



IUGONET

Inter-university Upper atmosphere Global Observation NETwork

2024年度研究集会「太陽地球環境データ解析に基づく超高層大気の空間・時間変動の解明」

解析セッションテキスト

- IUGONETツール概要
- pySPEDAS (CUI) を使ってみよう
- pySPEDAS/pyUDAS を使って、2024年5月の巨大磁気嵐を解析してみよう

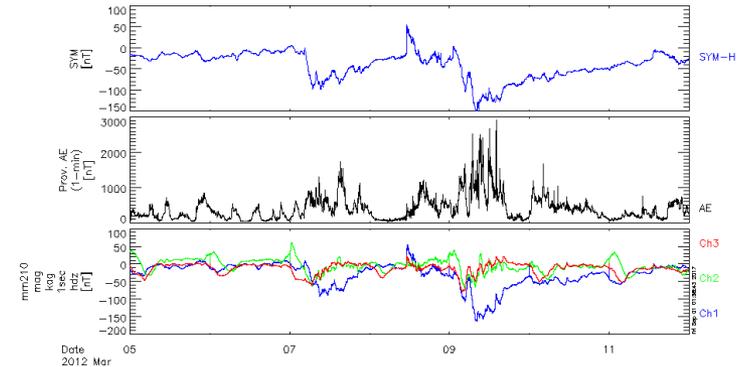
Published by IUGONET Project Team, September.
2024. <http://www.iugonet.org/>

SPEDAS

統合型解析ソフトウェア

SPEDAS

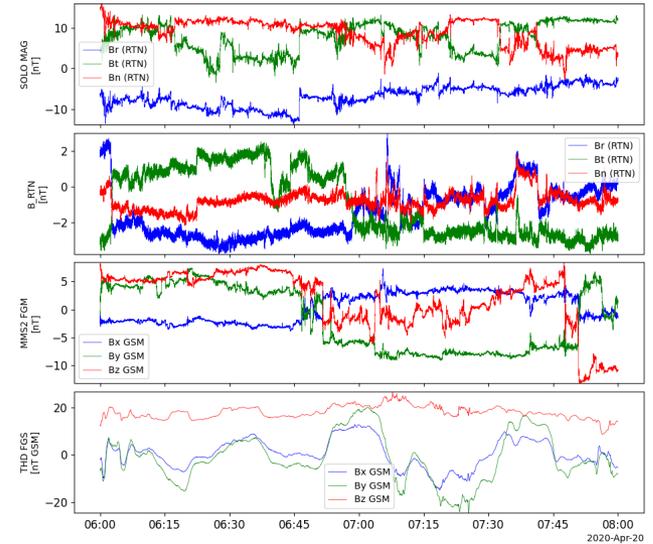
IDL言語で書かれた解析ツールです。多種多様な衛星・地上観測ミッションで取得されたデータを、CUI、GUI両方のインターフェースで解析できます。IDLライセンスが必要です。

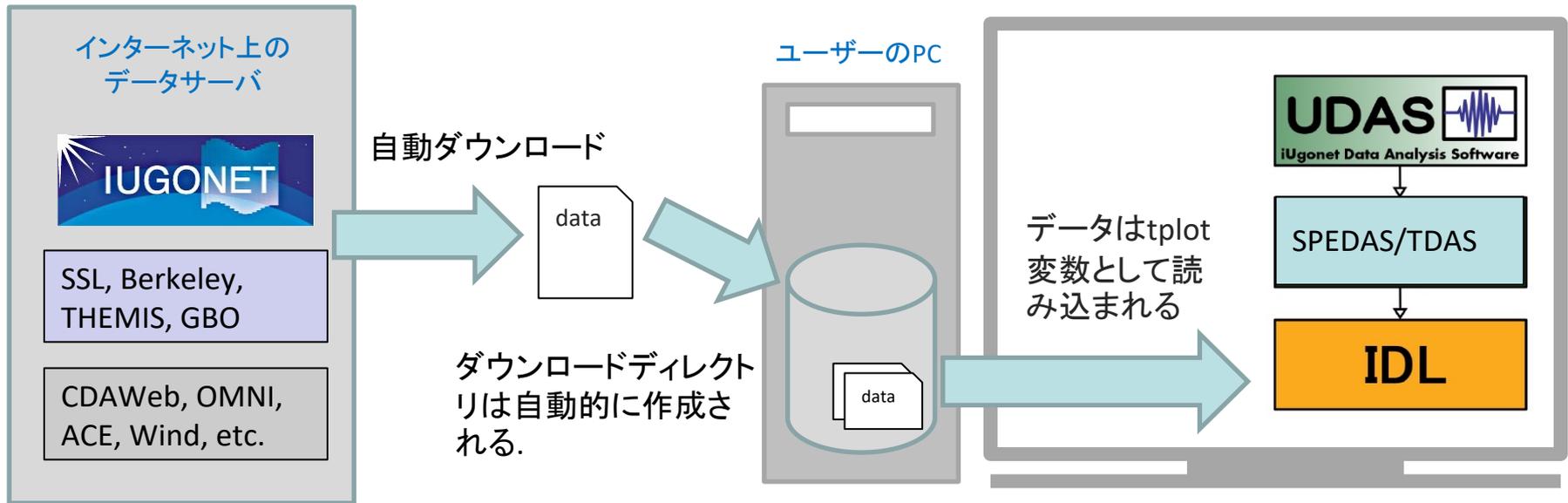


pySPEDAS

Pythonベースの統合型解析ソフトウェア pySPEDAS

Python言語で書かれた解析ツールです。多種多様な衛星・地上観測ミッションで取得されたデータを解析できます。誰でも無償で利用できます。





SPEDAS-CUI では、3つの基本コマンドでデータ読み込みと描画をすることができます。

- | | |
|------------------|-------------------------------------|
| 1. 時間幅を設定 | <code>timespan, 'yyyy-mm-dd'</code> |
| 2. ロードプロシージャを実行 | <code>iug_load_***</code> |
| 3. プロットプロシージャを実行 | <code>tplot, +++</code> |

GUIの場合でも、数回のマウスクリックで同じ結果を得ることができます。

timespan, 'YYYY-MM-DD/hh:mm:ss', N
読み込み開始の日時 読み込む日数

プロシージャ名 (, options, /keywords)

tplot_names

tplot, ['tplot変数名', ..]
プロットするtplot変数名

毎回使うコマンドとして、必ず覚えておきましょう。

読み込む時間幅を設定します。

データをロードします。

ロードしたデータは、tplot変数という、SPEDASの構造体に自動的に格納されます。複数のデータをロードしたい場合は、この操作を繰り返します。また、実行時に表示されるデータの利用ポリシーを必ず確認してください。

ロードしたデータを一覧表示します。

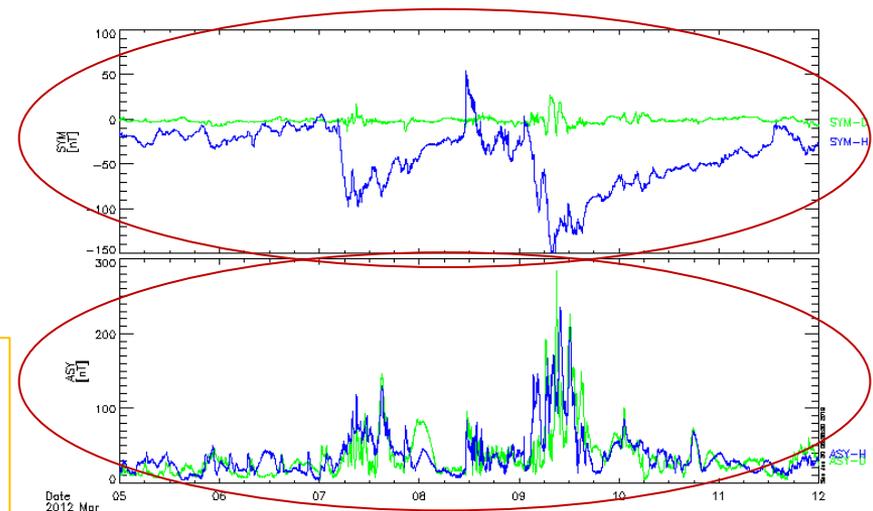
プロットを作成します。

(例) THEMIS> timespan, '2012-03-05', 7
 THEMIS> iug_load_gmag_wdc, site='sym asy'
 THEMIS> tplot_names
 1 wdc_mag_sym
 2 wdc_mag_asy
 THEMIS> tplot, ['wdc_mag_sym', 'wdc_mag_asy']



データをロードするためのプロシージャとその使い方は、SPEDASに属する各プロジェクトから提供されます。各プロジェクトのウェブサイト等を参照してください。

http://themis.ssl.berkeley.edu/socware/spedas_3_1/idl/_spd_doc.html
<http://search.iugonet.org/> 等。



指定した期間の地磁気指数SYM/ASYがプロットされました。

前ページの基本コマンドは、以下のようにも実行できます。

timespan

timespan, '2012-03-05/00:00:00', 7, /day	キーワードで指定した値が日数であることを明示します。他に、時間(/hour)、分(/min)、秒(/sec)を指定することができます。省略した場合は日(/day)が適用されます。
timespan, ['2018-04-01', '2018-04-14'] timespan, ['2018-04-01/00:00:00', '2018-04-01/12:00:00']	開始日時と終了日時を、具体的に与えることができます。

tplot_names

tplot_names, 'tplot変数名'	指定したtplot変数がロードされているかを確認することができます。
tplot_names, 'tplot変数名', /verbose	指定したtplot変数の詳細情報を見ることができます。



データ自体の注意事項、データを使う際に必要なPIへのコンタクト、論文に書くべき謝辞内容などは、/verbose オプションによる表示でも確認することができます。データロード時に表示される内容のほか、/verbose を使って、その内容を確認しておくようにしましょう。

tplot

tplot, [1, 2]	tplot変数名の代わりに、tplot変数に振られる通し番号を指定することもできます。通し番号は、tplot_names を実行すると見ることができます。
tplot	キーワードの省略により、直前に描画したデータをもう一度描画することができます。

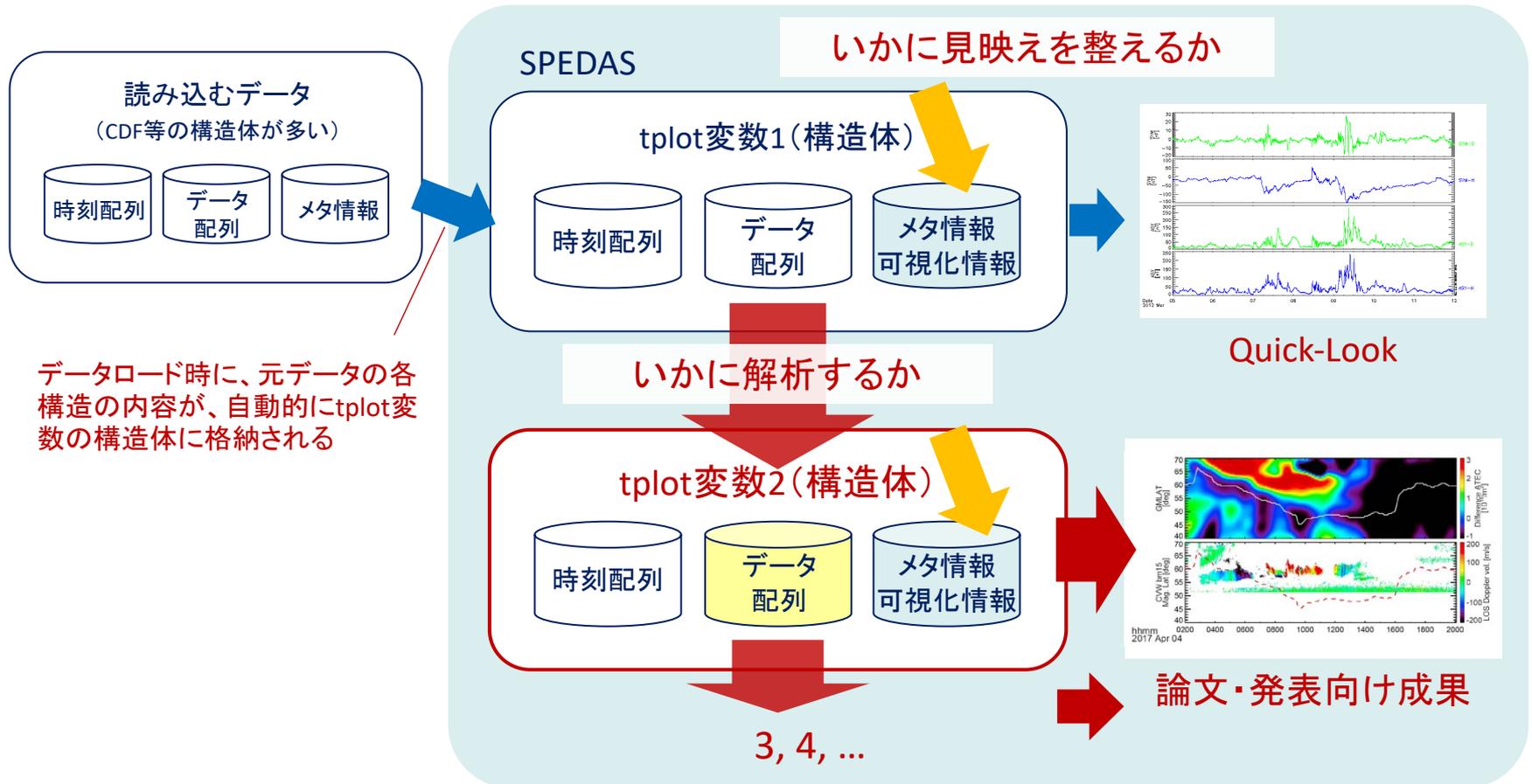
(その他の基本コマンド)

del_data, 'tplot変数'	指定したtplot変数を削除します。すべて削除する場合は、'*'を与えます。
exit	SPEDASとIDLを終了します。

tplot変数とは？

tplot変数とは、時刻情報、緯度・経度情報、値、データに関する情報(メタ情報)等を格納するための、SPEDAS構造体です。

SPEDASでは、tplot変数に格納さえすれば、簡単に描画することができます。いかに適した解析を行い、それをいかにtplot変数に格納していくかが、SPEDASを使いこなすうえでのポイントとなります。



pySPEDASのインストール

pySPEDASの動作環境

Windows、macOS、Linuxをサポートしています。

Python **3.8以上**のバージョンが必要です。

pySPEDASの公式ウェブサイトでは、Pythonの利用環境として、Anacondaを推奨していますが、利用に際し有償ライセンスが必要になる方もいるので、本講習ではPythonにJupyterLabをインストールして解析を進めます。

Pythonのインストール

Pythonをインストールしていない場合は、以下のサイトを参考に、Pythonをインストールします。

- macOS: <https://www.python.jp/install/macos/index.html>
- Windows: <https://www.python.jp/install/windows/index.html>
- Linux(Ubuntu): <https://www.python.jp/install/ubuntu/index.html>

仮想環境を作成する

他のPythonパッケージとの依存関係の問題を避けるために、PySPEDAS用の仮想環境を作成します。

各OSのターミナルで以下を実行します。

Windows, macOS, Linux共通:

```
python -m venv pyspedas
```

※python2のと共存などのため、pythonコマンドがpython3にシンボリックリンクされていない場合はpython3としてください。(以降の操作も同様)

仮想環境を開始する

ターミナルでactivateスクリプトを実行します。

Windows:

コマンドプロンプトの場合

```
¥pyspedas¥Scripts¥activate.bat
```

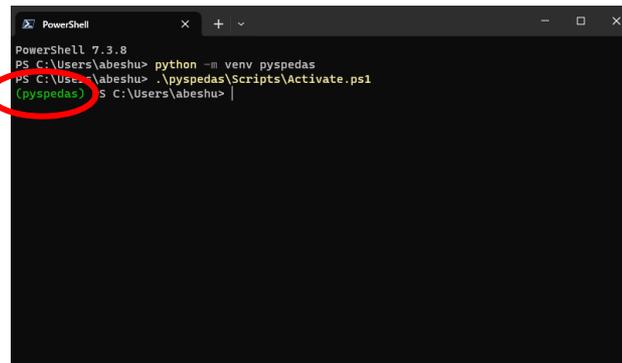
PowerShellの場合

```
¥pyspedas¥Scripts¥Activate.ps1
```

macOS, Linux:

```
source pyspedas/bin/activate
```

仮想環境が有効になると、プロンプトの先頭に
(pyspedas)と表示されます



```
PowerShell 7.3.8
PS C:\Users\abeshu> python -m venv pyspedas
PS C:\Users\abeshu> .\pyspedas\Scripts\Activate.ps1
(pyspedas) PS C:\Users\abeshu>
```

(参考)仮想環境を終了する

ターミナルでactivateスクリプトを実行します。

Windows:

コマンドプロンプトの場合

```
deactivate.bat
```

Windows: PowerShellの場合, macOS, Linux:

```
deactivate
```

仮想環境が終了すると、プロンプトの先頭の
(pyspedas)が表示されなくなります

JupyterLabのインストール

本講習では、ブラウザ上で動作する対話型プログラム実行環境のJupyterLabを使います。
ターミナル(仮想環境有効状態)で以下を実行してください。

Windows, macOS, Linux:

```
python -m pip install jupyterlab
```

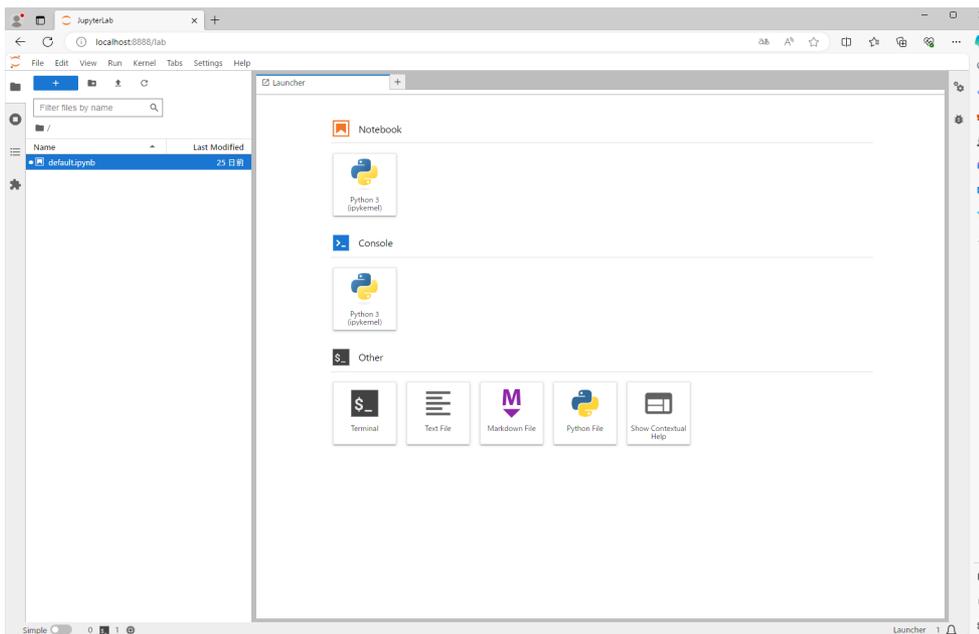
JupyterLabの起動

本講習では、ブラウザ上で動作する対話型プログラム実行環境のJupyterLabを使います。
ターミナル(仮想環境有効状態)で以下を実行してください。

Windows, macOS, Linux:

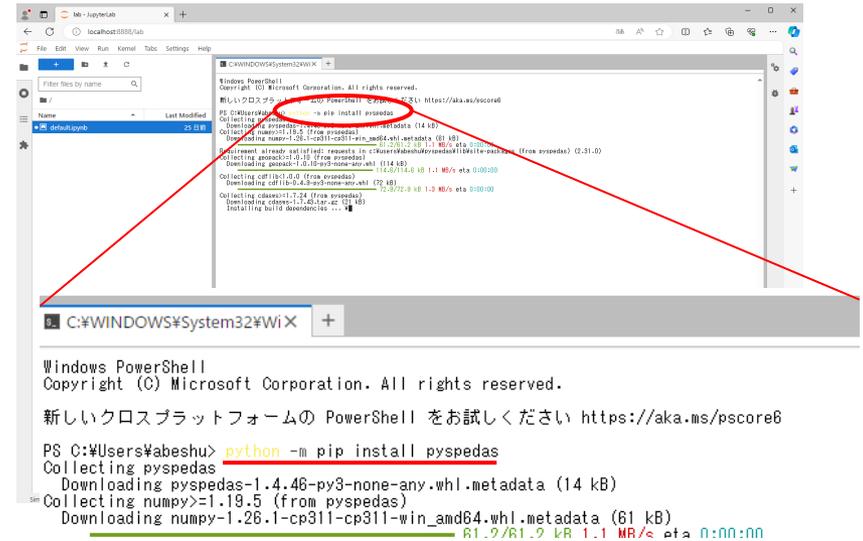
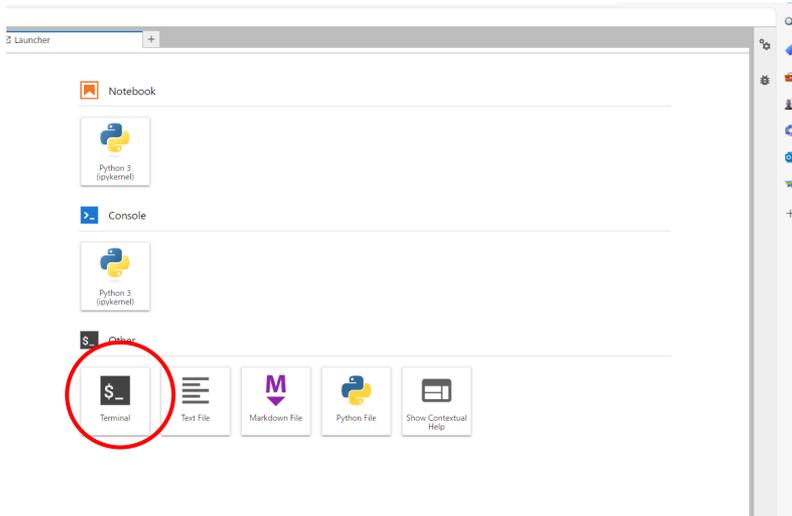
```
jupyter lab
```

ブラウザが自動で開き、JupyterLabの
ウィンドウが現れます。



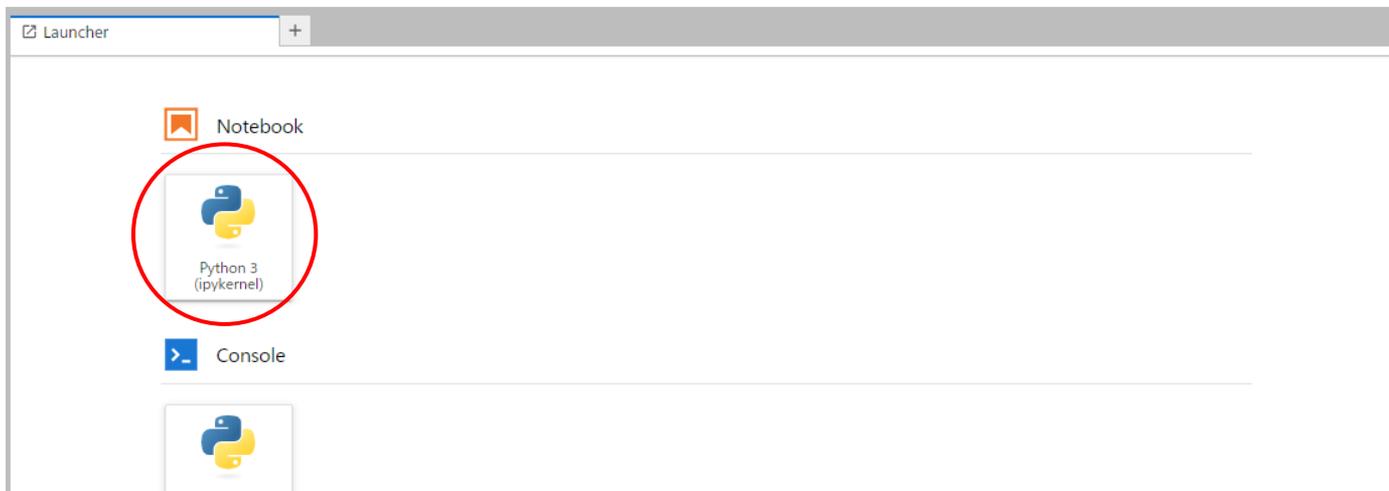
pySPEDASのインストール

1. Launcher > Othersの「Terminal」をクリックします。
2. 新しいタブでターミナルが開くので、以下を実行します。
`python -m pip install pypedas`
3. タブの×ボタンを押して、ターミナルを閉じます。

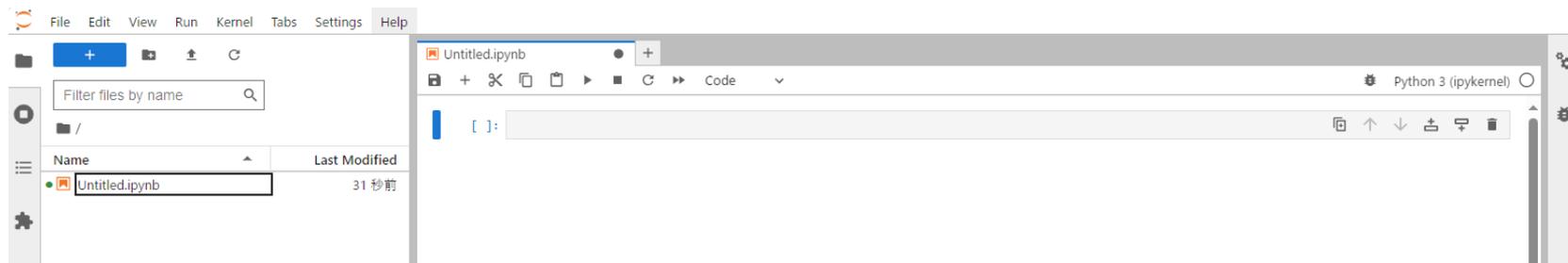


Notebookの起動

1. Launcher>NoteBookの「Python3(ipynkernel)」をクリックします。



以下のようなウィンドウが開けば、準備OKです。



pySPEDAS/pyUDAS を使って、
2024年5月の巨大磁気嵐を解析してみよう

```
import pyspedas
import pyplot
from pyplot import tplot
```

ロード関数(keyword=値)

```
pyplot.tplot_names()
```

```
tplot(['tplot変数名', ..])
```

プロットするtplot変数名

必要なパッケージをインポートします。

データをロードします。

ロードしたデータは、tplot変数という、SPEDASの構造体に自動的に格納されます。複数のデータをロードしたい場合は、この操作を繰り返します。また、実行時に表示されるデータの利用ポリシーを必ず確認してください。

ロードしたデータを一覧表示します。

プロットを作成します。

```
(例) > import pyspedas
      > import pyplot
      > from pyplot import tplot
      > pyspedas.omni.data(trange=['2024-05-10', '2024-05-14'])
      データがロードされ、メッセージが表示される。
      > pyplot.tplot_names()
      0 : IMF
      1 : PLS
      2 : IMFPTS
      :
      > tplot(['flow_speed', 'proton_density', 'BZ_GSM', 'SYM_H', 'AE_INDEX'])
```

タブによるコードの自動補完

```
In [1]: import pyspedas
import pytpplot
from pytpplot.MPLPlotter.tplot import tplot
```

```
In [ ]: pyspedas.
```

```
ace
analysis
avg_data
cdf_to_tplot
clean_spikes
cluster
cotrans
cotrans_get_coord
cotrans_set_coord
csswe
```

? コマンド

> `pyspedas.omni.data?`

```
Signature:
pyspedas.omni.data(
  trange=['2013-11-5', '2013-11-6'],
  datatype='lmin',
  level='hro2',
  suffix='',
  get_support_data=False,
  get_ignore_data=False,
  varformat=None,
  varnames=[],
  downloadonly=False,
  notplot=False,
  no_update=False,
  time_clip=True,
)
Docstring:
This function loads OMNI (Combined 1AU IP Data; Magnetic and Solar Indices) data
```

helpコマンド

> `help(pyspedas.omni.data)`

```
> import pyspedas
> import pyplot
> from pyplot import tplot
```

```
> pyspedas.omni.data(trange=['2024-05-10', '2024-05-14'])
```

※1行で書く。

データがロードされ、メッセージが表示される。

```
> pyplot.tplot_names()
```

0 : IMF

1 : PLS

2 : IMF_PTS

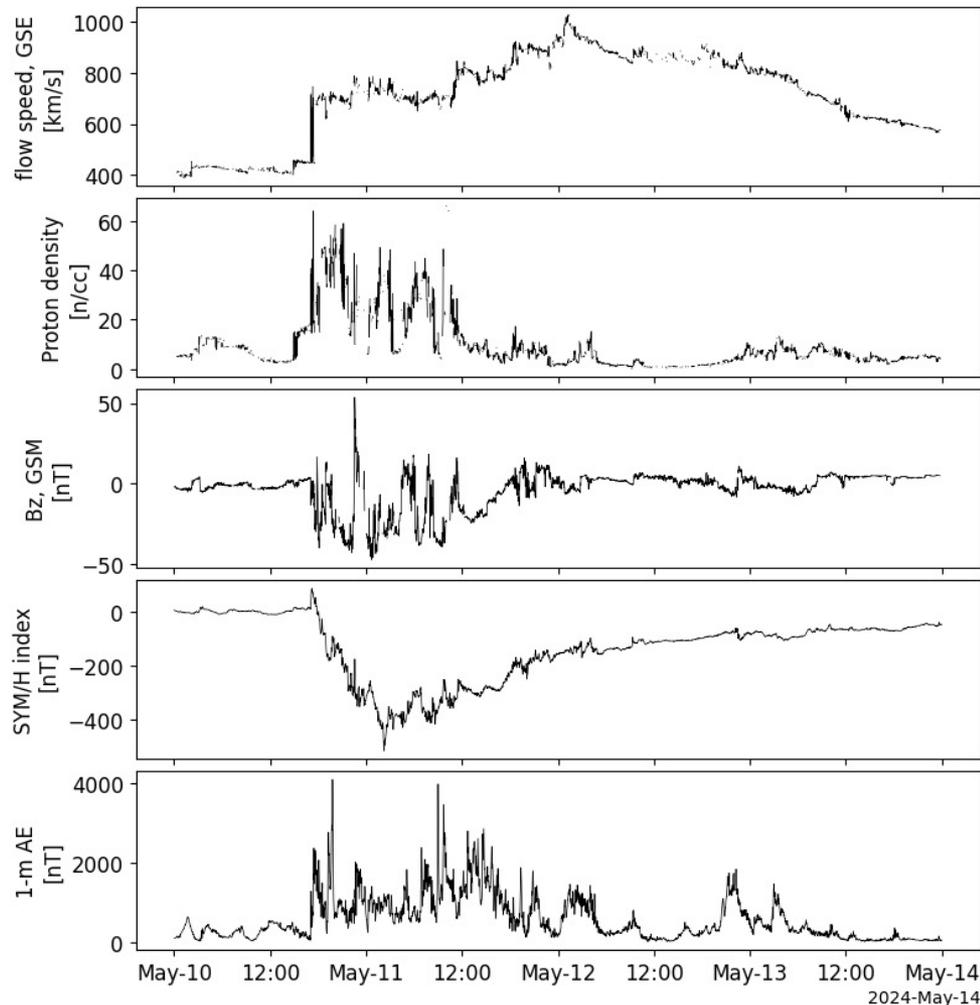
:

```
> tplot(['flow_speed', 'proton_density', 'BZ_GSM',
'SYM_H', 'AE_INDEX'])
```

※1行で書く。

変数名を使わず、通し番号で記述しても良い。

```
> tplot([17, 21, 14, 39, 35])
```



```
> import pyspedas
> import pyplot
> from pyplot import tplot
※これは、一度インポートしたら実行する必要無し。
```

```
> pyspedas.themis.gmag(trange=['2024-05-10',
'2024-05-14'], sites='brw')
※1行で書く。
```

```
> pyplot.tplot_names()
```

```
0 : IMF
```

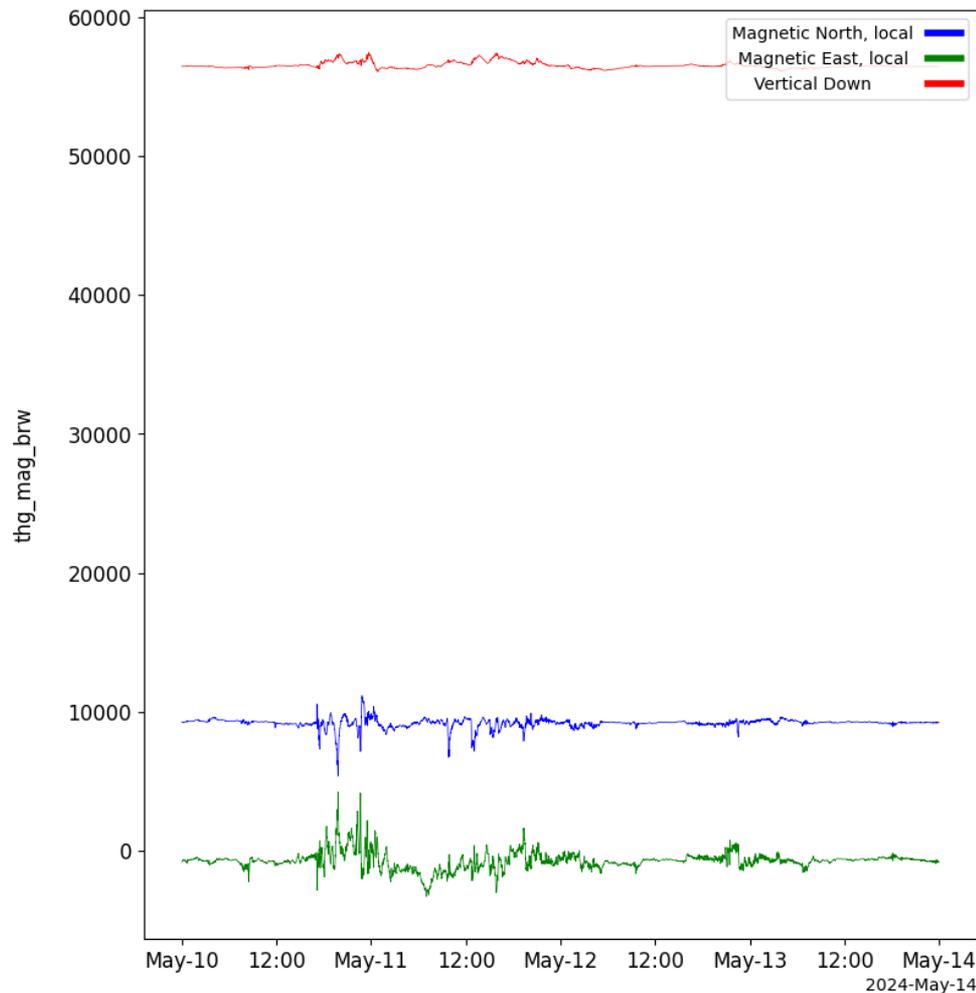
```
1 : PLS
```

```
2 : IMF_PTS
```

```
:
```

```
? : thg_mag_brw
```

```
> tplot(['thg_mag_brw'])
```



pyUDASのインストール

pyUDASとは？

IUGONETプロジェクトで開発中のpySPEDASのプラグインソフトウェアです。

pySPEDASと一緒にpyUDASをインストールすることで、IUGONETで公開している地上観測データをロード、描画することが可能となります。

公募型共同研究「ROIS-DS-JOINT」の支援で各大学の学生や開発支援業者とともに開発中です。
将来的には、pySPEDASに統合することを計画しています。

Gitのインストール

Gitがインストールされていない場合は、以下のサイトを参考に、Gitをインストールします。

- Windows: <https://prog-8.com/docs/git-env-win>
- Mac: <https://prog-8.com/docs/git-env>

Terminalから以下を実行して、

```
> git
```

usage: git...という表示ができればインストールされています。

pyUDASのインストール

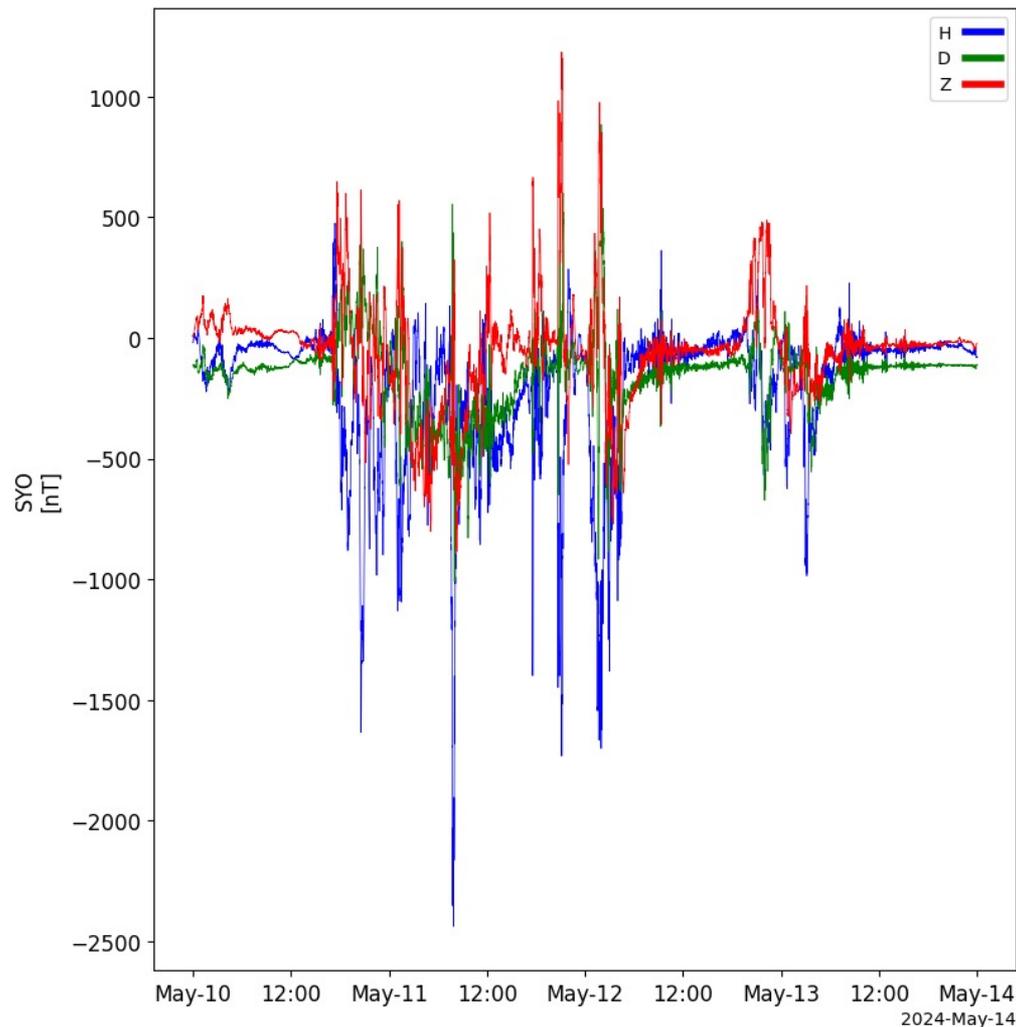
1. メニューからFile > New > Terminalを選択し、ターミナルを開きます。
2. 以下を実行します。
3. `python -m pip install git+https://github.com/iugonet/pyudas@koushu2024`
4. ターミナルを閉じます。

```
> from iugonet import gmag_nipr  
ロード関数のインポート。
```

```
> gmag_nipr(trange=['2024-05-10', '2024-05-14'],  
site='syo')  
※1行で書く。  
データがロードされ、メッセージが表示される。
```

```
> pyplot.tplot_names()  
0 : nipr_mag_syo_1sec
```

```
> tplot(['nipr_mag_syo_1sec'])
```



pytplot.split_vec('tplot変数名')

分解したいtplot変数

成分ごとに詳細な解析(相関解析や演算等)をしたい場合に使います。

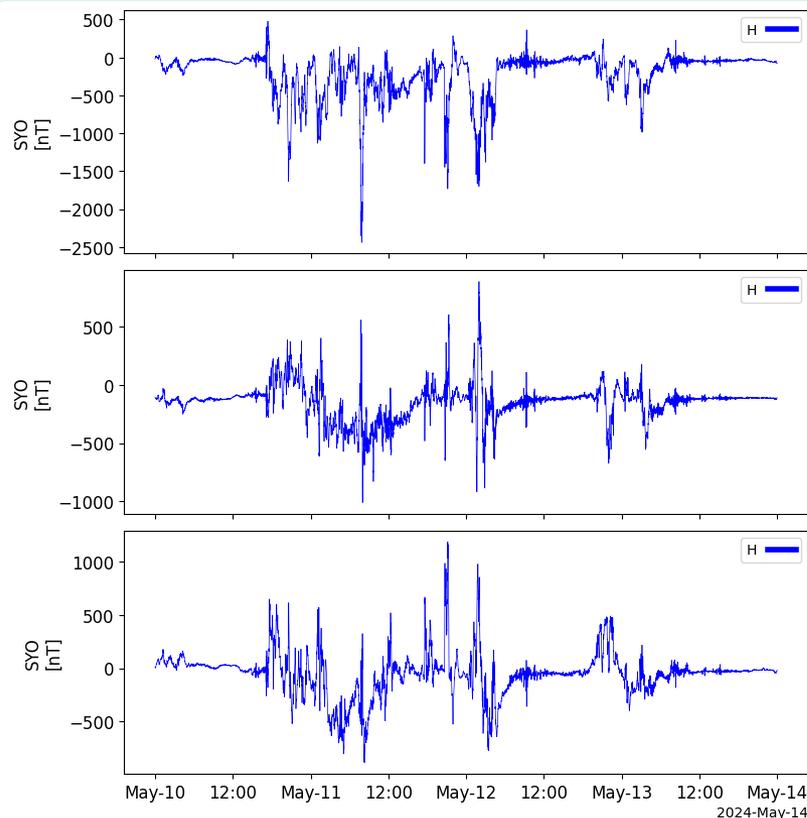
(例)

```
> pytplot.split_vec('nipr_mag_syo_1sec')
```

```
['nipr_mag_syo_1sec_x', 'nipr_mag_syo_1sec_y',  
'nipr_mag_syo_1sec_z']
```

成分分解されました。

```
> tplot(['nipr_mag_syo_1sec_x', 'nipr_mag_syo_1sec_y',  
'nipr_mag_syo_1sec_z'])
```



(その他の類似関数)

```
pytplot.join_vec(['tplot変数1', 'tplot変数2'], new_tvar='新しいtplot  
変数')
```

複数のtplot変数を、新しいtplot変数にまとめます。

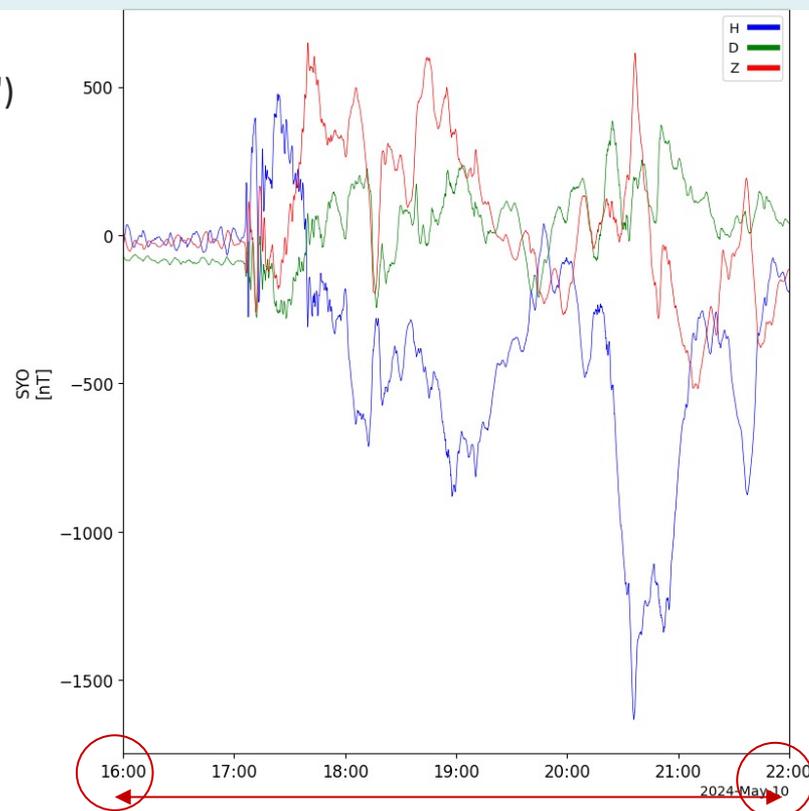
`pytplot.timespan('YYYY-MM-DD hh:mm:ss', 値,
keyword='days')`

時間軸を拡大して、時間変動を詳しく見たい場合に使います。他に、hours, minutes, secondsが使えます。

(例)

```
> pytplot.timespan('2024-05-10 16:00:00', 6, keyword='hours')
```

```
> tplot('nipr_mag_syo_1sec')
```



指定した時間幅に変更されました。

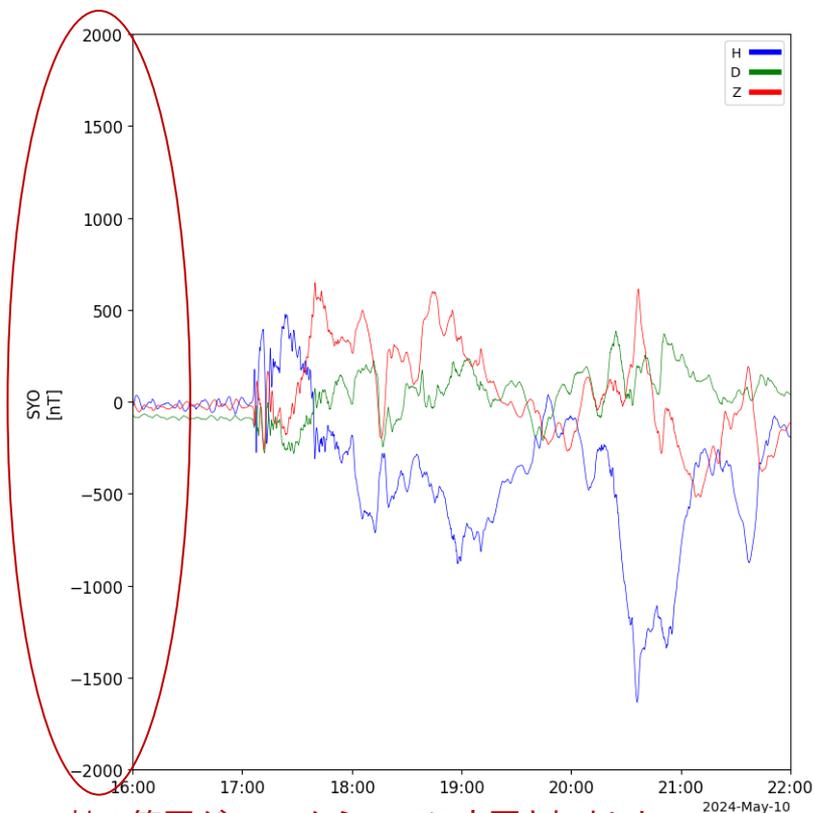
```
pytplot.options('tplot変数名', 'yrange', [最小値, 最大値])
```

縦軸を拡大して、変動量を詳しく見たい場合に使います。

(例)

```
> pytplot.options('nipr_mag_syo_1sec', 'yrange', [-2000, 2000])
```

```
> tplot('nipr_mag_syo_1sec')
```



Y軸の範囲が-2000から2000に変更されました。

(その他の類似関数)

```
pytplot.zlim('tplot変数', 最小値, 最大値)
```

データが三軸(パワースペクトル等)で構成されている場合に、同じように第三軸に対して適用します。

pytplot.tplot_options('オプション名', 値)

```
pytplot.tplot_options('title', '文字列')  
pytplot.tplot_options('axis_font_size', 値)
```

プロットの上にタイトルを表示します。
軸のフォントのサイズを変更します。
(デフォルトは10)

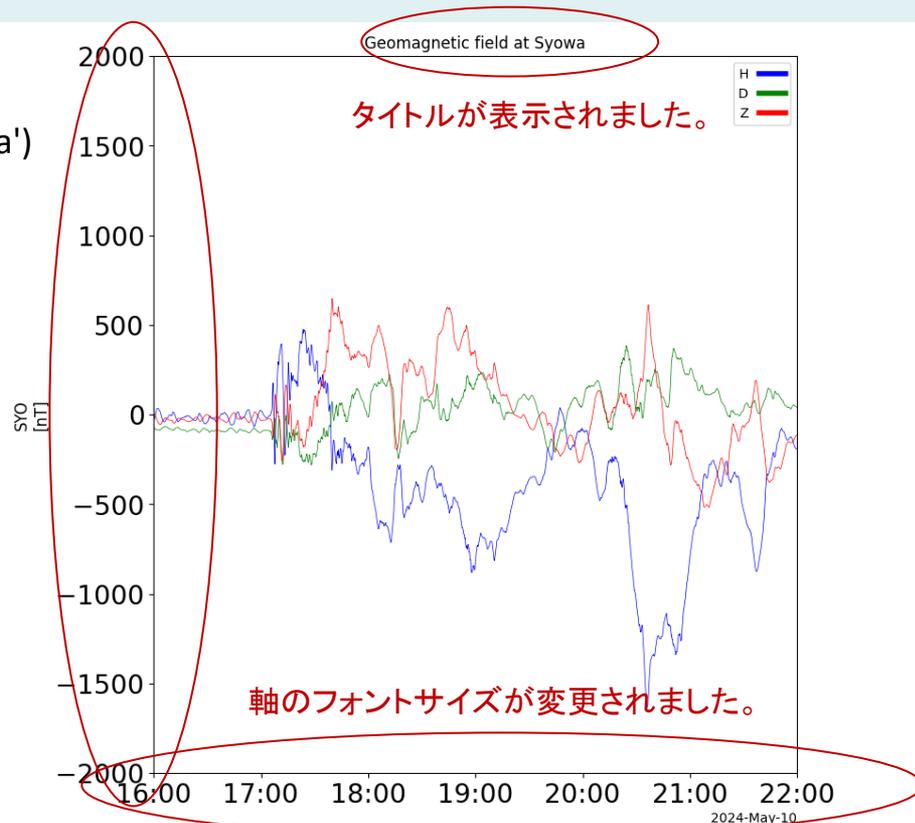
プロット全体をきれいに仕上げる場合に使います。

(例)

```
> pytplot.tplot_options('title', 'Geomagnetic field at Syowa')
```

```
> pytplot.tplot_options('axis_font_size', 20)
```

```
> tplot('nopr_mag_syo_1sec')
```



pytplot.options('tplot変数名', 'オプション名', 値)

pytplot.options('tplot変数', 'legend_names', 'ラベル文字列のリスト')

pytplot.options('tplot変数', 'yttitle', '文字列')

pytplot.options('tplot変数', 'color', '色')

プロットのラインのラベルを変更します。

Y軸のタイトルを変更します。

プロットのラインの色を変更します。

(red, green, blueなど。RGBのタプル(0, 255, 0)も使えます)

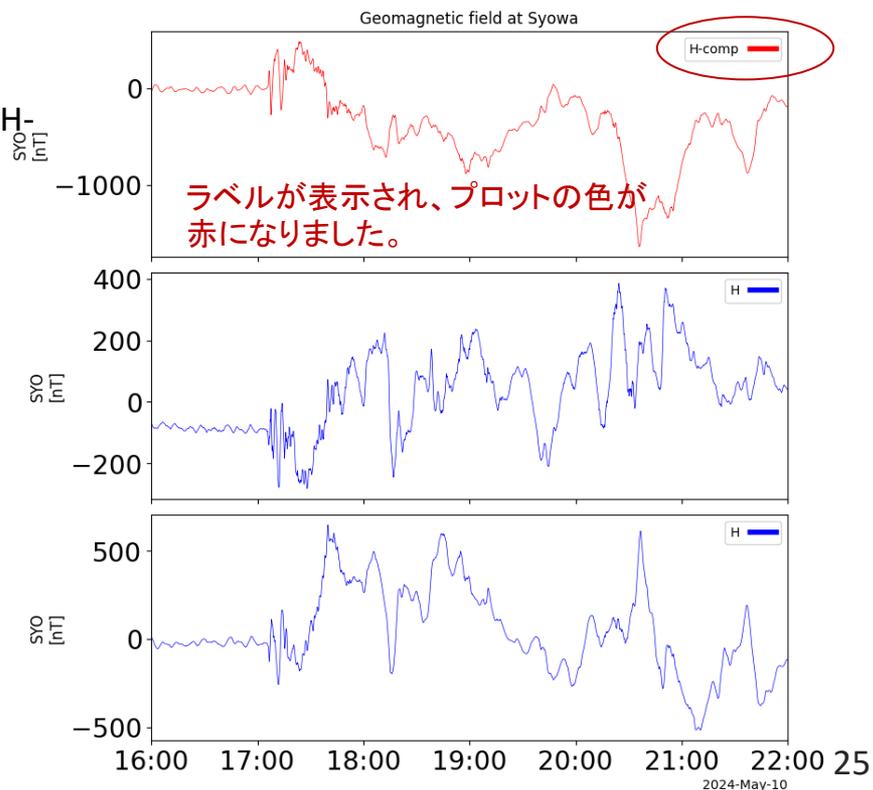
データごとに凡例や色を変更する場合に使います。

(例)

```
> pytplot.options('nipr_mag_syo_1sec_x', 'legend_names', 'H-comp')
```

```
> pytplot.options('nipr_mag_syo_1sec_x', 'color', red')
```

```
> tplot(['nipr_mag_syo_1sec_x', 'nipr_mag_syo_1sec_y', 'nipr_mag_syo_1sec_z'])
```



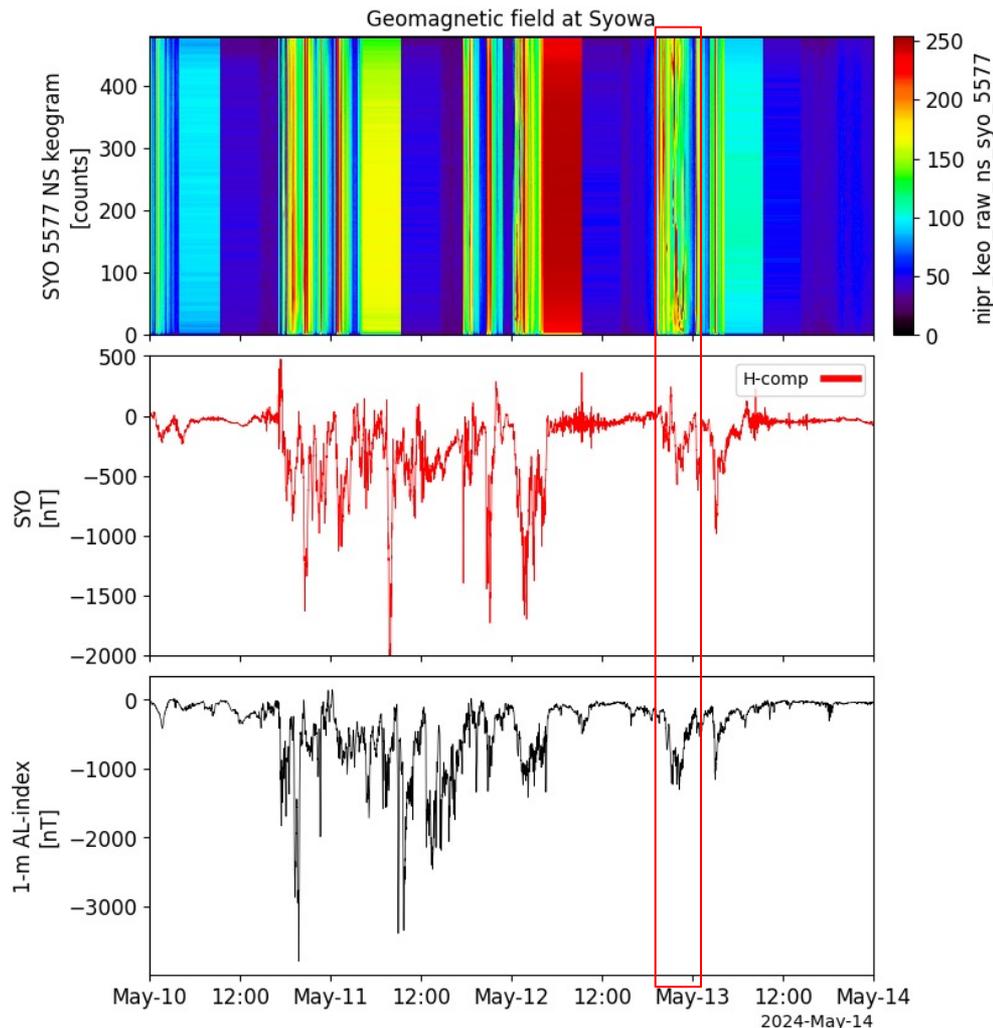
(例)

```
> from iugonet import ask_nipr
> ask_nipr( site='syo', trange=[ '2024-05-10', '2024-05-14' ] )
```

```
> pyplot.tplot_options('axis_font_size', 12)
> pyplot.timespan('2024-05-10 00:00:00', 4,
keyword='days')
```

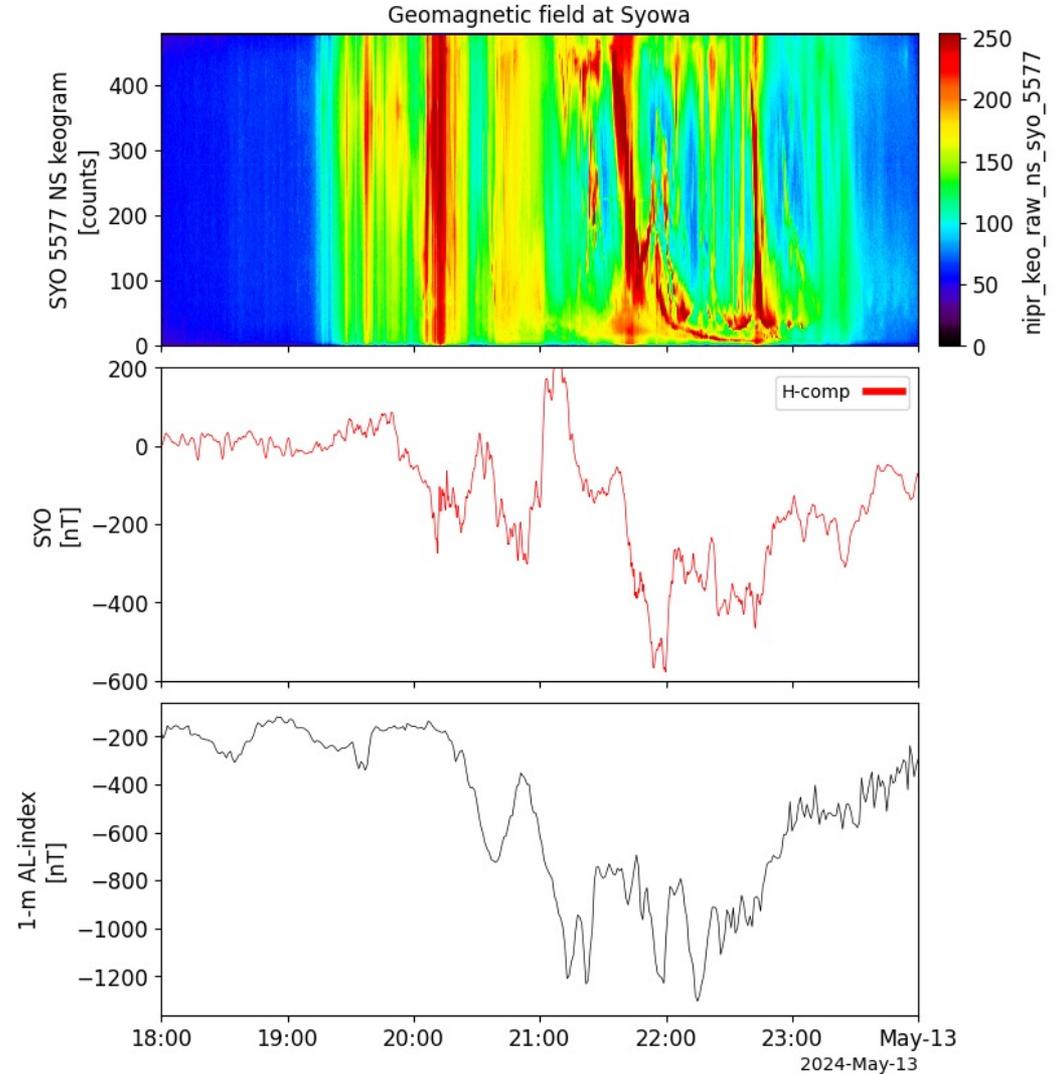
```
> pyspedas.options('nipr_mag_syo_1sec_x',
'yrange', [-2000,500])
```

```
> tplot( [ 'nipr_keo_syo_5577_ns',
'nipr_mag_syo_1sec_x', 'AL_INDEX' ] )
```



(例)

```
> pyplot.timespan('2024-05-12 18:00:00', 6,
keyword='hours')
pyspedas.options('nipr_mag_syo_1sec_x','yrange',[-
600,200])
tplot( ['nipr_keo_syo_5577_ns','nipr_mag_syo_1se
c_x','AL_INDEX'])
```



(例)

```
> pyspedas.goes.load(trange=['2024-05-05', '2024-05-07'],  
probe='16', instrument='xrs', datatype='avg1m')
```

```
> pyspedas.themis.gmag(trange=['2024-05-05', '2024-05-07'],  
sites='gua')
```

```
>
```

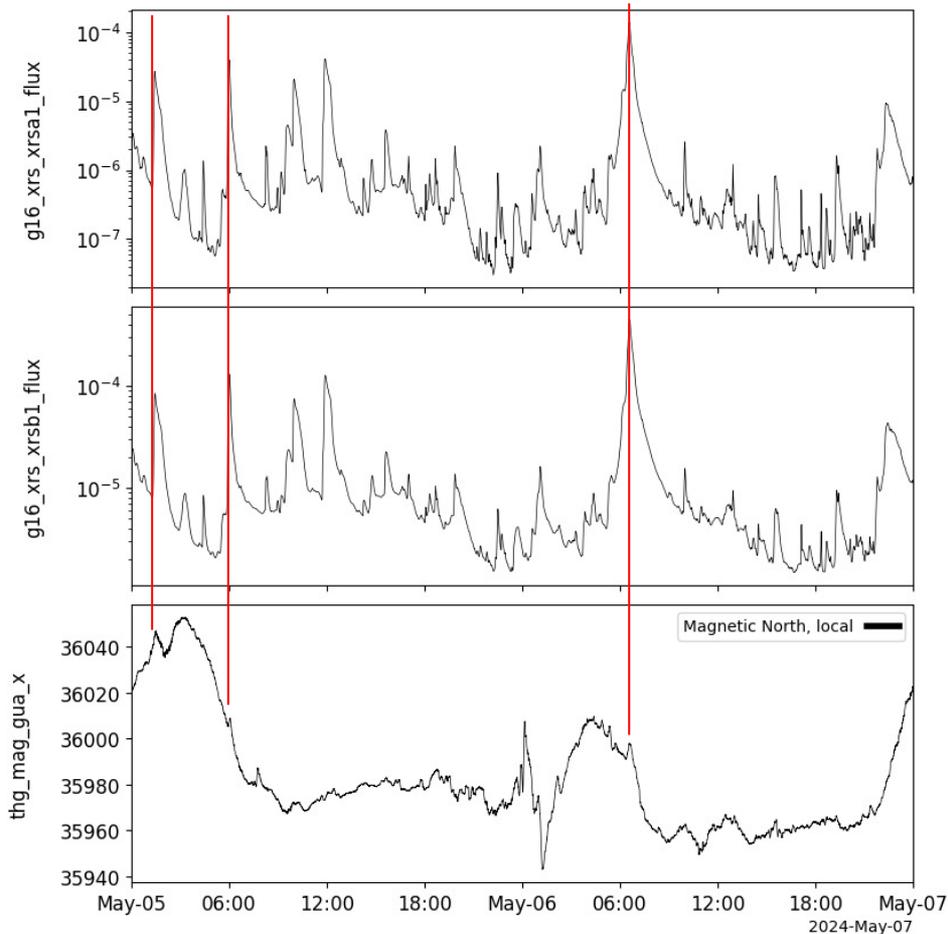
```
pytplot.options(['g16_xrs_xrsa1_flux', 'g16_xrs_xrsb1_flux'],  
'ylog', True)
```

```
> pytplot.tplot_options('title', '')
```

```
> pyspedas.split_vec('thg_mag_gua')
```

```
> pytplot.timespan('2024-05-05 00:00:00', 2,  
keyword='days')
```

```
> tplot(['g16_xrs_xrsa1_flux', 'g16_xrs_xrsb1_flux',  
'thg_mag_gua_x'])
```



```
pytplot.tplot_save('tplot変数名', filename='保存するファイル名')  
pytplot.tplot_restore(filename='再読み込みするファイル名')
```

後でまたそのtplot変数を使いたい場合に用います。

(例)

```
> pytplot.tplot_save('nipr_mag_syo_1sec_x', filename='nipr_mag_syo_1sec_x.dat')  
nipr_mag_syo_1sec_xをカレントディレクトリに保存します。
```

```
> pytplot.del_data('nipr_mag_syo_1sec_x')  
tplot変数「nipr_mag_syo_1sec_x」を削除します。
```

```
> pytplot.tplot_names()  
tplot変数「nipr_mag_syo_1sec_x」が削除されていることを確認します。
```

```
> pytplot.tplot_restore(filename='nipr_mag_syo_1sec_x.dat')  
tplot変数「nipr_mag_syo_1sec_x」を再読み込みします。
```

```
> pytplot.tplot_names()  
tplot変数「nipr_mag_syo_1sec_x」がロードされています。
```

```
data=pytplot.get_data('tplot変数名')
```

データ配列を入れる変数

ロードしたtplot変数を取り出して、pythonで自由に解析したい場合に使います。

(例)

```
> data=pytplot.get_data('nipr_mag_syo_1sec')
```

tplot変数からデータを取り出し、変数dataに格納します。

```
> time=data[0]
```

```
> ydata=data[1]
```

配列の0番目は日時、1番目は地磁気データです。

```
> pyspedas.time_string(time)
```

```
['2024-05-10 00:00:00.000000',
```

```
'2024-05-10 00:00:01.000000',
```

```
'2024-05-10 00:00:02.000000',
```

```
> ydata
```

```
array([[ -20.21408081, -110.0769043 ,  9.52148438],
```

```
       [ -20.27130127, -110.17990112,  9.46426392],
```

```
       [ -20.29037476, -110.29434204,  9.52911377],
```

```
       ...]
```

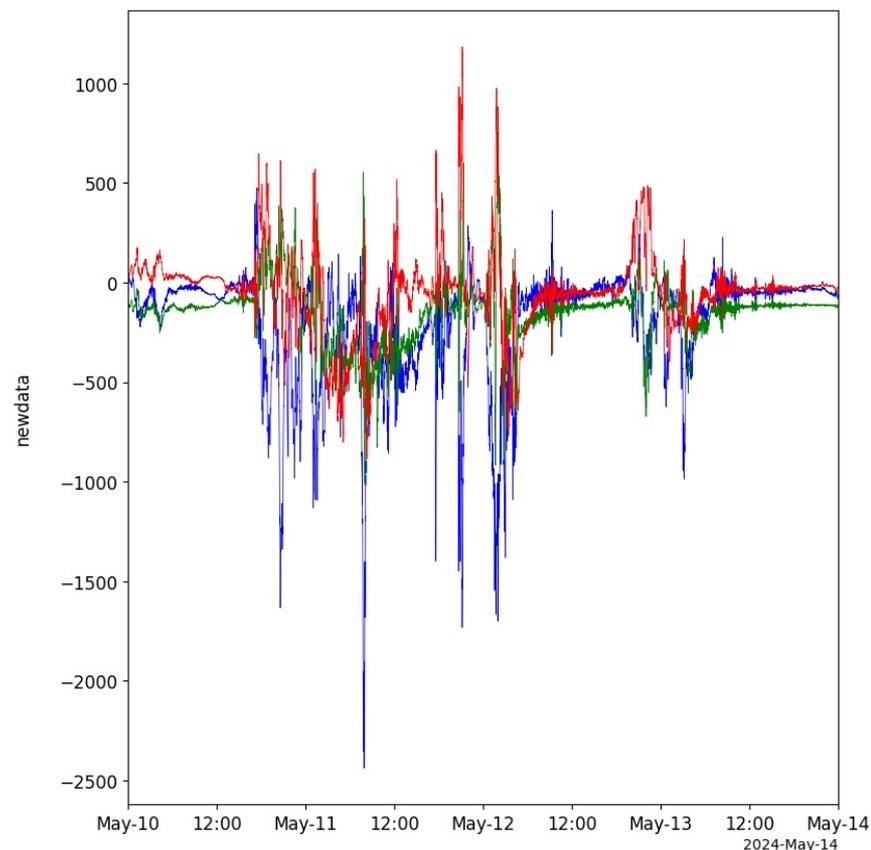
```
pytplot.store_data('新tplot変数名', data = {'x':time, 'y':data})
```

作成するtplot変数(構造体)の定義

計算した値を、tplot変数に格納するときに使います。

(例)

- > `pytplot.store_data('newdata', data={'x':time, 'y':ydata})`
変数time, ydataをtplot変数「newdata」に格納します。
- > `pytplot.timespan('2024-05-10 00:00:00', 4, keyword='days')`
- > `tplot ('newdata')`



`pyspedas.tdpwrspc('tplot変数', nboxpoints=値, nshiftpoints=値)`

FFTをかける
データ点数

時間方向にシフトする
データ点数

変動の周波数を算出する場合に使います。

(例)

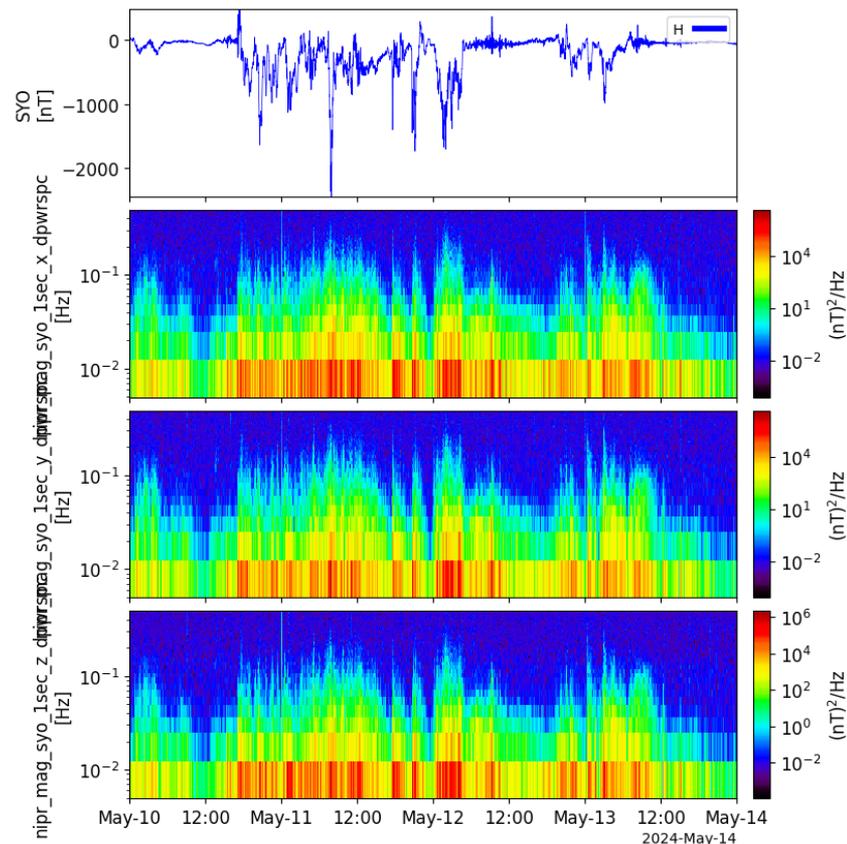
```
> pyspedas.tdpwrspc('nopr_mag_syo_1sec', nboxpoints=256,
nshiftpoints=128)
```

成分ごとにスペクトルのtplot変数が生成されます

```
> tplot(['nopr_mag_syo_1sec_x',
'nopr_mag_syo_1sec_?_dpwrspc'])
```

?は任意の一字のワイルドカードです

*は任意の文字数のワイルドカードです



ダウンロード元URL: <https://github.com/iugonet/pyudas/tree/koushu2024>

pyUDAS 講習会版に含まれているロード関数:

※開発途中であり、まだバグを含んでいる可能性が高いことにご留意ください。

※今年度中に複数のロード関数を追加していきます。

No.	データの種類	ロード関数名
1	極地研・全天イメージャデータ	asi_nipr
2	極地研・全天イメージャ・ケオグラムデータ	ask_nipr
3	極地研・フラックスゲート磁力計データ	gmag_nipr
4	極地研・誘導磁力計データ	gmag_nipr_induction
5	EISCATレーダーデータ	eiscat
6	北大・誘導磁力計データ	elf_hokudai
7	京大WDC・地磁気・指数データ	gmag_wdc
8	九大・GCMシミュレーションデータ	kyushugcm
9	東北大・HFデータ	hf_tohokuu
10	京大rish・流星レーダーデータ	meteor_rish
11	京大rish・GPS radio occultation FSI データ	gps_ro_rish
12	京大rish・MFレーダー	mf_rish
13	京大rish・ラジオゾンデデータ	radiosonde_rish