Introduction to pySPEDAS 1.0

July 6, 2020

Eric Grimes
egrimes@igpp.ucla.edu

Feel free to email me your questions!

This presentation is being recorded

Overview

- Introduction
- Projects Supported
- Getting Started
- Examples
- Getting Help
- How to Contribute

This presentation is being recorded

Introduction

- Requires Python 3.5 or later
- Depends on pyTplot (developed at LASP)
- We're doing development on GitHub

https://github.com/spedas/pyspedas

- If you have questions about instruments supported, or the names of their functions, please see the READMEs on GitHub

This presentation is being recorded

Projects Supported

- Advanced Composition Explorer (ACE)
- Arase (ERG)
- Cluster
- Colorado Student Space Weather Experiment (CSSWE)
- Deep Space Climate Observatory (DSCOVR)
- Equator-S
- Fast Auroral Snapshot Explorer (FAST)
- Geotail
- Geostationary Operational Environmental Satellite (GOES)
- Imager for Magnetopause-to-Aurora Global Exploration (IMAGE)
- Mars Atmosphere and Volatile Evolution (MAVEN)
- Magnetic Induction Coil Array (MICA)
- Magnetospheric Multiscale (MMS)
- OMNI
- Polar Orbiting Environmental Satellites (POES)
- Polar
- Parker Solar Probe (PSP)
- Van Allen Probes (RBSP)
- Solar Terrestrial Relations Observatory (STEREO)
- Time History of Events and Macroscale Interactions during Substorms (THEMIS)
- Two Wide-Angle Imaging Neutral-Atom Spectrometers (TWINS)
- Ulysses
- Wind

This presentation is being recorded

Getting Started

- Installing Anaconda
- Virtual Environments
- Installing pySPEDAS
- Local Data Directories

This presentation is being recorded

Installing Anaconda

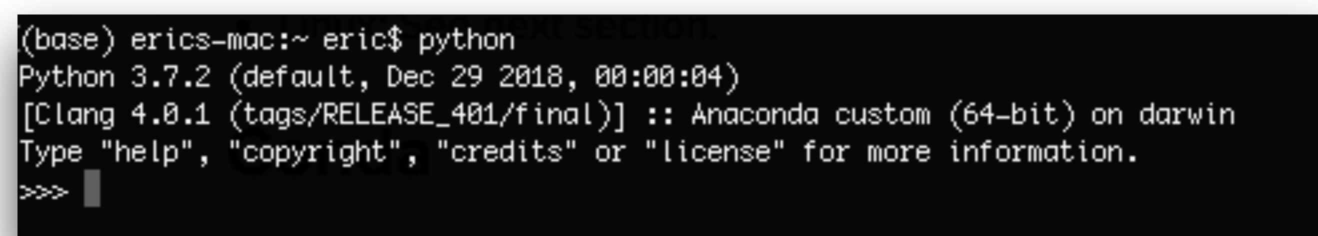
Step-by-step instructions for installing Anaconda can be found at:

- macOS
  - https://docs.anaconda.com/anaconda/install/mac-os/
- Windows
  - https://docs.anaconda.com/anaconda/install/windows/
- Linux
  - https://docs.anaconda.com/anaconda/install/linux/

This presentation is being recorded

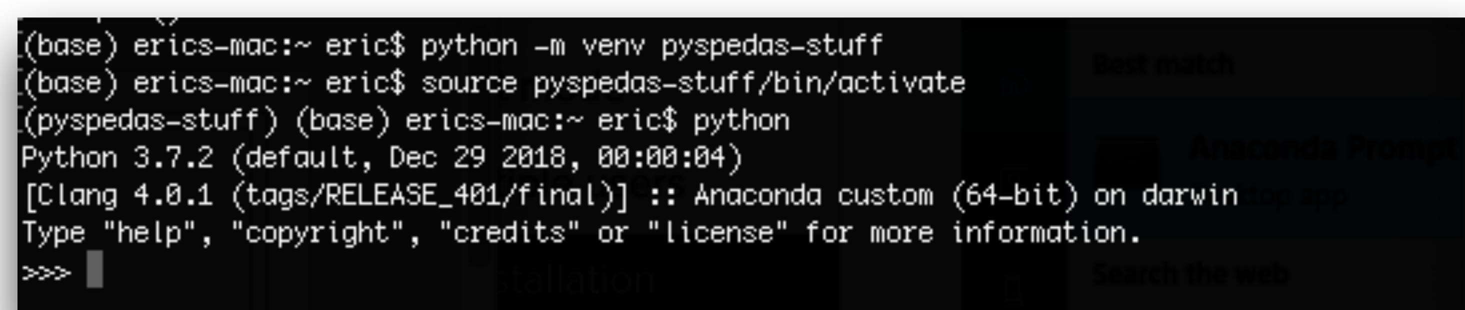
Installing Anaconda

- Once Anaconda is installed, you should be able to open Python in your terminal window by typing “python”.
  - note: your Python version will be the first line displayed



Virtual Environments

- To avoid potential dependency issues with other Python packages, it’s best to create a virtual environment in Python
- You can create a virtual environment in your terminal with:
    - python -m venv environment-name
- And enter into that virtual environment by running the ‘activate’ script with:
    - source environment-name/bin/activate (macOS and Linux)
    - .\environment-name\Scripts\activate (Windows)
- e.g.,



This presentation is being recorded

Installing pySPEDAS

- The first time you enter your virtual environment, you'll have to install pyspedas; this is as simple as:
    - pip install pyspedas
- This should go out and find all of the required libraries and install them inside the virtual environment.
- If you would like to upgrade your copy of your pySPEDAS libraries inside of your virtual environment, use:
    - pip install pyspedas --upgrade

Local Data Directories

- Your data directory can be set using the **SPEDAS_DATA_DIR** environment variable. Each mission also has its own data directory, e.g., **MMS_DATA_DIR**, **THM_DATA_DIR**, etc.

- Note: mission data directories will override the root data directory set in SPEDAS_DATA_DIR.

Importing pySPEDAS

- To get started, import pyspedas:

```
(july6tutorial) erics-mac:~ eric$ python
Python 3.7.6 (default, Jan  8 2020, 13:42:34)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyspedas
>>>
```

- You can also access the load routines by importing the mission modules:

```
>>>
>>> from pyspedas import mms, themis, cluster
>>>
```

- You can also import the instrument load routines, e.g.:

```
>>>
>>> from pyspedas.mms import fgm, fpi
>>> from pyspedas.themis import esa, sst
>>>
```

This presentation is being recorded

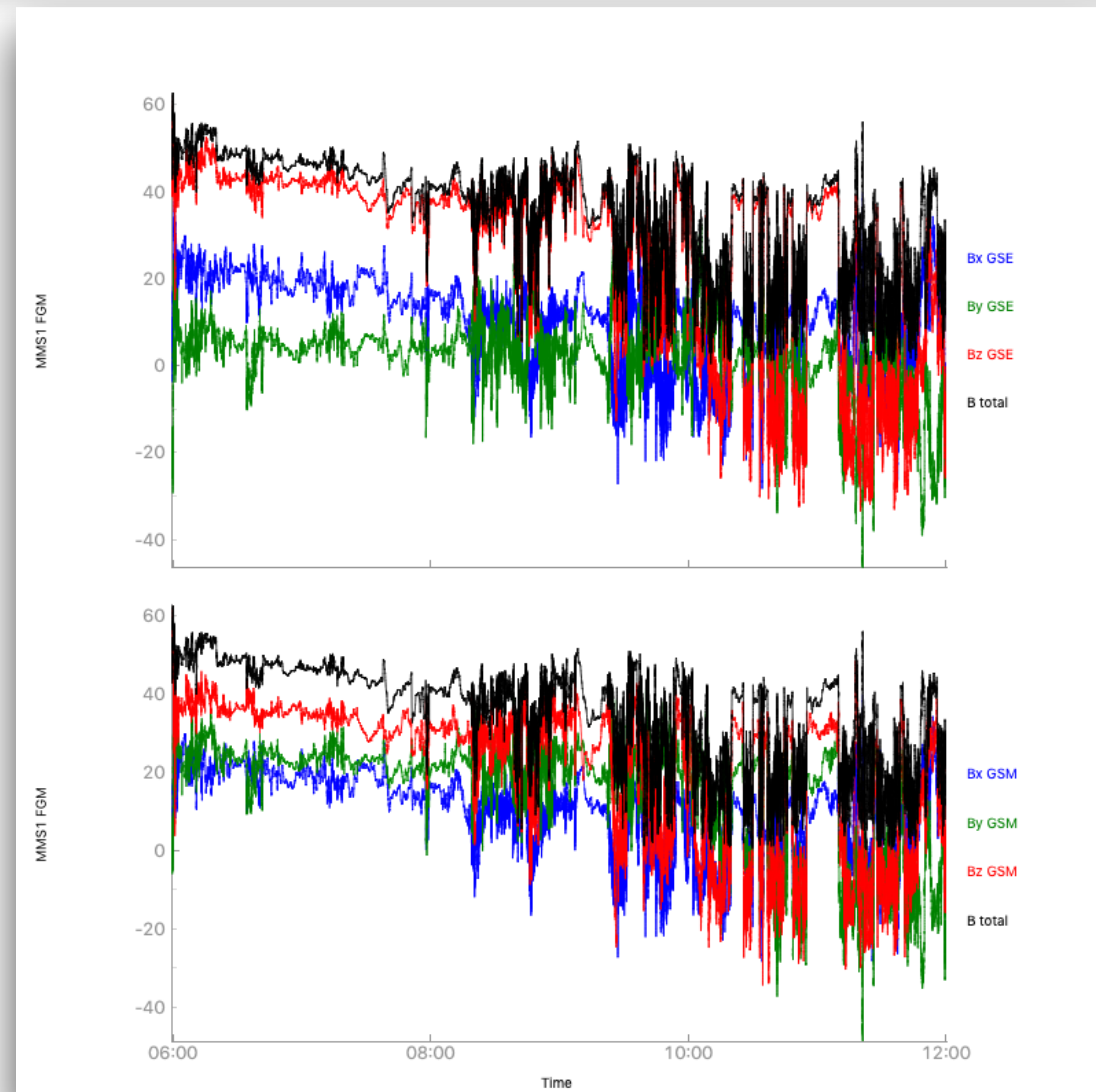
Loading MMS FGM Data

```
>>> data = pyspedas.mms.fgm(trange=['2015-10-16/6:00', '2015-10-16/12:00'], time_clip=True)
04-Jul-20 21:04:15: Loading /Volumes/data/data/mms_test_folder/mms1/fgm/srvy/l2/2015/10/mms1_fgm_srvy_l2_20151016_v4.18.0.cdf
The lengths of x and y do not match!
mms1_fgm_rdeltahalf_srvy_l2 is currently not in pytplot.
Time clip was applied to: mms1_fgm_b_gse_srvy_l2
Time clip was applied to: mms1_fgm_b_gsm_srvy_l2
Time clip was applied to: mms1_fgm_b_dmpa_srvy_l2
Time clip was applied to: mms1_fgm_b_bcs_srvy_l2
Time clip was applied to: mms1_fgm_flag_srvy_l2
Time clip was applied to: mms1_fgm_r_gse_srvy_l2
Time clip was applied to: mms1_fgm_r_gsm_srvy_l2
Time clip was applied to: mms1_fgm_hirange_srvy_l2
Time clip was applied to: mms1_fgm_bdeltahalf_srvy_l2
Time clip was applied to: mms1_fgm_stemp_srvy_l2
Time clip was applied to: mms1_fgm_etemp_srvy_l2
Time clip was applied to: mms1_fgm_mode_srvy_l2
Time clip error: No pytplot names were provided.
Loaded variables:
mms1_fgm_b_gse_srvy_l2
mms1_fgm_b_gsm_srvy_l2
mms1_fgm_b_dmpa_srvy_l2
mms1_fgm_b_bcs_srvy_l2
mms1_fgm_flag_srvy_l2
mms1_fgm_r_gse_srvy_l2
mms1_fgm_r_gsm_srvy_l2
mms1_fgm_hirange_srvy_l2
mms1_fgm_bdeltahalf_srvy_l2
mms1_fgm_stemp_srvy_l2
mms1_fgm_etemp_srvy_l2
mms1_fgm_mode_srvy_l2
mms1_fgm_rdeltahalf_srvy_l2
>>>
```

This presentation is being recorded

Plotting MMS FGM Data

```
>>>
>>> from pytplot import tplot
>>>
>>> tplot(['mms1_fgm_b_gse_srvy_l2', 'mms1_fgm_b_gsm_srvy_l2'])
>>> _
```



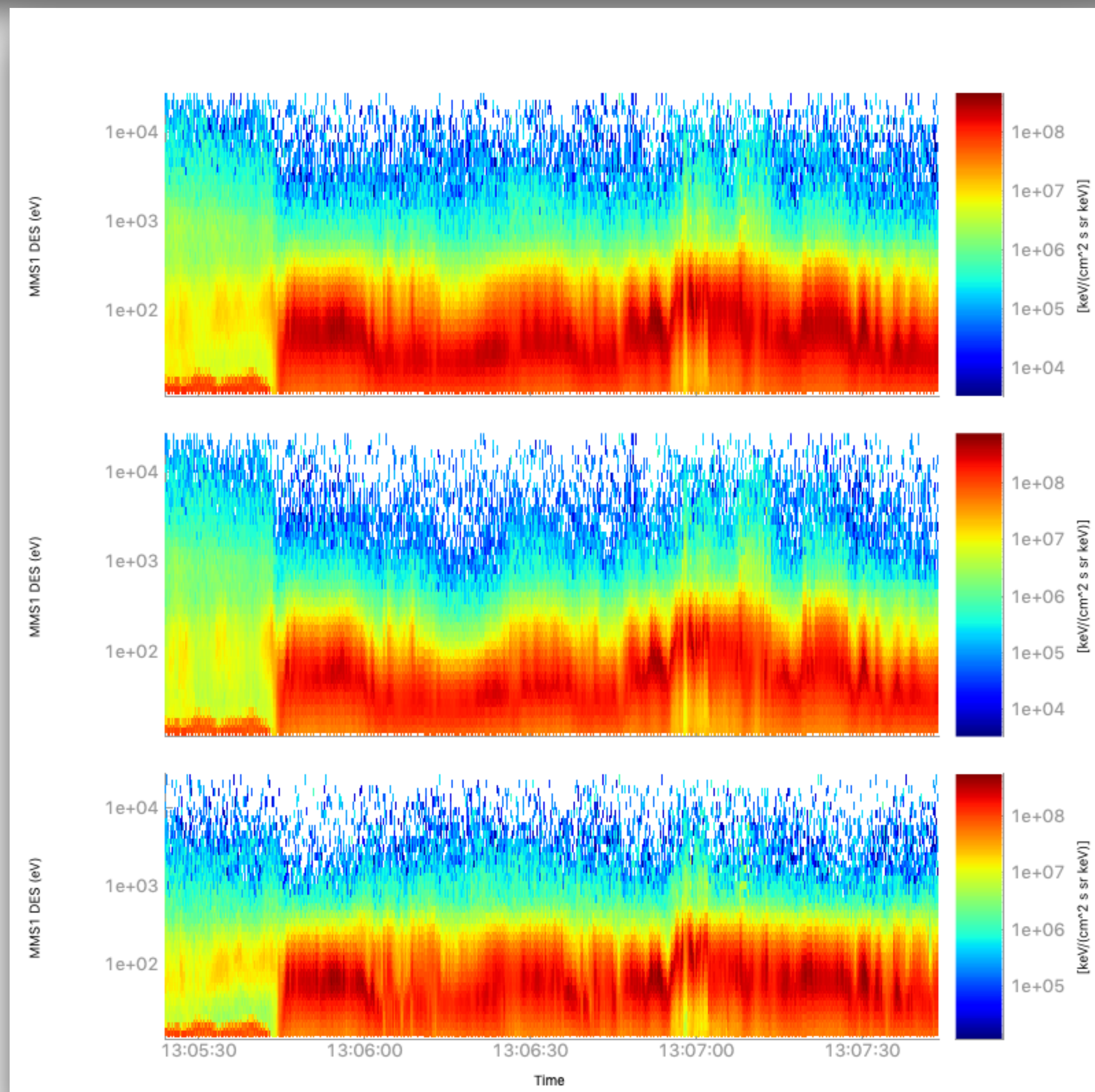
This presentation is being recorded

Loading MMS FPI Data

```
>>>
>>> pyspedas.mms.fpi(trange=['2015-10-16/13:06', '2015-10-16/13:07'], data_rate='brst', datatype='des-moms')
04-Jul-20 21:14:49: Downloading mms1_fpi_brst_l2_des-moms_20151016130524_v3.3.0.cdf to /Volumes/data/data/mms_test_folder/mms1/fpi/brst/l2/des-moms/2015/10/16
Loaded variables:
mms1_des_errorflags_brst
mms1_des_compressionloss_brst
mms1_des_startdelphi_count_brst
mms1_des_startdelphi_angle_brst
mms1_des_sector_despinp_brst
mms1_des_pitchangdist_lowen_brst
mms1_des_pitchangdist_miden_brst
mms1_des_pitchangdist_highen_brst
mms1_des_energyspectr_px_brst
mms1_des_energyspectr_mx_brst
mms1_des_energyspectr_py_brst
mms1_des_energyspectr_my_brst
mms1_des_energyspectr_pz_brst
mms1_des_energyspectr_mz_brst
mms1_des_energyspectr_par_brst
mms1_des_energyspectr_anti_brst
mms1_des_energyspectr_perp_brst
mms1_des_energyspectr_omni_brst
mms1_des_numberdensity_brst
mms1_des_densityextrapolation_low_brst
mms1_des_densityextrapolation_high_brst
mms1_des_bulkv_dbcs_brst
mms1_des_bulkv_spintone_dbcs_brst
mms1_des_bulkv_gse_brst
mms1_des_bulkv_spintone_gse_brst
mms1_des_prestensor_dbcs_brst
mms1_des_prestensor_gse_brst
mms1_des_temptensor_dbcs_brst
mms1_des_temptensor_gse_brst
mms1_des_heatq_dbcs_brst
mms1_des_heatq_gse_brst
mms1_des_temppara_brst
mms1_des_tempperp_brst
```

Plotting MMS FPI Data

```
>>>
>>> tplot(['mms1_des_energyspectr_omni_brst', 'mms1_des_energyspectr_perp_brst', 'mms1_des_energyspectr_par_brst'])
>>>
```



This presentation is being recorded

Finding the Loaded Variables

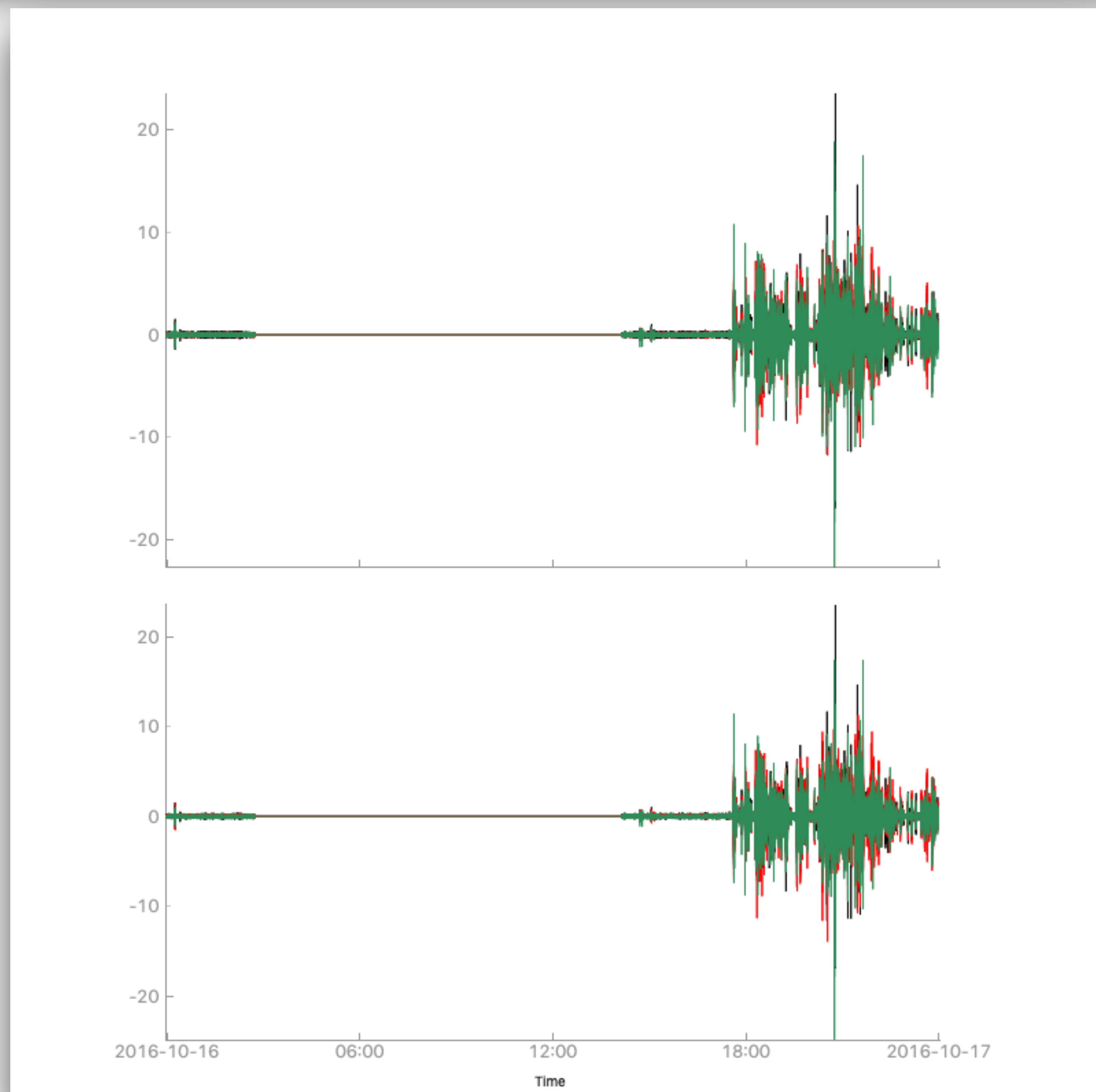
```
>>> from pytplot import tplot_names
>>> tplot_names()
0 : mms1_fgm_b_gse_srvy_l2
1 : mms1_fgm_b_gsm_srvy_l2
2 : mms1_fgm_b_dmpa_srvy_l2
3 : mms1_fgm_b_bcs_srvy_l2
4 : mms1_fgm_flag_srvy_l2
5 : mms1_fgm_r_gse_srvy_l2
6 : mms1_fgm_r_gsm_srvy_l2
7 : mms1_fgm_hirange_srvy_l2
8 : mms1_fgm_bdeltahalf_srvy_l2
9 : mms1_fgm_stemp_srvy_l2
10 : mms1_fgm_etemp_srvy_l2
11 : mms1_fgm_mode_srvy_l2
12 : mms1_des_errorflags_brst
13 : mms1_des_compressionloss_brst
14 : mms1_des_startdelphi_count_brst
15 : mms1_des_startdelphi_angle_brst
16 : mms1_des_sector_despinp_brst
17 : mms1_des_pitchangdist_lowen_brst
18 : mms1_des_pitchangdist_miden_brst
19 : mms1_des_pitchangdist_highen_brst
20 : mms1_des_energyspectr_px_brst
21 : mms1_des_energyspectr_mx_brst
22 : mms1_des_energyspectr_py_brst
23 : mms1_des_energyspectr_my_brst
24 : mms1_des_energyspectr_pz_brst
25 : mms1_des_energyspectr_mz_brst
26 : mms1_des_energyspectr_par_brst
27 : mms1_des_energyspectr_anti_brst
28 : mms1_des_energyspectr_perp_brst
29 : mms1_des_energyspectr_omni_brst
30 : mms1_des_numberdensity_brst
```

Loading THEMIS SCM Data

```
>>>
>>> pyspedas.themis.scm(probe='d', trange=['2016-10-16', '2016-10-17'])
05-Jul-20 11:36:30: Downloading remote index: http://themis.ssl.berkeley.edu/data/themis/thd/l2/scm/2016/
05-Jul-20 11:36:30: File is current: /Volumes/data/data/themis/thd/l2/scm/2016/thd_l2_scm_20161016_v01.cdf
['thd_scf_btotal', 'thd_scf_gse', 'thd_scf_gsm', 'thd_scf_dsl', 'thd_scp_btotal', 'thd_scp_gse', 'thd_scp_gsm', 'thd_scp_dsl', 'thd_scw_btotal',
 'thd_scw_gse', 'thd_scw_gsm', 'thd_scw_dsl']
>>>
```

Plotting THEMIS SCM Data

```
>>>
>>> tplot(['thd_scf_gse', 'thd_scf_gsm'])
>>>
```



This presentation is being recorded

Updating Plot Metadata

```
>>>
>>> from pytplot import options
>>>
```

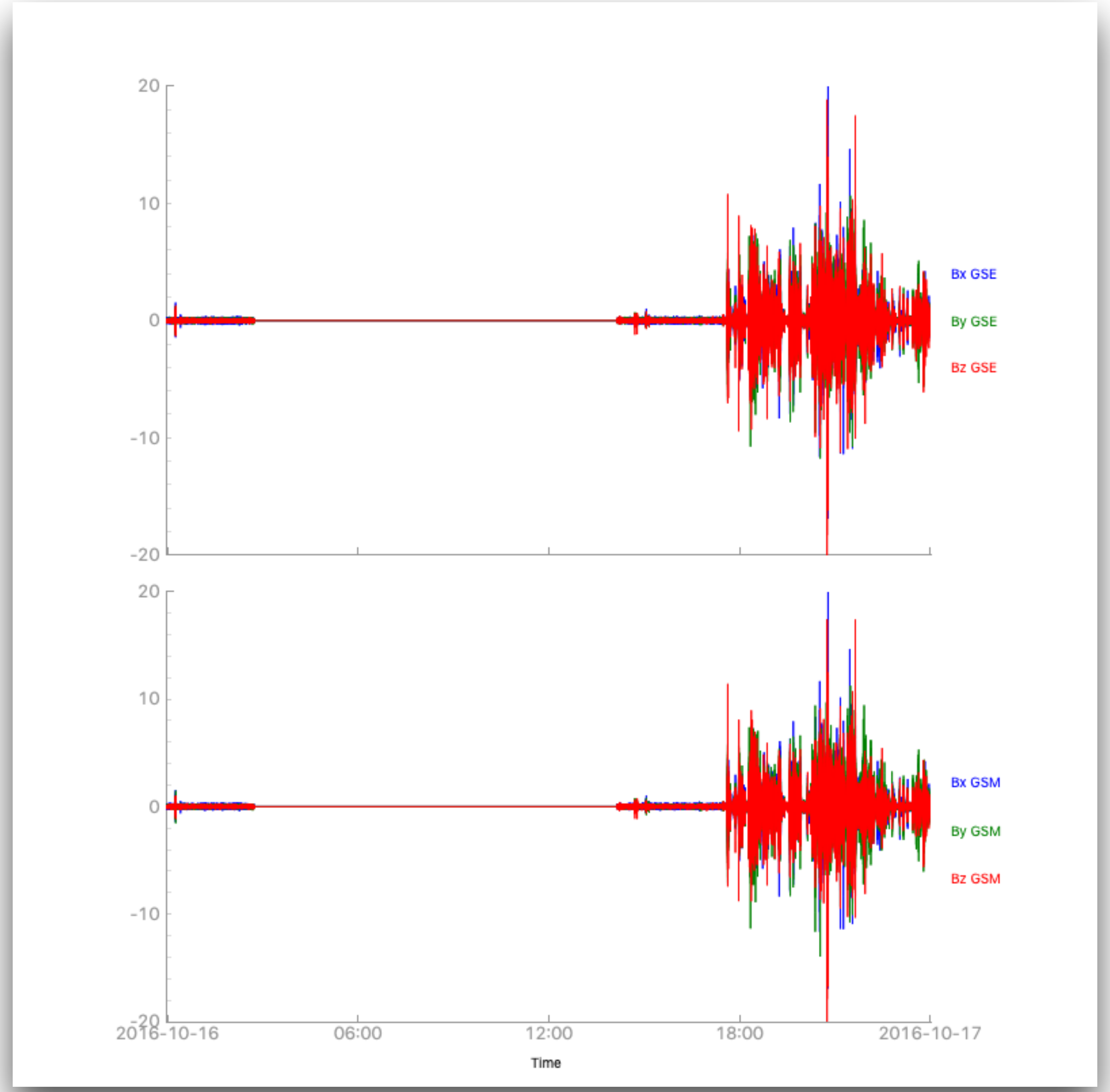
```
Help on function options in module pytplot.options:

options(name, option=None, value=None, opt_dict=None)
    This function allows the user to set a large variety of options for individual plots.

    Parameters:
        name : str
            Name or number of the tplot variable
        option : str
            The name of the option.  See section below.
        value : str/int/float/list
            The value of the option.  See section below.
        dict : dict
            This can be a dictionary of option:value pairs.  Option and value
            will not be needed if this dictionary item is supplied.

    Options:
        ================== ==========   =====
        Options            Value type   Notes
        ================== ==========   =====
        Color              str/list     Red, Orange, Yellow, Green, Blue, etc.
        Colormap           str/list     https://matplotlib.org/examples/color/colormaps_reference.html.
        Spec               int          1 sets the Tplot Variable to spectrogram mode, 0 reverts.
        Alt                int          1 sets the Tplot Variable to altitude plot mode, 0 reverts.
        Map                int          1 sets the Tplot Variable to latitude/longitude mode, 0 reverts.
        link               list         Allows a user to reference one tplot variable to another.
        ylog               int          1 sets the y axis to log scale, 0 reverts.
        zlog               int          1 sets the z axis to log scale, 0 reverts (spectrograms only).
        legend_names       list         A list of strings that will be used to identify the lines.
        xlog_slice         bool         Sets x axis on slice plot to log scale if True.
        ylog               bool         Set y axis on main plot window to log scale if True.
        ylog_slice         bool         Sets y axis on slice plot to log scale if True.
        zlog               bool         Sets z axis on main plot window to log scale if True.
        line_style         str          scatter (to make scatter plots), or solid_line, dot, dash, dash_dot, dash_dot_dot_dot, long_dash.
        char_size          int          Defines character size for plot labels, etc.
        name               str          The title of the plot.
        panel_size         flt          Number between (0,1], representing the percent size of the plot.
        basemap            str          Full path and name of a background image for "Map" plots.
        alpha              flt          Number between [0,1], gives the transparancy of the plot lines.
        thick              flt          Sets plot line width.
        yrange             flt list     Two numbers that give the y axis range of the plot.
        zrange             flt list     Two numbers that give the z axis range of the plot.
        xrange_slice       flt list     Two numbers that give the x axis range of spectrogram slicing plots.
        yrange_slice       flt list     Two numbers that give the y axis range of spectrogram slicing plots.
        ytitle             str          Title shown on the y axis.
        ztitle             str          Title shown on the z axis.  Spec plots only.
        ysubtitle          str          Subtitle shown on the y axis.
        zsubtitle          str          Subtitle shown on the z axis.  Spec plots only.
```

This presentation is being recorded

Updating Plot Metadata

```
>>> options('thd_scf_gsm', 'legend_names', ['Bx GSM', 'By GSM', 'Bz GSM'])
>>> options('thd_scf_gse', 'legend_names', ['Bx GSE', 'By GSE', 'Bz GSE'])
>>>
>>> options('thd_scf_gsm', 'Color', ['blue', 'green', 'red'])
>>> options('thd_scf_gse', 'Color', ['blue', 'green', 'red'])
>>>
>>> options('thd_scf_gsm', 'yrange', [-20, 20])
>>> options('thd_scf_gse', 'yrange', [-20, 20])
>>>
>>> tplot(['thd_scf_gse', 'thd_scf_gsm'])
```

Updating Plot Metadata



This presentation is being recorded

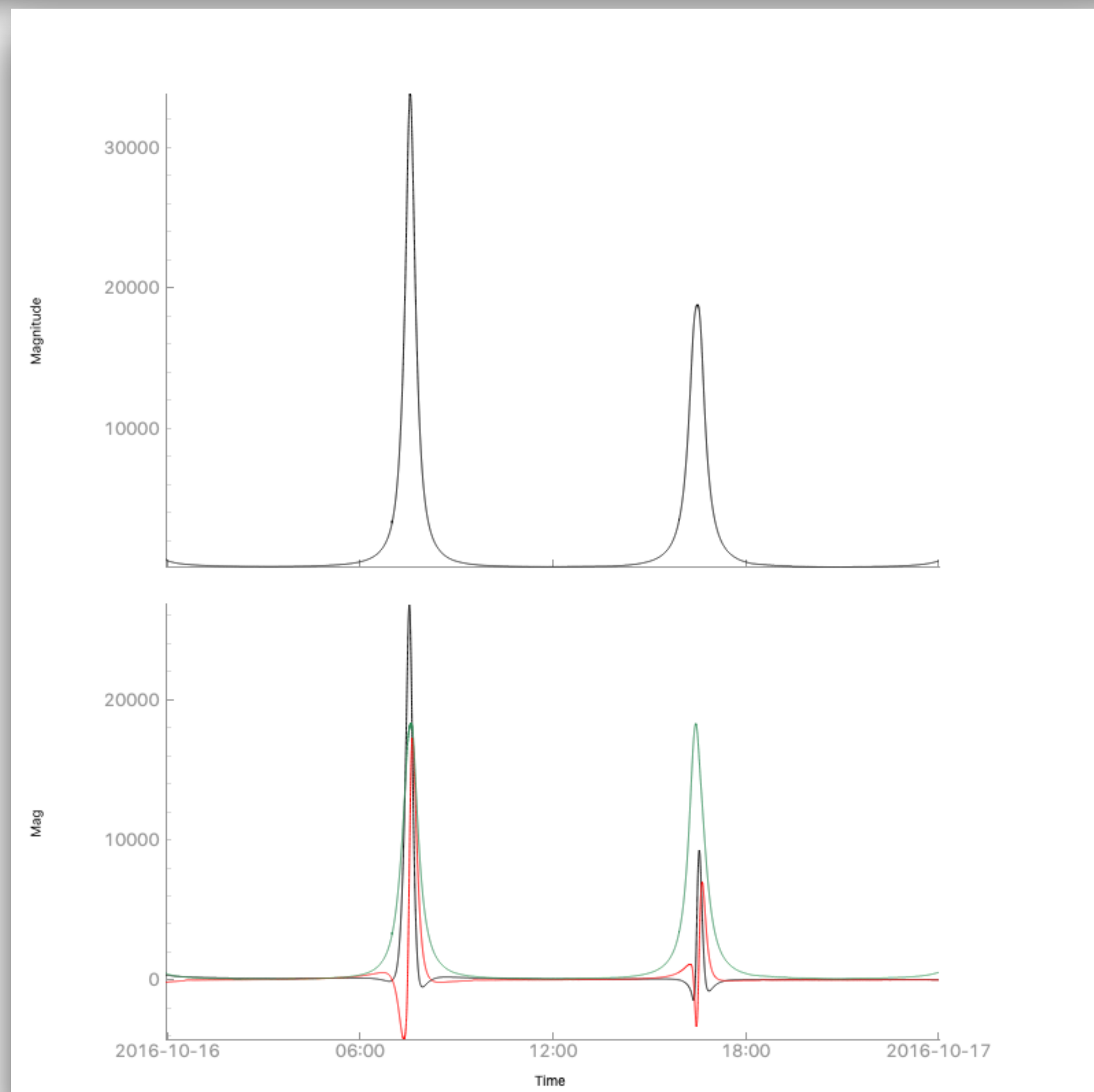
Loading RBSP Data

```
>>>
>>> pyspedas.rbsp.emfisis(trange=['2016-10-16', '2016-10-17'])
05-Jul-20 11:49:40: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/rbsp/rbspa/l3/emfisis/magnetometer/4sec/sm/2016/
05-Jul-20 11:49:41: File is current: /Volumes/data/data/rbsp/rbspa/l3/emfisis/magnetometer/4sec/sm/2016/rbsp-a_magnetometer_4sec-sm_emfisis-l3_2
0161016_v1.6.1.cdf
['Mag', 'Magnitude', 'delta', 'lambda', 'rms', 'coordinates']
>>>
```

Plotting RBSP Data

```
>>>
>>> tplot(['Magnitude', 'Mag'])
>>>
```

• https://do



This presentation is being recorded

Finding Load Routine Options

```
Help on function mageis in module pyspedas.rbsp:

mageis(trange=['2015-11-5', '2015-11-6'], probe='a', datatype='', level='l3', rel='rel04', suffix='', get_support_data=False, varformat=None, do
wnloadonly=False, notplot=False, no_update=False, time_clip=False)
    This function loads data from the Energetic Particle, Composition, and Thermal Plasma Suite (ECT)

    Parameters:
        trange : list of str
            time range of interest [starttime, endtime] with the format
            'YYYY-MM-DD','YYYY-MM-DD'] or to specify more or less than a day
            ['YYYY-MM-DD/hh:mm:ss','YYYY-MM-DD/hh:mm:ss']

        probe: str or list of str
            Spacecraft probe name ('a' or 'b'); default: a

        datatype: str
            Data type; Valid options:

        suffix: str
            The tplot variable names will be given this suffix.  By default,
            no suffix is added.

        get_support_data: bool
            Data with an attribute "VAR_TYPE" with a value of "support_data"
            will be loaded into tplot.  By default, only loads in data with a
            "VAR_TYPE" attribute of "data".

        varformat: str
            The file variable formats to load into tplot.  Wildcard character
            "*" is accepted.  By default, all variables are loaded in.

        downloadonly: bool
            Set this flag to download the CDF files, but not load them into
            tplot variables

        notplot: bool
            Return the data in hash tables instead of creating tplot variables

        no_update: bool
            If set, only load data from your local cache

        time_clip: bool
            Time clip the variables to exactly the range specified in the trange keyword

    Returns:
        List of tplot variables created.
```

This presentation is being recorded

Loading RBSP Data

```
>>>
>>> files = pyspedas.rbsp.mageis(trange=['2018-11-5', '2018-11-6'], probe=['a', 'b'], downloadonly=True)
05-Jul-20 11:54:52: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/rbsp/rbspa/l3/ect/mageis/sectors/rel04/2018/
05-Jul-20 11:54:53: File is current: /Volumes/data/data/rbsp/rbspa/l3/ect/mageis/sectors/rel04/2018/rbspa_rel04_ect-mageis-l3_20181105_v8.1.0.cdf
05-Jul-20 11:54:53: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/rbsp/rbspb/l3/ect/mageis/sectors/rel04/2018/
05-Jul-20 11:54:54: File is current: /Volumes/data/data/rbsp/rbspb/l3/ect/mageis/sectors/rel04/2018/rbspb_rel04_ect-mageis-l3_20181105_v8.1.0.cdf
>>>
>>> files
['/Volumes/data/data/rbsp/rbspa/l3/ect/mageis/sectors/rel04/2018/rbspa_rel04_ect-mageis-l3_20181105_v8.1.0.cdf', '/Volumes/data/data/rbsp/rbspb/l3/ect
/mageis/sectors/rel04/2018/rbspb_rel04_ect-mageis-l3_20181105_v8.1.0.cdf']
>>> _
```

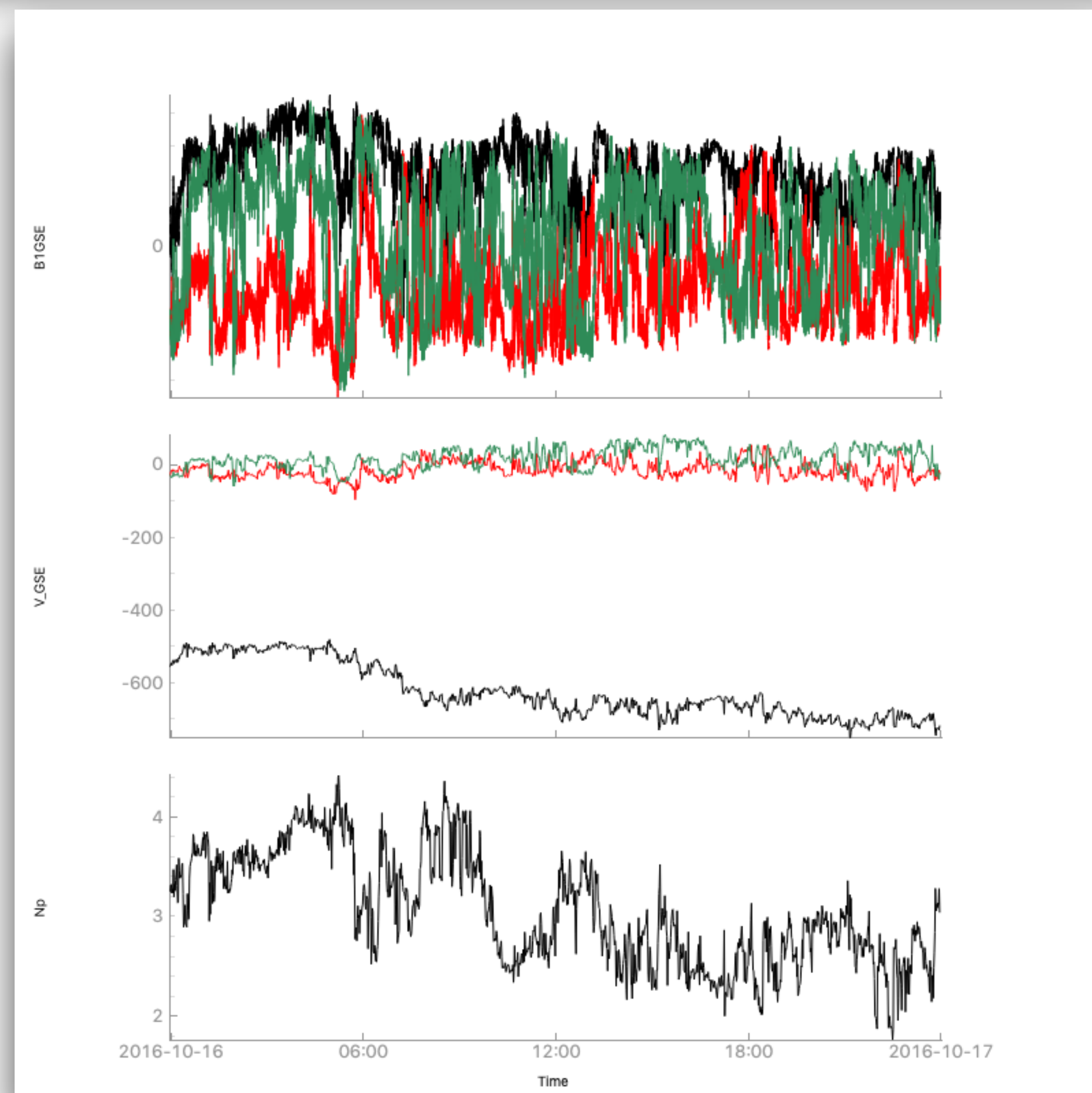
Loading DSCOVR Data

```
>>>
>>> from pyspedas import dscovr
>>>
>>> dscovr.mag(trange=['2016-10-16', '2016-10-17'])
05-Jul-20 12:17:11: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/dscovr/h0/mag/2016/
05-Jul-20 12:17:12: File is current: /Volumes/data/data/dscovr/h0/mag/2016/dscovr_h0_mag_20161016_v01.cdf
['B1F1', 'B1SDF1', 'B1GSE', 'B1SDGSE', 'B1RTN', 'B1SDRTN']
>>>
>>> dscovr.fc(trange=['2016-10-16', '2016-10-17'])
05-Jul-20 12:17:24: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/dscovr/h1/faraday_cup/2016/
05-Jul-20 12:17:25: File is current: /Volumes/data/data/dscovr/h1/faraday_cup/2016/dscovr_h1_fc_20161016_v06.cdf
['V_GSE', 'THERMAL_SPD', 'Np', 'THERMAL_TEMP']
>>>
```

This presentation is being recorded

Plotting DSCOVR Data

```
>>>
>>> tplot(['B1GSE', 'V_GSE', 'Np'])
>>>
```



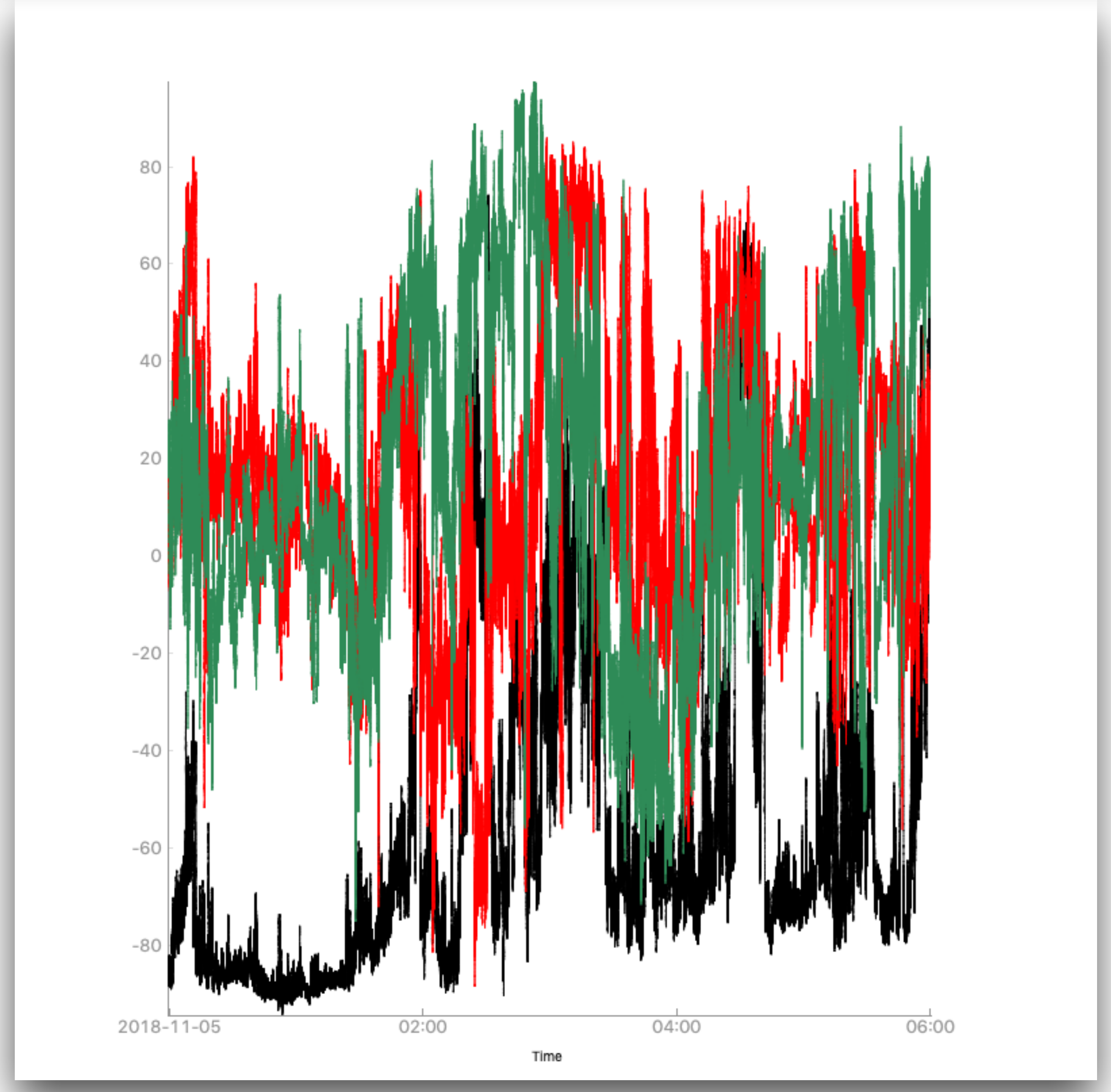
This presentation is being recorded

Loading PSP Data

```
>>>
>>> fields_vars = pyspedas.psp.fields(trange=['2018-11-5', '2018-11-5/06:00'], datatype='mag_rtn', level='l2')
05-Jul-20 12:45:46: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/psp/fields/l2/mag_rtn/2018/
05-Jul-20 12:45:48: File is current: /Volumes/data/data/psp/fields/l2/mag_rtn/2018/psp_fld_l2_mag_rtn_2018110500_v01.cdf
>>>
>>> fields_vars
['psp_fld_l2_mag_RTN']
>>>
```

Plotting PSP Data

```
>>>
>>> tplot('psp_fld_l2_mag_RTN')
>>>
```



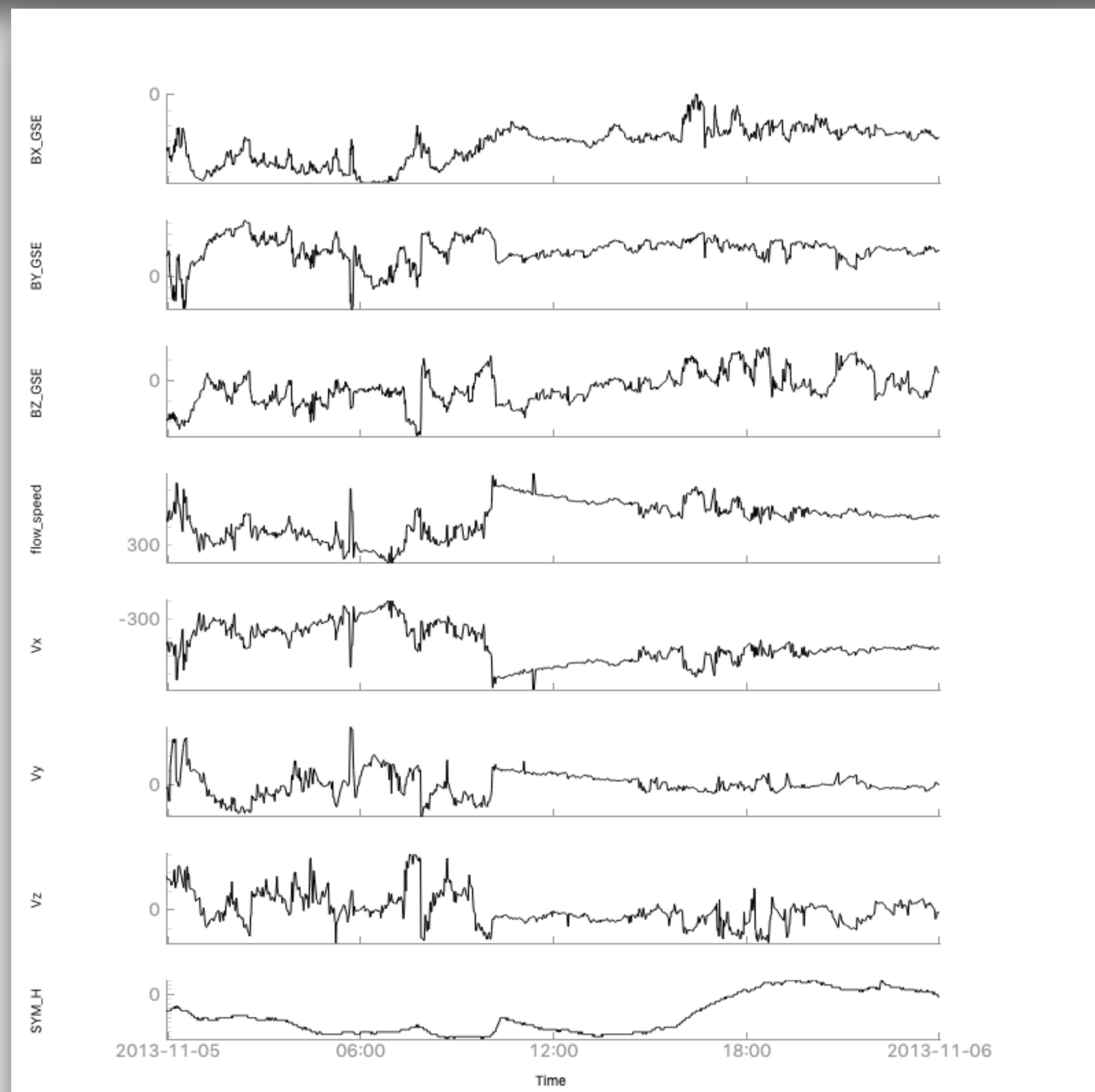
This presentation is being recorded

Loading OMNI Data

```
>>> pyspedas.omni.data(trange=['2013-11-5', '2013-11-6'])
05-Jul-20 12:20:21: Downloading remote index: https://spdf.sci.gsfc.nasa.gov/pub/data/omni/omni_cdaweb/hro2_1min/2013/
05-Jul-20 12:20:22: File is current: /Volumes/data/data/omni/hro2_1min/2013/omni_hro2_1min_20131101_v01.cdf
Time clip was applied to: IMF
Time clip was applied to: PLS
Time clip was applied to: IMF_PTS
Time clip was applied to: PLS_PTS
Time clip was applied to: percent_interp
Time clip was applied to: Timeshift
Time clip was applied to: RMS_Timeshift
Time clip was applied to: RMS_phase
Time clip was applied to: Time_btwn_obs
Time clip was applied to: F
Time clip was applied to: BX_GSE
Time clip was applied to: BY_GSE
Time clip was applied to: BZ_GSE
Time clip was applied to: BY_GSM
Time clip was applied to: BZ_GSM
Time clip was applied to: RMS_SD_B
Time clip was applied to: RMS_SD_fld_vec
Time clip was applied to: flow_speed
Time clip was applied to: Vx
Time clip was applied to: Vy
Time clip was applied to: Vz
Time clip was applied to: proton_density
Time clip was applied to: T
Time clip was applied to: NaNp_Ratio
Time clip was applied to: Pressure
Time clip was applied to: E
Time clip was applied to: Beta
Time clip was applied to: Mach_num
Time clip was applied to: Mgs_mach_num
```

Plotting OMNI Data

```
>>>
>>> tplot(['BX_GSE', 'BY_GSE', 'BZ_GSE', 'flow_speed', 'Vx', 'Vy', 'Vz', 'SYM_H'])
>>> _
```



This presentation is being recorded

Working with the Data Values

```
>>> from pytplot import get_data
>>>
>>> bx_times, bx_values = get_data('BX_GSE')
>>>
>>> bx_times
array([1.38360960e+09, 1.38360966e+09, 1.38360972e+09, ...,
       1.38369588e+09, 1.38369594e+09, 1.38369600e+09])
>>>
>>> bx_values
array([-3.47, -3.65, -3.52, ..., -2.8 , -2.77, -2.77], dtype=float32)
>>> _
```

This presentation is being recorded

Working with the Times

```
>>> from pyspedas import time_string
>>>
>>> time_string(bx_times[0:5])
['2013-11-05 00:00:00.000000', '2013-11-05 00:01:00.000000', '2013-11-05 00:02:00.000000', '2013-11-05 00:03:00.000000', '2013-11-05 00:04:00.000000']
>>>
>>> from pyspedas import time_double
>>>
>>> time_double(time_string(bx_times[0:5]))
[1383609600.0, 1383609660.0, 1383609720.0, 1383609780.0, 1383609840.0]
>>>
```

This presentation is being recorded

Working with Energy Spectra

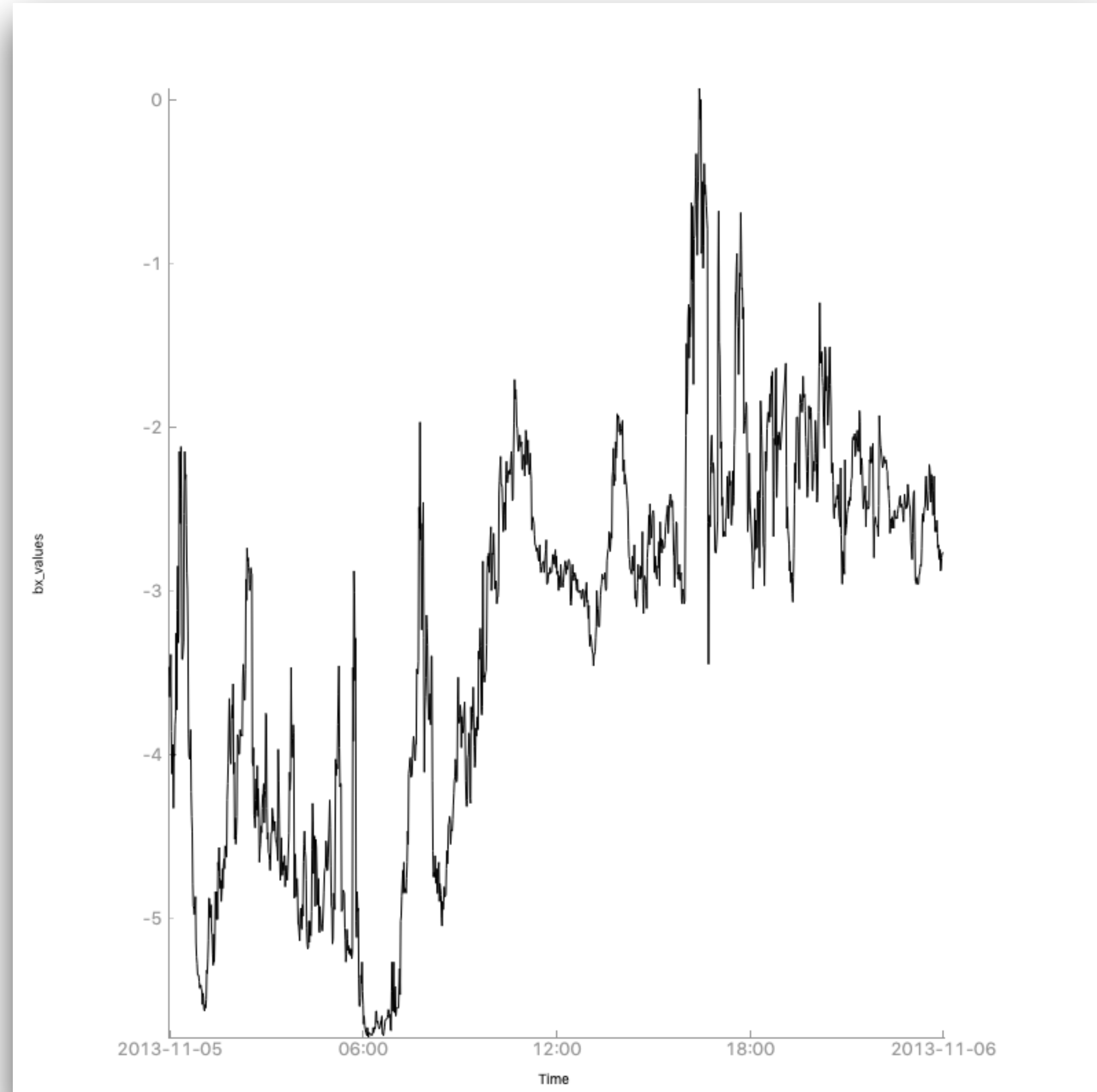
```
>>>
>>> times, data, energies = get_data('mms1_des_energyspectr_omni_brst')
>>>
>>> times
array([1.44500016e+09, 1.44500016e+09, 1.44500016e+09, ...,
       1.44500098e+09, 1.44500098e+09, 1.44500098e+09])
>>>
>>> data
array([[5.9108852e+07, 6.9532248e+07, 9.2644320e+07, ..., 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00],
       [6.3552028e+07, 8.2698200e+07, 1.1583492e+08, ..., 3.3417566e+04,
        3.6519996e+04, 0.0000000e+00],
       [5.9101072e+07, 6.9920048e+07, 9.6920600e+07, ..., 6.9254941e+03,
        2.9787959e+04, 0.0000000e+00],
       ...,
       [5.7632320e+07, 7.7443400e+07, 1.1962530e+08, ..., 1.3054982e+06,
        2.1576712e+05, 0.0000000e+00],
       [4.9851772e+07, 6.6963148e+07, 9.6925824e+07, ..., 5.0306168e+04,
        0.0000000e+00, 5.7740699e+04],
       [5.8684188e+07, 8.0578224e+07, 1.2334721e+08, ..., 0.0000000e+00,
        0.0000000e+00, 0.0000000e+00]], dtype=float32)
>>>
>>> energies
array([[1.096000e+01, 1.405000e+01, 1.802000e+01, ..., 1.481758e+04,
        1.899954e+04, 2.436178e+04],
       [1.241000e+01, 1.591000e+01, 2.040000e+01, ..., 1.677878e+04,
        2.151425e+04, 2.758621e+04],
       [1.096000e+01, 1.405000e+01, 1.802000e+01, ..., 1.481758e+04,
        1.899954e+04, 2.436178e+04],
       ...,
       [1.241000e+01, 1.591000e+01, 2.040000e+01, ..., 1.677878e+04,
        2.151425e+04, 2.758621e+04],
       [1.096000e+01, 1.405000e+01, 1.802000e+01, ..., 1.481758e+04,
        1.899954e+04, 2.436178e+04],
       [1.241000e+01, 1.591000e+01, 2.040000e+01, ..., 1.677878e+04,
        2.151425e+04, 2.758621e+04]], dtype=float32)
>>>
```

This presentation is being recorded

Creating Variables

```
>>>
>>> from pytplot import store_data
>>>
>>> store_data('bx_values', data={'x': bx_times, 'y': bx_values})
True
>>>
>>> tplot('bx_values')
>>>
```

Creating Variables

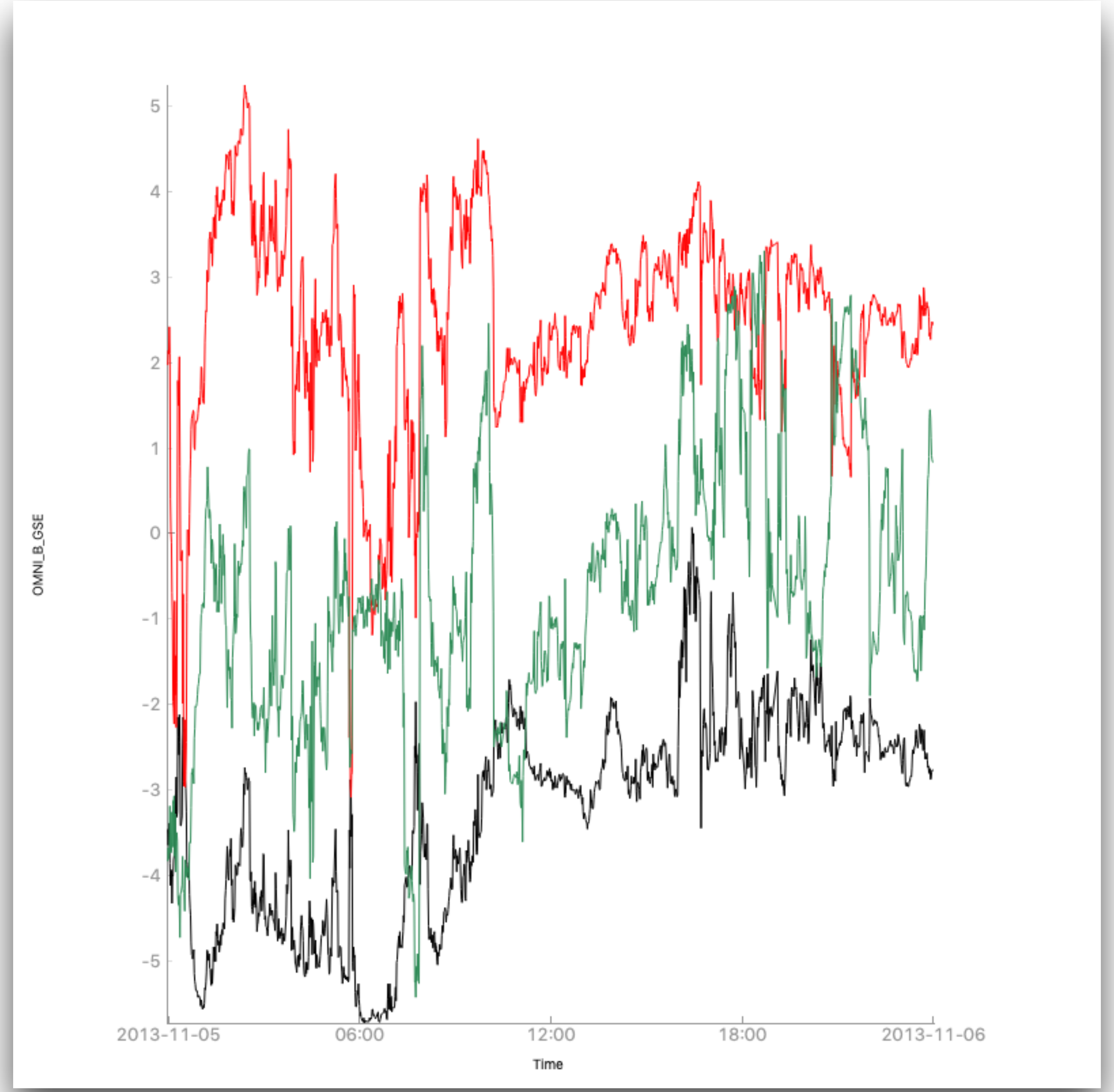


This presentation is being recorded

Creating a Vector from Components

```
>>>
>>> from pytplot import join_vec
>>>
>>> join_vec(['BX_GSE', 'BY_GSE', 'BZ_GSE'], new_tvar='OMNI_B_GSE')
'OMNI_B_GSE'
>>>
>>> tplot('OMNI_B_GSE')
>>>
```

Creating a Vector



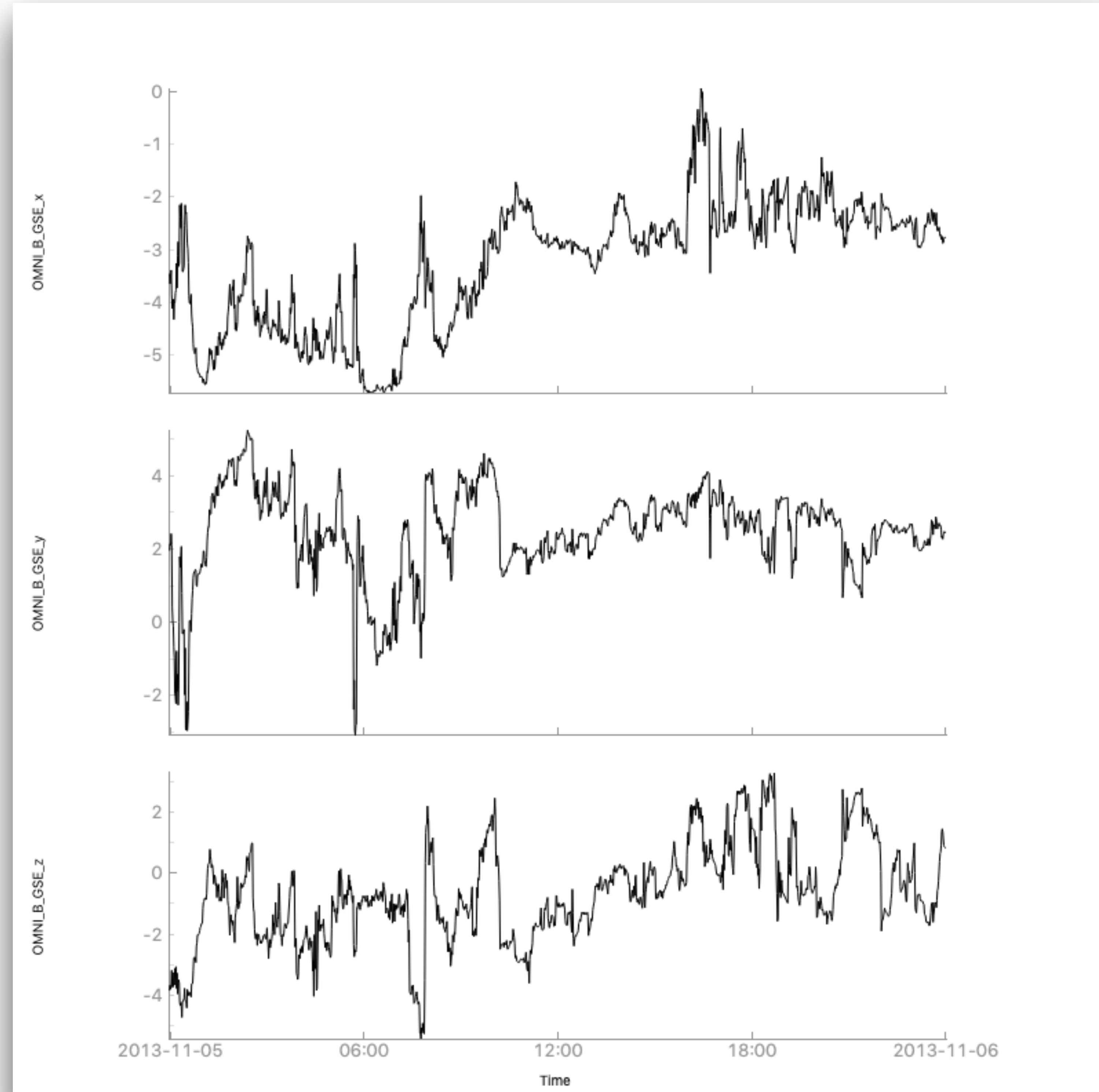
This presentation is being recorded

Splitting a Vector into Components

```
>>>
>>> from pytplot import split_vec
>>>
>>> split_vec('OMNI_B_GSE')
['OMNI_B_GSE_x', 'OMNI_B_GSE_y', 'OMNI_B_GSE_z']
>>>
>>> tplot(['OMNI_B_GSE_x', 'OMNI_B_GSE_y', 'OMNI_B_GSE_z'])
>>>
```

This presentation is being recorded

Splitting a Vector into Components



This presentation is being recorded

Getting Help

- Examples can be found in the READMEs on GitHub
- We also have Jupyter notebooks on GitHub:
    - https://github.com/spedas
- Feel free to email me: egrimes@igpp.ucla.edu

This presentation is being recorded

How to Contribute

- Try it out!
- Please report bugs, missing documentation, or any other issues so that we can fix them (feel free to email me or submit them through GitHub issues)
- Submit changes through pull requests, or email them to me (whichever is easiest for you)
- If there's a missing dataset or analysis tool that you would like to see included, please let us know!

This presentation is being recorded