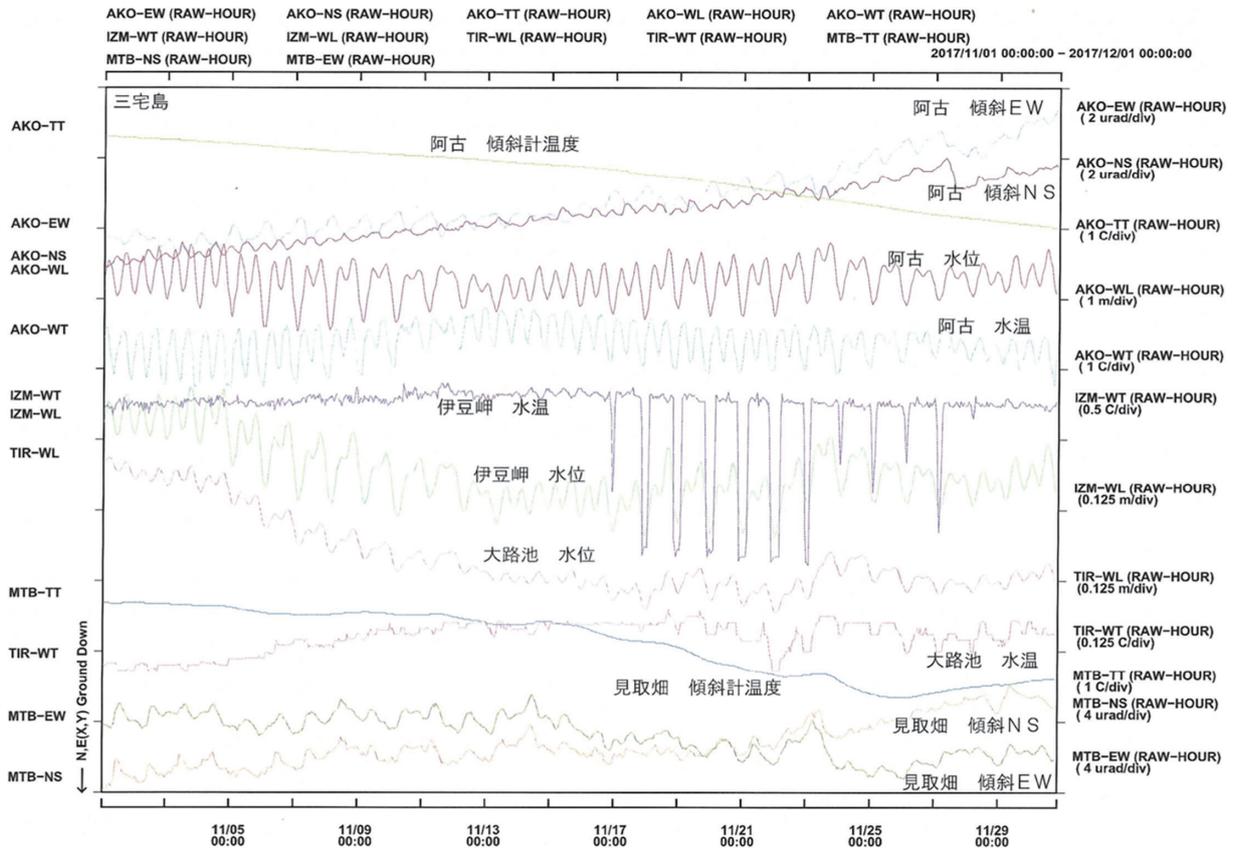


図 1. 2017 年 11 月における三宅島低速データの紙へのプロットの例

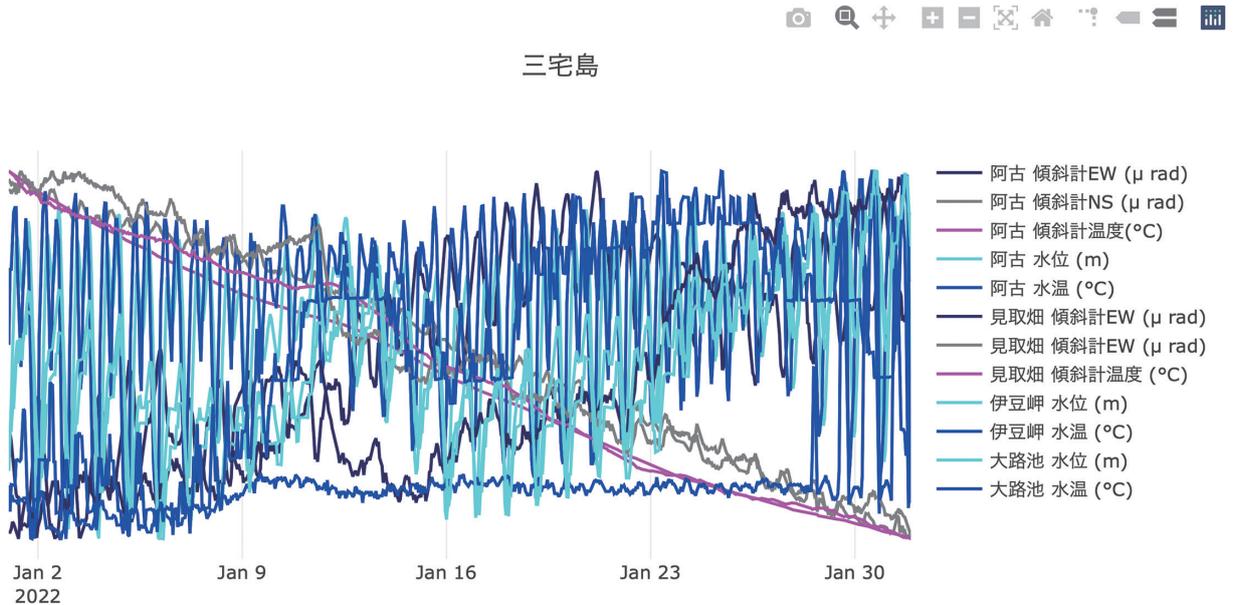


図 2. 2022 年 1 月における三宅島低速データの Plotly によるプロット例

表 1. 低速データのフォーマット

X-Location: AKO
X-Component: EW
X-Data-type: RAW
X-Date-oldest: 20171101000000
X-Date-latest: 20171130230000
X-Interval: 1hour
X-Organization: STA
X-Timezone: JST
X-Unit: urad
X-Count: 720
X-Content-length: 6333
8.44412
8.42138
8.39106
8.35316
8.32284
8.33042
8.34558
8.36074
8.31526
8.29252
8.25462
.....
.....

ただし、多数の低速データを同時に表示するとかなり見にくくなるために、Perlスクリプトを作成して、図3の表示を実現した。こちらの詳細については次節以降で述べる。さらに、低速データは三宅島以外の島でのデータもあり、それらへ簡単にリンクされたトップページ例を図4に示した。なお、観測データを複数項目表示する際には縦軸の目盛は省略されているが、個別の観測データを表示する場合は、三宅島阿古観測点傾斜計東西成分の例を図5に示したように縦軸の目盛が表示されるようになっている。生成されたHTMLファイルはローカル環境においてWebブラウザでアクセスすれば表示が可能であるのはもちろんのこと、公開サーバにアップロードすることによって簡単に複数ユーザでの共有が可能である。

### データ表示作成用 Perl スクリプトおよび データ資料ページ作成スクリプト

図3のようなデータ表示のために作成したPerlスクリプトは以下の2つである。

- (1) 1行に1列のデータを入力として、その最小値、最大値を出力する。ただし、追加の引数を入力することにより、Y軸上のプロット位置を調整する最小値、最大値を出力することが可能なスクリプト。
- (2) 二つの最小値、最大値のセットの入力値に対して、それぞれのセットが同一スケールになるように最小値、最大値を調整して出力するスクリプト。

(1)のスクリプトを表2に、(2)のスクリプトを表3に示した。表4に図4全体の出力を行なったシェルスクリプトを示した。スクリプト内の変数を編集することによって簡単

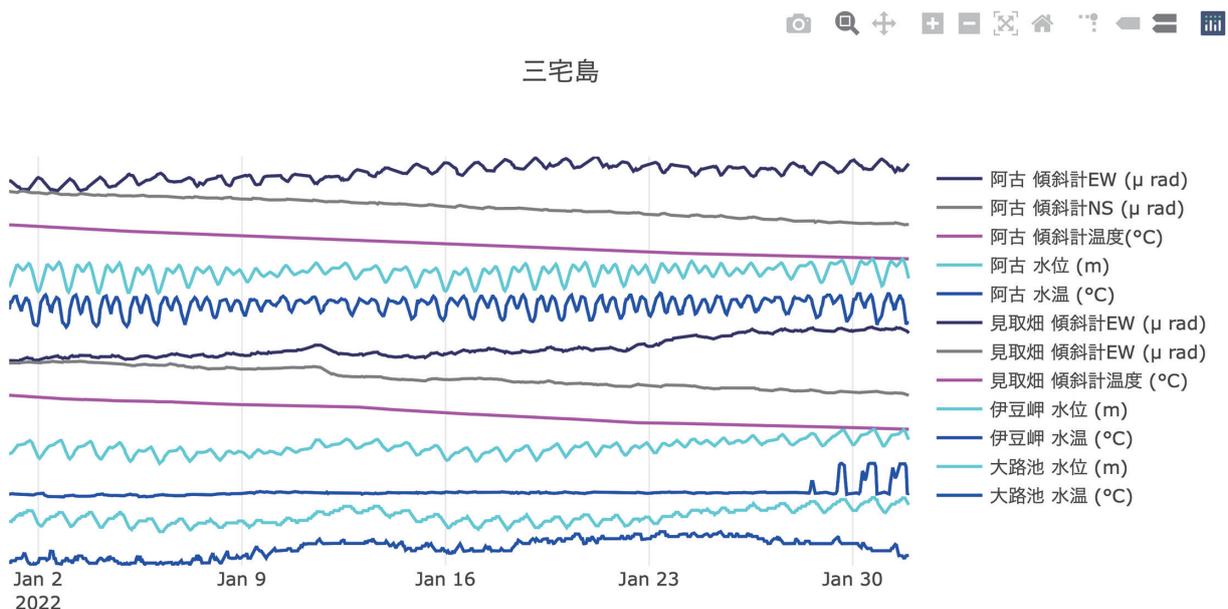


図 3. Perl スクリプトを用いた工夫によるプロット例

(青ヶ島) / 三曾根ヶ崎 / [地温] [地温1] [地温2] [地温3] / 向里 / [気圧] [気温] [雨量] (八丈島) / 三根 / 傾斜計: [EW] [NS] [温度]  
 (三宅島) / 阿古 / 傾斜計: [EW] [NS] [温度] [水位] [水温] / 見取畑 / 傾斜計: [EW] [NS] [温度] / 伊豆岬 / [水位] [水温] / 大路池 / [水位] [水温]

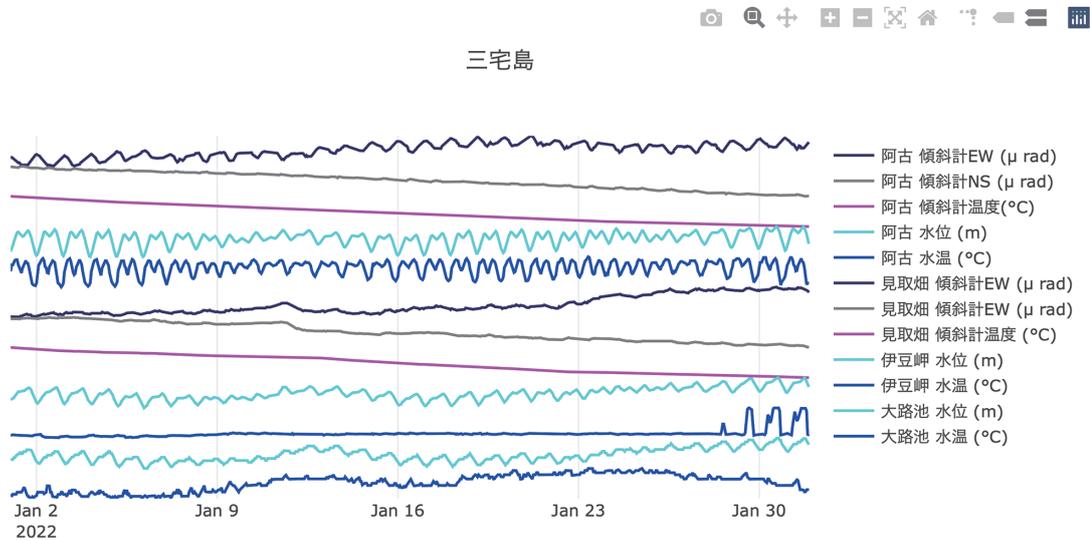


図 4. 東京都庁データ検討会低速データ資料の 2022 年 1 月の表示例

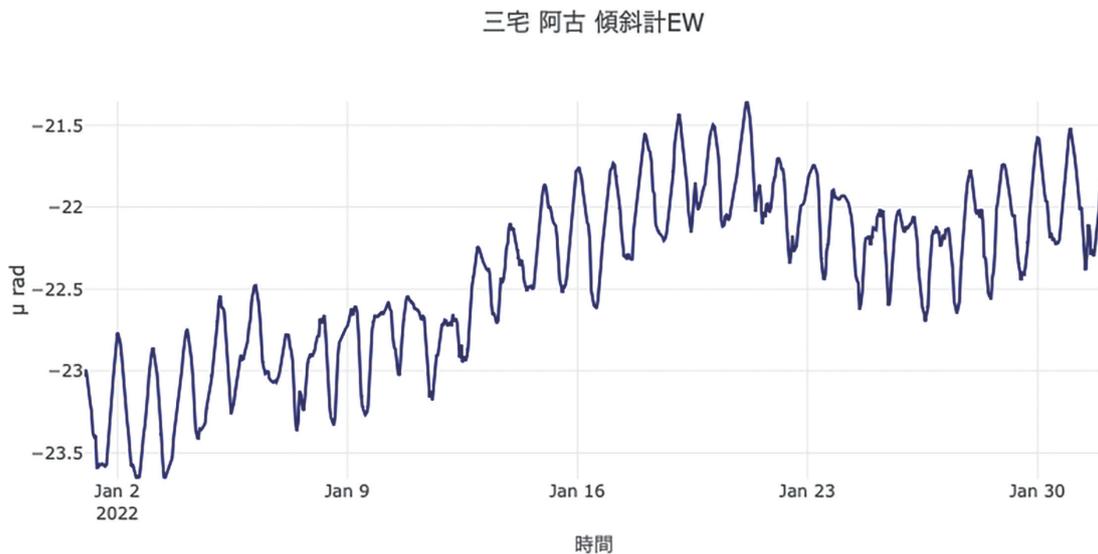


図 5. 三宅島阿古観測点傾斜計東西成分の表示例

に表示用 HTML ファイルを生成できるようになっている。また、表 5 に八丈島用の Plotly 表示用 HTML ファイル作成の скриプトも示した。さらに、低速データから Plotly 表示用の HTML ファイルを作成する скриプトを表 6 に示した。ここで強調しておきたいことは、表 4 に記載された скриプト先頭付近の 8 つのパラメータを変更してその

スクリプトを実行すれば、図 4 を表示できる HTML ファイルが作成される点である。

### ま と め

傾斜、気温、気圧および雨量などの低速データを Plotly というツールを用いて可視化するデータ表示システムを開

表 2. 1 行に 1 列のデータ入力から, その最小値, 最大値を出力する Perl スクリプト

```
#!/usr/bin/perl
# 1 行に 1 列のデータを入力としてその最小値・最大値を出力する
# 使用方法
# cat DATA | ./fminmax.pl
# -> 0.0 10.0 (最小値, 最大値を空白で区切って出力
# ただし, 追加の引数を入力することにより, Y 軸上のプロット位置を調整する最大値
# を出力することが可能である
# たとえば, データファイルの最小値, 最大値が [0,10] の時に
# cat DATA | ./fminmax.pl 2 1 (2 分割をして, 上の部分にデータを表示する)
# という場合には, 出力は, [-10,10] になる
# なお, データの欠測値は null あるいは NA で記載すること
$xmin= 9999999.0;
$xmax=-9999999.0;
$N=int($ARGV[0]);
$M=int($ARGV[1]);
$NARG=$#ARGV;
$nn=0;
while(<STDIN>){
  chop;
  if ( /null/ || /NA/ ) {
    ;
  } else {
    $data = sprintf("%f",$_);
    if ( $data < $xmin ) { $xmin = $data;}
    if ( $data > $xmax ) { $xmax = $data;}
    $nn=$nn+1;
  }
}
if ( $NARG > 0 ) {
  $dx=$xmax-$xmin;
  $xmin=$xmin-($N-$M)*$dx;
  $xmax=$xmax+($M-1.0)*$dx;
  printf("$xmin $xmax\n");
} else {
  printf("$xmin $xmax\n");
}
}
```

表 3. 最小値, 最大値を調整する Perl スクリプト

```
#!/usr/bin/perl
#二つの最小値, 最大値のセットの入力値に対
#してそれぞれのセットが同一スケールになる
#ように最小値,最大値を調整して出力する.
# ./samescale.pl 0 10 -5 0 --> 0 10 -7.5 2.5
# ./samescale.pl -5 0 0 10 --> 7.5 2.5 0 10
$X1=$ARGV[0];
$X2=$ARGV[1];
$Y1=$ARGV[2];
$Y2=$ARGV[3];
$DX=$X2-$X1;
$DY=$Y2-$Y1;
if ($DX>=$DY) {
  $DD=($DX-$DY)/2.0;
  $Y1=$Y1-$DD; $Y2=$Y2+$DD;
} else {
  $DD=($DY-$DX)/2.0;
  $X1=$X1-$DD; $X2=$X2+$DD;
}
printf("$X1 $X2 $Y1 $Y2\n");
```

発した. 現在, 東京都のデータ検討会においてデータの説明等に活用されている. 今回は 1 ヶ月分のデータを例に可視化するためのスクリプト等の説明を行なったが, 1 年分以上のデータをコンパイルするなどをすれば簡単に 1 年分以上のデータを図示できるようになる. なお, 今回作成したスクリプト等の修正・改善等は今後発生するとともに Plotly のバージョンアップなどによっても修正が必要になることは指摘しておきたい.

謝 辞: 技術研究報告編集委員会と 2 名の匿名査読者の方々には, 有益なご指摘を頂きました. ここに記して感謝申し上げます.

#### 文 献

Plotly ウェブサイト, Plotly : Low-Code Data Apps, <https://plotly.com>, (参照 2022-11-02).

表 4. 2022 年 1 月のデータ検討会資料を作成するスクリプト

```
#!/bin/sh
# 東京都庁低速データを グラフィックライブラリ Plotly で表示するプログラム
# 使い方
# パラメータの説明
# YMD1 から YMD2 までの期間のデータをプロットする
# YMD1: データ開始期間
# YMD2: データ終了期間
# KIKAN: 低速データファイルの suffix の名前
# たとえば, AKO-EW-RAW-202001 であるなら 202001
# YR0 MO0 DY0 は, データの開始時間の年月日
# YR0: データ開始時間(年)
# MO0: データ開始時間(月)
# DY0: データ開始時間(日)
# DAT: 低速データファイルが存在するディレクトリ
# OUT: HTML 等を入力するディレクトリ
YMD1="2022-01-01"
YMD2="2022-01-31"
KIKAN=202201
YR0=2022
MO0=01
DY0=1
DAT=data/2022/01
OUT=202201
if [ ! -e $OUT ]; then
  mkdir -p $OUT
fi
cp index.html plotly.min.js $OUT
cat index.html | sed -e 's/_COLOR_/0000ff/' | \
  sed -e 's/_DATAKIKAN_/2022 年 01 月/' > $OUT/index.html
sh mkhtmlAOG.sh $YMD1 $YMD2 $KIKAN $YR0 $MO0 $DY0 $DAT $OUT
sh mkhtmlHCJ.sh $YMD1 $YMD2 $KIKAN $YR0 $MO0 $DY0 $DAT $OUT
sh mkhtmlIMYK.sh $YMD1 $YMD2 $KIKAN $YR0 $MO0 $DY0 $DAT $OUT
```

表 5. 八丈島用の Plotly 表示用 HTML ファイル作成のスクリプト

```
#!/bin/sh
#YMD1="2020-01-01" YMD2="2020-12-31" KIKAN=202001-202012
#YR0=2020 MO0=1 DY0=1
#DAT=data OUT=output
YMD1=$1
YMD2=$2
KIKAN=$3
YR0=$4
MO0=$5
DY0=$6
DAT=$7
OUT=$8
HTML="$OUT/HCJ_KND_EW.html"
cat head.html > $HTML
tail -n+13 $DAT/KND-EW-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 1 "三根 傾斜計 EW" "navy" >> $HTML
MINMAX=`tail -n+13 $DAT/KND-EW-RAW-$KIKAN | ./fminmax.pl`
./mkfoot.pl 1 "八丈島 三根 傾斜計 EW" "u rad" $MINMAX $YMD1 $YMD2 1 >> $HTML
HTML="$OUT/HCJ_KND_NS.html"
cat head.html > $HTML
tail -n+13 $DAT/KND-NS-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 1 "三根 傾斜計 NS" "gray" >> $HTML
MINMAX=`tail -n+13 $DAT/KND-NS-RAW-$KIKAN | ./fminmax.pl`
./mkfoot.pl 1 "八丈島 三根 傾斜計 NS" "u rad" $MINMAX $YMD1 $YMD2 1 >> $HTML
HTML="$OUT/HCJ_KND_TT.html"
cat head.html > $HTML
tail -n+13 $DAT/KND-TT-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 1 "三根 傾斜計温度" "violet" >> $HTML
MINMAX=`tail -n+13 $DAT/KND-TT-RAW-$KIKAN | ./fminmax.pl`
./mkfoot.pl 1 "八丈島 三根 傾斜計温度" "(C)" $MINMAX $YMD1 $YMD2 1 >> $HTML
HTML="$OUT/HCJ_knd.html"
cat head.html > $HTML
tail -n+13 $DAT/KND-EW-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 1 "三根 傾斜計 EW (μ rad)" "navy" >> $HTML
tail -n+13 $DAT/KND-NS-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 2 "三根 傾斜計 NS (μ rad)" "gray" >> $HTML
tail -n+13 $DAT/KND-TT-RAW-$KIKAN | ./mkplotly.pl $YR0 $MO0 $DY0 3 "三根 傾斜計温度(C)" "violet" >> $HTML
MINMAX1=`tail -n+13 $DAT/KND-EW-RAW-$KIKAN | ./fminmax.pl 3 1`
MINMAX2=`tail -n+13 $DAT/KND-NS-RAW-$KIKAN | ./fminmax.pl 3 2`
MINMAX3=`tail -n+13 $DAT/KND-TT-RAW-$KIKAN | ./fminmax.pl 3 3`
SAME=`./samescale.pl $MINMAX1 $MINMAX2`
./mkfootx.pl "八丈島 三根" $YMD1 $YMD2 3 0 $SAME $MINMAX3 >> $HTML
```

表 6. 低速データから Plotly 表示用の HTML ファイルを作成するスクリプト

```
#!/usr/bin/perl
$YR0=$ARGV[0]; $MO0=$ARGV[1]; $DY0=$ARGV[2];
$NUMBER=$ARGV[3]; $NAME=$ARGV[4]; $COL=$ARGV[5];
$yr=$YR0; $mo=$MO0; $dy=$DY0; $hr=0; $mi=0; $sc=0;
$n=0;
while(<STDIN>){
  chop;
  $data=$_;
  $tm[$n]=sprintf("%04d-%02d-%02d %02d:%02d:%02d",$yr,$mo,$dy,$hr,$mi,$sc);
  $dt[$n]=$data;
  $hr=$hr+1;$n=$n+1;
  if ( $hr >= 24 ) {
    $hr = 0; $dy=$dy+1;
    $jul = 8julday($mo, $dy, $yr); @mdy = &caldat($jul);
    $yr = $mdy[2]; $mo = $mdy[0]; $dy = $mdy[1]; }
}
$n=$n-1;
print<<EOM;
var trace$NUMBER = {
  EOM
  printf("    x:[]");
  for($i=0;$i<$n-1;$i++){
    printf("%s",$tm[$i]);
  }
  printf("%s",$n,$tm[$n-1]); printf("    y:[]");
  for($i=0;$i<$n-1;$i++){
    $_=$dt[$i];
    if ( /null/ || /NA/ ) { printf("null,"); } else { printf("%f",$dt[$i]); }
  }
  $_=$dt[$n-1];
  if ( /null/ || /NA/ ) { printf("null",$n); } else { printf("%f",$n,$dt[$n-1]); }
  printf("name: '%s',$n",$NAME);
  if ( $NUMBER > 1 ) { printf("yaxis: '%d',$n",$NUMBER); }
  printf("connectgaps: false,$n");
  if ( $COL eq 'red' ) { printf("    marker: {color: 'rgb(255,0,0)'},$n"); }
  if ( $COL eq 'orange' ) { printf("    marker: {color: 'rgb(255,128,0)'},$n"); }
  if ( $COL eq 'pink' ) { printf("    marker: {color: 'rgb(255,0,128)'},$n"); }
  if ( $COL eq 'green' ) { printf("    marker: {color: 'rgb(0,255,0)'},$n"); }
  if ( $COL eq 'darkgreen' ) { printf("    marker: {color: 'rgb(0,192,0)'},$n"); }
  if ( $COL eq 'lightblue' ) { printf("    marker: {color: 'rgb(0,255,255)'},$n"); }
  if ( $COL eq 'blue' ) { printf("    marker: {color: 'rgb(0,0,255)'},$n"); }
  if ( $COL eq 'navy' ) { printf("    marker: {color: 'rgb(0,0,128)'},$n"); }
  if ( $COL eq 'gray' ) { printf("    marker: {color: 'rgb(128,128,128)'},$n"); }
  if ( $COL eq 'black' ) { printf("    marker: {color: 'rgb(0,0,0)'},$n"); }
  if ( $COL eq 'violet' ) { printf("    marker: {color: 'rgb(255,0,255)'},$n"); }
  if ( $COL eq 'yellow' ) { printf("    marker: {color: 'rgb(255,255,0)'},$n"); }
  printf("type: 'scatter'$n");
  printf("}");
}
```