🔥 Wildfire Modeling in Yosemite National Park

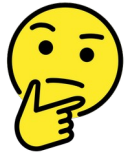
Source: https://abc7.com

Abraham Coiman

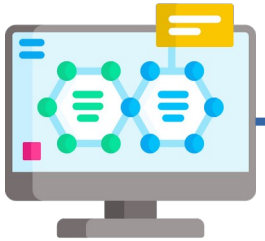
Agenda

1. Getting data 
2. Importing data 
3. Wildfire simulation 
4. Conclusions 

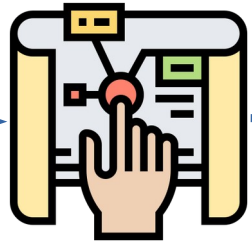
Introduction



Why do we model wildfires?



Computational science



Simulation

Fire behavior

Wildfire modeling applications:

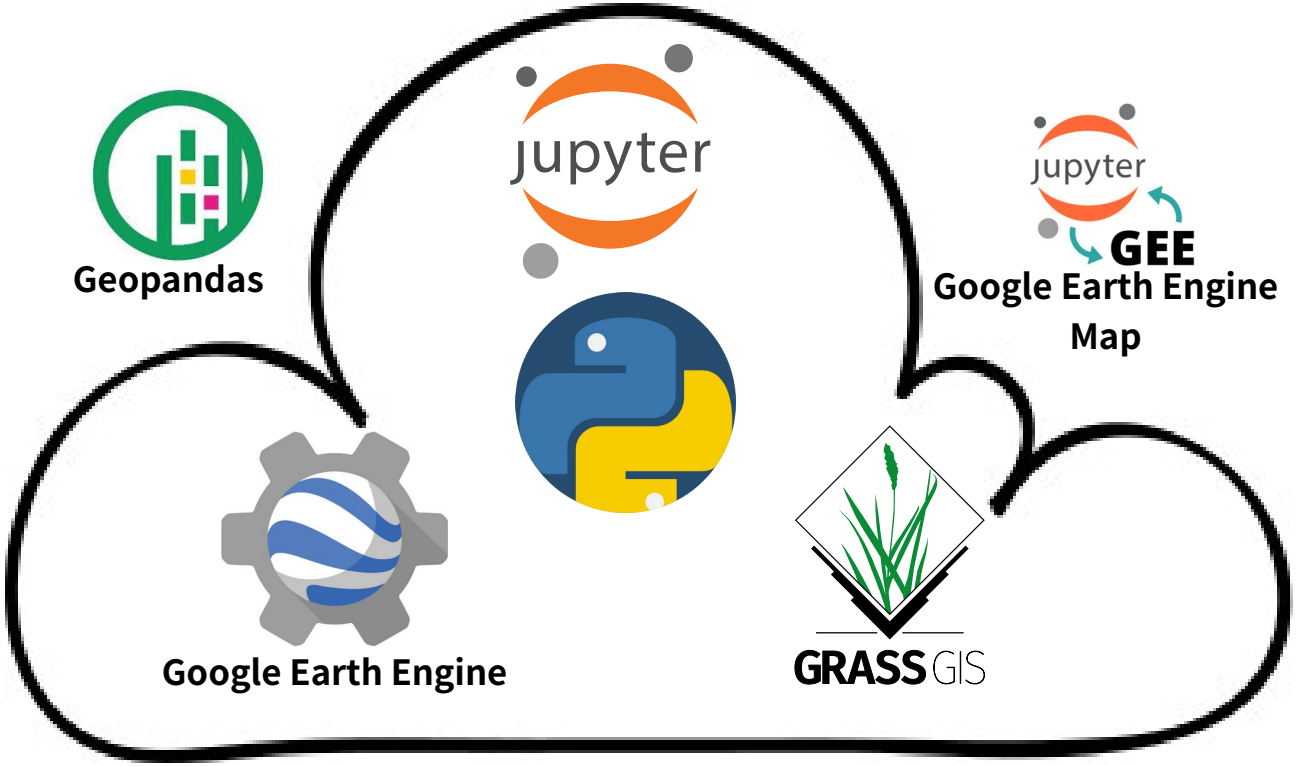
- Firefighter security
- Damage reduction
- Protection of ecosystem services

Introduction

Data for wildfire simulations

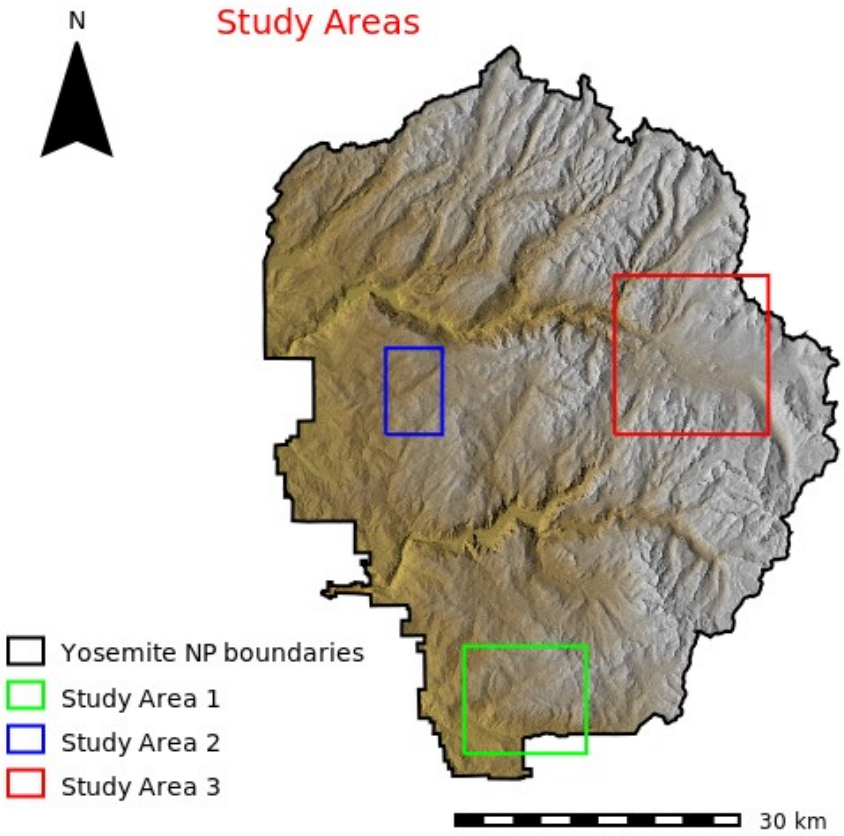
- Daunting activity
- Data are not readily available
- Data are sparse
- Data are spread out

Introduction



Development environment

Introduction



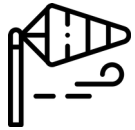
- Summer season in 2020
- June - September

Getting data

Datasets

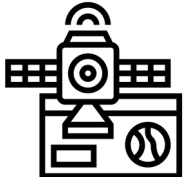

Protected area (polygon)


Dead fuel moisture


Wind direction


Wind Speed


Enhanced Vegetation Index (EVI)

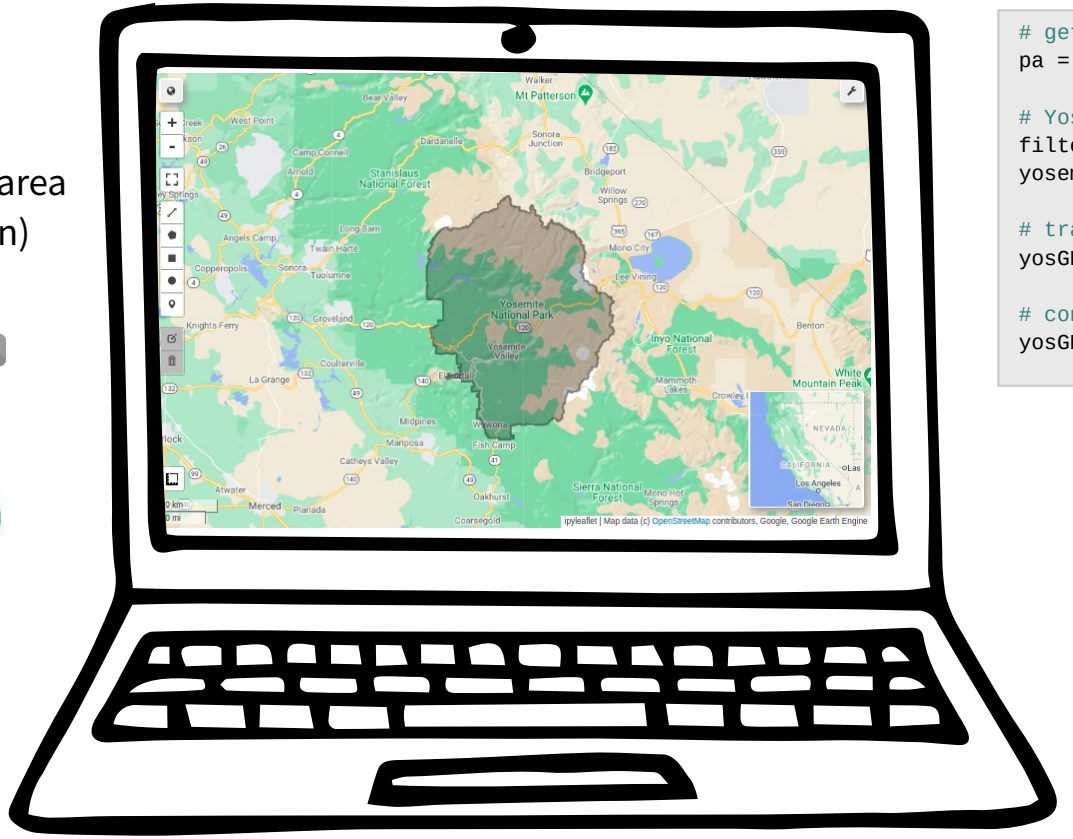

Landsat 8 image


High resolution image


Fire behavior fuel model

Getting data

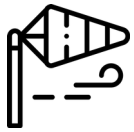
Protected area (polygon)




```
# get World Data Protected Areas (WDPA) FeatureCollection
pa = ee.FeatureCollection("WCMC/WDPA/current/polygons")

# Yosemite National Park polygon
filter = ee.Filter.inList('NAME', ['Yosemite National Park'])
yosemite = pa.filter(filter)

# transform Yosemite fc into gdf
yosGDF = eeconvert.fcToGdf(yosemite)

# convert gdf into shp
yosGDF.to_file("gisdata/yosemite/shp/yosemite.shp")
```

Getting data

Dead fuel moisture



Wind direction

Wind speed


```
# load GRIDMET ImageCollection
gridmet = ee.ImageCollection('IDAHO_EPSCOR/GRIDMET').filter(ee.Filter.date('2020-06-20', '2020-09-22'))

# select 100-hour dead fuel moisture (fm100) and calculate the mode
fm100 = gridmet.select('fm100').mode()

# clip data to rectangle
fm100_clip =fm100.clip(bbox)

# sample points using the mean
samplefm100 = fm100_clip.reduceRegions(**{
  'collection':fm100_points,
  'reducer':ee.Reducer.mean(),
  'scale': 4000,
  });

# convert gdf into shp
samplefm100GDF.to_file("gisdata/yosemite/shp/samplefm100.shp")
```

Getting data

Source: Earth Engine Data Catalog

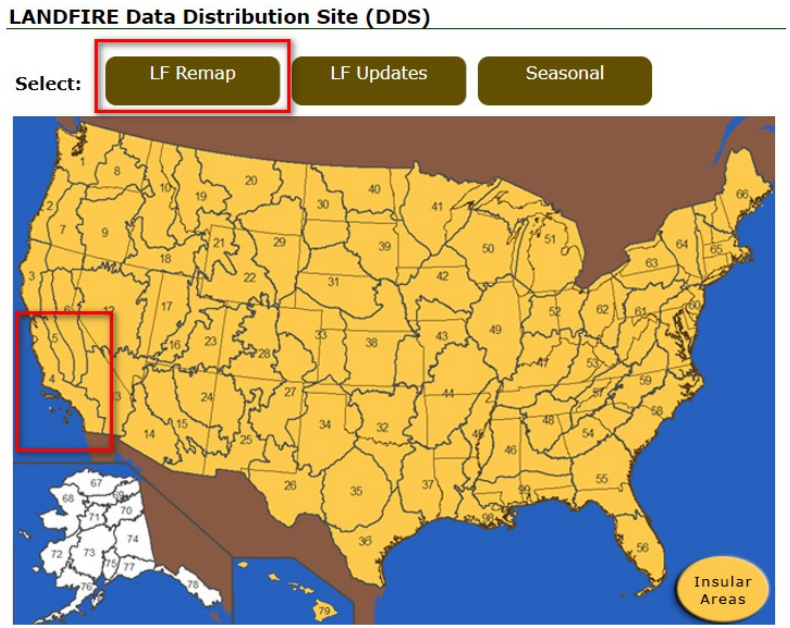
Enhanced Vegetation Index (EVI)

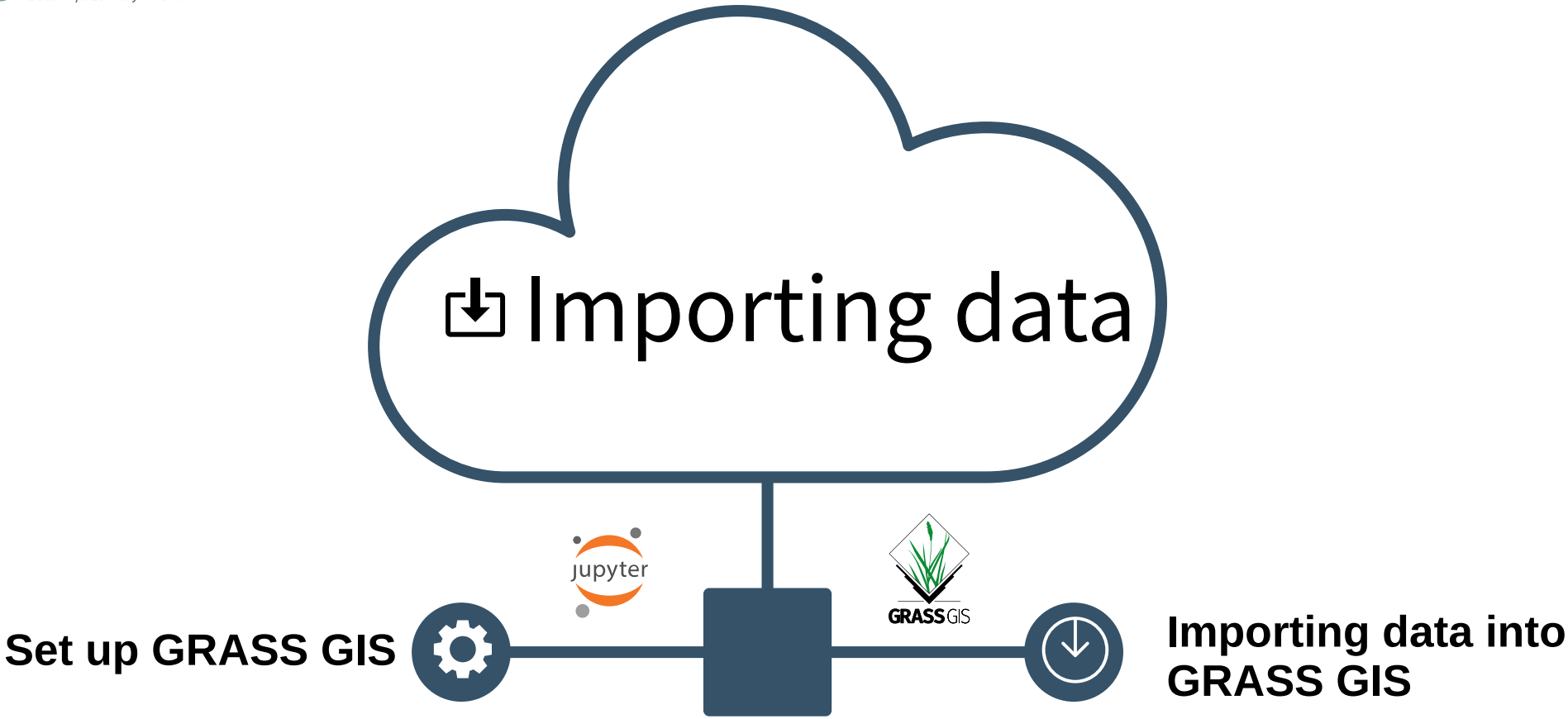
- Call the Landsat 8 Collection EVI Composite.
- Filter by date and bounds.
- Calculate the mean Image and clip.
- Export to Google Drive and download to local directory

Getting data

Fire behavior fuel model

- 13 Anderson Fire Behavior Fuel Model.
- 13 classes of fuel.
- Available at LANDFIRE program website : https://landfire.cr.usgs.gov




Importing data
jupyter
GRASS GIS
Set up GRASS GIS
Importing data into GRASS GIS

Importing data

Setting up GRASS in Jupyter Notebook

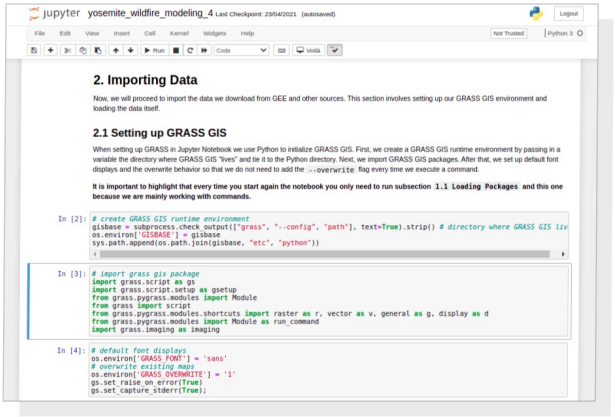
Python to initialize GRASS GIS.

- GRASS GIS runtime environment.

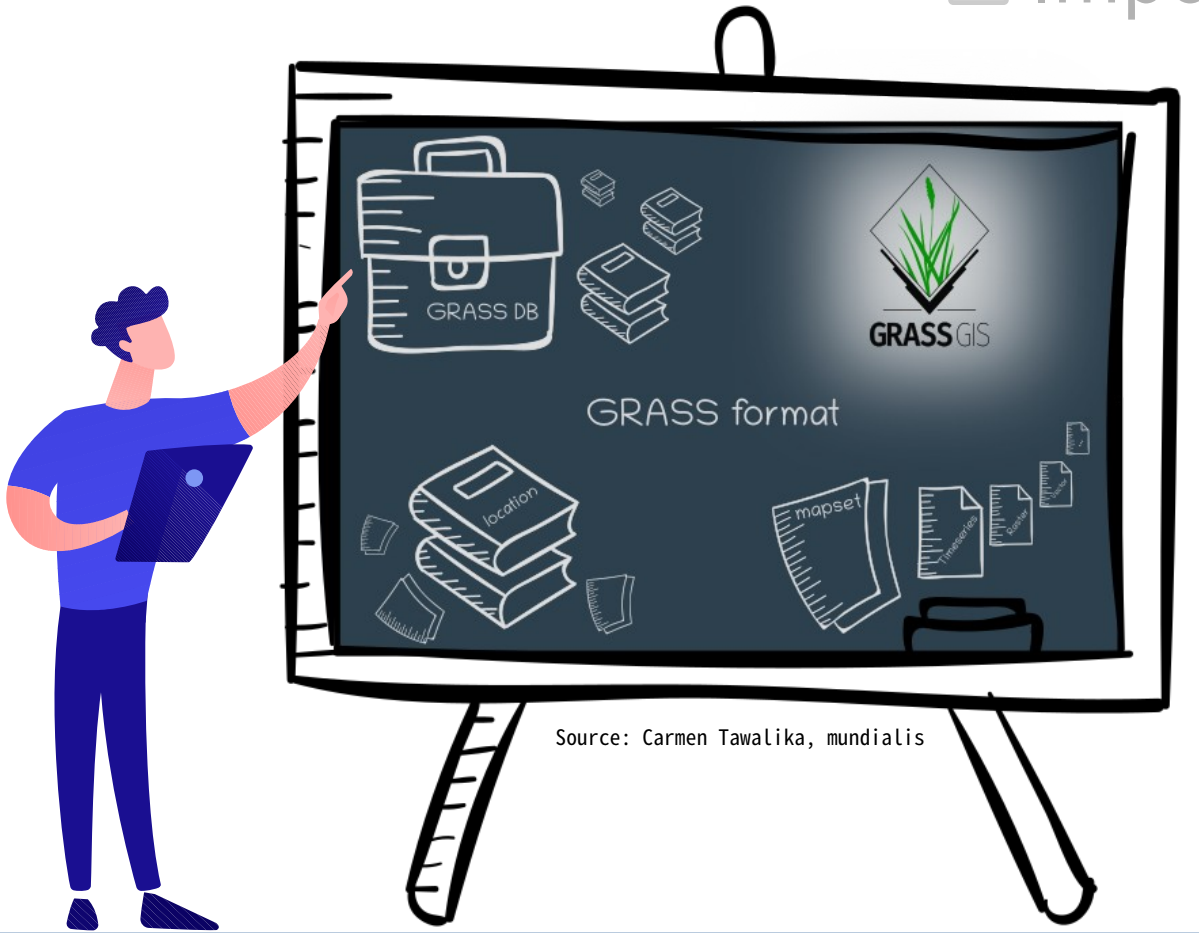
```
# create GRASS GIS runtime environment
gisbase = subprocess.check_output(["grass", "--config", "path"], text=True).strip()
os.environ['GISBASE'] = gisbase
sys.path.append(os.path.join(gisbase, "etc", "python"))
```

- Import GRASS GIS packages

```
# import grass gis packages
import grass.script as gs
import grass.script.setup as gsetup
from grass.pygrass.modules import Module
...
```

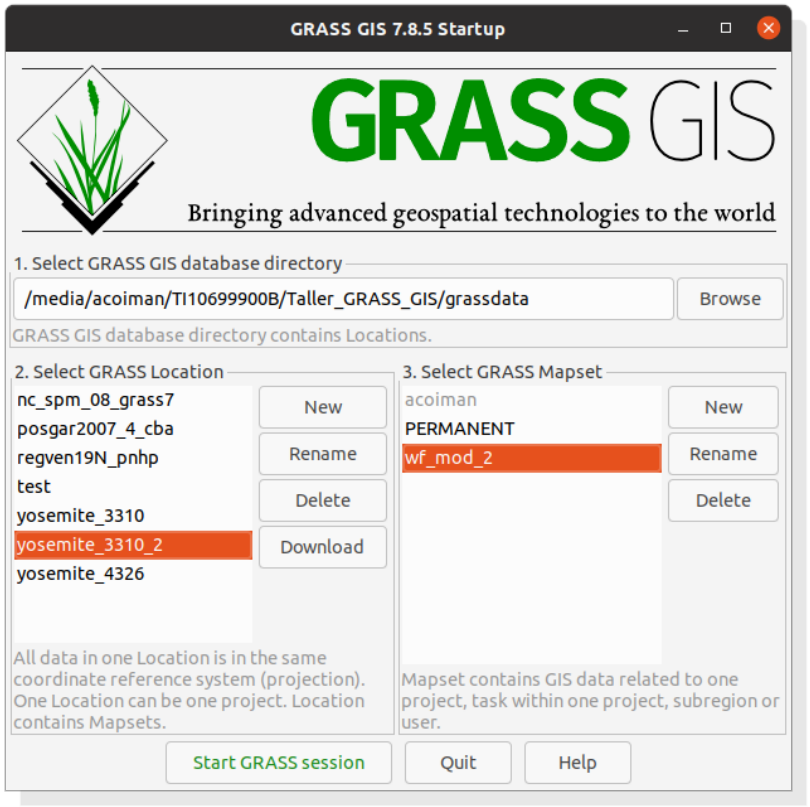


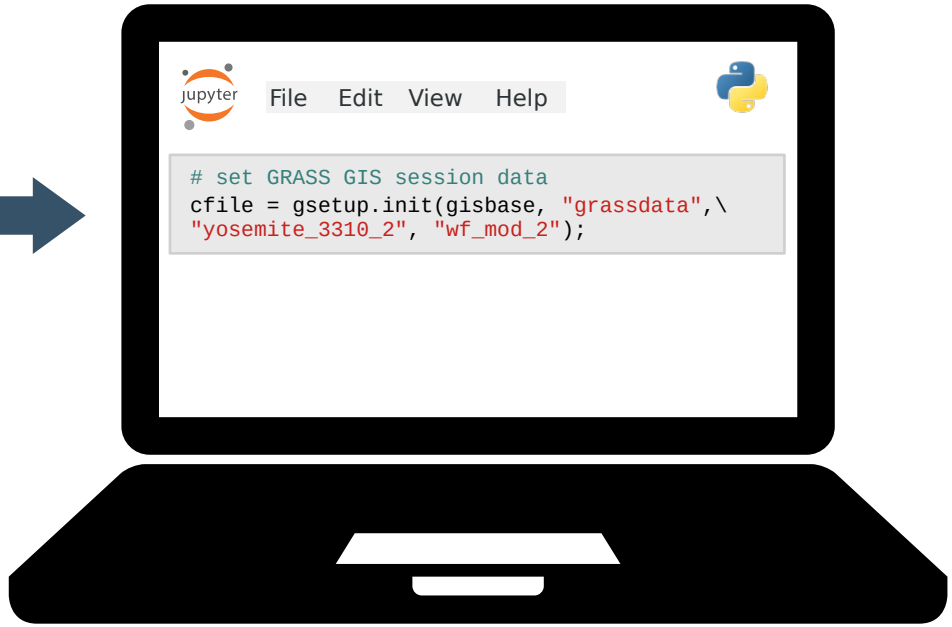
Importing data



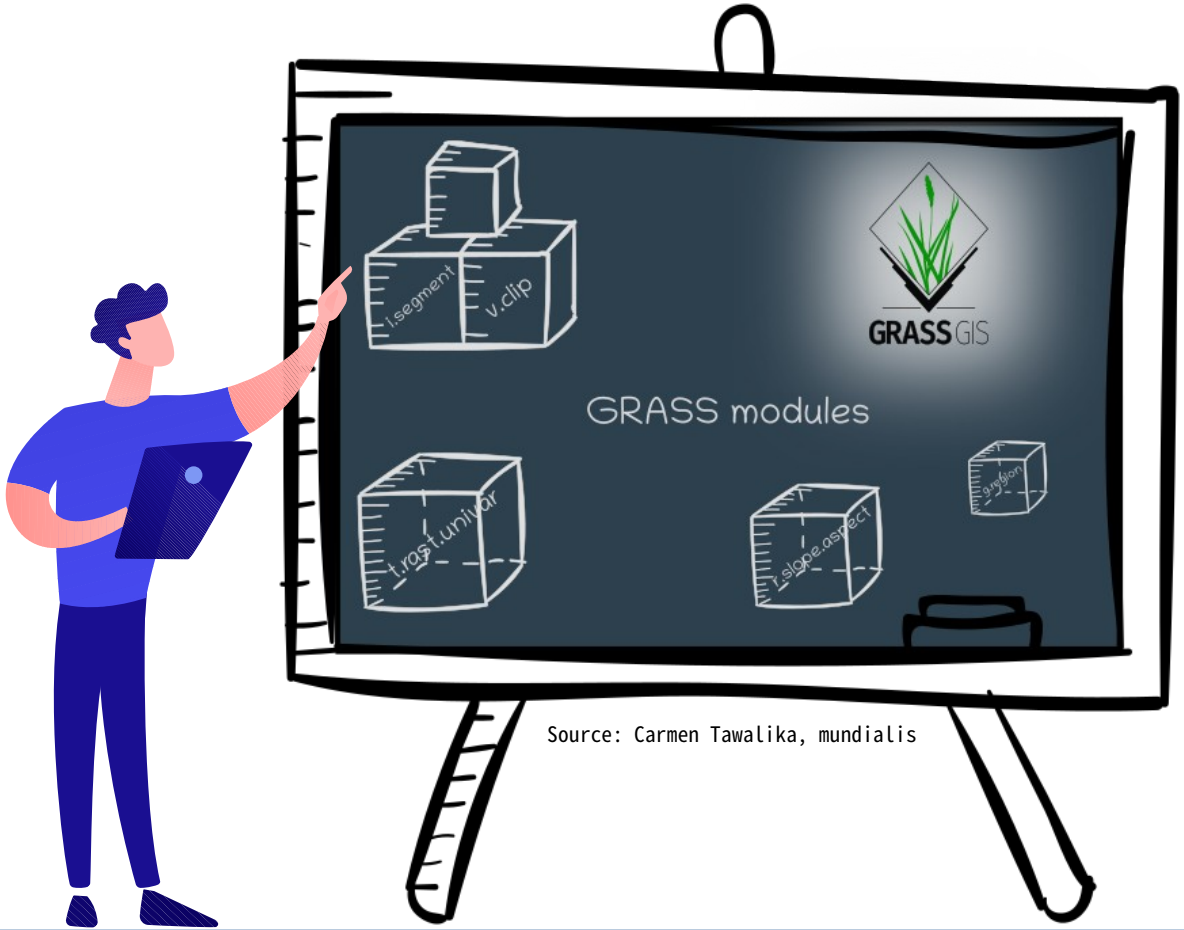
Source: Carmen Tawalika, mundialis

Importing data





Importing data



Source: Carmen Tawalika, mundialis

group	prefix	examples
general	g.*	g.list, g.remove, g.copy
raster	r.*	r.univar, r.contour
vector	v.*	v.info, v.select, v.db
imagery	I.*	I.segment, I.cluster
3D raster	r3.*	r3.info, r3.colors
temporal	t.*	t.list, t.rast, t.vect

Importing data

Importing data into GRASS GIS

```
script.run_command('v.import', input='gisdata/yosemite/shp/yosemite.shp',output='yosemite')
```

GRASS GIS Python Function

GRASS module

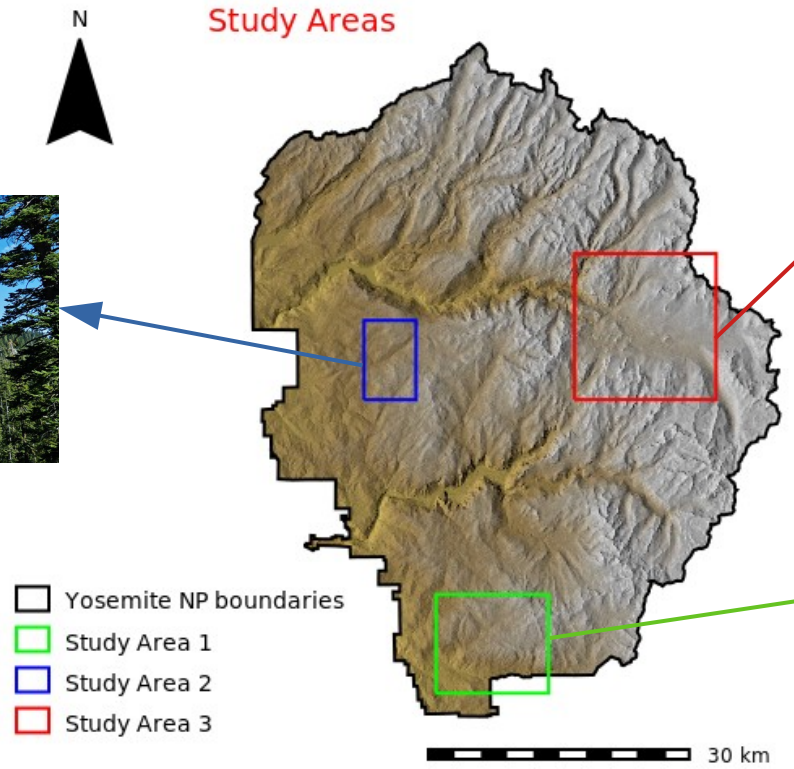
Name and path of the dataset to be imported

Name for output dataset

```
# import evi raster
script.run_command('r.import', input='gisdata/yosemite/raster/evi.tif', output='evi', resolution='value', resolution_value=30.0)

# import fm 100h samples
script.run_command('v.import', input='gisdata/yosemite/shp/samplefm100.shp',output='samplefm100')

# import vs (wind speed) samples
script.run_command('v.import', input='gisdata/yosemite/shp/samplevs.shp', output='samplevs')
```

Wildfire modeling

Source: Google Maps Photos

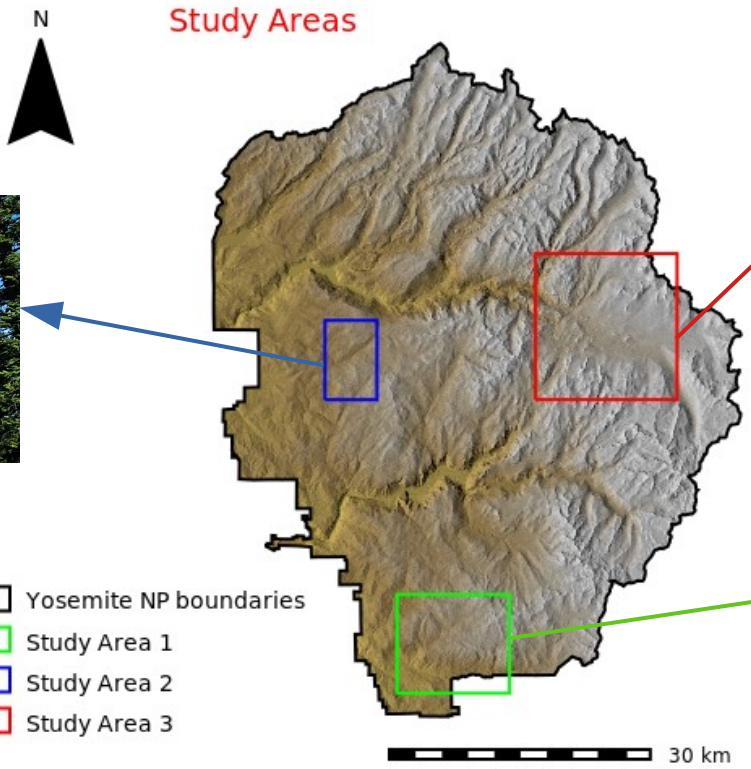


Source: Google Maps Photos

Source: Google Maps Photos

Wildfire modeling

Questions

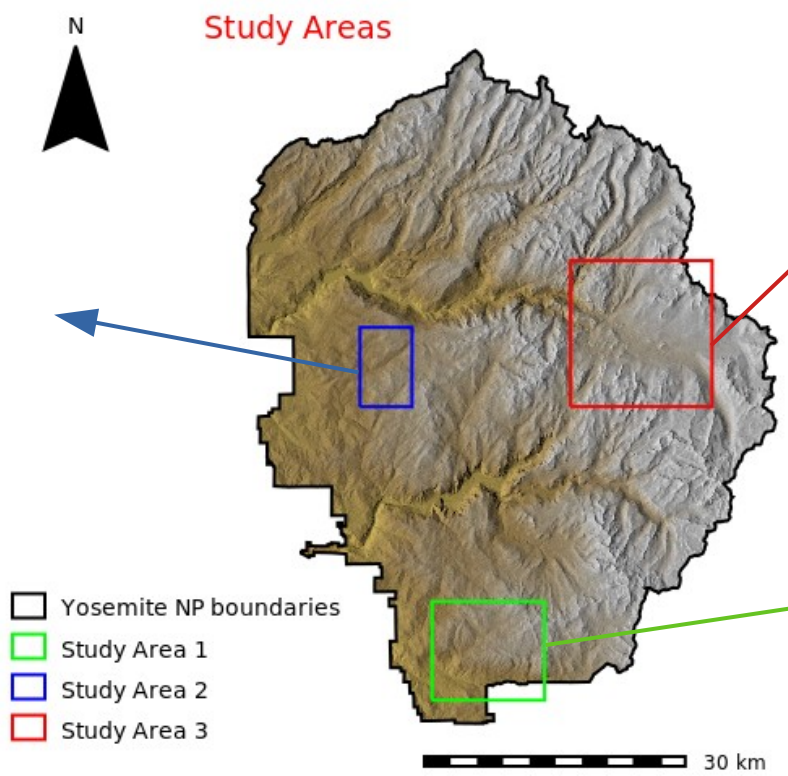


Does our simulated fire spread the same as a past fire event?

Wildfire modeling

Questions

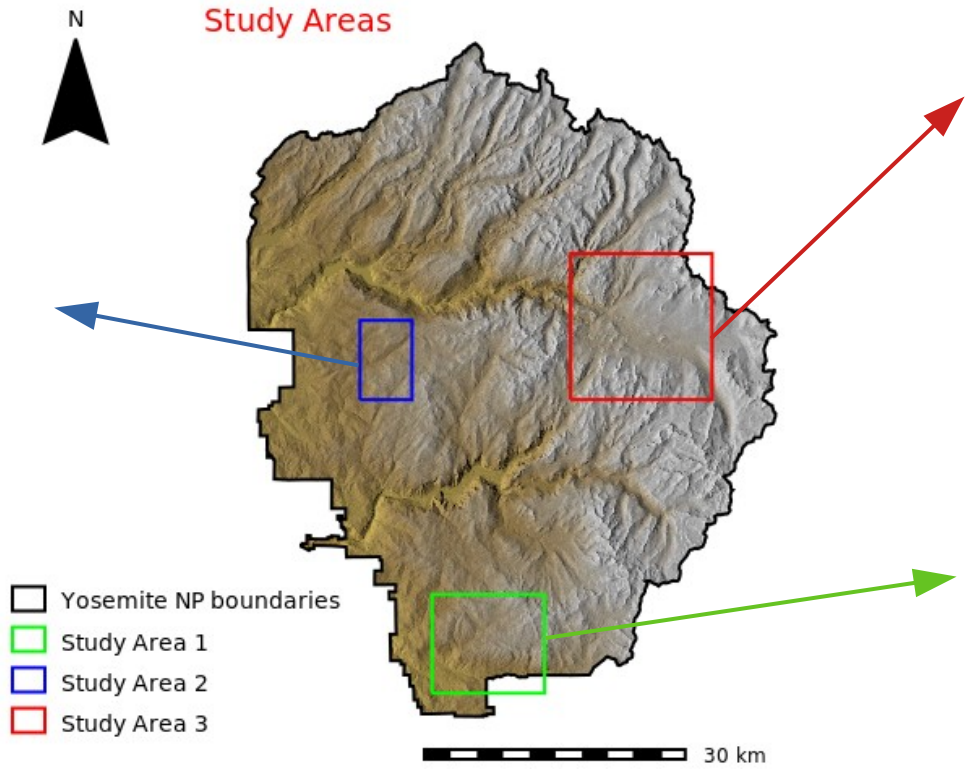
How our wildfire simulation behaves in a small area



Does our simulated fire spread the same as a past fire event?

Wildfire modeling

Questions



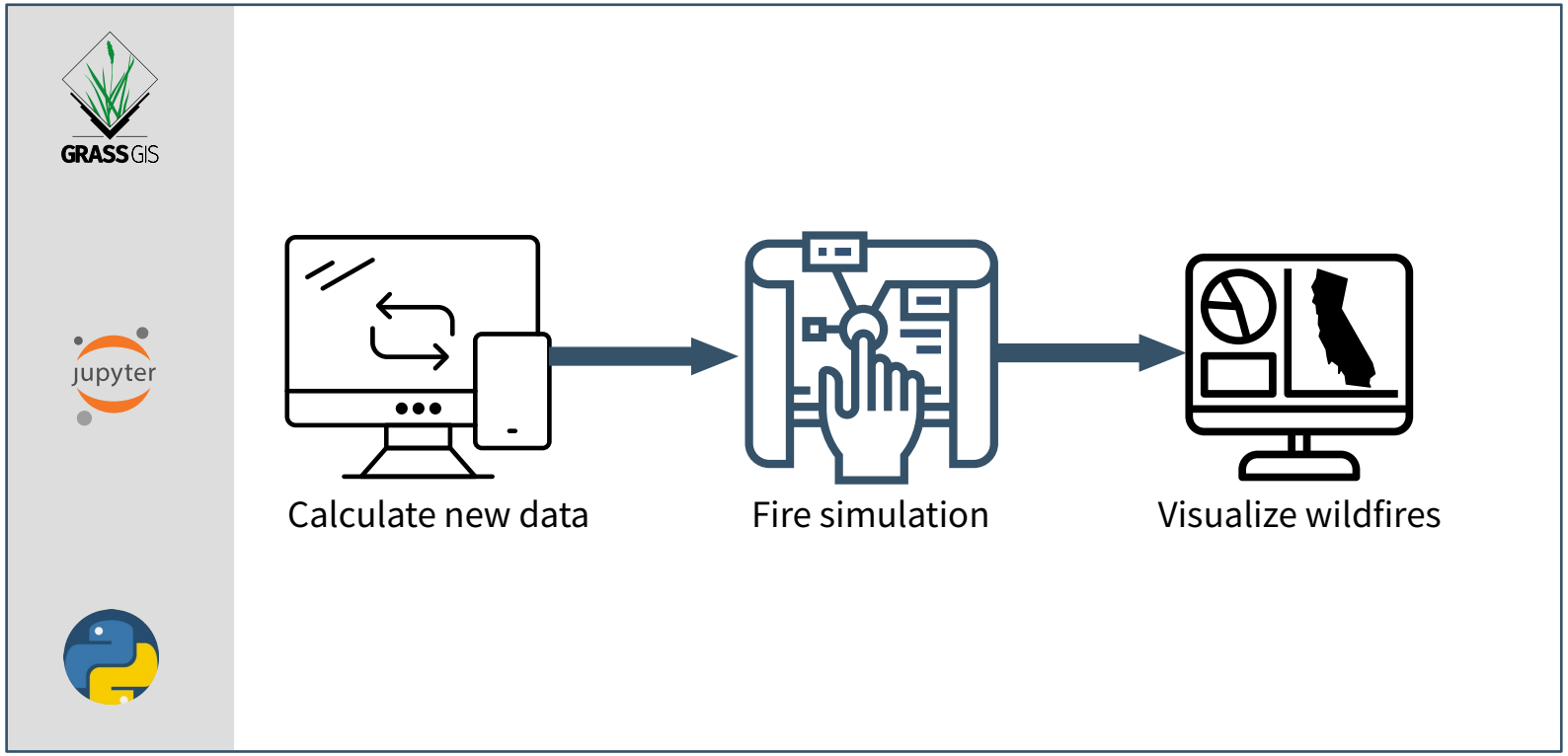
How our wildfire simulation behaves in a small area

How our wildfire simulation behaves in a large area

Does our simulated fire spread the same as a past fire event?

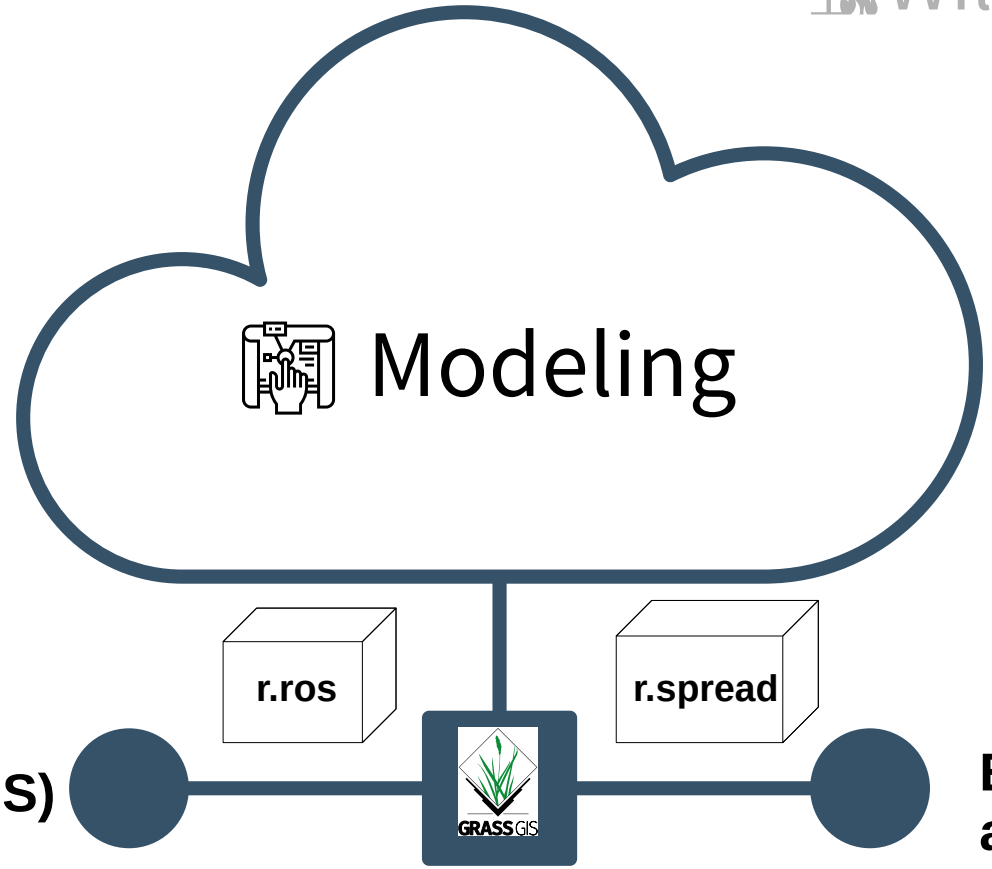
Wildfire modeling

Wildfire modeling workflow



Wildfire modeling

Calculate new data



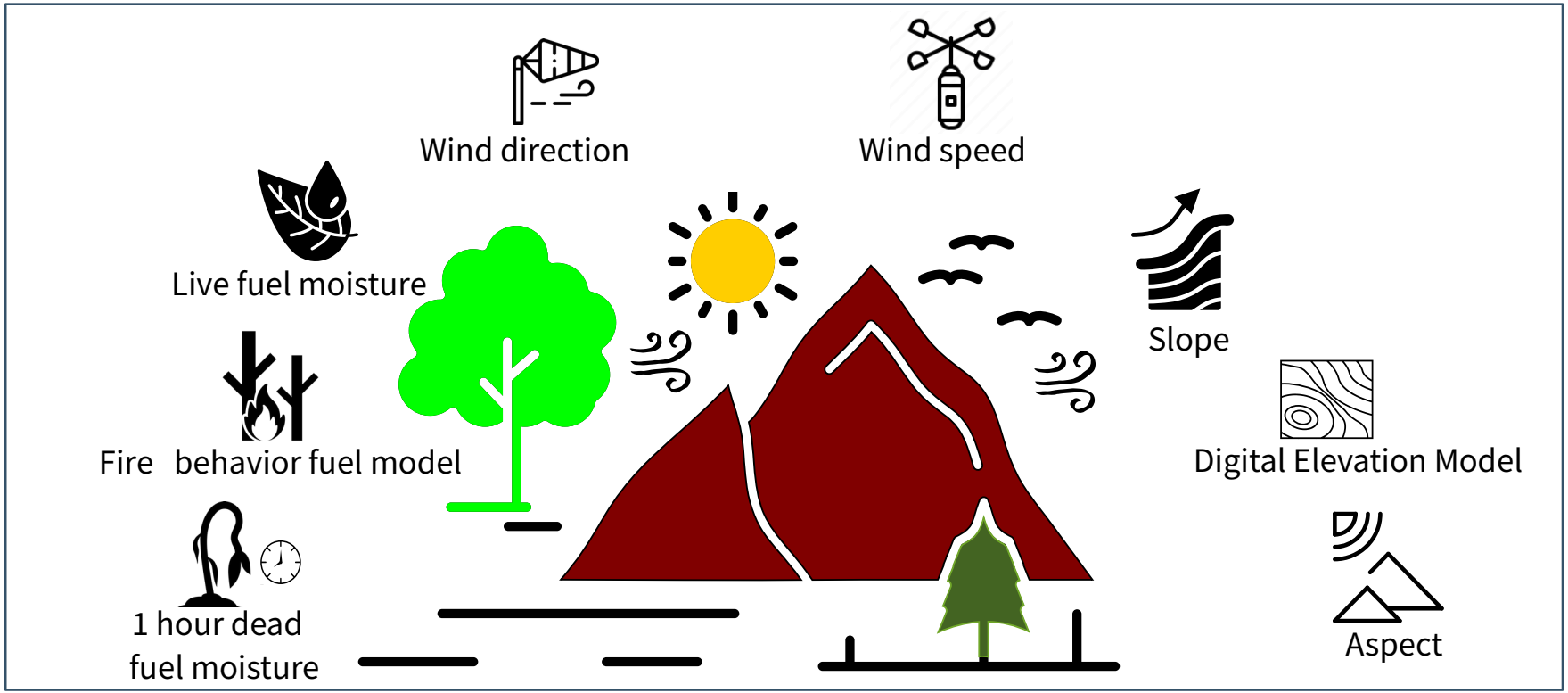
Rate of Spread (ROS)

Elliptically anisotropic spread

Wildfire modeling

Calculate new data

r.ros module inputs



Wildfire modeling

Calculate new data

Imported data

Data type	Name
	100 hour dead fuel moisture 
	Enhanced Vegetation Index 
	Wind direction 
	Wind speed 
	Digital Elevation Model 
	Digital Elevation Model 

r.ros module new data

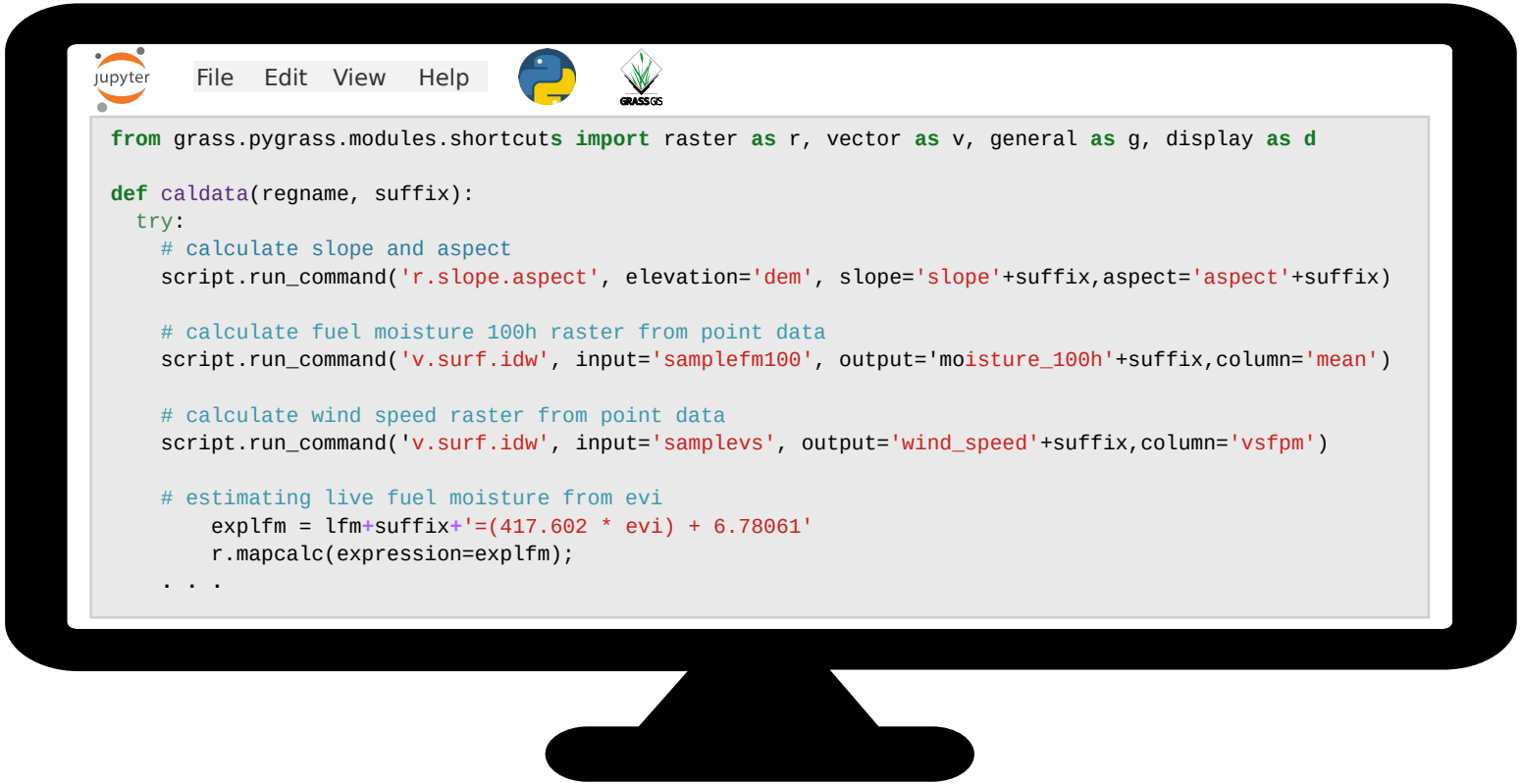
Data type	Name
	1 hour dead fuel moisture 
	Live fuel moisture 
	Wind direction 
	Wind speed 
	Slope 
	Aspect 

 = raster data

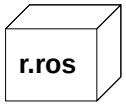
 = point vector data

Wildfire modeling

Calculate new data



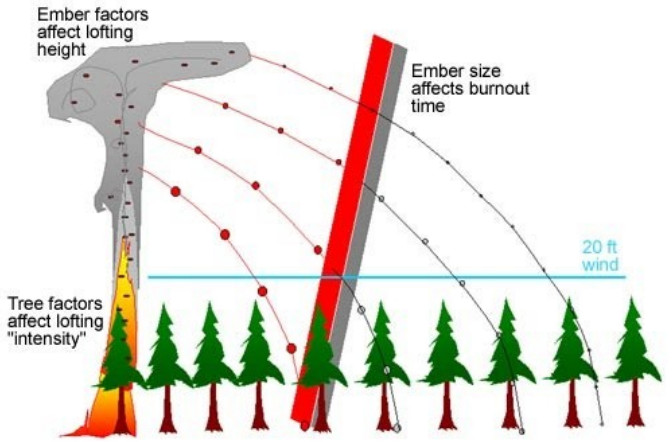

```python
from grass.pygrass.modules.shortcuts import raster as r, vector as v, general as g, display as d

def caldata(regname, suffix):
  try:
    # calculate slope and aspect
    script.run_command('r.slope.aspect', elevation='dem', slope='slope'+suffix,aspect='aspect'+suffix)

    # calculate fuel moisture 100h raster from point data
    script.run_command('v.surf.idw', input='samplefm100', output='moisture_100h'+suffix,column='mean')

    # calculate wind speed raster from point data
    script.run_command('v.surf.idw', input='samplevs', output='wind_speed'+suffix,column='vsfpm')

    # estimating live fuel moisture from evi
        explfm = lfm+suffix+'=(417.602 * evi) + 6.78061'
        r.mapcalc(expression=explfm);
    . . .
```



Rate of Spread (ROS)

r.ros outputs

- base_ros: raster map containing base ROS (cm/min)
- max_ros: raster map containing maximal ROS (cm/min)
- direction_ros: raster map containing directions of maxima ROS (degree)
- spotting_distance: raster map containing maximal spotting distance (m)

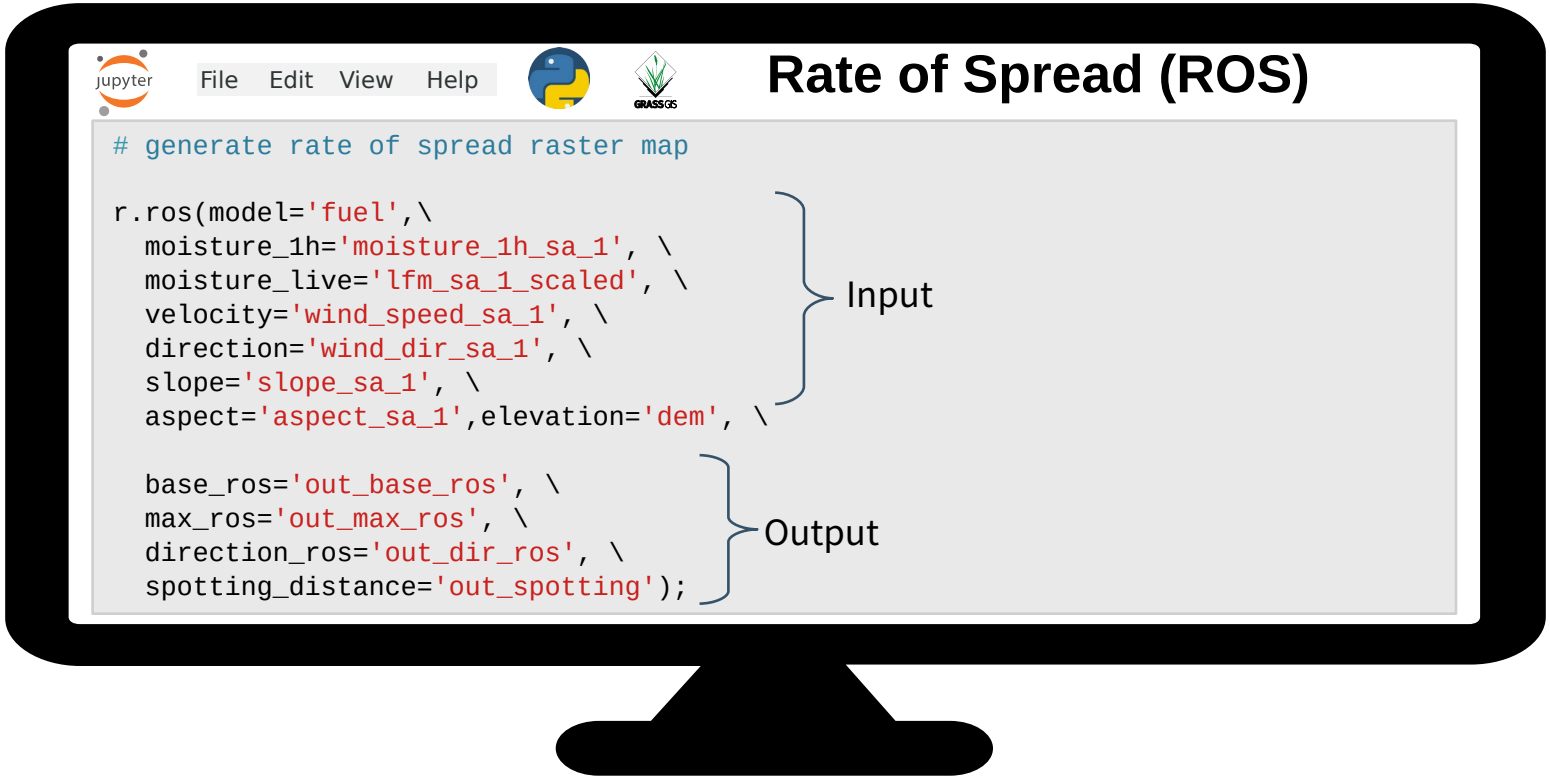


Factors Affecting Spotting

Source: https://www.nwcg.gov/

Wildfire modeling

Fire simulation



r.spread

Elliptically anisotropic spread

r.spread inputs

- Map containing base ROS
- Map containing maximal ROS
- Map containing directions of maxima ROS
- Map containing maximal spotting distance

- Map containing starting source

- Map containing wind speed

- Map containing fuel moisture 1hour

r.spread output

Map of the cumulative time of spread

Wildfire modeling

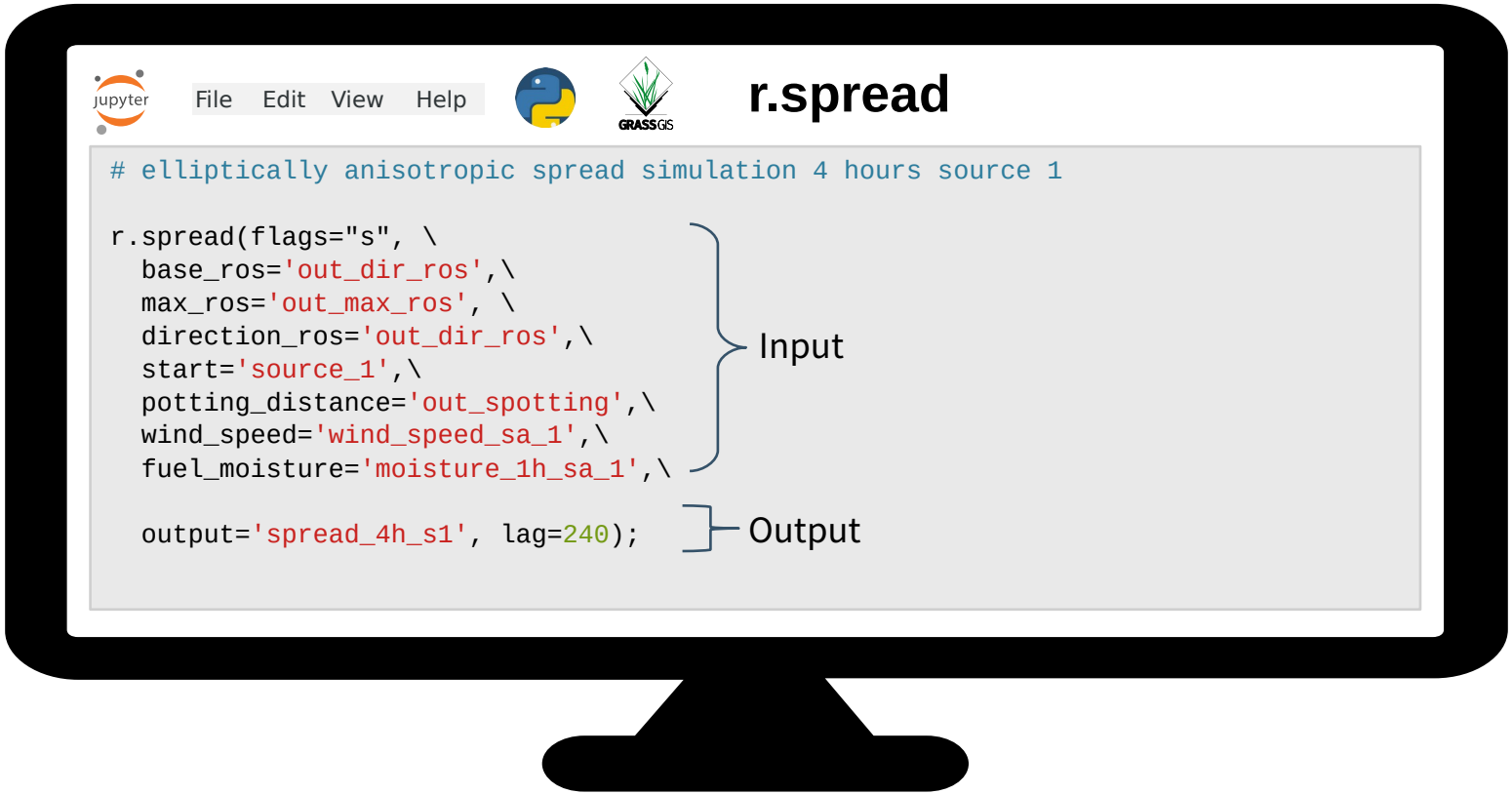
Fire simulation

Starting source

Sources: 123rf.com / Dreamsite.com /dfes.wa.gov.au

Wildfire modeling

Fire simulation



Wildfire modeling

Fire simulation

r.spread → **Raster map of the cumulative time of spread**

First iteration: start 0 h, end 3 h

```
#  elliptically anisotropic spread simulation 3 hours source 2
r.spread(flags="s", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='source_2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_3h_s2', lag=180);
```

Second iteration: start 3 h, end 6 h

```
#  elliptically anisotropic spread simulation 6 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_3h_s2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_6h_s2',
  lag=180, init_time=180);
```

Third iteration: start 6 h, end 9 h

```
#  elliptically anisotropic spread simulation 9 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_6h_s2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_9h_s2',
  lag=180, init_time=360);
```

Wildfire modeling

Fire simulation

r.spread → **Raster map of the cumulative time of spread**

First iteration

```
#  elliptically anisotropic spread simulation 3 hours source 2
r.spread(flags="s", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='source_2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_3h_s2', lag=180);
```

Second iteration

```
#  elliptically anisotropic spread simulation 6 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_3h_s2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_6h_s2',
  lag=180, init_time=180);
```

Third iteration

```
#  elliptically anisotropic spread simulation 9 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_6h_s2", \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_9h_s2',
  lag=180, init_time=360);
```

Wildfire modeling

Fire simulation

r.spread → **Raster map of the cumulative time of spread**

First iteration

```
#  elliptically anisotropic spread simulation 3 hours source 2
r.spread(flags="s", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='source_2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_3h_s2', lag=180);
```

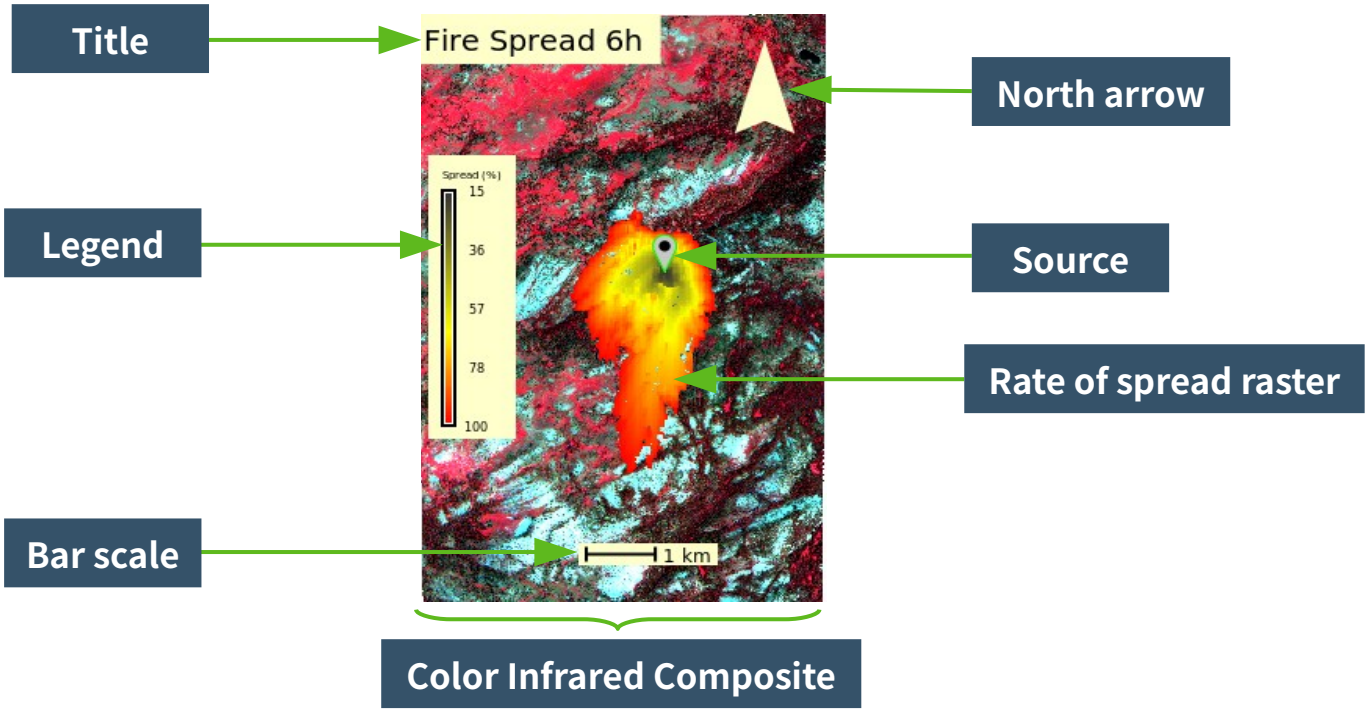
Second iteration

```
#  elliptically anisotropic spread simulation 6 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_3h_s2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_6h_s2',
  lag=180, init_time=180);
```

Third iteration

```
#  elliptically anisotropic spread simulation 9 hours source 2
r.spread(flags="si", base_ros='out_dir_ros', max_ros='out_max_ros', direction_ros='out_dir_ros', start='spread_6h_s2', \
  spotting_distance='out_spotting', wind_speed='wind_speed_sa_2', fuel_moisture='moisture_1h_sa_2', output='spread_9h_s2',
  lag=180, init_time=360);
```

Wildfire visualization



Wildfire modeling

Wildfire visualization

jupyter File Edit View Help GRASS GIS

```
d.frame(flags='c', frame='first', at=(0,100,0,33))
```
← Frame

```
d.rast(map='subset_cir')
```
← Color Infrared Composite

```
d.rast(map='spread_3h_s2')
```
← Rate of spread raster

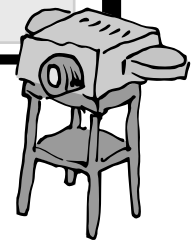
```
d.vect(map='source_2', size=25, icon='basic/pin_dot', color='green',legend_label='Source 2')
```
← Source

```
d.text(text='Fire Spread 3h', bgcolor=('255:255:204'), color='black', size=6)
```
← Title

```
script.run_command('d.legend', flags='sb', raster='spread_9h_s2_reclas, ...)
```
← Legend

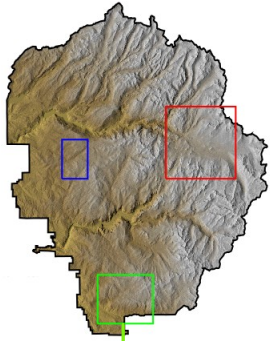
```
script.run_command('d.northarrow', flags='t', style='9',at=(85,85), ...)
```
← North arrow

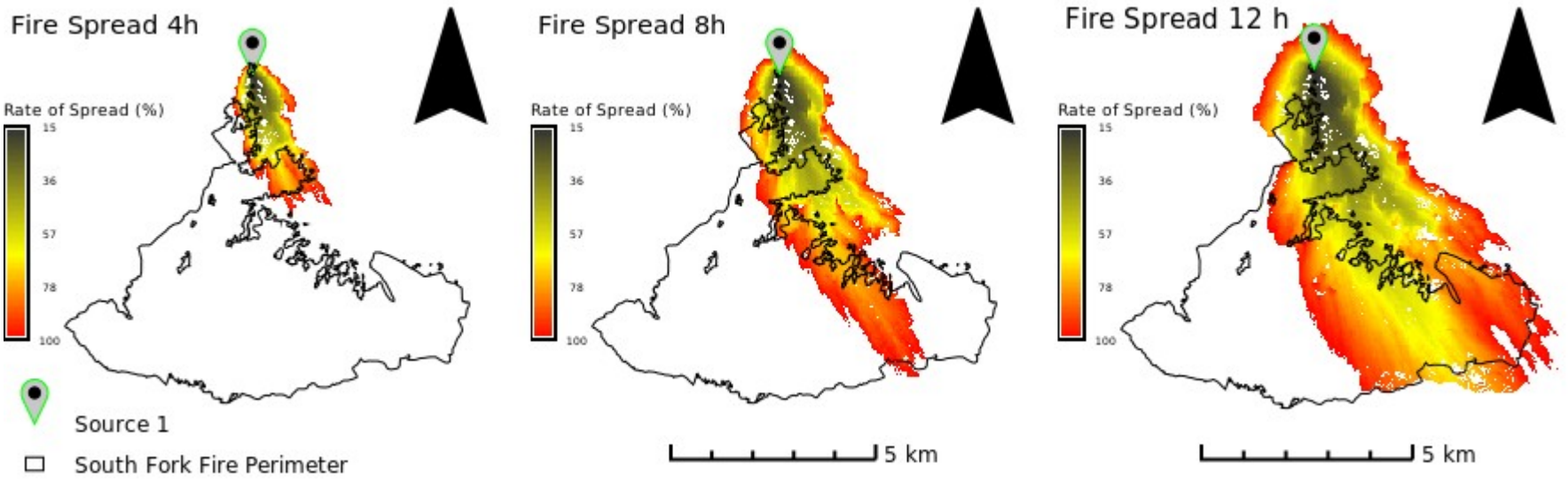
```
script.run_command('d.barscale', bgcolor=('255:255:204'), at=(50,30),...)
```
← Bar scale



Wildfire modeling

Wildfire visualization

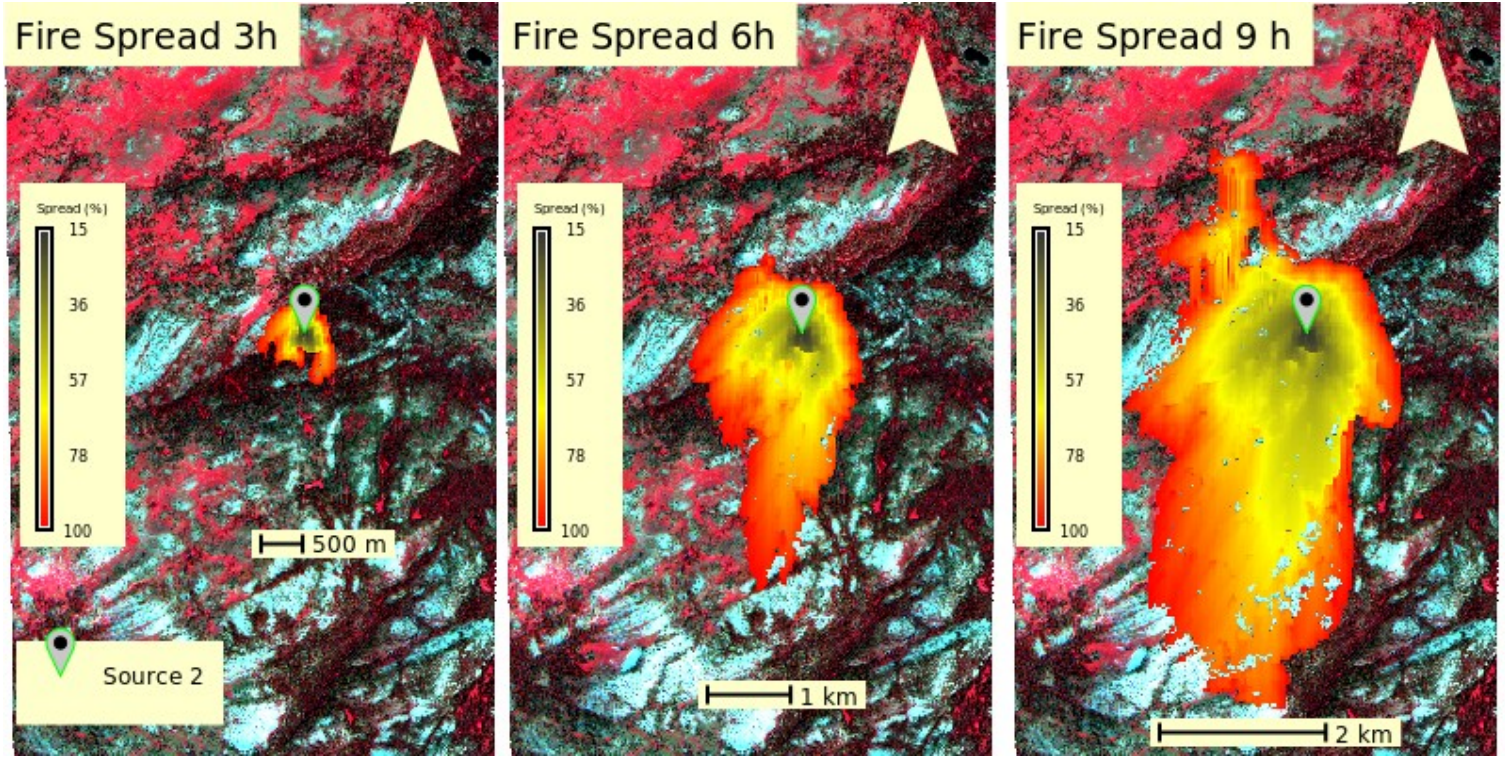


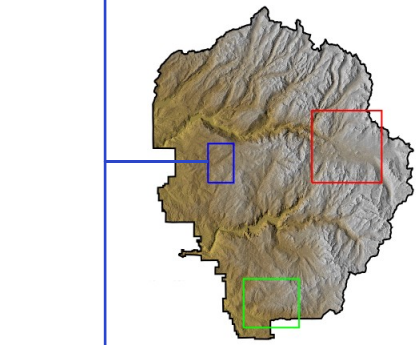


Wildfire simulation area 1

Wildfire modeling

Wildfire visualization

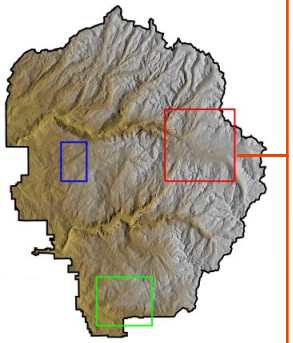


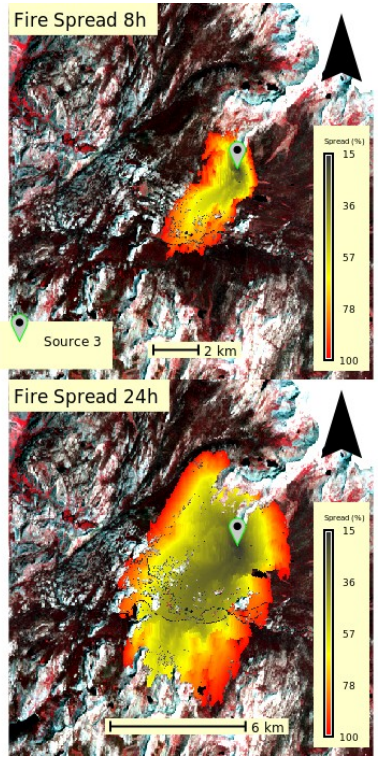


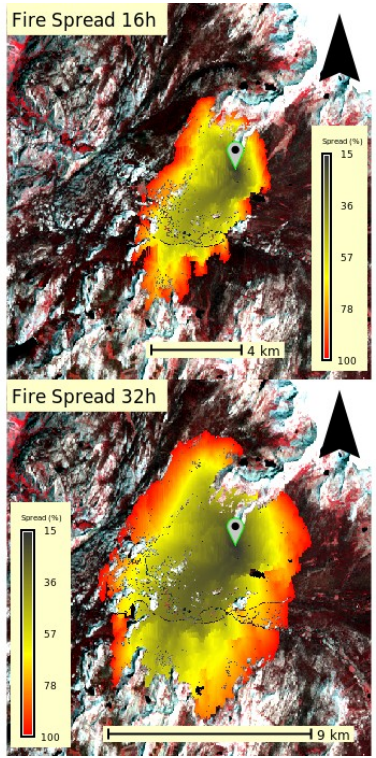
Wildfire simulation area 2

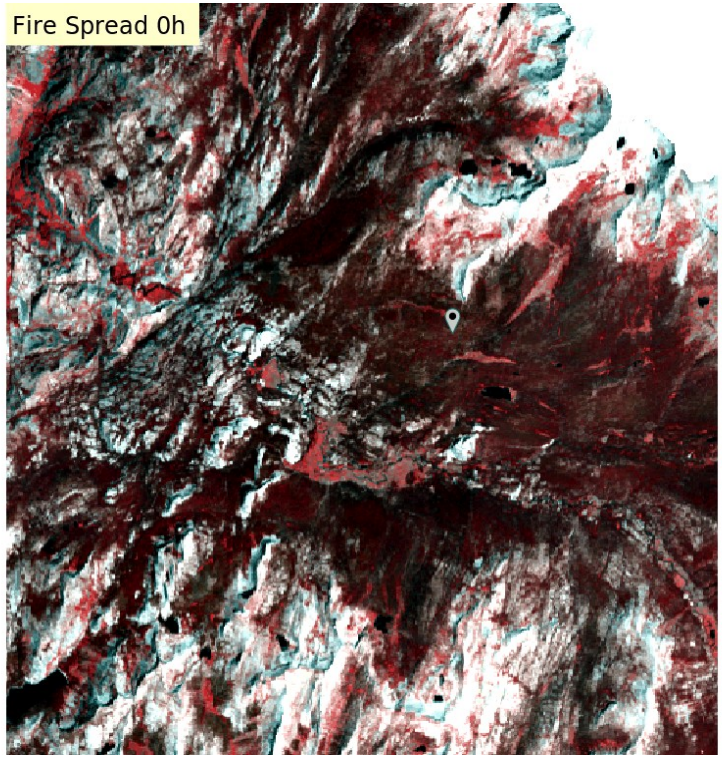
Wildfire modeling

Wildfire visualization









Wildfire simulation area 3

Conclusions

- Integration of GRASS GIS, Google Earth Engine, and Jupyter Notebook.
- Comparison of past and simulated wildfire events.
- Fire simulations avoid bare soils.
- Try out different interpolation parameters.

References

GRASS Development Team(2021). GRASS GIS 7.8.6dev Reference Manual. Retrieved April 17, 2021, from https://grass.osgeo.org/grass78/manuals/.

Myoung, B., Kim, S. H., Nghiem, S. V., Jia, S., Whitney, K., & Kafatos, M. C. (2018). Estimating live fuel moisture from MODIS satellite data for wildfire danger assessment in Southern California USA. Remote Sensing, 10(1), 87.

Northern Rockies Coordination Center(n.d). PSA NFDRS Component Glossary. Retrieved April 16, 2021, from https://gacc.nifc.gov/nrcc/predictive/fuels_fire-danger/psa_component_glossary.htm

Wikipedia. 2021. "Wildfire modeling." Last modified January 10,2021. https://en.wikipedia.org/wiki/Wildfire_modeling.

. . .

Icons taken from: Noun Project, Flaticon, Freepik, and Free SVG.


thank you,
questions?