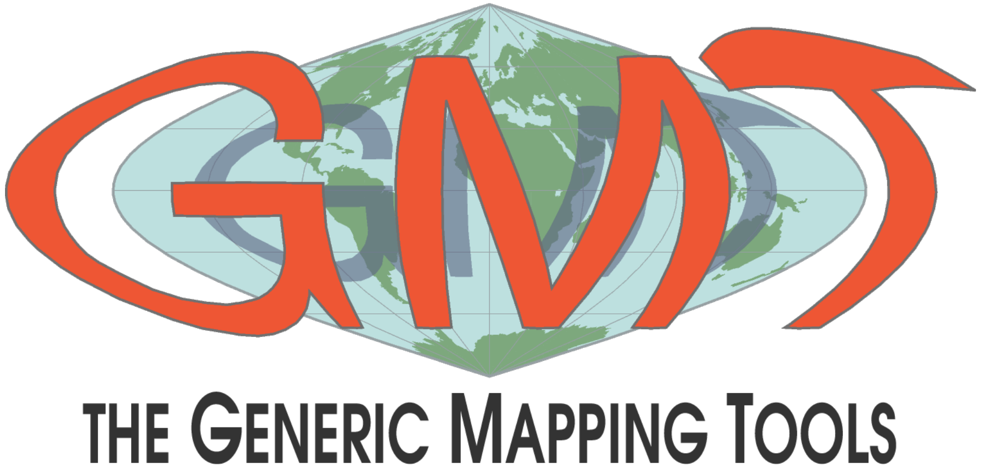
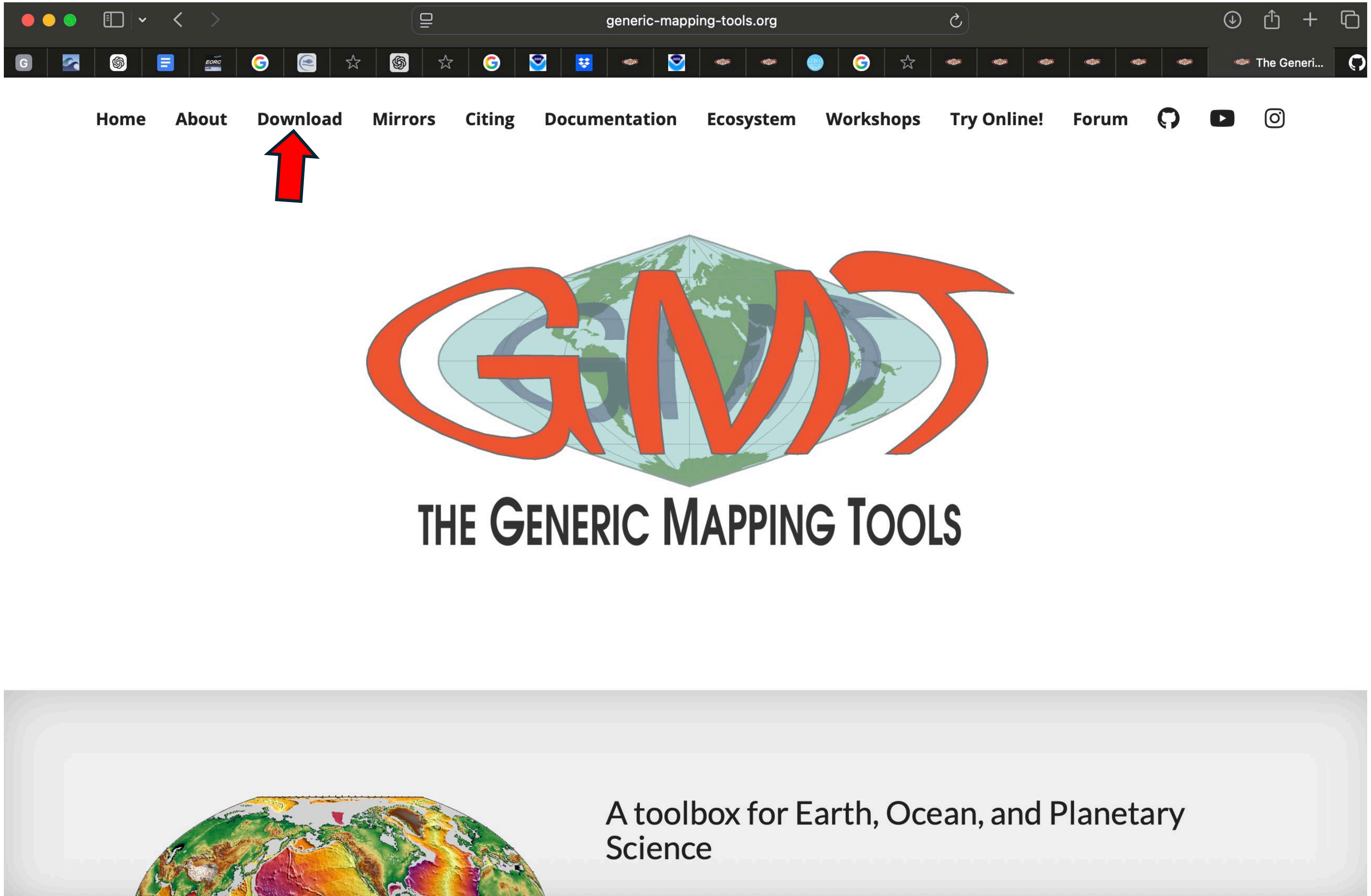
THE GENERIC MAPPING TOOLS

INTRO AND REVIEW

- Open source
- Command line tools
- For geographic and cartesian data

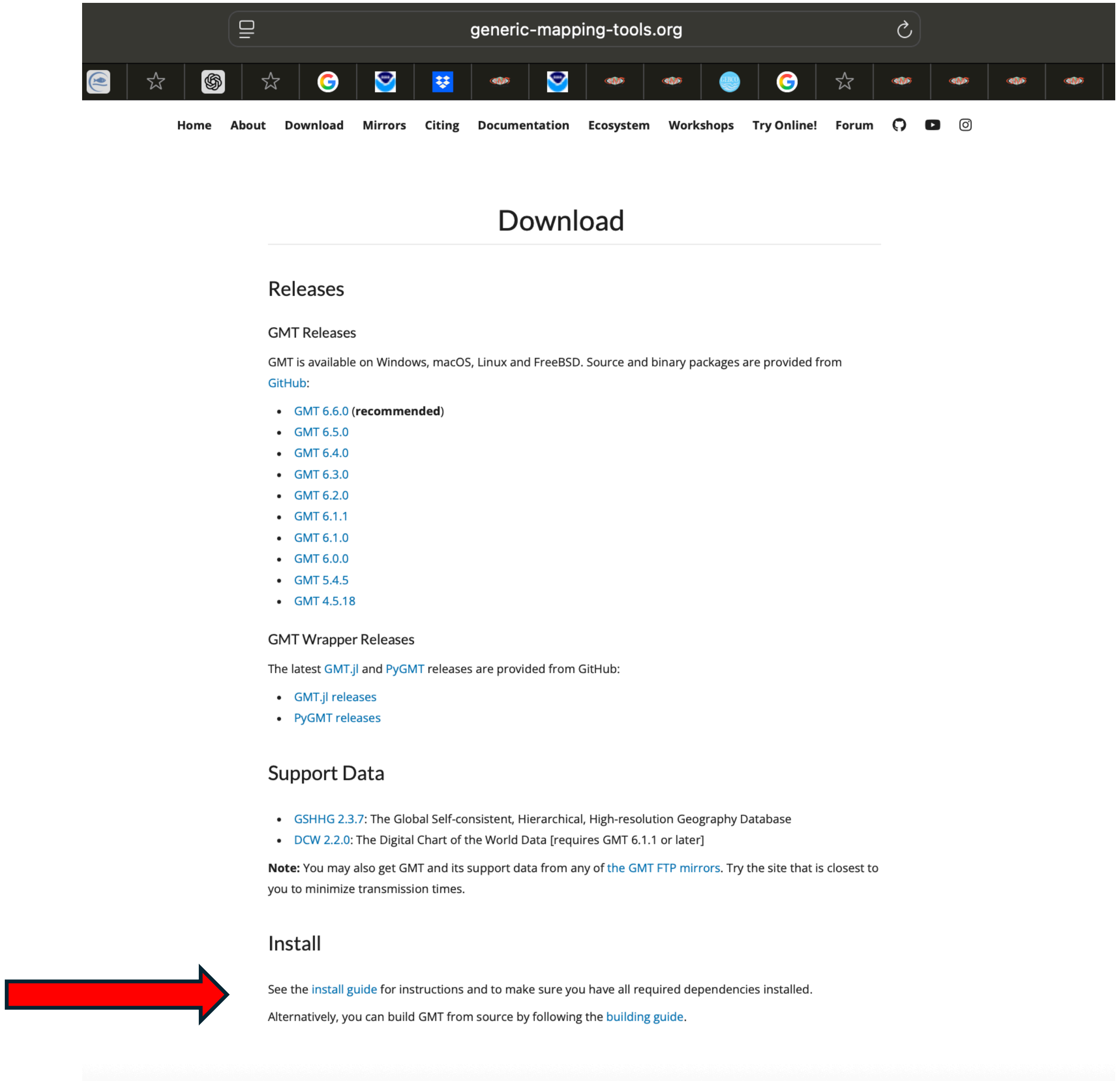


https://www.generic-mapping-tools.org



A toolbox for Earth, Ocean, and Planetary Science

https://www.generic-mapping-tools.org/download/



Download

Releases

GMT Releases

GMT is available on Windows, macOS, Linux and FreeBSD. Source and binary packages are provided from GitHub:

- GMT 6.6.0 (**recommended**)
- GMT 6.5.0
- GMT 6.4.0
- GMT 6.3.0
- GMT 6.2.0
- GMT 6.1.1
- GMT 6.1.0
- GMT 6.0.0
- GMT 5.4.5
- GMT 4.5.18

GMT Wrapper Releases

The latest GMT.jl and PyGMT releases are provided from GitHub:

- GMT.jl releases
- PyGMT releases

Support Data

- GSHHG 2.3.7: The Global Self-consistent, Hierarchical, High-resolution Geography Database
- DCW 2.2.0: The Digital Chart of the World Data [requires GMT 6.1.1 or later]

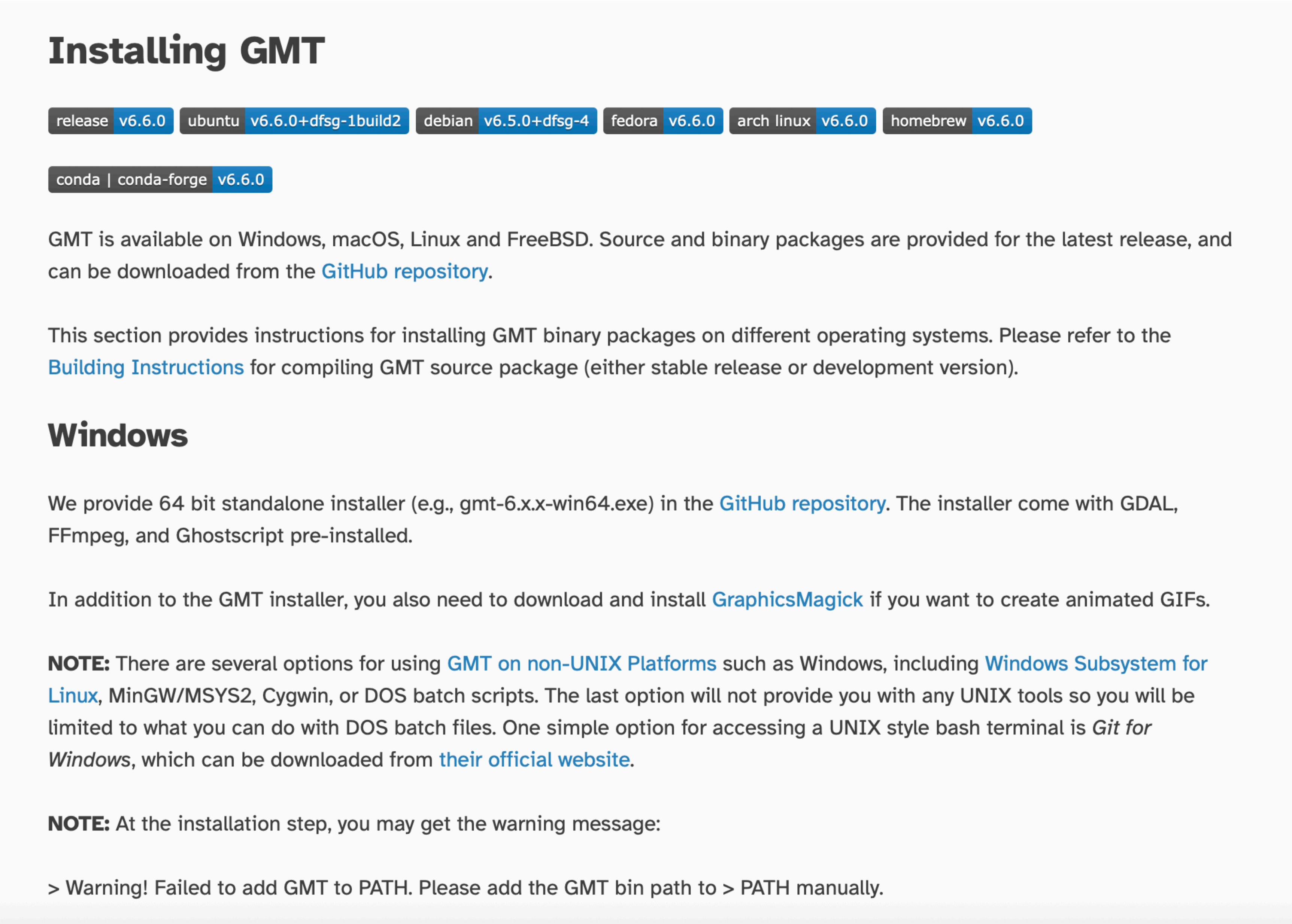
Note: You may also get GMT and its support data from any of the GMT FTP mirrors. Try the site that is closest to you to minimize transmission times.

Install

See the install guide for instructions and to make sure you have all required dependencies installed.

Alternatively, you can build GMT from source by following the building guide.

https://docs.generic-mapping-tools.org/latest/install.html



Installing GMT

release v6.6.0 ubuntu v6.6.0+dfsg-1build2 debian v6.5.0+dfsg-4 fedora v6.6.0 arch linux v6.6.0 homebrew v6.6.0

conda | conda-forge v6.6.0

GMT is available on Windows, macOS, Linux and FreeBSD. Source and binary packages are provided for the latest release, and can be downloaded from the GitHub repository.

This section provides instructions for installing GMT binary packages on different operating systems. Please refer to the Building Instructions for compiling GMT source package (either stable release or development version).

Windows

We provide 64 bit standalone installer (e.g., gmt-6.x.x-win64.exe) in the GitHub repository. The installer come with GDAL, FFmpeg, and Ghostscript pre-installed.

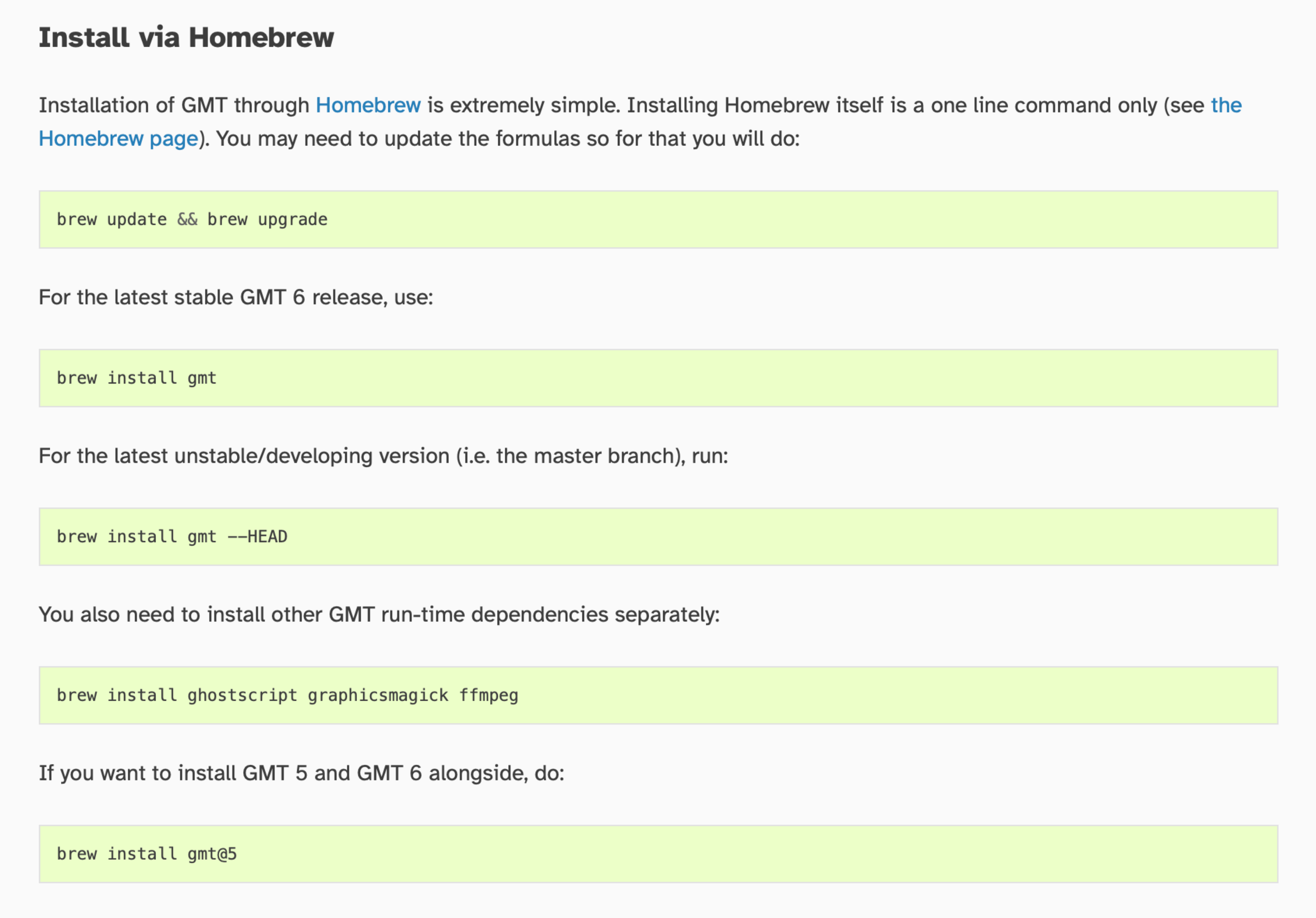
In addition to the GMT installer, you also need to download and install GraphicsMagick if you want to create animated GIFs.

NOTE: There are several options for using GMT on non-UNIX Platforms such as Windows, including Windows Subsystem for Linux, MinGW/MSYS2, Cygwin, or DOS batch scripts. The last option will not provide you with any UNIX tools so you will be limited to what you can do with DOS batch files. One simple option for accessing a UNIX style bash terminal is *Git for Windows*, which can be downloaded from their official website.

NOTE: At the installation step, you may get the warning message:

> Warning! Failed to add GMT to PATH. Please add the GMT bin path to > PATH manually.

https://docs.generic-mapping-tools.org/latest/install.html



Install via Homebrew

Installation of GMT through Homebrew is extremely simple. Installing Homebrew itself is a one line command only (see the Homebrew page). You may need to update the formulas so for that you will do:

```
brew update && brew upgrade
```

For the latest stable GMT 6 release, use:

```
brew install gmt
```

For the latest unstable/developing version (i.e. the master branch), run:

```
brew install gmt --HEAD
```

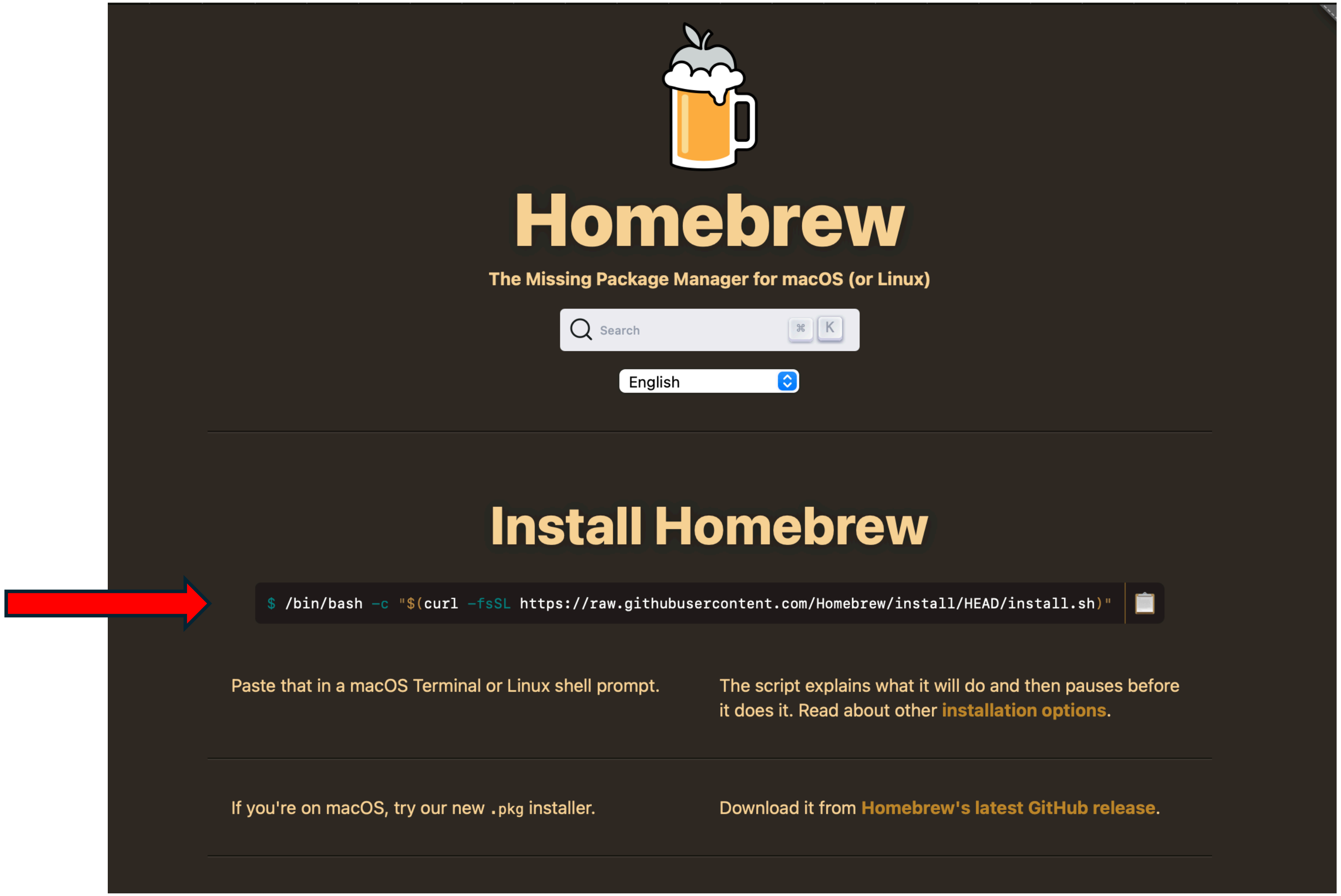
You also need to install other GMT run-time dependencies separately:

```
brew install ghostscript graphicsmagick ffmpeg
```

If you want to install GMT 5 and GMT 6 alongside, do:

```
brew install gmt@5
```

https://brew.sh



Homebrew

The Missing Package Manager for macOS (or Linux)

Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

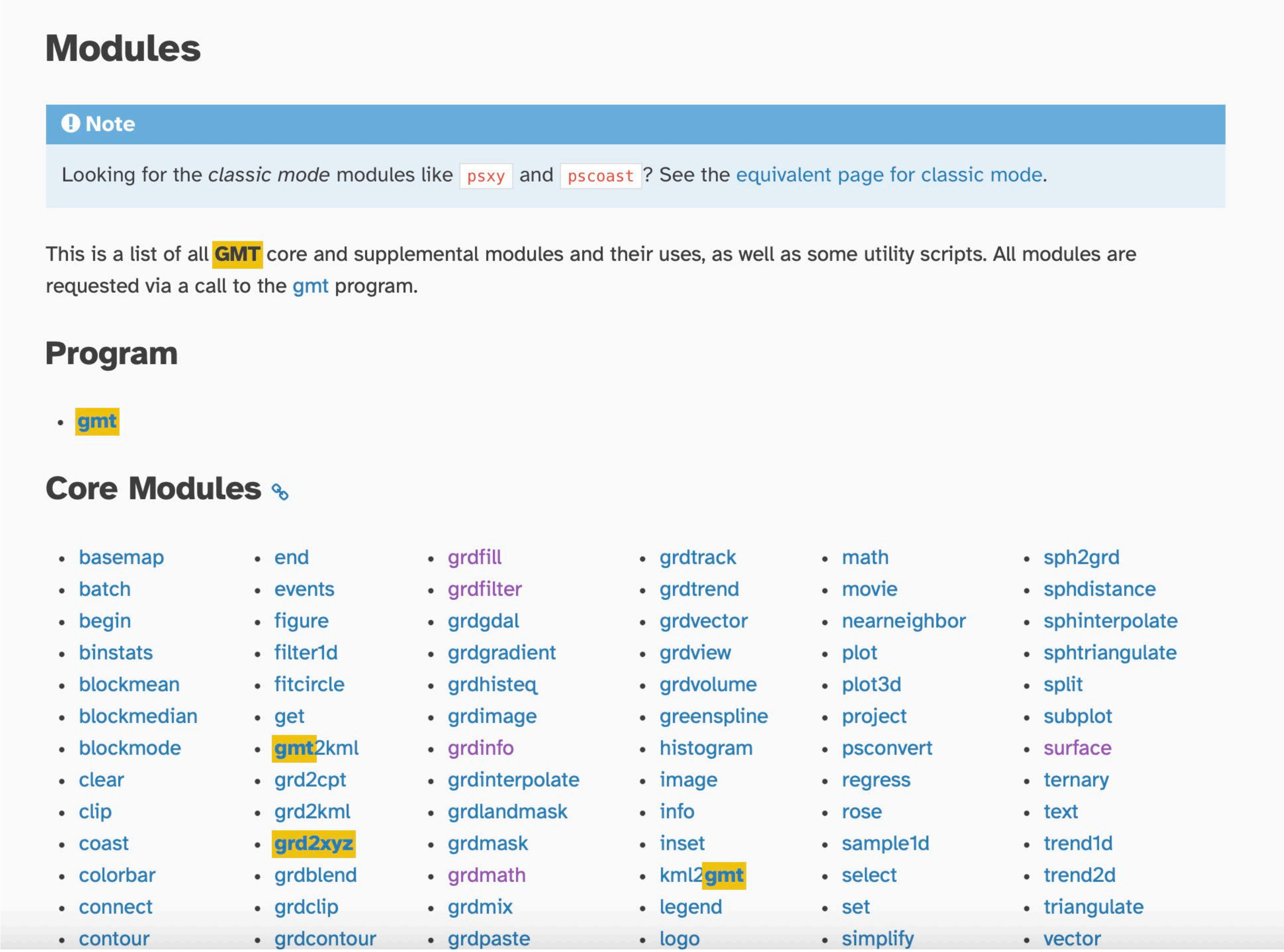
Paste that in a macOS Terminal or Linux shell prompt.

The script explains what it will do and then pauses before it does it. Read about other **installation options**.

If you're on macOS, try our new `.pkg` installer.

Download it from **Homebrew's latest GitHub release**.

https://docs.generic-mapping-tools.org/dev/modules.html



Modules

Note

Looking for the *classic mode* modules like `psxy` and `pscoast`? See the equivalent page for classic mode.

This is a list of all **GMT** core and supplemental modules and their uses, as well as some utility scripts. All modules are requested via a call to the gmt program.

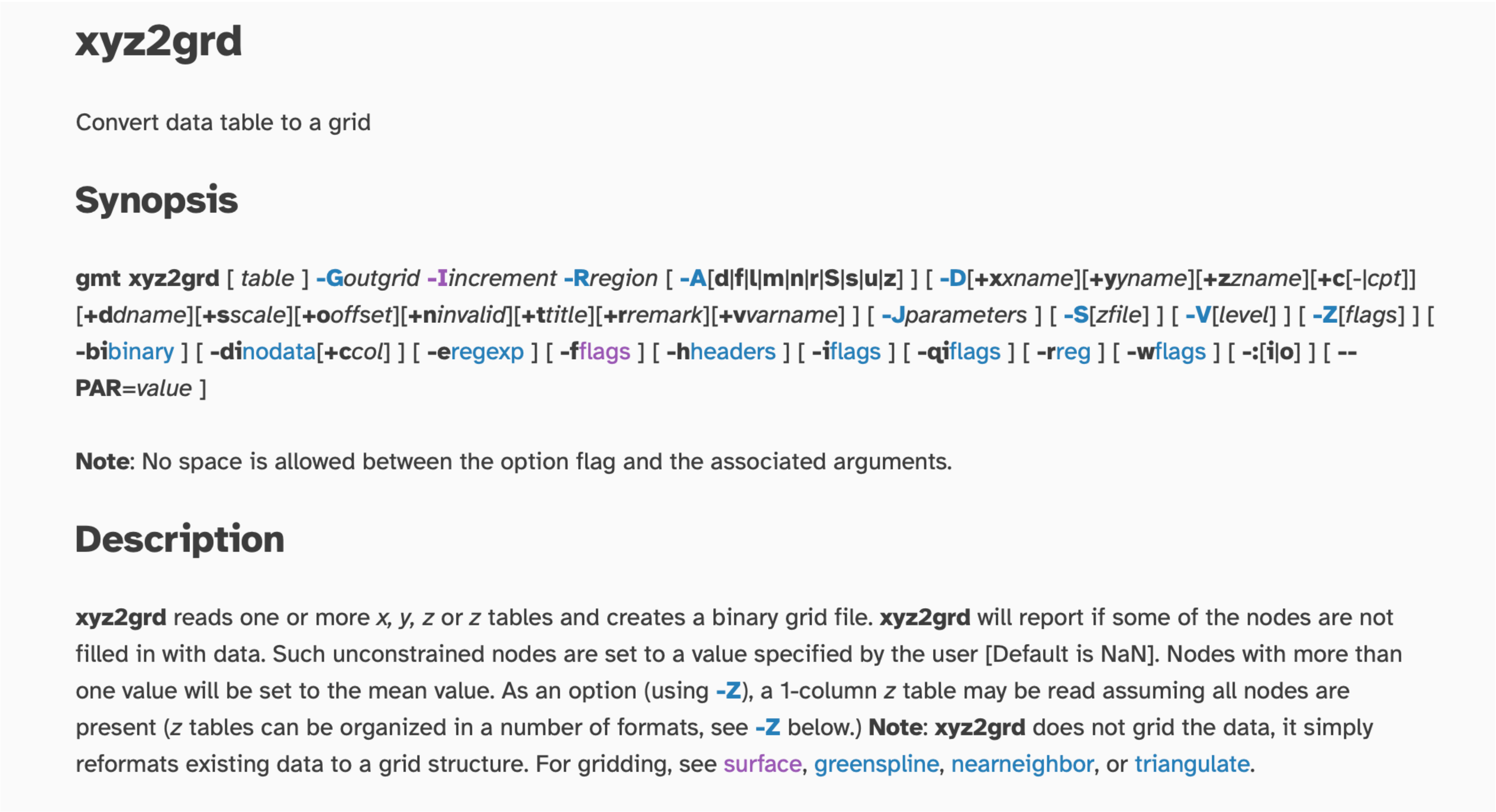
Program

- gmt

Core Modules

- basemap
- batch
- begin
- binstats
- blockmean
- blockmedian
- blockmode
- clear
- clip
- coast
- colorbar
- connect
- contour
- end
- events
- figure
- filter1d
- fitcircle
- get
- gmt2kml
- grd2cpt
- grd2kml
- grd2xyz
- grdblend
- grdclip
- grdcontour
- grdfill
- grdfilter
- grdgdal
- grdgradient
- grdhisteq
- grdimage
- grdinfo
- grdinterpolate
- grdlandmask
- grdmask
- grdmath
- grdmix
- grdpaste
- grdtrack
- grdtrend
- grdvector
- grdview
- grdvolume
- greenspline
- histogram
- image
- info
- inset
- kml2gmt
- legend
- logo
- math
- movie
- nearneighbor
- plot
- plot3d
- project
- psconvert
- regress
- rose
- sample1d
- select
- set
- simplify
- sph2grd
- sphdistance
- sphinterpolate
- sphtriangulate
- split
- subplot
- surface
- ternary
- text
- trend1d
- trend2d
- triangulate
- vector

TOOL



xyz2grd

Convert data table to a grid

Synopsis

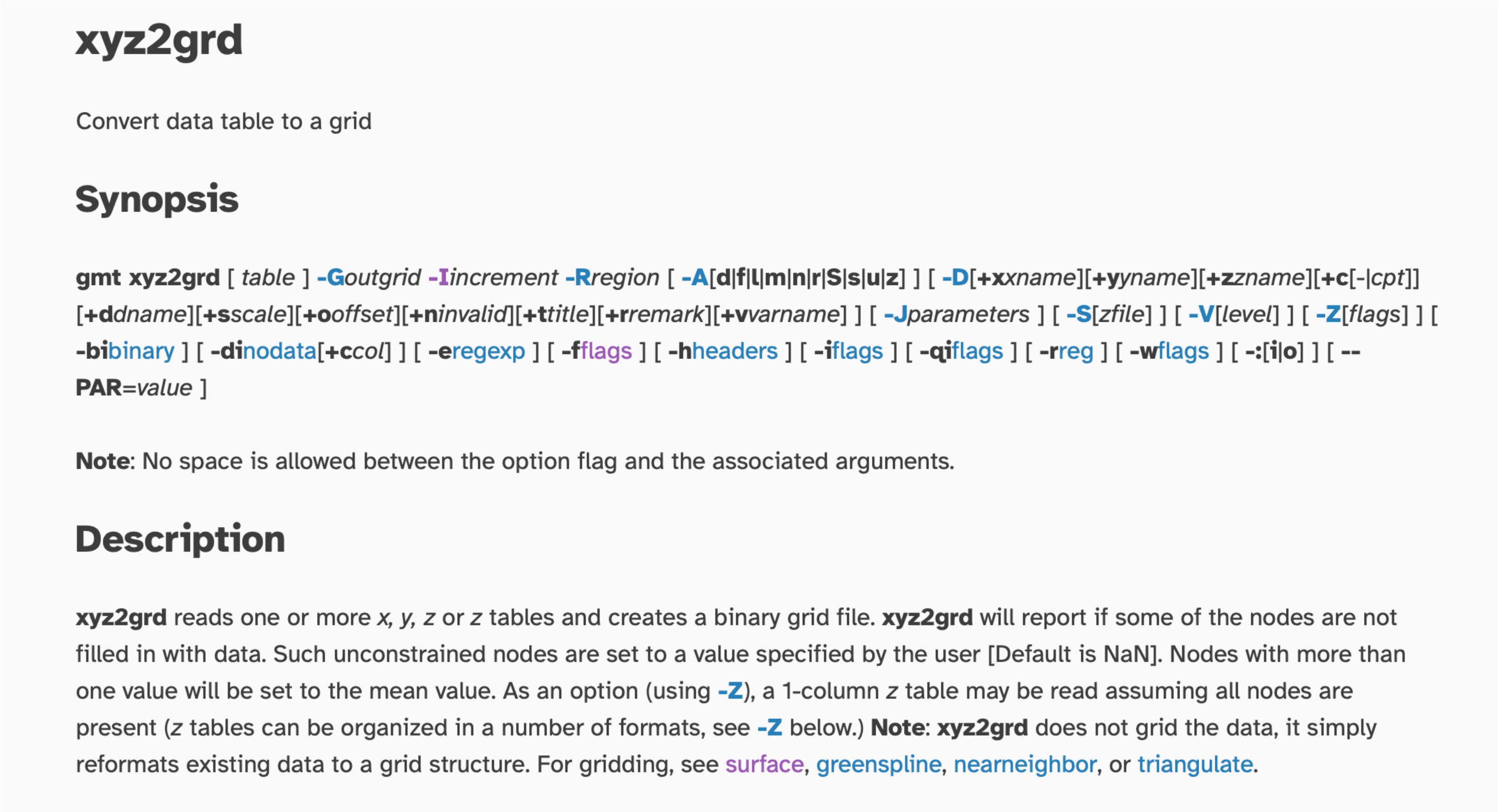
gmt xyz2grd [*table*] **-G***outgrid* **-I***increment* **-R***region* [**-A**[**d**|**f**|**l**|**m**|**n**|**r**|**S**|**s**|**u**|**z**]] [**-D**[**+x***xname*][**+y***yname*][**+z***zname*][**+c**[-|*cpt*]][**+d***dname*][**+s***scale*][**+o***offset*][**+n***invalid*][**+t***title*][**+r***remark*][**+v***varname*]] [**-J***parameters*] [**-S**[*zfile*]] [**-V**[*level*]] [**-Z**[*flags*]] [**-bi**binary] [**-di**nodata[**+c***col*]] [**-e**regexp] [**-f**flags] [**-h**headers] [**-i**flags] [**-qi**flags] [**-r**reg] [**-w**flags] [**-:**[**i**|**o**]] [**--PAR**=*value*]

Note: No space is allowed between the option flag and the associated arguments.

Description

xyz2grd reads one or more *x, y, z* or *z* tables and creates a binary grid file. **xyz2grd** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the mean value. As an option (using **-Z**), a 1-column *z* table may be read assuming all nodes are present (*z* tables can be organized in a number of formats, see **-Z** below.) **Note**: **xyz2grd** does not grid the data, it simply reformats existing data to a grid structure. For gridding, see surface, greenspline, nearneighbor, or triangulate.

FUNCTION



xyz2grd

Convert data table to a grid

Synopsis

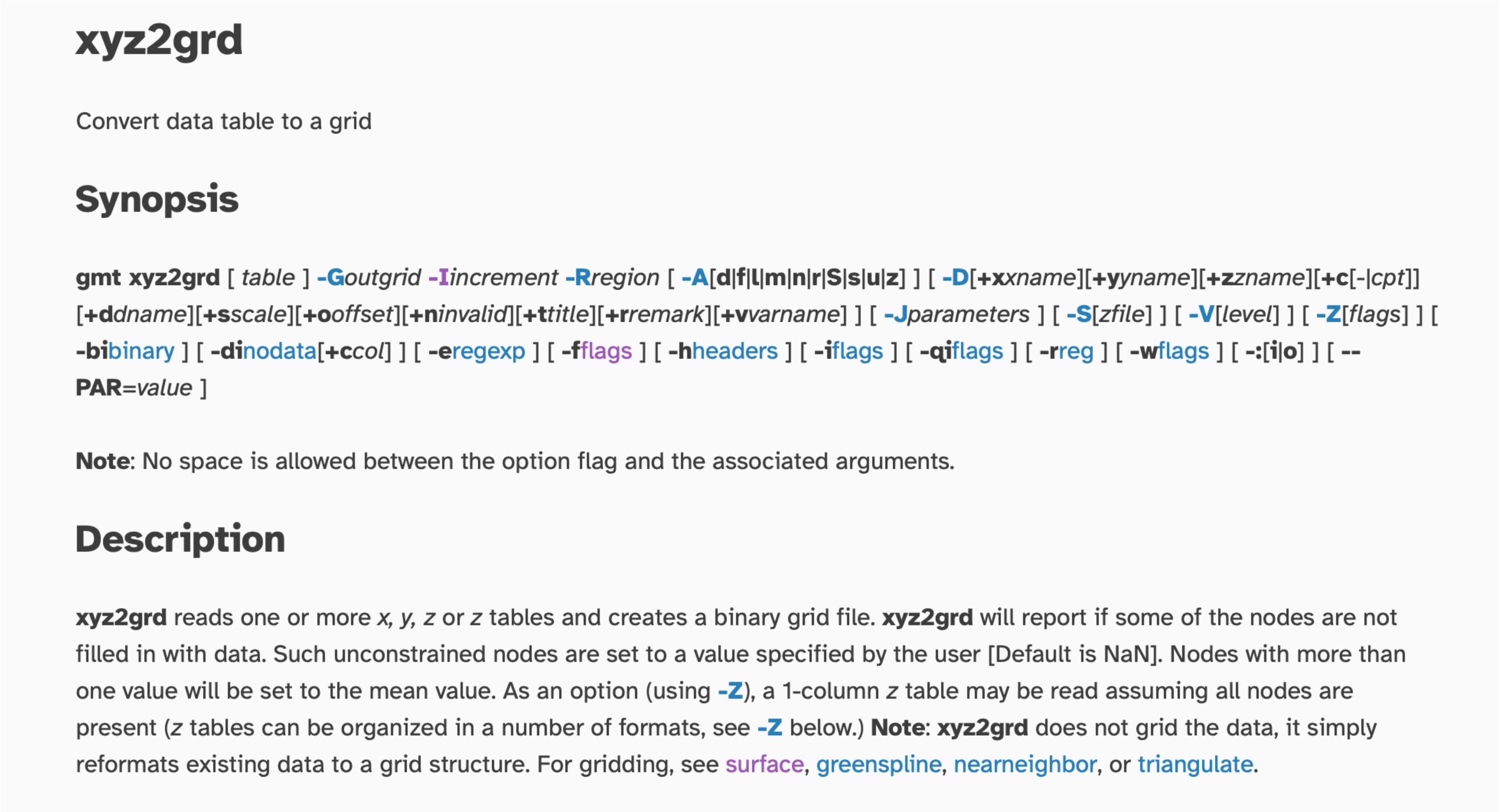
gmt xyz2grd [*table*] **-G***outgrid* **-I***increment* **-R***region* [**-A**[**d**|**f**|**l**|**m**|**n**|**r**|**S**|**s**|**u**|**z**]] [**-D**[**+x***xname*][**+y***yname*][**+z***zname*][**+c**[-|*cpt*]][**+d***dname*][**+s***scale*][**+o***offset*][**+n***invalid*][**+t***title*][**+r***remark*][**+v***varname*]] [**-J***parameters*] [**-S**[*zfile*]] [**-V**[*level*]] [**-Z**[*flags*]] [**-bi**binary] [**-di**nodata[**+c***col*]] [**-e**regexp] [**-f**flags] [**-h**headers] [**-i**flags] [**-qi**flags] [**-r**reg] [**-w**flags] [**-:**[**i**|**o**]] [**--PAR**=*value*]

Note: No space is allowed between the option flag and the associated arguments.

Description

xyz2grd reads one or more *x, y, z* or *z* tables and creates a binary grid file. **xyz2grd** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the mean value. As an option (using **-Z**), a 1-column *z* table may be read assuming all nodes are present (*z* tables can be organized in a number of formats, see **-Z** below.) **Note**: **xyz2grd** does not grid the data, it simply reformats existing data to a grid structure. For gridding, see surface, greenspline, nearneighbor, or triangulate.

INPUTS



xyz2grd

Convert data table to a grid

Synopsis

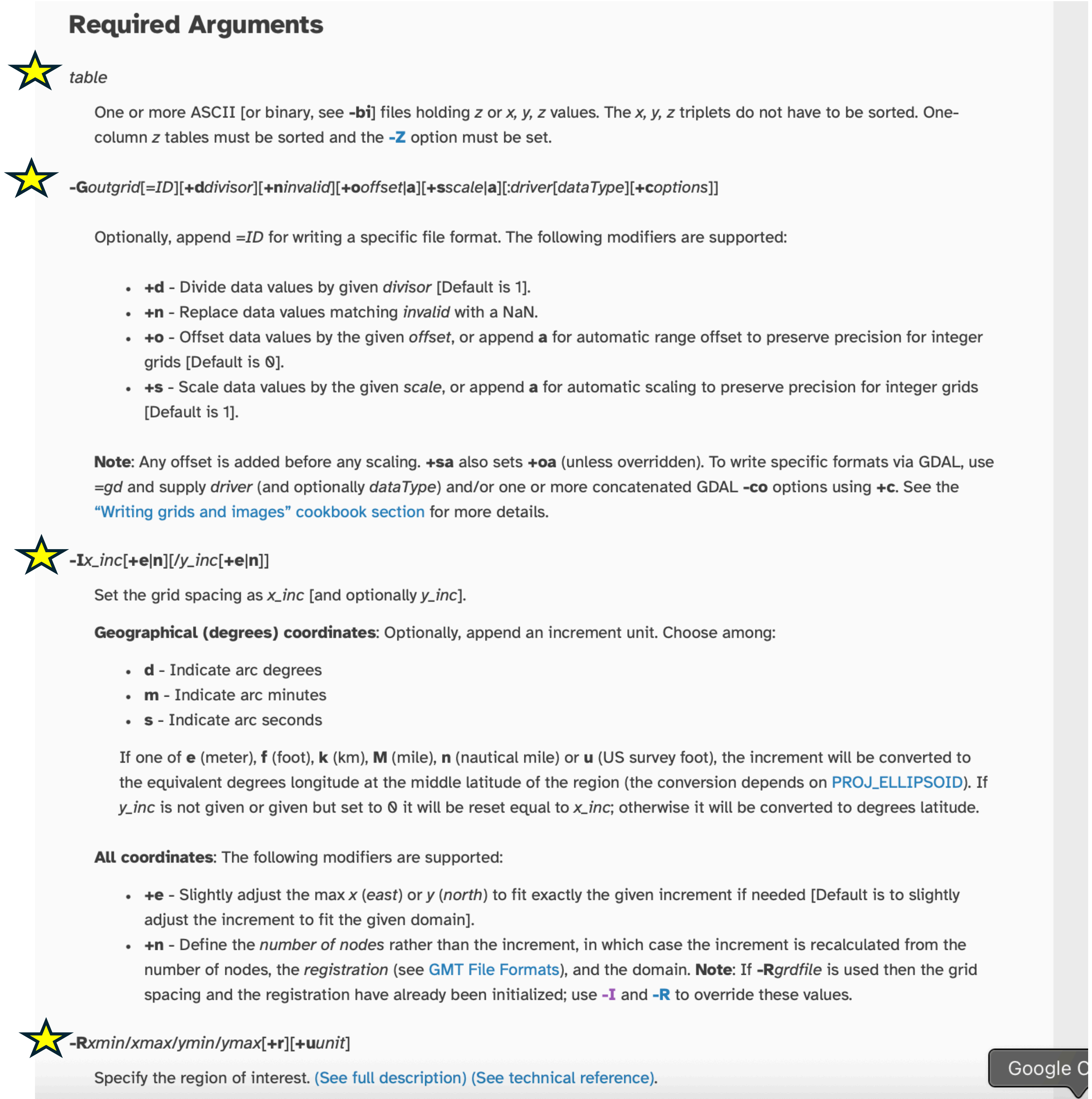
gmt xyz2grd [*table*] **-G***outgrid* **-I***increment* **-R***region* [**-A**[**d**|**f**|**l**|**m**|**n**|**r**|**S**|**s**|**u**|**z**]] [**-D**[**+x***xname*][**+y***yname*][**+z***zname*][**+c**[-|*cpt*]][**+d***dname*][**+s***scale*][**+o***offset*][**+n***invalid*][**+t***title*][**+r***remark*][**+v***varname*]] [**-J***parameters*] [**-S**[*zfile*]] [**-V**[*level*]] [**-Z**[*flags*]] [**-bi**binary] [**-di**nodata[**+c***col*]] [**-e**regexp] [**-f**flags] [**-h**headers] [**-i**flags] [**-qi**flags] [**-r**reg] [**-w**flags] [**-:**[**i**|**o**]] [**--PAR**=*value*]

Note: No space is allowed between the option flag and the associated arguments.

Description

xyz2grd reads one or more *x, y, z* or *z* tables and creates a binary grid file. **xyz2grd** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the mean value. As an option (using **-Z**), a 1-column *z* table may be read assuming all nodes are present (*z* tables can be organized in a number of formats, see **-Z** below.) **Note**: **xyz2grd** does not grid the data, it simply reformats existing data to a grid structure. For gridding, see surface, greenspline, nearneighbor, or triangulate.

BARE MINIMUM INPUTS



Required Arguments

table

One or more ASCII [or binary, see **-bi**] files holding *z* or *x, y, z* values. The *x, y, z* triplets do not have to be sorted. One-column *z* tables must be sorted and the **-Z** option must be set.

-G*outgrid*[=*ID*][**+d***divisor*][**+n***invalid*][**+o***offset*|**a**][**+s***scale*|**a**][:*driver*[*dataType*][**+c***options*]]

Optionally, append =*ID* for writing a specific file format. The following modifiers are supported:

- **+d** - Divide data values by given *divisor* [Default is 1].
- **+n** - Replace data values matching *invalid* with a NaN.
- **+o** - Offset data values by the given *offset*, or append **a** for automatic range offset to preserve precision for integer grids [Default is 0].
- **+s** - Scale data values by the given *scale*, or append **a** for automatic scaling to preserve precision for integer grids [Default is 1].

Note: Any offset is added before any scaling. **+sa** also sets **+oa** (unless overridden). To write specific formats via GDAL, use =*gd* and supply *driver* (and optionally *dataType*) and/or one or more concatenated GDAL **-co** options using **+c**. See the "Writing grids and images" cookbook section for more details.

-I*x_inc*[**+e**|**n**][/*y_inc*[**+e**|**n**]]

Set the grid spacing as *x_inc* [and optionally *y_inc*].

Geographical (degrees) coordinates: Optionally, append an increment unit. Choose among:

- **d** - Indicate arc degrees
- **m** - Indicate arc minutes
- **s** - Indicate arc seconds

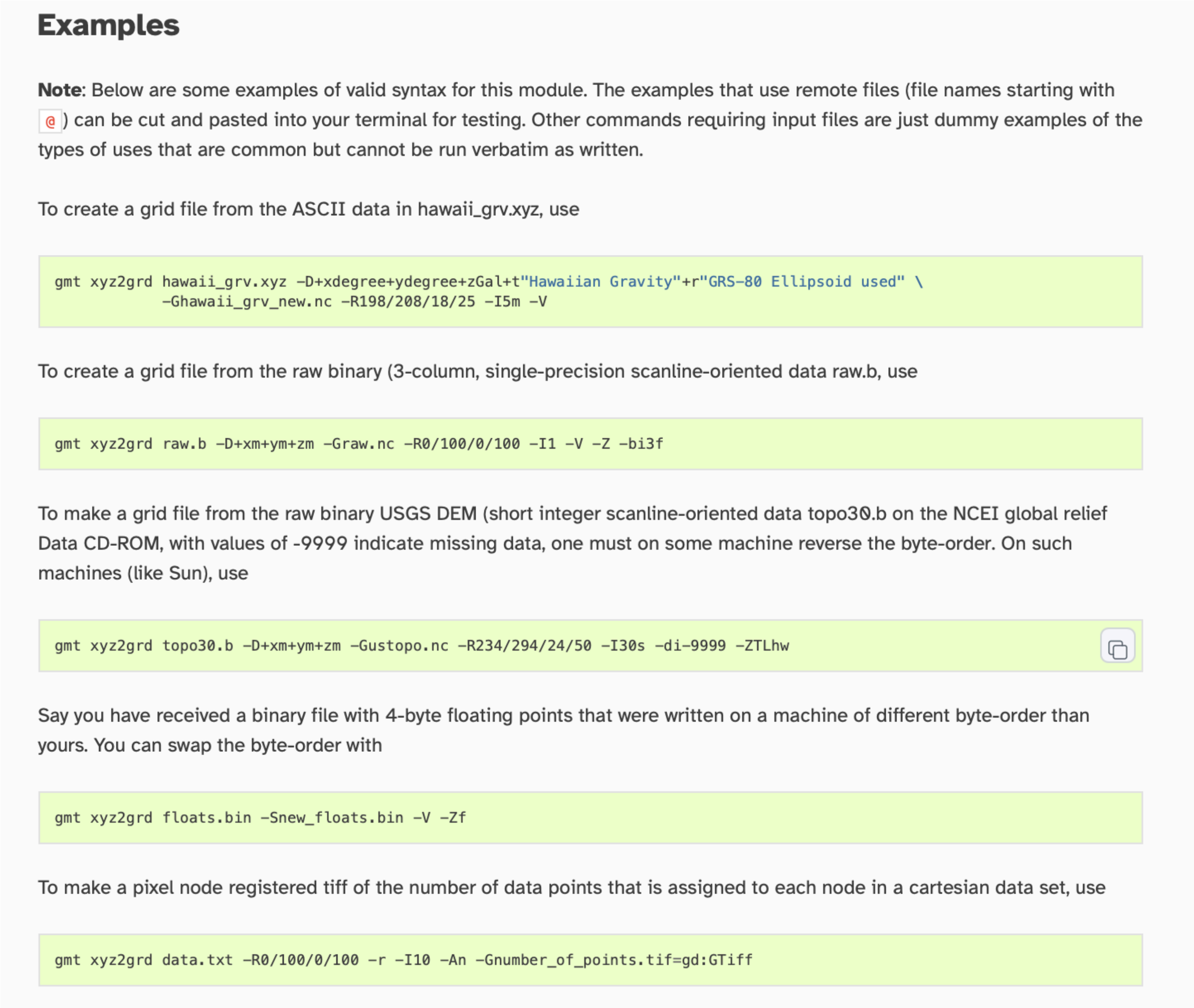
If one of **e** (meter), **f** (foot), **k** (km), **M** (mile), **n** (nautical mile) or **u** (US survey foot), the increment will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on PROJ_ELLIPSOID). If *y_inc* is not given or given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude.

All coordinates: The following modifiers are supported:

- **+e** - Slightly adjust the max *x* (*east*) or *y* (*north*) to fit exactly the given increment if needed [Default is to slightly adjust the increment to fit the given domain].
- **+n** - Define the *number of nodes* rather than the increment, in which case the increment is recalculated from the number of nodes, the *registration* (see GMT File Formats), and the domain. **Note**: If **-R***grdfile* is used then the grid spacing and the registration have already been initialized; use **-I** and **-R** to override these values.

-R*xmin*/*xmax*/*ymin*/*ymax*[**+r**][**+u***unit*]

Specify the region of interest. (See full description) (See technical reference).



Examples

Note: Below are some examples of valid syntax for this module. The examples that use remote files (file names starting with `@`) can be cut and pasted into your terminal for testing. Other commands requiring input files are just dummy examples of the types of uses that are common but cannot be run verbatim as written.

To create a grid file from the ASCII data in hawaii_grv.xyz, use

```
gmt xyz2grd hawaii_grv.xyz -D+xdegree+ydegree+zGal+t"Hawaiian Gravity"+r"GRS-80 Ellipsoid used" \
            -Ghawaii_grv_new.nc -R198/208/18/25 -I5m -V
```

To create a grid file from the raw binary (3-column, single-precision scanline-oriented data raw.b, use

```
gmt xyz2grd raw.b -D+xm+ym+zm -Graw.nc -R0/100/0/100 -I1 -V -Z -bi3f
```

To make a grid file from the raw binary USGS DEM (short integer scanline-oriented data topo30.b on the NCEI global relief Data CD-ROM, with values of -9999 indicate missing data, one must on some machine reverse the byte-order. On such machines (like Sun), use

```
gmt xyz2grd topo30.b -D+xm+ym+zm -Gustopo.nc -R234/294/24/50 -I30s -di-9999 -ZTLhw
```

Say you have received a binary file with 4-byte floating points that were written on a machine of different byte-order than yours. You can swap the byte-order with

```
gmt xyz2grd floats.bin -Snew_floats.bin -V -Zf
```

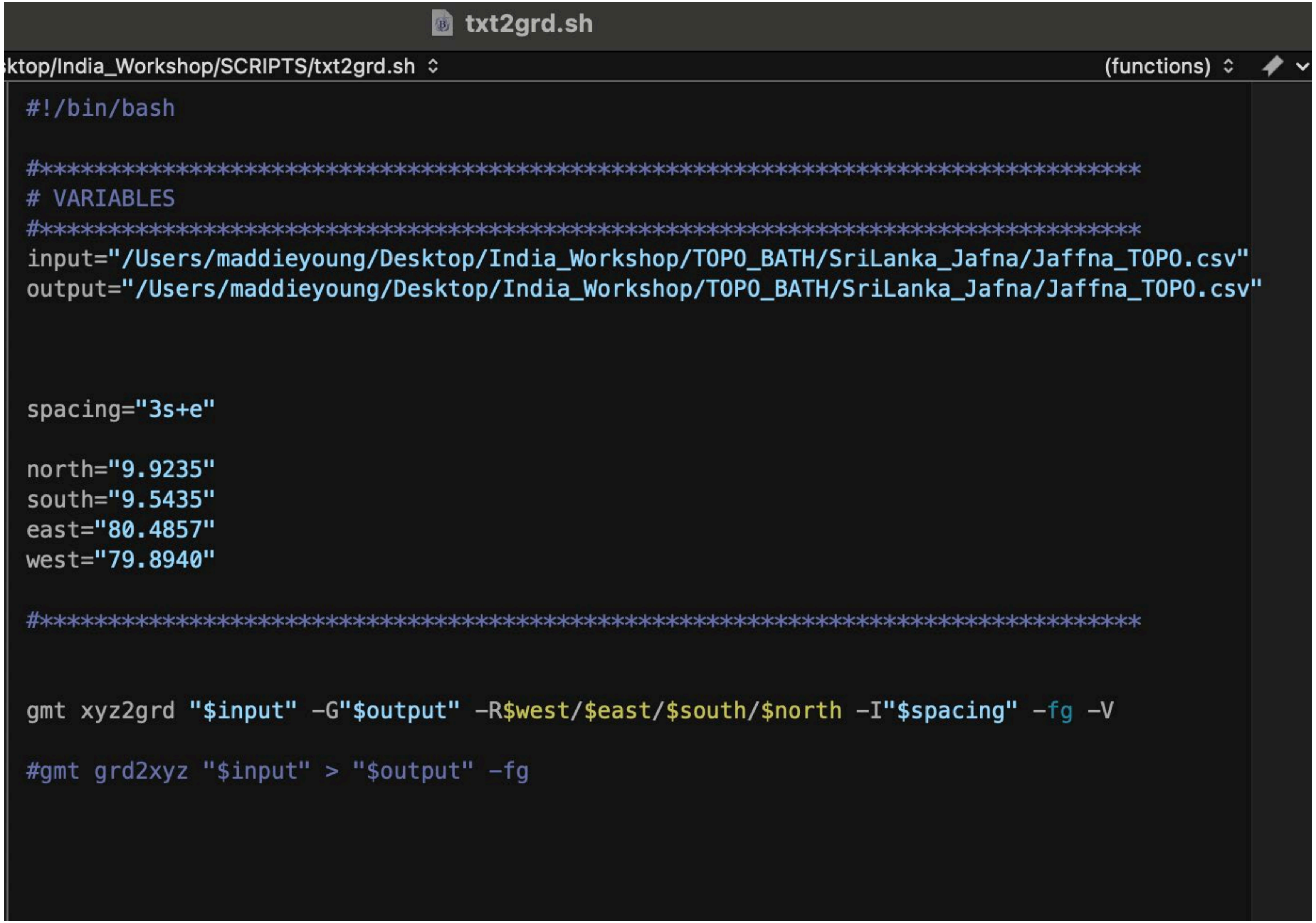
To make a pixel node registered tiff of the number of data points that is assigned to each node in a cartesian data set, use

```
gmt xyz2grd data.txt -R0/100/0/100 -r -I10 -An -Gnumber_of_points.tif=gd:GTiff
```

What I’ve used in this session

- grd2xyz
- xyz2grd
- grdinfo
- grdsample
- grdcut
- surface
- grdblend
- grdfill
- grdmath
- grdfilter
- grdedit
- grdconvert

How I use



```
#!/bin/bash

#*********************************************************************************
# VARIABLES
#*********************************************************************************
input="/Users/maddieyoung/Desktop/India_Workshop/TOPO_BATH/SriLanka_Jafna/Jaffna_TOPO.csv"
output="/Users/maddieyoung/Desktop/India_Workshop/TOPO_BATH/SriLanka_Jafna/Jaffna_TOPO.csv"



spacing="3s+e"

north="9.9235"
south="9.5435"
east="80.4857"
west="79.8940"

#*********************************************************************************


gmt xyz2grd "$input" -G"$output" -R$west/$east/$south/$north -I"$spacing" -fg -V

#gmt grd2xyz "$input" > "$output" -fg
```