Analyzing space-time satellite data with GRASS GIS for environmental monitoring

Verónica Andreo

OpenGeoHub Summer School 2019, Münster

ABOUT ME

PhD in Biology

MSc in Remote Sensing and GIS

Researcher @ Argentinean Space Agency

Applications of RS & GIS for disease ecology

Keywords: RS, GIS, Time series, SDM,
Disease Ecology, Rodents, Hantavirus

GRASS GIS Dev Team

OSGeo Charter member

FOSS4G enthusiast and advocate

veroandreo.gitlab.io

veroandreo @VeronicaAndreo

veroandreo@gmail.com

GRASS GIS is **the first FOSS GIS** that incorporated capabilities to **manage, analyze, process and visualize spatio-temporal data**, as well as the temporal relationships among time series.

The TGRASS framework

The TGRASS framework

- TGRASS is the temporal enabled GRASS GIS designed to easily handle time series data

The TGRASS framework

- TGRASS is the temporal enabled GRASS GIS designed to easily handle time series data
- TGRASS is fully based on metadata and does not duplicate any dataset

The TGRASS framework

- TGRASS is the temporal enabled GRASS GIS designed to easily handle time series data
- TGRASS is fully based on metadata and does not duplicate any dataset
- Snapshot approach, i.e., adds time stamps to maps

The TGRASS framework

- TGRASS is the temporal enabled GRASS GIS designed to easily handle time series data
- TGRASS is fully based on metadata and does not duplicate any dataset
- Snapshot approach, i.e., adds time stamps to maps
- A collection of time stamped maps (snapshots) of the same variable are called space-time datasets or STDS

The TGRASS framework

- TGRASS is the temporal enabled GRASS GIS designed to easily handle time series data
- TGRASS is fully based on metadata and does not duplicate any dataset
- Snapshot approach, i.e., adds time stamps to maps
- A collection of time stamped maps (snapshots) of the same variable are called space-time datasets or STDS
- Maps in a STDS can have different spatial and temporal extents

Space-time datasets

- Space time raster datasets (**STRDS**)
- Space time 3D raster datasets (**STR3DS**)
- Space time vector datasets (**STVDS**)

Suppot for **image collections** is on the way!

Other TGRASS notions

Other TGRASS notions

- Time can be defined as intervals (start and end time) or instances (only start time)

Other TGRASS notions

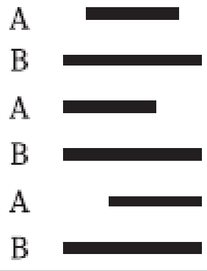
- Time can be defined as intervals (start and end time) or instances (only start time)
- Time can be absolute (e.g., 2017-04-06 22:39:49) or relative (e.g., 4 years, 90 days)

Other TGRASS notions

- Time can be defined as intervals (start and end time) or instances (only start time)
- Time can be absolute (e.g., 2017-04-06 22:39:49) or relative (e.g., 4 years, 90 days)
- Granularity is the greatest common divisor of the temporal extents (and possible gaps) of all maps in the space-time cube

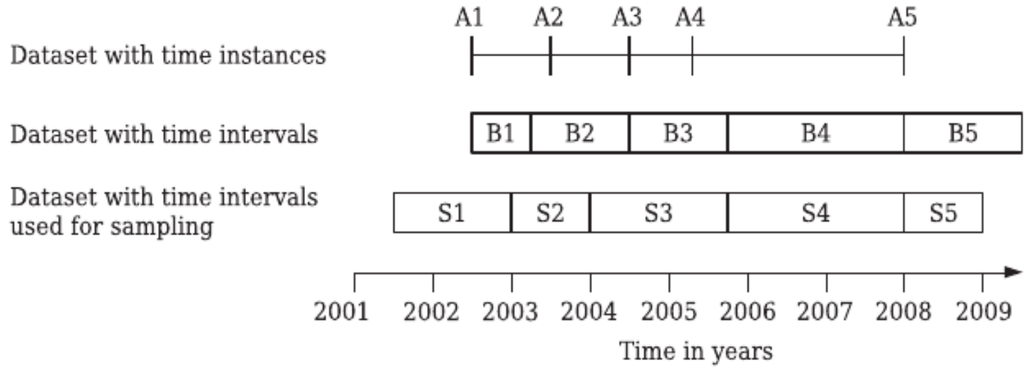
Other TGRASS notions

- Topology refers to temporal relations between time intervals in a STDS.

	A in relation to B	B in relation to A
	equivalent	equivalent
	overlap	overlap
	during	contain
	follows	precedes

Other TGRASS notions

- Temporal sampling is used to determine the state of one process during a second process.



Sampling time instances

	start
S1	A1
S2	A2
S3	A3,A4
S4	—
S5	A5

Sampling time intervals

	start	during	contain	overlap	equal
S1	B1	—	—	B1	—
S2	B2	—	—	B1,B2	—
S3	B3	B3	—	B2	—
S4	B4	—	—	—	B4
S5	B5	—	B5	—	—

Spatio-temporal modules

- **t.***: General modules to handle STDS of all types
- **t.rast.***: Modules that deal with STRDS
- **t.rast3d.***: Modules that deal with STR3DS
- **t.vect.***: Modules that deal with STVDS

TGRASS framework and workflow

SPATIO-TEMPORAL GIS MODULES

GRASS DATABASE

3D rasters | Rasters | Vectors

Create STDS
t.create

Assign time stamps
t.register

t.rast.to.vect

STR3DS **STRDS** **STVDS**

TEMPORAL DATABASE

General utility: t.list, t.info, t.remove, t.select, t.sample, t.topology, etc.

Other useful modules and Add-ons

g.gui.mapswipe

Gap-fill:
r.hants,
r.series.lwr

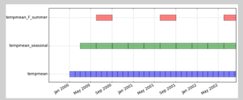
Nulls:
t.rast.nulls

Data extraction:
v.what.strds,
v.strds.stats,
t.rast.what.aggr

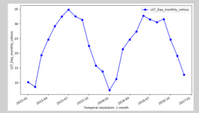
Others:
r.seasons,
r.regression.series,
r.bioclim

t.rast.out.xyz

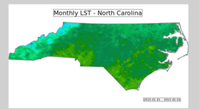
Visualization



g.gui.timeline



g.gui.tplot



g.gui.animation

List, extract, and statistics

t.rast.list,
t.rast.extract,
t.rast.univar

t.vect.list,
t.vect.extract,
t.vect.univar

Pre-processing

t.rast.colors,
t.rast.gapfill

Processing and analysis

t.rast.series

t.rast.aggregate

t.vect.what.strds
t.vect.observe.strds

t.rast.algebra

```
t.rast.algebra "C = A {+, contains} B"
t.rast.algebra "C = A + map(b)"
t.rast.algebra "C = (A[-1] + A[0] + A[1])/3"
```

t.rast.accumulate + t.rast.accdetect

ST algebra

t.vect.algebra

```
t.vect.algebra \
"C = A {|,equal|overlap,u} B"
t.vect.algebra \
"C = buff_p(A,1) {&,overlap,i} buff_p(B,1)"
```

Import/Export

t.rast.import
t.rast.export

t.vect.import
t.vect.export

Hands-on to NDVI time series for environmental monitoring

Overview

Overview

- Data for the session

Overview

- Data for the session
- Get familiar with the data

Overview

- Data for the session
- Get familiar with the data
- Use of reliability band

Overview

- Data for the session
- Get familiar with the data
- Use of reliability band
- Create NDVI time series

Overview

- Data for the session
- Get familiar with the data
- Use of reliability band
- Create NDVI time series
- Gap-filling: HANTS

Overview

- Data for the session
- Get familiar with the data
- Use of reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Phenological indices

Overview

- Data for the session
- Get familiar with the data
- Use of reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Phenological indices
- NDWI time series

Overview

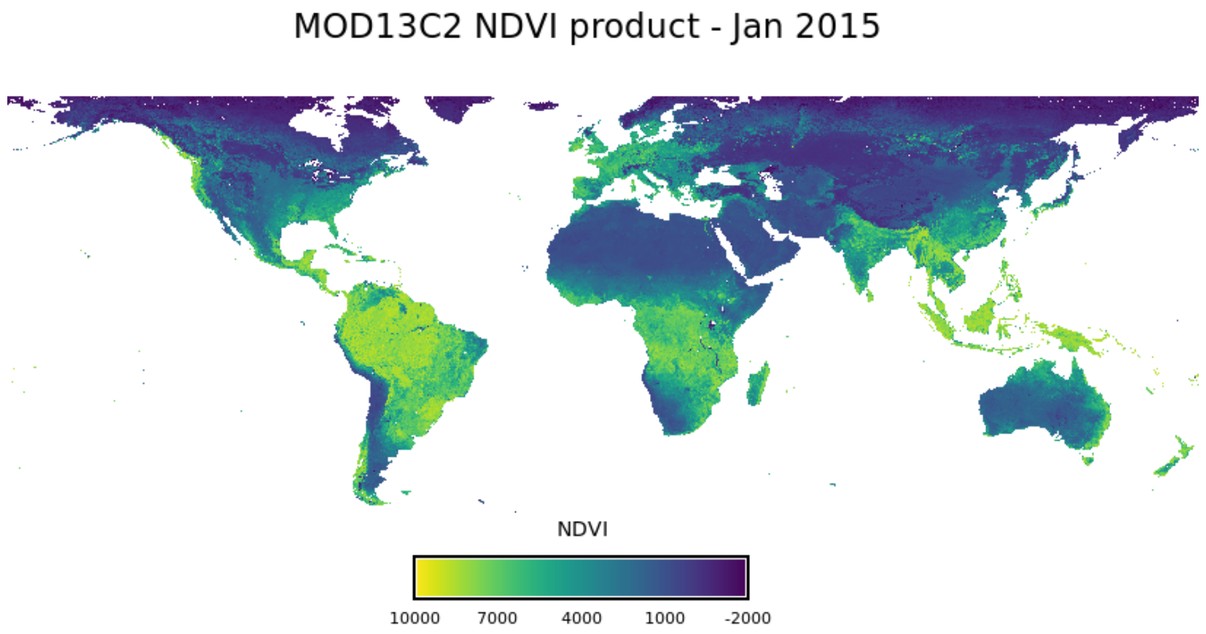
- Data for the session
- Get familiar with the data
- Use of reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Phenological indices
- NDWI time series
- Regression between NDVI and NDWI

Sample location: North Carolina

- Download the **North Carolina location**
- Create a folder in your `$HOME` directory (or Documents) and name it `grassdata`
- Unzip the file `nc_basic_ogh_2019.zip` within `grassdata`
- Download the GRASS script to follow the session

Data for the session

- MODIS Vegetation product: MOD13C2 Collection 6
- Global monthly composites
- Spatial resolution: 5600m



Let's start GRASS GIS!

Get familiar with NDVI data

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Get familiar with NDVI data

```
### END OF DO NOT RUN ###


#
# Get familiar with NDVI data
#



# Download ready to use mapset and unzip in NC location


# List maps and get basic info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

# Set computational region
g.region -p vector=nc_state \
  align=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI


# Set mask
r.mask vector=nc_state
```

Get familiar with NDVI data

```
# Get familiar with NDVI data
#



# Download ready to use mapset and unzip in NC location

# List maps and get basic info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

# Set computational region
g.region -p vector=nc_state \
  align=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI


# Set mask
r.mask vector=nc_state



#
```

Get familiar with NDVI data

```
# Download ready to use mapset and unzip in NC location

# List maps and get basic info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

# Set computational region
g.region -p vector=nc_state \
  align=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI


# Set mask
r.mask vector=nc_state


#
# Use of reliability band
#
```

Task:

- *Display EVI, NIR and QA maps and get information about minimum and maximum values*
- *What do you notice about the values?*

Use of reliability band

Task:

- *Read about this reliability band at the MOD13 User guide (pag 27)*
- *Display one of the pixel reliability bands along with NDVI band of the same date*
- *Select only pixels with value 0 (Good quality) in the pixel reliability band. What do you notice?*

Use of reliability band

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Use of reliability band

```
# Set mask
r.mask vector=nc_state


#
# Use of reliability band
#



# *nix: Keep only NDVI most reliable pixels (one map)
PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
  expression="${NDVI}_filt = if(${PR} != 0, null(), ${NDVI})"

# Windows: Keep only NDVI most reliable pixels (one map)
SET PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
SET NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
```

Use of reliability band

```
#

# *nix: Keep only NDVI most reliable pixels (one map)
PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
  expression="${NDVI}_filt = if(${PR} != 0, null(), ${NDVI})"

# Windows: Keep only NDVI most reliable pixels (one map)
SET PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
SET NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
  expression="%NDVI%_filt = if(%PR% != 0, null(), %NDVI%)"

# For all NDVI maps (Windows users run bash.exe and once done, exit)

# List of maps
PR=$(g.list type=raster pattern="*_pixel_reliability" separator=" ")
```

Use of reliability band

```
SET NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
  expression="%NDVI%_filt = if(%PR% != 0, null(), %NDVI%)"

# For all NDVI maps (Windows users run bash.exe and once done, exit)

# List of maps
PR=$(g.list type=raster pattern="*_pixel_reliability" separator=" ")
NDVI=$(g.list type=raster pattern="*_Monthly_NDVI" separator=" ")

# Convert list to array
PR=($PR)
NDVI=($NDVI)


# Iterate over the 2 arrays
for ((i=0;i<${#PR[@]};i++)) ; do
  echo ${PR[$i]} ${NDVI[$i]};
  r.mapcalc \
    expression="${NDVI[$i]}_filt = if(${PR[$i]} != 0, null(), ${NDVI[$i]})"
```

Task: *Compare stats among original and filtered NDVI maps for the same date using r.univar. Do stats differ?*

To decode QA bits from the QA band there's a specific GRASS GIS module: i.modis.qc

Create time series

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Create time series

```
  echo ${PR[$i]} ${NDVI[$i]};
  r.mapcalc \
    expression="${NDVI[$i]}_filt = if(${PR[$i]} != 0, null(), ${NDVI[$i]})"
done



#
# Create NDVI time series
#



# Create STRDS
t.create output=ndvi_monthly \
  type=strds temporaltype=absolute \
  title="Filtered monthly NDVI" \
  description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# Check if it was created
t.list type=strds

# List NDVI filtered files
```

Create time series

```
#
# Create NDVI time series
#



# Create STRDS
t.create output=ndvi_monthly \
  type=strds temporaltype=absolute \
  title="Filtered monthly NDVI" \
  description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# Check if it was created
t.list type=strds

# List NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# Register maps
t.register -i input=ndvi_monthly \
```

Create time series

```
# Create NDVI time series
#


# Create STRDS
t.create output=ndvi_monthly \
  type=strds temporaltype=absolute \
  title="Filtered monthly NDVI" \
  description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# Check if it was created
t.list type=strds

# List NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# Register maps
t.register -i input=ndvi_monthly \
  type=raster file=ndvi_list.txt \
  start="2015-01-01" increment="1 months"
```

Create time series

```
# Create STRDS
t.create output=ndvi_monthly \
  type=strds temporaltype=absolute \
  title="Filtered monthly NDVI" \
  description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# Check if it was created
t.list type=strds

# List NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# Register maps
t.register -i input=ndvi_monthly \
  type=raster file=ndvi_list.txt \
  start="2015-01-01" increment="1 months"

# Print time series info
t.info input=ndvi_monthly

# Print list of maps in the space-time raster
```

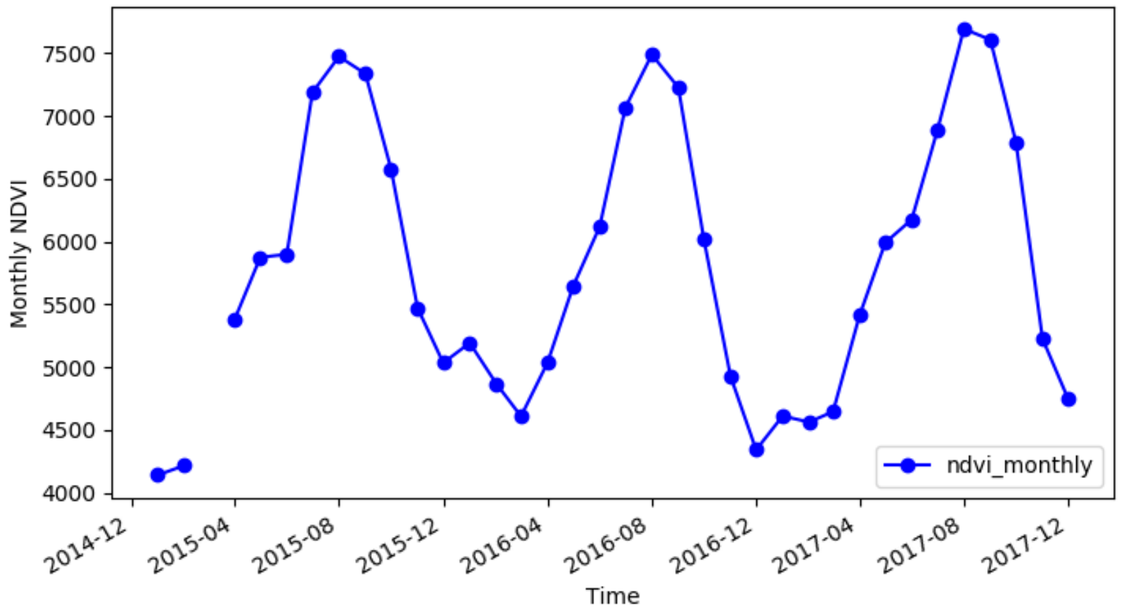
Create time series

```
  description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# Check if it was created
t.list type=strds

# List NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# Register maps
t.register -i input=ndvi_monthly \
  type=raster file=ndvi_list.txt \
  start="2015-01-01" increment="1 months"

# Print time series info
t.info input=ndvi_monthly

# Print list of maps in time series
t.rast.list input=ndvi_monthly
```

Create time series

```
t.list type=strds

# List NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# Register maps
t.register -i input=ndvi_monthly \
  type=raster file=ndvi_list.txt \
  start="2015-01-01" increment="1 months"

# Print time series info
t.info input=ndvi_monthly


# Print list of maps in time series
t.rast.list input=ndvi_monthly


#
# Estimate percentage of missing data
#
```

Task: *Visually explore the values of the time series in different points. Use g.gui.tplot and select different points interactively.*



Missing data

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Missing data

```
# Print time series info
t.info input=ndvi_monthly


# Print list of maps in time series
t.rast.list input=ndvi_monthly



#
# Estimate percentage of missing data
#



# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# Count valid data
t.rast.series input=ndvi_monthly \
  method=count \
  output=ndvi_count_valid
```

Missing data

```
# Print list of maps in time series
t.rast.list input=ndvi_monthly



#
# Estimate percentage of missing data
#



# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# Count valid data
t.rast.series input=ndvi_monthly \
  method=count \
  output=ndvi_count_valid

# Get total number of maps
eval $(t.info -g type=strds input=ndvi_monthly)
echo $number_of_maps
```

Missing data

```
#
# Estimate percentage of missing data
#



# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# Count valid data
t.rast.series input=ndvi_monthly \
  method=count \
  output=ndvi_count_valid

# Get total number of maps
eval $(t.info -g type=strds input=ndvi_monthly)
echo $number_of_maps

# Estimate percentage of missing data
r.mapcalc \
  expression="ndvi_missing = \
  (($number_of_maps - ndvi_count_valid) * 100.0)/$number_of_maps"
```

Missing data

```
# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# Count valid data
t.rast.series input=ndvi_monthly \
  method=count \
  output=ndvi_count_valid

# Get total number of maps
eval $(t.info -g type=strds input=ndvi_monthly)
echo $number_of_maps

# Estimate percentage of missing data
r.mapcalc \
  expression="ndvi_missing = \
  (($number_of_maps - ndvi_count_valid) * 100.0)/$number_of_maps"


#
# Temporal gap-filling: HANTS
```

Task:

- *Display the map representing the percentage of missing data and explore values*
- *Get univariate statistics of this map*

Hint: r.univar

Temporal gap-filling

- Harmonic Analysis of Time Series (HANTS).
- Implemented in r.hants add-on

See Roerink et al. 2000 for more details

Temporal gap-filling: HANTS

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Temporal gap-filling: HANTS

```
echo $number_of_maps

# Estimate percentage of missing data
r.mapcalc \
  expression="ndvi_missing = \
  (($number_of_maps - ndvi_count_valid) * 100.0)/$number_of_maps"


#
# Temporal gap-filling: HANTS
#


# install extension
g.extension extension=r.hants

# *nix
# List maps
maplist=$(t.rast.list input=ndvi_monthly method=comma)
```

Temporal gap-filling: HANTS

```
  expression="ndvi_missing = (
  (($number_of_maps - ndvi_count_valid) * 100.0)/$number_of_maps"


#
# Temporal gap-filling: HANTS
#



# install extension

g.extension extension=r.hants

# *nix
# List maps
maplist=$(t.rast.list input=ndvi_monthly method=comma)

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12

# Windows
# list maps
```

Temporal gap-filling: HANTS

```
# install extension
g.extension extension=r.hants

# *nix
# List maps
maplist=$(t.rast.list input=ndvi_monthly method=comma)

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12

# Windows
# list maps
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly" "method=comma"') DO SET maplis

r.hants in=%maplist% range=-2000,10000 nf=5 fet=500 base_period=12

# Patch original with filled maps
# Windows users run bash.exe, once done type exit
```

Temporal gap-filling: HANTS

```
# Windows
# list maps
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly" "method=comma"') DO SET maplis

r.hants in=%maplist% range=-2000,10000 nf=5 fet=500 base_period=12

# Patch original with filled maps
# Windows users run bash.exe, once done type exit

# Get list of maps
ORIG=$(g.list type=raster pattern="*_filt" separator=" ")
FILL=$(g.list type=raster pattern="*_hants" separator=" ")

# Convert list to array
ORIG=($ORIG)
FILL=($FILL)

# Iterate over the 2 arrays
for ((i=0;i<${#ORIG[@]};i++)) ; do
  echo ${ORIG[$i]} ${FILL[$i]};
```

Temporal gap-filling: HANTS

```
ORIG=$(g.list type=raster pattern="*_fill" separator=" ")
FILL=$(g.list type=raster pattern="*_hants" separator=" ")

# Convert list to array
ORIG=($ORIG)
FILL=($FILL)

# Iterate over the 2 arrays
for ((i=0;i<${#ORIG[@]};i++)) ; do
  echo ${ORIG[$i]} ${FILL[$i]};
  r.patch input=${ORIG[$i]},${FILL[$i]} output=${FILL[$i]}_patch --o
done

# Create new time series

t.create output=ndvi_monthly_patch \
  type=strds temporaltype=absolute \
  title="Patched monthly NDVI" \
  description="Filtered, gap-filled and patched monthly NDVI - MOD13C2 - 2015-2017"

# List NDVI patched files
  list type=raster pattern="*patch" \
```

Temporal gap-filling: HANTS

```
for ((i=0;i<${#ORIG[@]};i++)) ; do
  echo ${ORIG[$i]} ${FILL[$i]};
  r.patch input=${ORIG[$i]},${FILL[$i]} output=${FILL[$i]}_patch --o
done

# Create new time series

t.create output=ndvi_monthly_patch \
  type=strds temporaltype=absolute \
  title="Patched monthly NDVI" \
  description="Filtered, gap-filled and patched monthly NDVI - MOD13C2 - 2015-2017"

# List NDVI patched files
g.list type=raster pattern="*patch" \
  output=list_ndvi_patched.txt

# Register maps
t.register -i input=ndvi_monthly_patch \
  type=raster file=list_ndvi_patched.txt \
  start="2015-01-01" increment="1 months"
```

Temporal gap-filling: HANTS

```
t.create output=ndvi_monthly_patch \
  type=strds temporaltype=absolute \
  title="Patched monthly NDVI" \
  description="Filtered, gap-filled and patched monthly NDVI - MOD13C2 - 2015-2017"

# List NDVI patched files
g.list type=raster pattern="*patch" \
  output=list_ndvi_patched.txt

# Register maps
t.register -i input=ndvi_monthly_patch \
  type=raster file=list_ndvi_patched.txt \
  start="2015-01-01" increment="1 months"

# Print time series info
t.info input=ndvi_monthly_patch


#
# Obtain phenological information
```

Task:

- *Graphically assess the results of HANTS reconstruction in pixels with higher percentage of missing data*
- *Obtain univariate statistics for the new time series*

Hints: g.gui.tplot and t.rast.univar

Phenology

- The study of periodic plant and animal life cycle events and how these are influenced by seasonal and interannual variations in climate
- We will estimate indices to characterize different aspects of phenology:
  - min and max NDVI values
  - dates of min and max NDVI values
  - rate of change
  - length, start and end of growing season

Phenological indices

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Phenological indices

```
  type=raster file=first_ndvi_parsed.txt \
  start="2015-01-01" increment="1 months"

# Print time series info
t.info input=ndvi_monthly_patch


#
# Obtain phenological information
#


# Get NDVI maximum and minimum
t.rast.series input=ndvi_monthly_patch \
  method=maximum \
  output=ndvi_max

t.rast.series input=ndvi_monthly_patch \
  method=minimum \
  output=ndvi_min
```

Phenological indices

```
# Obtain phenological information
#



# Get NDVI maximum and minimum
t.rast.series input=ndvi_monthly_patch \
  method=maximum \
  output=ndvi_max

t.rast.series input=ndvi_monthly_patch \
  method=minimum \
  output=ndvi_min

# Get month of maximum

t.rast.mapcalc -n inputs=ndvi_monthly_patch \
  output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())" \
  basename=month_max_ndvi

# Get the earliest month in which the maximum appeared
```

Phenological indices

```
t.rast.series input=ndvi_monthly_patch \
  method=maximum \
  output=ndvi_max

t.rast.series input=ndvi_monthly_patch \
  method=minimum \
  output=ndvi_min

# Get month of maximum

t.rast.mapcalc -n inputs=ndvi_monthly_patch \
  output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())" \
  basename=month_max_ndvi

# Get the earliest month in which the maximum appeared
t.rast.series input=month_max_ndvi \
  method=minimum \
  output=max_ndvi_date

# Get NDVI minimum per year starting in December
```

Phenological indices

```
# Get month of maximum

t.rast.mapcalc -n inputs=ndvi_monthly_patch \
  output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())" \
  basename=month_max_ndvi

# Get the earliest month in which the maximum appeared
t.rast.series input=month_max_ndvi \
  method=minimum \
  output=max_ndvi_date

# Get NDVI minimum per year starting in December
t.rast.aggregate input=ndvi_monthly_patch \
  where="start_time >= '2015-12-01' AND start_time <= '2017-11-30'" \
  method=min_raster \
  granularity="12 months" \
  output=annual_index_min_ndvi \
  basename=index_ndvi_min \
  suffix=gran
```

Phenological indices

```
# Get the earliest month in which the maximum appeared
t.rast.series input=month_max_ndvi \
  method=minimum \
  output=max_ndvi_date

# Get NDVI minimum per year starting in December
t.rast.aggregate input=ndvi_monthly_patch \
  where="start_time >= '2015-12-01' AND start_time <= '2017-11-30'" \
  method=min_raster \
  granularity="12 months" \
  output=annual_index_min_ndvi \
  basename=index_ndvi_min \
  suffix=gran

# Get index of minimum raster
t.rast.series input=annual_index_min_ndvi \
  method=minimum \
  output=min_ndvi_index

# Index to month reclass rules
```

Phenological indices

```
  granularity="12 months" \
  output=annual_index_min_ndvi \
  basename=index_ndvi_min \
  suffix=gran

# Get index of minimum raster
t.rast.series input=annual_index_min_ndvi \
  method=minimum \
  output=min_ndvi_index

# Index to month reclass rules
echo "0 = 12
1 = 1
2 = 2
3 = 3
4 = 4
5 = 5
6 = 6
7 = 7

8 = 8
```

Phenological indices

```
# Index to month reclass rules
echo "0 = 12
1 = 1
2 = 2
3 = 3
4 = 4
5 = 5
6 = 6
7 = 7

8 = 8
9 = 9
10 = 10
11 = 11" >> index2month.txt

# Reclass index to month
r.reclass input=min_ndvi_index output=min_ndvi_date \
  rules=index2month.txt

# Remove intermediate strds
```

Phenological indices

```
1 = 1
2 = 2
3 = 3
4 = 4
5 = 5
6 = 6
7 = 7

8 = 8
9 = 9
10 = 10
11 = 11" >> index2month.txt

# Reclass index to month
r.reclass input=min_ndvi_index output=min_ndvi_date \
  rules=index2month.txt

# Remove intermediate strds
t.remove -rf inputs=month_max_ndvi,annual_index_min_ndvi

# Add `modis_lst` to accessible mapsets path
mapsets -p
```

GRASS GIS

Task: *Display `max_ndvi_date` and `min_ndvi_date` maps from the terminal using wx monitors*

Hints: d.mon and d.rast

Task: *Associate max and min LST with max and min NDVI, and max and min LST dates with max and min NDVI dates.*

Hints: g.mapsets and r.covar

Phenological indices

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Phenological indices

```
5 = 5
6 = 6
7 = 7

8 = 8
9 = 9
10 = 10
11 = 11" >> index2month.txt

# Reclass index to month
r.reclass input=min_ndvi_index output=min_ndvi_date \
  rules=index2month.txt

# Remove intermediate strds
t.remove -rf inputs=month_max_ndvi,annual_index_min_ndvi

# Add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# Estimate global correlations
r.covar -r map=LST_Day_max,LST_Day_min,ndvi_max,ndvi_min
```

Phenological indices

```
8 = 8
9 = 9
10 = 10
11 = 11" >> index2month.txt

# Reclass index to month
r.reclass input=min_ndvi_index output=min_ndvi_date \
  rules=index2month.txt

# Remove intermediate strds
t.remove -rf inputs=month_max_ndvi,annual_index_min_ndvi

# Add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# Estimate global correlations
r.covar -r map=LST_Day_max,LST_Day_min,ndvi_max,ndvi_min
r.covar -r map=max_lst_date,min_lst_date,max_ndvi_date,min_ndvi_date

# Time series of slopes
```

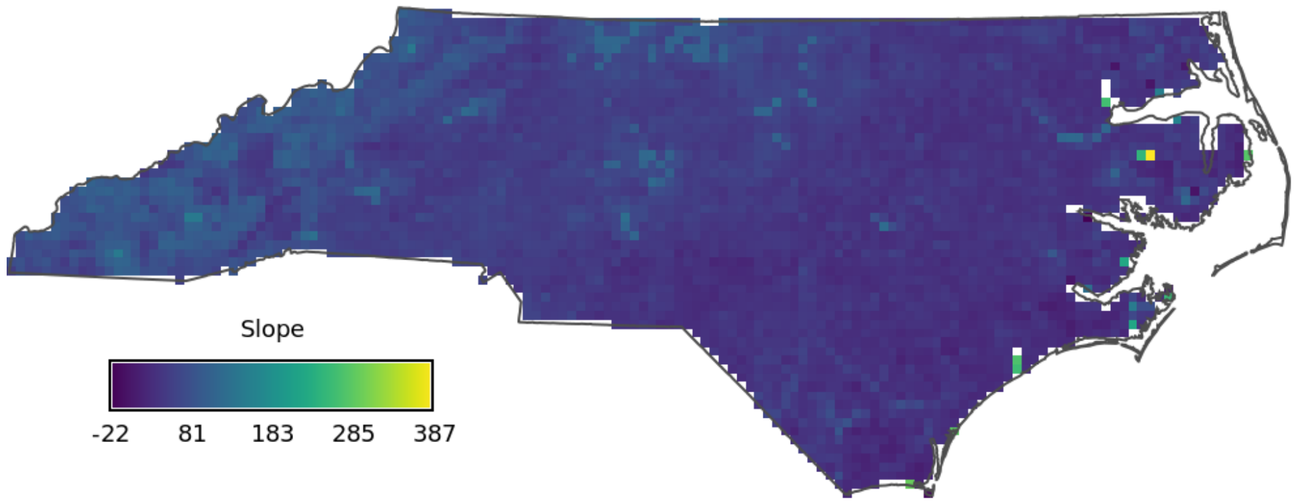
Phenological indices

```
# Reclass index to month
r.reclass input=min_ndvi_index output=min_ndvi_date \
  rules=index2month.txt

# Remove intermediate strds
t.remove -rf inputs=month_max_ndvi,annual_index_min_ndvi

# Add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# Estimate global correlations
r.covar -r map=LST_Day_max,LST_Day_min,ndvi_max,ndvi_min
r.covar -r map=max_lst_date,min_lst_date,max_ndvi_date,min_ndvi_date

# Time series of slopes
t.rast.algebra \
  expression="slope_ndvi = (ndvi_monthly_patch[1] - ndvi_monthly_patch[0]) / td(ndv
  basename=slope_ndvi \
  suffix=gran
```

Phenological indices

```
# Add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# Estimate global correlations
r.covar -r map=LST_Day_max,LST_Day_min,ndvi_max,ndvi_min
r.covar -r map=max_lst_date,min_lst_date,max_ndvi_date,min_ndvi_date

# Time series of slopes
t.rast.algebra \
  expression="slope_ndvi = (ndvi_monthly_patch[1] - ndvi_monthly_patch[0]) / td(ndv
  basename=slope_ndvi \
  suffix=gran

# Get max slope per year
t.rast.aggregate input=slope_ndvi \
  output=ndvi_slope_yearly \
  basename=NDVI_max_slope_year \
  suffix=gran \
  method=maximum \
  granularity="1 years"
```

☑☰ ***Task***: *Obtain a map with the highest growing rate per pixel in the period 2015-2017 and display it from the terminal*



Hint: t.rast.series

Phenological indices

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Phenological indices

```
r.covar -r map=max_lst_date,min_lst_date,max_ndvi_date,min_ndvi_date

# Time series of slopes
t.rast.algebra \
  expression="slope_ndvi = (ndvi_monthly_patch[1] - ndvi_monthly_patch[0]) / td(ndv
  basename=slope_ndvi \
  suffix=gran

# Get max slope per year
t.rast.aggregate input=slope_ndvi \
  output=ndvi_slope_yearly \
  basename=NDVI_max_slope_year \
  suffix=gran \
  method=maximum \
  granularity="1 years"


# Install extension
g.extension extension=r.seasons

# *nix: Start, end and length of growing season
```

Phenological indices

```
  basename=slope_ndvi \
  suffix=gran

# Get max slope per year
t.rast.aggregate input=slope_ndvi \
  output=ndvi_slope_yearly \
  basename=NDVI_max_slope_year \
  suffix=gran \
  method=maximum \
  granularity="1 years"


# Install extension
g.extension extension=r.seasons

# *nix: Start, end and length of growing season
r.seasons input=$(t.rast.list -u input=ndvi_monthly_patch method=comma) \
  prefix=ndvi_season \
  n=3 \
  nout=ndvi_season \
  threshold_value=3000 \
  min_length=5 \
```

Phenological indices

```
  suffix=gran \
  method=maximum \
  granularity="1 years"


# Install extension
g.extension extension=r.seasons

# *nix: Start, end and length of growing season
r.seasons input=$(t.rast.list -u input=ndvi_monthly_patch method=comma) \
  prefix=ndvi_season \
  n=3 \
  nout=ndvi_season \
  threshold_value=3000 \
  min_length=5 \
  max_gap=4

# Windows: Start, end and length of growing season
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=," "method=co

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season threshold_value
```

Task: *Display some of the resulting maps. What do they represent?*

Check r.seasons manual page

Phenological indices

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

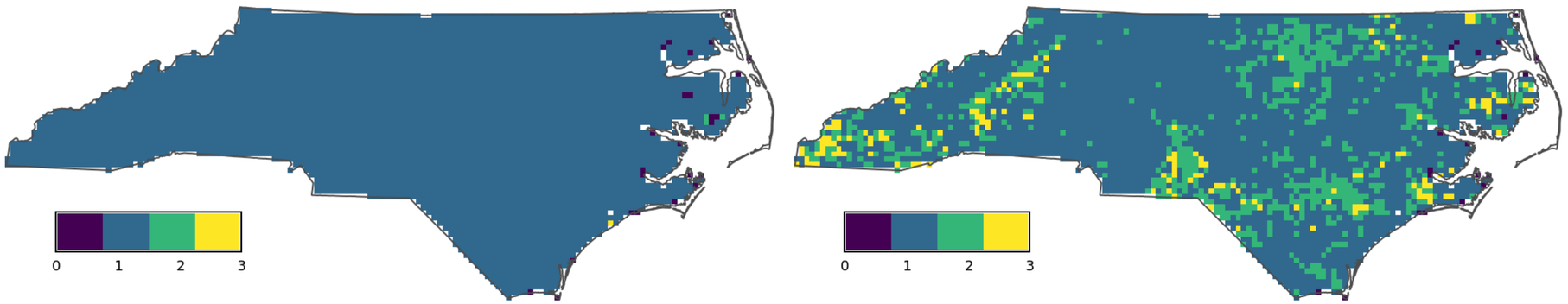
Phenological indices

```
# Install extension
g.extension extension=r.seasons

# *nix: Start, end and length of growing season
r.seasons input=$(t.rast.list -u input=ndvi_monthly_patch method=comma) \
  prefix=ndvi_season \
  n=3 \
  nout=ndvi_season \
  threshold_value=3000 \
  min_length=5 \
  max_gap=4

# Windows: Start, end and length of growing season
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=," "method=co

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season threshold_value

# Create a threshold map: min ndvi + 0.1*ndvi
r.mapcalc expression="threshold_ndvi = ndvi_min*1.1"

r.seasons input=$(t.rast.list -u input=ndvi_monthly_patch method=comma) \
```

Phenological indices

```
  prefix=ndvi_season \
  n=3 \
  nout=ndvi_season \
  threshold_value=3000 \
  min_length=5 \
  max_gap=4

# Windows: Start, end and length of growing season
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=," "method=co

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season threshold_value

# Create a threshold map: min ndvi + 0.1*ndvi
r.mapcalc expression="threshold_ndvi = ndvi_min*1.1"

r.seasons input=$(t.rast.list -u input=ndvi_monthly_patch method=comma) \
  prefix=ndvi_season_thres \
  n=3 \
  nout=ndvi_season_thres \
  threshold_map=threshold_ndvi \
  min_length=5 \
```

Task: *Use the threshold map in r.seasons and compare output maps with the outputs of using a unique threshold value*



Number of seasons with fixed threshold and using a varying threshold map

Water index time series

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Water index time series

```
  prefix=ndvi_season_thres \
  n=3 \
  nout=ndvi_season_thres \
  threshold_map=threshold_ndvi \
  min_length=5 \
  max_gap=4



#
# Estimate NDWI
#



# Create time series of NIR and MIR
t.create output=NIR \
  type=strds temporaltype=absolute \
  title="NIR monthly" \
  description="NIR monthly - MOD13C2 - 2015-2017"

t.create output=MIR \
  type=strds temporaltype=absolute \
```

Water index time series

```
#
# Estimate NDWI
#



# Create time series of NIR and MIR
t.create output=NIR \
  type=strds temporaltype=absolute \
  title="NIR monthly" \
  description="NIR monthly - MOD13C2 - 2015-2017"

t.create output=MIR \
  type=strds temporaltype=absolute \
  title="MIR monthly" \
  description="MIR monthly - MOD13C2 - 2015-2017"

# List NIR and MIR files
g.list type=raster pattern="*NIR*" output=list_nir.txt
g.list type=raster pattern="*MIR*" output=list_mir.txt
```

Water index time series

```
t.create output=NIR \
  type=strds temporaltype=absolute \
  title="NIR monthly" \
  description="NIR monthly - MOD13C2 - 2015-2017"

t.create output=MIR \
  type=strds temporaltype=absolute \
  title="MIR monthly" \
  description="MIR monthly - MOD13C2 - 2015-2017"

# List NIR and MIR files
g.list type=raster pattern="*NIR*" output=list_nir.txt
g.list type=raster pattern="*MIR*" output=list_mir.txt

# Register maps
t.register -i input=NIR \
  type=raster file=list_nir.txt \
  start="2015-01-01" \
  increment="1 months"

t.register -i input=MIR \
```

Water index time series

```
  title="MIR monthly" \
  description="MIR monthly - MOD13C2 - 2015-2017"

# List NIR and MIR files
g.list type=raster pattern="*NIR*" output=list_nir.txt
g.list type=raster pattern="*MIR*" output=list_mir.txt

# Register maps
t.register -i input=NIR \
  type=raster file=list_nir.txt \
  start="2015-01-01" \
  increment="1 months"

t.register -i input=MIR \
  type=raster file=list_mir.txt \
  start="2015-01-01" \
  increment="1 months"

# Print time series info
t.info input=NIR
t.info input=MIR
```

Water index time series

```
g.list type=raster pattern="*MIR*" output=list_mir.txt

# Register maps
t.register -i input=NIR \
  type=raster file=list_nir.txt \
  start="2015-01-01" \
  increment="1 months"

t.register -i input=MIR \
  type=raster file=list_mir.txt \
  start="2015-01-01" \
  increment="1 months"

# Print time series info
t.info input=NIR
t.info input=MIR

# Estimate NDWI time series

t.rast.algebra basename=ndwi_monthly suffix=gran \
  expression="ndwi_monthly = if(NIR > 0 && MIR > 0, \
```

GRASS GIS

Task: *Get maximum and minimum values for each NDWI map and explore the time series plot in different points interactively*

Hints: t.rast.list and g.gui.tplot

Regression analysis

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018. Edits: December 2018, April 2019
############################################################################



#
# Data download and preparation
#



### DO NOT RUN ###

#~ # start GRASS GIS in NC full location and create a new mapset
#~ g.mapset -c mapset=modis_ndvi

#~ # add modis_lst to path
#~ g.mapsets -p
#~ g.mapsets mapset=modis_lst operation=add
```

Regression analysis

```
t.info input=MIR

# Estimate NDWI time series

t.rast.algebra basename=ndwi_monthly suffix=gran \
  expression="ndwi_monthly = if(NIR > 0 && MIR > 0, \
  (float(NIR - MIR) / float(NIR + MIR)), null())"

# List maps in NDWI time series
t.rast.list ndwi_monthly \
  columns=name,start_time,end_time,min,max


#
# Regression between NDWI and NDVI
#



# Install extension
g.extension extension=r.regression.series
```

Regression analysis

```
t.rast.algebra basename=ndwi_monthly suffix=gran \
  expression="ndwi_monthly = if(NIR > 0 && MIR > 0, \
  (float(NIR - MIR) / float(NIR + MIR)), null())"

# List maps in NDWI time series
t.rast.list ndwi_monthly \
  columns=name,start_time,end_time,min,max



#
# Regression between NDWI and NDVI
#



# Install extension
g.extension extension=r.regression.series

# Rescale NDVI values
t.rast.algebra \
  expression="ndvi_monthly_rescaled = ndvi_monthly_patch * 0.0001" \
  basename=ndvi_rescaled \
```

Regression analysis

```
  columns=name,start_time,end_time,min,max


#
# Regression between NDWI and NDVI
#



# Install extension
g.extension extension=r.regression.series

# Rescale NDVI values
t.rast.algebra \
  expression="ndvi_monthly_rescaled = ndvi_monthly_patch * 0.0001" \
  basename=ndvi_rescaled \
  suffix=gran

# *nix
xseries=$(t.rast.list input=ndvi_monthly_rescaled method=comma)
yseries=$(t.rast.list input=ndwi_monthly method=comma)
```

Regression analysis

```
#
# Regression between NDWI and NDVI
#


# Install extension
g.extension extension=r.regression.series

# Rescale NDVI values
t.rast.algebra \
  expression="ndvi_monthly_rescaled = ndvi_monthly_patch * 0.0001" \
  basename=ndvi_rescaled \
  suffix=gran

# *nix
xseries=$(t.rast.list input=ndvi_monthly_rescaled method=comma)
yseries=$(t.rast.list input=ndwi_monthly method=comma)

r.regression.series xseries=$xseries yseries=$yseries \
```

Task: *Where is the highest correlation among NDVI and NDWI?*

QUESTIONS?

Other (very) useful resources

- Temporal data processing wiki
- GRASS GIS and R for time series processing wiki
- GRASS GIS temporal workshop at NCSU
- GRASS GIS workshop held in Jena 2018
- GRASS GIS course IRSAE 2018
- GRASS GIS course in Argentina 2018

References

- Gebbert, S., Pebesma, E. (2014). *A temporal GIS for field based environmental modeling*. Environmental Modelling & Software, 53, 1-12. DOI
- Gebbert, S., Pebesma, E. (2017). *The GRASS GIS temporal framework*. International Journal of Geographical Information Science 31, 1273-1292. DOI
- Gebbert, S., Leppelt, T. and Pebesma, E. (2019). *A Topology Based Spatio-Temporal Map Algebra for Big Data Analysis*. Data, 4, 86. DOI

Thanks for your attention!!

Verónica Andreo

veroandreo

@VeronicaAndreo

veroandreo@gmail.com

Presentation powered by

GitPitch