# Software libre para investigación: Una (muy breve) introducción

Mario Gavidia-Calderón

# Hola!


IAGUSP
Instituto de Astronomia, Geofísica
e Ciências Atmosféricas
IAG USP

# Modelos de calidad del aire

# Análisis de datos atmosféricos y dataviz

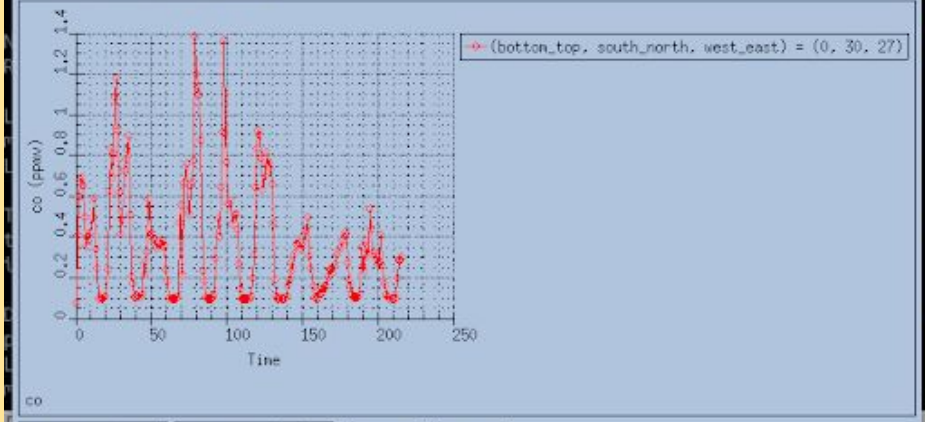
# Todo lo hago con Software libre

mgavidia@svante2: /scr2/mgavidia/WRF4/WRF/test/em_real

mgavidia@svante2: ~ 74x19

co (on svante2)

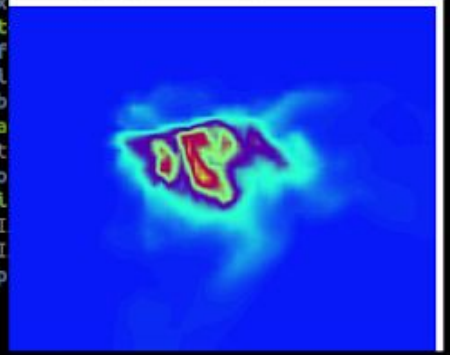
Close | Print | Dump | Locked




X Axis: Time | Use Log: X Y | X Range | Y Range

```
examples.desktop                snap
Firefox_wallpaper.png           somex
id_dsa.mgavidia                 svant
latexperiment                   texmf
LDOPE-1.7                       Thesi
miniconda3                      thumb
mozilla.pdf                     to_ia
Music                           Video
OpenJUMP-1.14.1-r6147-PLUS      vpn_i
openrefine                      VPN-I
Pictures                        VPN-I
pm25_zambela_summary_plot.png   vprmp
Production
quishqa@wall-e:~$ cd projects/
quishqa@wall-e:~/projects$ ls
AAS4WRF.py  qualR_old  qualR.py  wrf_sp_eval
quishqa@wall-e:~/projects$
```

wrfout_d02_2018-06-21_00:00:00 (on sv...



mgavidia@svante2: /scr2/mgavidia/WRF4/WRF/test/em_real 74x19

```
windturbines.txt
wrfbdy_d01
wrfchemi_12z_d01
wrfchemi_d01_2018-06-21_00:00:00
wrfchemi_d02_2018-06-21_00:00:00
wrf_chem_val
wrf.exe
wrfinput_d01
wrfinput_d02
wrfout_d01_2018-06-21_00:00:00
wrfout_d01_2018-06-21_00:00:00_2way_test
wrfout_d01_2018-06-21_00:00:00_aas4wrf_test
wrfout_d01_2018-06-21_00:00:00_first_a4w
wrfout_d02_2018-06-21_00:00:00
```

Ncview 1.93 (on svante2)

Ncview 1.93g David W. Pierce 24 February 2009

variable=co

frame 26/217

displayed range: 0.034448 to 1.26827 ppmv

Current: x=319.343, y=1.55556

Quit | ->1 | Edit | ? | Delay: | Opts

3gauss | Inv P | Inv C | M X5 | Linear | Axes | Range | Bi-lin | Print

0.2 0.4 0.6 0.8 1 1.2

(28) 1d vars | (25) 2d vars | (128) 3d vars | (98) 4d vars

| Dim: | Name: | Min: | Current: | Max: | Units: |
|---|---|---|---|---|---|
| Scan: | Time | 0 | 25 | 216 | - |
| | bottom_top | 0 | 0 | 33 | - |
| Y: | south_north | 0 | -Y- | 50 | - |
| X: | west_east | 0 | -X- | 62 | - |

*“**...**es todo **software** cuyo código fuente puede ser estudiado, modificado, y **utilizado libremente** con cualquier fin y **redistribuido** con cambios y/o **mejoras** o sin ellas.”*

https://es.wikipedia.org/wiki/Software_libre

# Por qué estamos aquí?

1. Cuáles son las softwares libres más utilizados?

1. Cuáles son las softwares libres más utilizados?
2. Cuáles son los pro y contras?

1. Cuáles son las softwares libres más utilizados?
2. Cuáles son los pro y contras?
3. Por qué son importantes para la investigación?

# Software libres + utilizados en investigación

# Hacer el experimento

Hacer el experimento

## Calcular/Analizar resultados (datos)

Hacer el experimento

Calcular/Analizar resultados (datos)

**Visualizar resultados**

Hacer el experimento

Calcular/Analizar resultados (datos)

Visualizar resultados

**Reportar**

Hacer el experimento

**Calcular/Analizar resultados (datos)**

**Visualizar resultados**

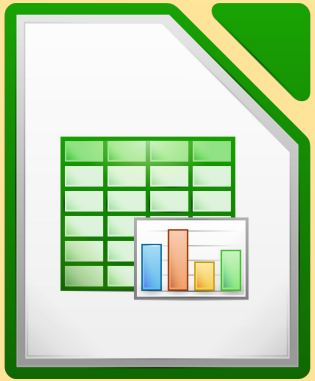
**Reportar**

# Analizar Resultados

# Microsoft Excel

# Microsoft Excel

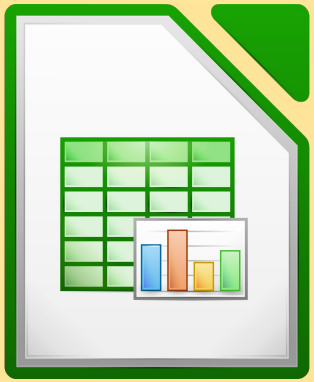
≈



LibreOffice
The Document Foundation

# Microsoft Excel

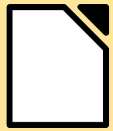
≈



LibreOffice
The Document Foundation

Google Sheets

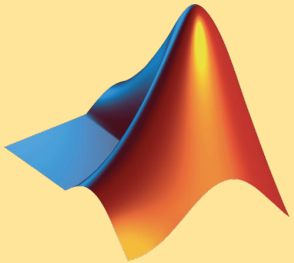
# (Un paréntesis personal)

LibreOffice
The Document Foundation


Booo!!

# A veces necesitamos armas más pesadas

# A veces necesitamos armas más pesadas

ArcGIS

AUTOCAD

# Pero hay software libre para todos los gustos

# Pero hay software libre para todos los gustos

# Y si no es necesario tanto software???

# Lenguajes de programación al rescate!


# (Otro paréntesis personal)

## R & Python

- Lenguajes de programación
- Opensource
- Multiplataforma
- Los más usados en ciencia de datos
- Muchos paquetes para diferentes necesidades

Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | lima_cov

_data_cleaning.R | draft.R | 01_read_files.R | 02_plots.R*

Source on Save | Run | Source

```
# The story until now plot ------------------------------------------------

StoryPlot <- function(df, pol, station){
  plot(df$date, df[[pol]], t = 'l', lwd = 0,
       xlab = '2020',
       ylab = expression("PM"[2.5] * " ("*mu*"g/m"^3*")"),
       main = station)
  for (i in dates$dateI[2:7]){
    abline(v = i, col = 'Blue', lwd = 2.)
  }
  abline(v = dates$dateI[1], col = 'Red', lwd = 2.)
  lines(df$date, df[[pol]], lwd =2.5)
  legend('bottomleft', legend = c('Inicio 40tena', 'Prolongaciones'),
         bty = 'n', col = c('red', 'blue'), lwd = c(2.5))
}

StoryPlot(cdm20, 'data', 'Campo de Marte')
StoryPlot(cdm20o3, 'data', 'Campo de Marte')

usa20q <- selectByDate(usa20, '15/2/2020', '2/6/2020')
StoryPlot(usa20q, 'pm25', 'USA Embassy')


# Daily Profile Plots -----------------------------------------------------

cdm_hq <- selectByDate(cdm20, '1/3/2020', '15/3/2020')
```

1 # The story until now plot | R Script

Terminal | Jobs

tests/lima_covid19/

```
lot(df$date, df[[pol]], t = 'l', lwd = 0,
    xlab = '2020',
    ylab = expression("PM"[2.5] * " ("*mu*"g/m"^3*")"),
    main = station)
or (i in dates$dateI[2:7]){
 abline(v = i, col = 'Blue', lwd = 2.)

bline(v = dates$dateI[1], col = 'Red', lwd = 2.)
ines(df$date, df[[pol]], lwd =2.5)
egend('bottomleft', legend = c('Inicio 40tena', 'Prolongaciones'),
      bty = 'n', col = c('red', 'blue'), lwd = c(2.5))


ryPlot(cdm20, 'data', 'Campo de Marte')
ryPlot(cdm20o3, 'data', 'Campo de Marte')

20q <- selectByDate(usa20, '15/2/2020', '2/6/2020')
ryPlot(usa20q, 'pm25', 'USA Embassy')
```

Environment | History | Connections | Git

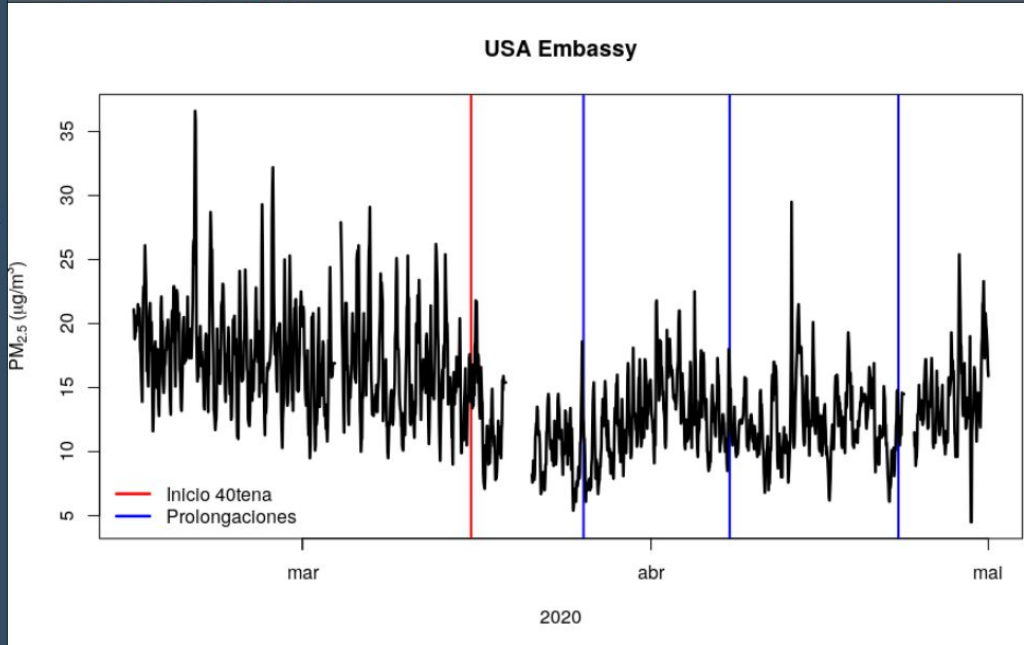
Import Dataset | List

Global Environment

| | |
|---|---|
| usa20q | 1759 obs. of 3 variables |
| Values | |
| after_aq_mean | 12.045 |
| aqs_code | chr [1:5] "112194" "112193" "112208" "112267" "112265" |
| aqs_name | chr [1:5] "Campo de Marte" "San Borja" "Santa Anita" ... |
| before_aq_mean | 13.7872222222222 |
| date_end | "13052020" |
| date_end_19 | "13052019" |
| date_end_20 | "13052020" |
| date_start | "01032020" |

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



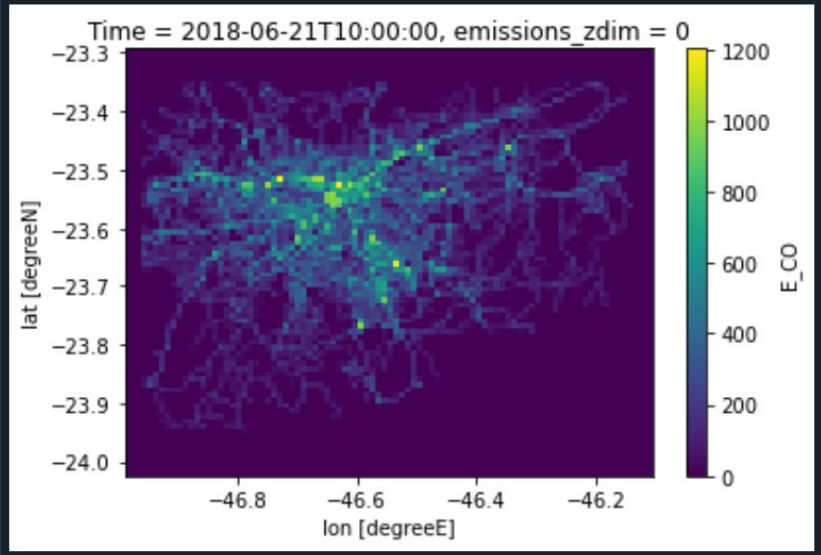
dit Search Source Run Debug Consoles Projects Tools View Help

/scr2/mgavidia/python_stunts/wrf_eval

/scr2/mgavidia/python_stunts/test/read_emissions.py

_stats.py | qualar_py.py | data_preparation.py | model_eval_sp.py | read_emissions.py

```
# -*- coding: utf-8 -*-
"""
Editor Spyder

Este é um arquivo de script temporário.
"""

import pandas as pd
import xarray as xr

file_name1km = '/scr2/mgavidia/WRF4/in_out/emissions_1km.txt'
file_name3km = '/scr2/mgavidia/WRF4/in_out/emissions_3km.txt'
file_name9km = '/scr2/mgavidia/WRF4/in_out/emissions_9km.txt'


df = pd.read_csv(file_name1km,
                 delimiter='\t',
                 names=['i', 'lon', 'lat', 'E_CO', 'E_HCHO', 'E_C2H5OH',
                        'E_KET', 'E_NH3', 'E_XYL', 'E_TOL',  'E_ISO',
                        'E_OLI', 'E_OLT', 'E_OL2', 'E_HC8', 'E_HC5', 'E_ORA2',
                        'E_ETH', 'E_ALD', 'E_CSL', 'E_SO2', 'E_HC3', 'E_NO2',
                        'E_NO', 'E_CH3OH', 'E_PM25I', 'E_PM25J', 'E_SO4I',
                        'E_SO4J', 'E_NO3I', 'E_NO3J', 'E_ORGI', 'E_ORGJ',
                        'E_ECI', 'E_ECJ', 'E_SO4C', 'E_NO3C', 'E_ORGC',
                        'E_ECC'])

n_lon = len(df.lon.unique())
n_lat = len(df.lat.unique())

n_points = n_lon * n_lat
lon1d = df["lon"][: n_lon].values
lat1d = df["lat"][: n_points: n_lon].values[::-1]

date = pd.date_range('2018-06-21 00:00',
                     '2018-06-30 23:00',
                     freq='H')



emiss_names = df.columns[3:]


def create_dataaaray_per_emi(emiss_df, pol, lat, lon, date):
    '''
    Create a xarray dataarray for each emission species
    by reshaping the emission dataframe (emiss_df) according the lat, lon
    and date, add emission_zdim dimension and add attribute names to lat and
    lon dimensions.

    Parameters
    ----------
    emiss_df : pandas DataFrame
        create by read_csv emission_file
    pol : string
        name of emitted pollutant.
    lat : numpy ndarray
        latitudes of emission_file.
    lon : numpy ndarray
```




Help | Variable explorer | Plots | Files

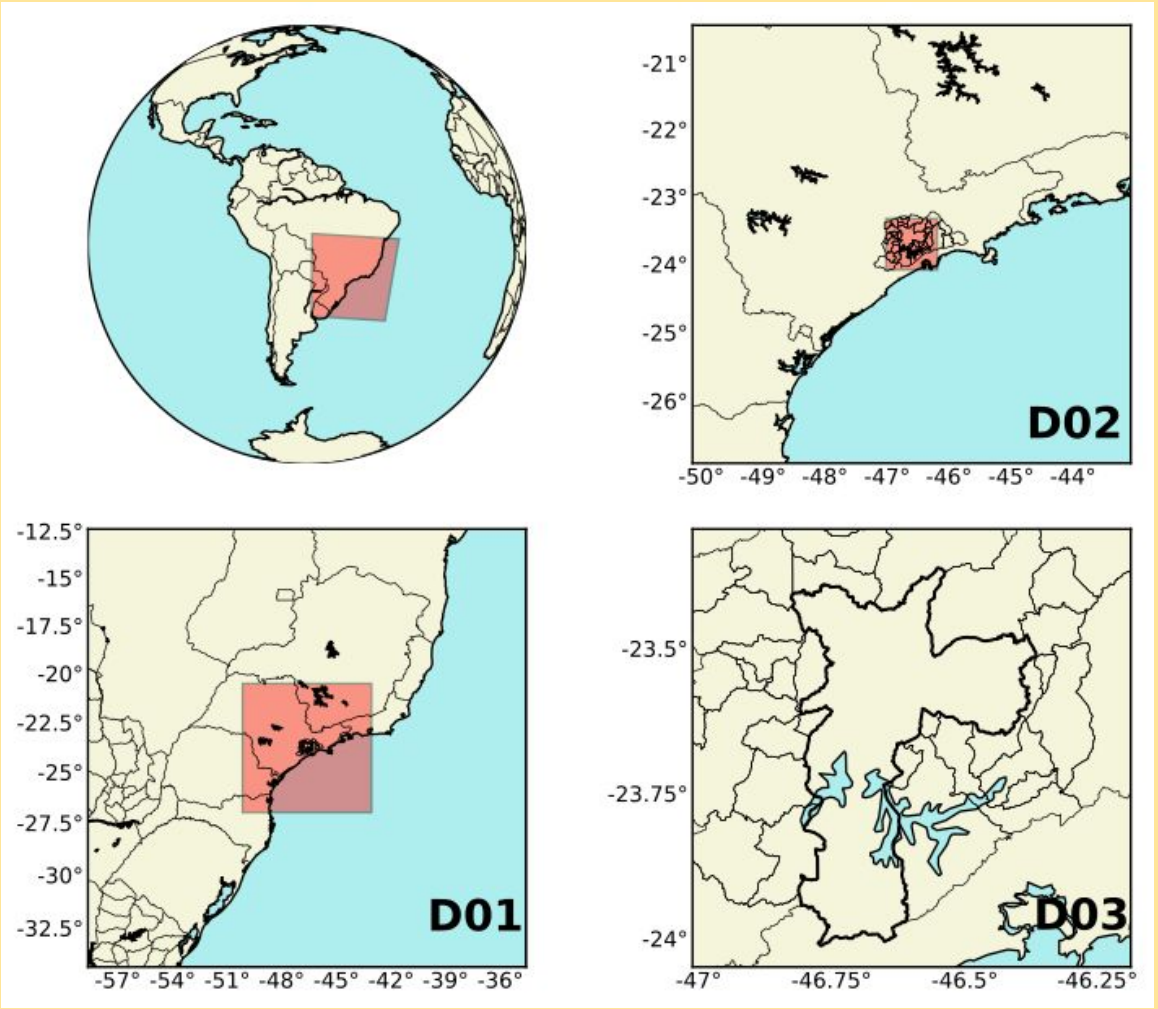
Console 1/A

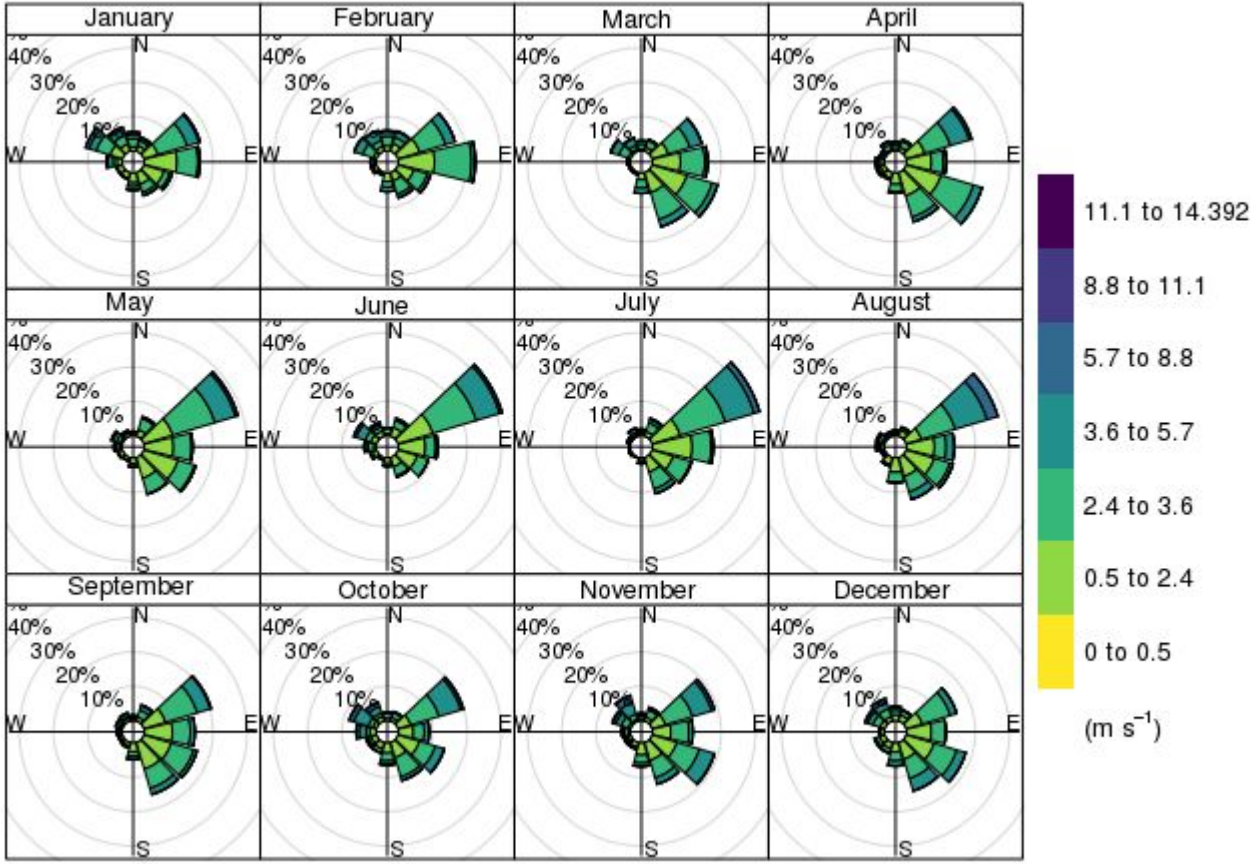
```
array([-46.982305, -46.972505, -46.962705, -46.952905, -46.943104, -46.933304,
       -46.923504, -46.913704, -46.903903, -46.894103, -46.884302, -46.874502,
       -46.864701, -46.8549  , -46.845099, -46.835299, -46.825498, -46.815697,
       -46.805896, -46.796095, -46.786293, -46.776492, -46.766691, -46.756889,
       -46.747088, -46.737287, -46.727485, -46.717683, -46.707882, -46.69808 ,
       -46.688278, -46.678477, -46.668675, -46.658873, -46.649071, -46.639269,
       -46.629467, -46.619664, -46.609862, -46.60006 , -46.590258, -46.580455,
       -46.570653, -46.56085 , -46.551048, -46.541245, -46.531443, -46.52164 ,
       -46.511837, -46.502034, -46.492232, -46.482429, -46.472626, -46.462823,
       -46.45302 , -46.443217, -46.433413, -46.42361 , -46.413807, -46.404004,
       -46.3942  , -46.384397, -46.374593, -46.36479 , -46.354986, -46.345183,
       -46.335379, -46.325576, -46.315772, -46.305968, -46.296164, -46.28636 ,
       -46.276557, -46.266753, -46.256949, -46.247145, -46.237341, -46.227536,
       -46.217732, -46.207928, -46.198124, -46.18832 , -46.178515, -46.168711,
       -46.158907, -46.149102, -46.139298, -46.129493, -46.119689, -46.109884])
Coordinates:
```

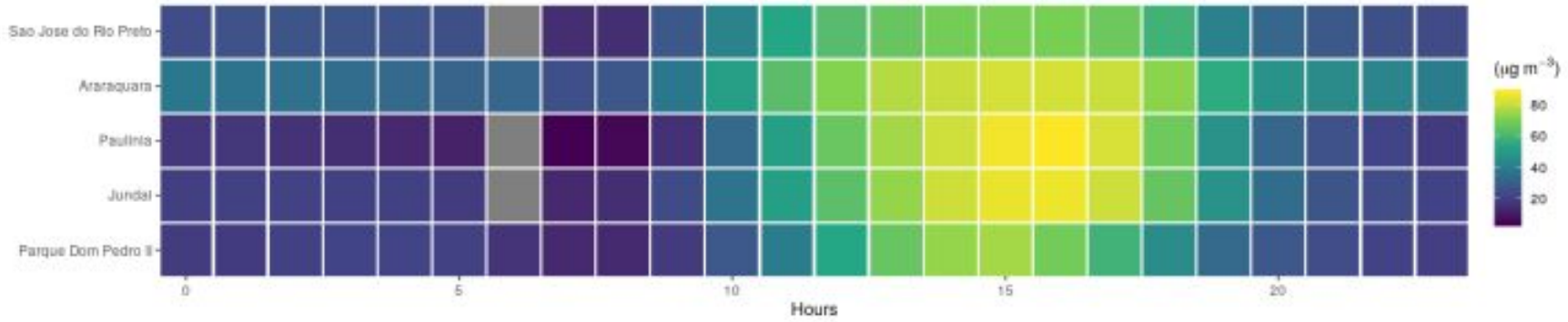
IPython console | History

# Visualizar datos

# R y Python tienen excelentes paquetes para dataviz


-21°
-22°
-23°
-24°
-25°
-26°
D02
-50° -49° -48° -47° -46° -45° -44°
-12.5°
-15°
-17.5°
-20°
-22.5°
-25°
-27.5°
-30°
-32.5°
D01
-57° -54° -51° -48° -45° -42° -39° -36°
-23.5°
-23.75°
-24°
D03
-47°
-46.75°
-46.5°
-46.25°


January
February
March
April
May
June
July
August
September
October
November
December
N
S
E
W
40%
30%
20%
10%
11.1 to 14.392
8.8 to 11.1
5.7 to 8.8
3.6 to 5.7
2.4 to 3.6
0.5 to 2.4
0 to 0.5
(m s$^{-1}$)


Sao Jose do Rio Preto
Araraquara
Paulinia
Jundiai
Parque Dom Pedro II
0
5
10
15
20
Hours
(μg m⁻³)
80
60
40
20

# Reportar

# Word puede hacerte enojar

# Moving a picture in Microsoft Word

It actually does what you want

You the mess up whole document


# al rescate!

**What you see is what you get (WYSIWYG)**

**What you see is what you mean (WYSIWYM)**

```
\begin{sidewaysfigure}[ht]
    \includegraphics[width=\textwidth]{figs/interp_method_eval_pop_dens.pdf}
    \caption{Test of different interpolation methods to calculate population density.}
    \label{fig:pop_den}
\end{sidewaysfigure}


Another solution is to upscale PM\textsubscript{2.5} simulation to the Gridded Population of the World
resolutions, because it has data at 30 sec, 2.5 min, 15 min, 30 min and 60 min resolution.


\subsection{Exposition curves}

The methodology was based on \cite{Gao2018}. First we have to calculate the Relative Risk ($RR$) by using
Integrated Exposure Response (IER) functions. With RR, we can calculate Population Attributable Fractions is
($PAF$), which will be used to calculate Mortality ($\Delta M$).

\begin{equation*}
    RR_{i,j,k}(C_l) = \begin{cases}
                      1 + \alpha_{i,j,k}(1-e^{-\beta_{i,j,K}(C_l-C_0)^{\gamma_{i,j,k}}})  & Cl > C_0 \\
                      1             & Cl < C_0
                    \end{cases}
\label{eq:1}
\end{equation*}

\begin{equation*}
    PAF_{i,j,k} = \dfrac{RR_{i,j,k}(C_l) - 1}{RR_{i,j,k}(C_l)}
\end{equation*}

\begin{equation*}
    \Delta M = PAF_{i,j,k.l} * y_{0_{i,j,k,l}} * Pop_{i,j,l}
\end{equation*}

In these equations $C_l$ is the annual PM\textsubscript{2.5} concentration from WRF-Chem, $C_0$, the
counterfactual concentration (in \cite{Cohen2017} is the Theoretical minimum-risk exposure level);
$\alpha_{i,j,k}$, $ \beta_{i,j,k}$ and $ \gamma_{i,j,k}$ are the parameters that describe the IER curves in
i\textsuperscript{th} age  and j\textsuperscript{th}  sex group for the k\textsuperscript{th} disease. In the
mortality equation $y_{0_{i,j,k,l}}$ is the current age-sex-specific  mortality rate for the
k\textsuperscript{th} and $Pop_{i,j,l}$ is the exposed population in that grid cell ($l$).

In \cite{Gao2018}, they calculate the mortality for the following diseases:"\textit{ischemic heart disease
(IHD), stroke (STK, including both ischemic and hemorrhagic stroke), lung cancer (LC), and chronic obstructive
pulmonary disease (COPD), and for one disease among young children, acute lower respiratory infections
(LRI)}".

Right now, we are still trying to get the parameters that shape the IER curves.


\subsection{Mortality Calculation}
On stand-by.
```

### 3.2 Spatial interpolation methods

Works that estimates mortality using model results used different spatial resolutions, going from 30 km [2] to 0.5 degrees [4]. So, to get comparable results with other studies, a process of **upscalling** has to be made upon WRF-Chem simulation and also population count data(Gridded Population of the World (GPW)) [2].

This **upscalling** process rose problems. Because interpolation methods, such as bi-linear and nearest-neighbor, can create *new population*, we can end with more population than the original data. Other problem was that the spatial distribution of population doesn't preserves, getting less dense population hot spots.

One solution was to, instead of interpolation population count, it's best to interpolate population density, and calculate population count by multiplying it by the cell (grid) area (Figure 5, at the end of the document). Further, there is a **conservative method** for spatial interpolation that gives better results, but its implementation is not as easy as the other mentioned methods and it doesn't conserve 100%. This conservative method was used to upscale population count and $PM_{2.5}$ (Figure 4).

Another solution is to upscale $PM_{2.5}$ simulation to the Gridded Population of the World resolutions, because it has data at 30 sec, 2.5 min, 15 min, 30 min and 60 min resolution.

### 3.3 Exposition curves

The methodology was based on [2]. First we have to calculate the Relative Risk ($RR$) by using Integrated Exposure Response (IER) functions. With RR, we can calculate Population Attributable Fractions is ($PAF$), which will be used to calculate Mortality ($\Delta M$).

$$RR_{i,j,k}(C_l) = \begin{cases} 1 + \alpha_{i,j,k}(1 - e^{-\beta_{i,j,K}(C_l - C_0)^{\gamma_{i,j,k}}}) & Cl > C_0 \\ 1 & Cl < C_0 \end{cases}$$

$$PAF_{i,j,k} = \frac{RR_{i,j,k}(C_l) - 1}{RR_{i,j,k}(C_l)}$$

$$\Delta M = PAF_{i,j,k.l} * y_{0_{i,j,k,l}} * Pop_{i,j,l}$$

In these equations $C_l$ is the annual $PM_{2.5}$ concentration from WRF-Chem, $C_0$, the counterfactual concentration (in [1] is the Theoretical minimum-risk exposure level); $\alpha_{i,j,k}$, $\beta_{i,j,k}$ and $\gamma_{i,j,k}$ are the parameters that describe the IER curves in $i^{th}$ age and $j^{th}$ sex group for the $k^{th}$ disease. In the mortality equation $y_{0_{i,j,k,l}}$ is the current age-sex-specific mortality rate for the $k^{th}$ and $Pop_{i,j,l}$ is the exposed population in that grid cell ($l$).

In [2], they calculate the mortality for the following diseases:"*ischemic heart disease (IHD), stroke (STK, including both ischemic and hemorrhagic stroke), lung cancer (LC), and chronic obstructive pulmonary disease (COPD), and for one disease among young children, acute lower respiratory infections (LRI)*".

Right now, we are still trying to get the parameters that shape the IER curves.

[2] Retrieved from: https://sedac.ciesin.columbia.edu/data/collection/gpw-v4

# Pros y Contras

# Contras

- No son muy populares en todos los ambientes (debería)
- A veces pueden ser difíciles de instalar
- Hay una curva de aprendizaje

# Pros

- Son Gratis!
- No hay límite de licencias
- Hay mucha pero mucha ayuda en línea
- Son versátiles
- Puedes contribuir
- Todos para uno y uno para todos!

# Importancia en investigación

# Sabían que no siempre es posible reproducir los resultados de una investigación?

Lowndes et al., 2017

# Incluso las hechas por tu yo del pasado?

Lowndes et al., 2017

# Ayudan a replicar la investigación (Reproducibility)

**Como no hay licencia, todos pueden acceder al lenguaje de programación**

**Se comparte el script, la data original y tu reporte para llegar al mismo resultado**

# Esto hace el proceso más transparente.

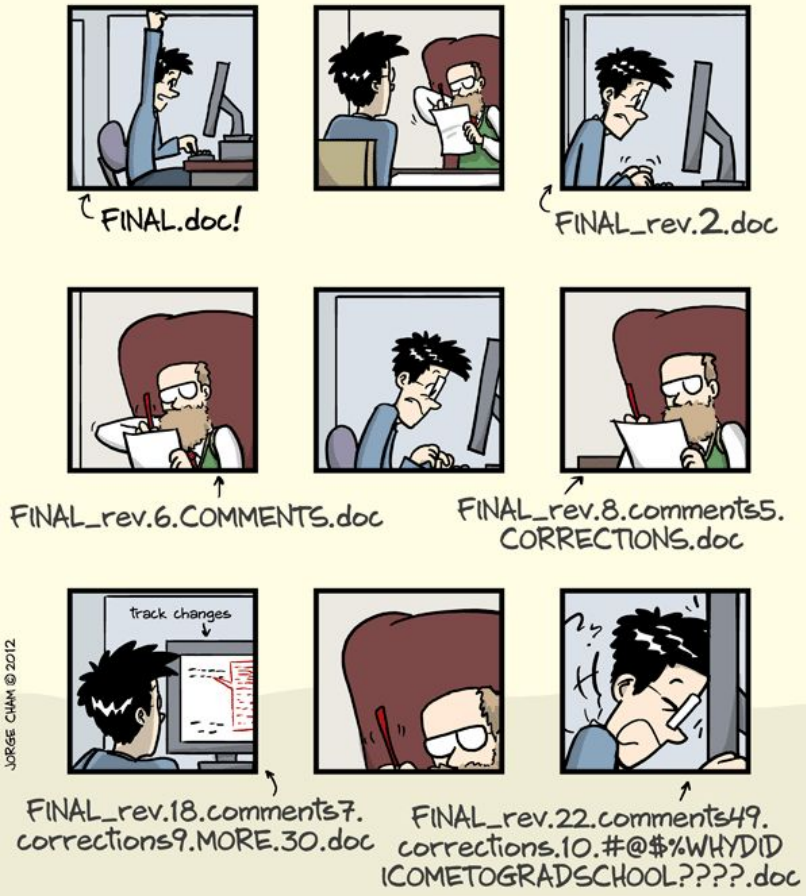
# Aquí entra a la cancha...

# Git & GitHub

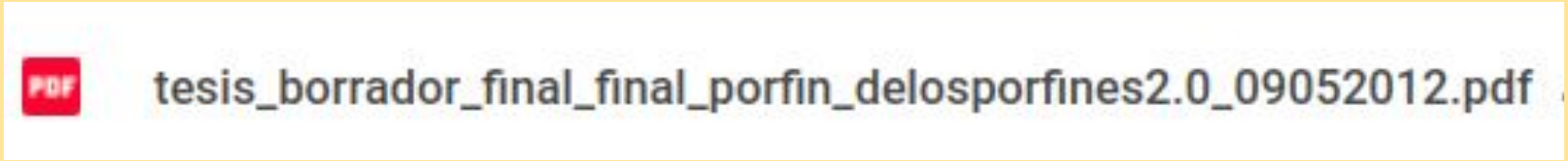
**Software de control de cambios y plataforma de desarrollo colaborativo, respectivamente.**

# Esto suena complicado...

# "FINAL".doc


FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012

# (Otro paréntesis personal)



# (Otro paréntesis personal)

# Git evita esta situación

# Mejor un ejemplo práctico

quishqa/AAS4WRF.py: An

github.com/quishqa/AAS4WRF.py

Apps | EI | Proyecto | NCL | powercomma... | LAB_AQ | modelos | editcommands | USP_meteoro | Modelagem | Booze app | Other bookmarks

# quishqa / AAS4WRF.py

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

Another Assimilation System for WRF-Chem, python flavored.

Edit

Manage topics

20 commits | 1 branch | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

quishqa Now, conservative method is well implemented, cell area is not requir... — Latest commit 1d512ff 2 days ago

| | | |
|---|---|---|
| .gitignore | Correction in conservative method, multiply by cell area, regrid, and… | 2 days ago |
| README.md | Now, conservative method is well implemented, cell area is not requir… | 2 days ago |
| aas4wrf.py | Now, conservative method is well implemented, cell area is not requir… | 2 days ago |
| aas4wrf.yml | Change to YAML config file to better emission file set up, Print diff… | 15 days ago |
| aas4wrf_example.svg | Update README.md to use aas4wrf.yml, add img example | 15 days ago |
| emissions.txt | README.md updated and emissions.txt uplodad | 21 days ago |
| requirements.txt | Change to YAML config file to better emission file set up, Print diff… | 15 days ago |
| wrfchemi_zeros.py | script to create a wrfchemi with zero emissions | 21 days ago |

README.md

# aas4wrf.py

`aas4wrf.py` is a Python script to create the `wrfchemi` file from local emissions needed to run WRF-Chem model. It's based on his older broder AAS4WRF.ncl. Currently, It works with CBMZ/MOSAIC chemical mechanism and for surface emissions.

## Installation

You need to install the packages that `aas4wrf.py` needs. We recommend to use miniconda or anaconda

You can download this repo or clone it by:

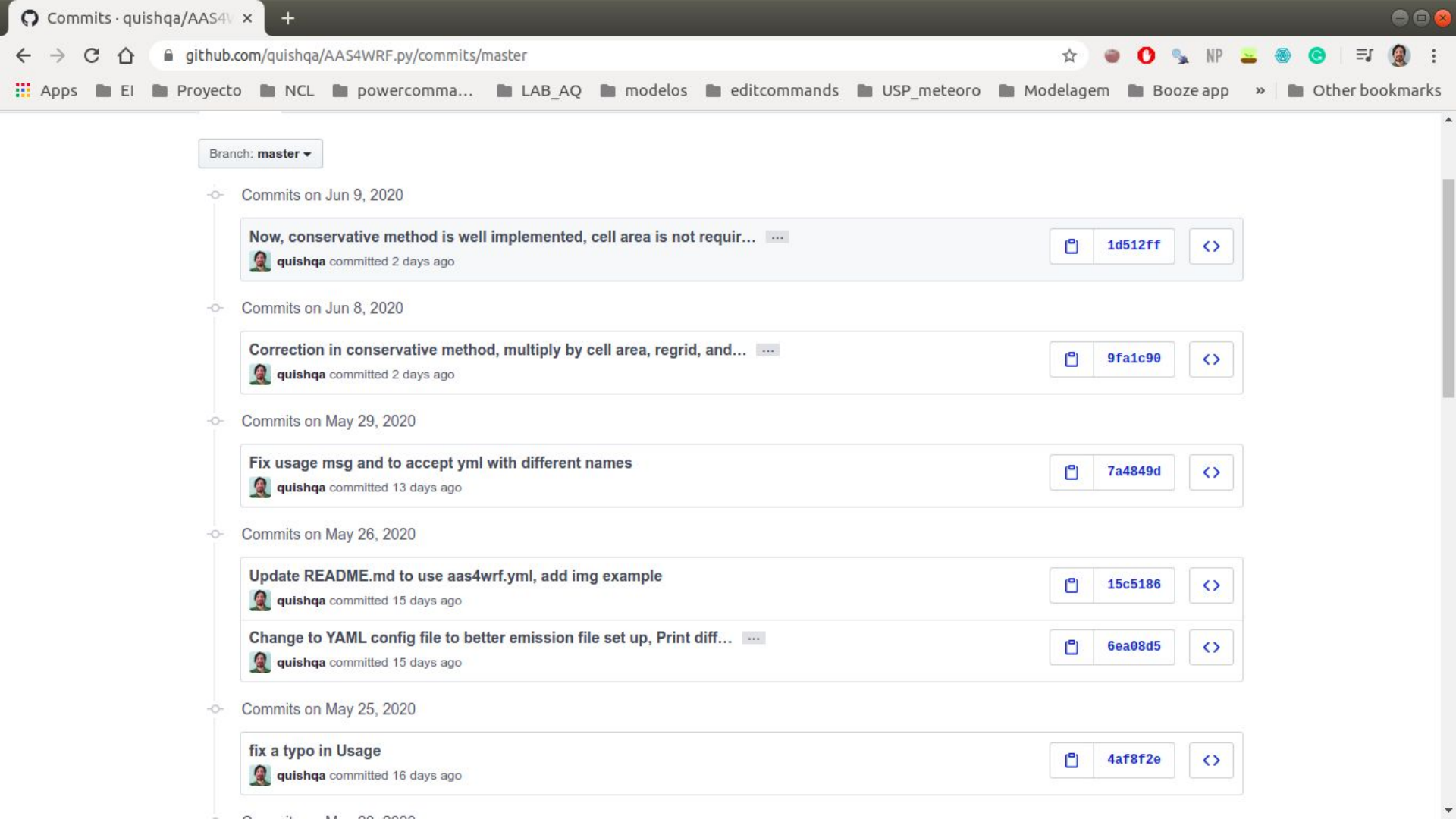
```
git clone https://github.com/quishqa/AAS4WRF.py.git
```

Then add `conda-forge` channel by:

```
conda config --add channels conda-forge
```

To avoid conflicts during the Installation, we also recommend create a new environment to run `aas4wrf.py` :

```
conda create --name aas4wrf
conda activate aas4wrf
```


Commits · quishqa/AAS4
github.com/quishqa/AAS4WRF.py/commits/master
Apps
EI
Proyecto
NCL
powercomma...
LAB_AQ
modelos
editcommands
USP_meteoro
Modelagem
Booze app
Other bookmarks
Branch: master
Commits on Jun 9, 2020
Now, conservative method is well implemented, cell area is not requir...
quishqa committed 2 days ago
1d512ff
Commits on Jun 8, 2020
Correction in conservative method, multiply by cell area, regrid, and...
quishqa committed 2 days ago
9fa1c90
Commits on May 29, 2020
Fix usage msg and to accept yml with different names
quishqa committed 13 days ago
7a4849d
Commits on May 26, 2020
Update README.md to use aas4wrf.yml, add img example
quishqa committed 15 days ago
15c5186
Change to YAML config file to better emission file set up, Print diff...
quishqa committed 15 days ago
6ea08d5
Commits on May 25, 2020
fix a typo in Usage
quishqa committed 16 days ago
4af8f2e

## Fix usage msg and to accept yml with different names

Browse files

master

**quishqa** committed 13 days ago — 1 parent 15c5186 — commit 7a4849dbf75b679cb73f777b9a0f91b919f339ca

Showing **1 changed file** with **4 additions** and **2 deletions**.

Unified Split

6 aas4wrf.py

```
@@ -264,11 +264,13 @@ def print_conservation(wrfchemi, emiss_input, pol):
264 264     import sys
265 265     import yaml
266 266     if len(sys.argv) < 2:
267     -       print('usage: python {} aasf4wrf.cfg'.format(sys.argv[0]))
    267 +       print('usage: python {} aasf4wrf.yml'.format(sys.argv[0]))
268 268         sys.exit()
    269 +
    270 +   config_file = sys.argv[1]
269 271
270 272     # Retrieving parameters from config file
271     -   with open('aas4wrf.yml') as file:
    273 +   with open(config_file) as file:
272 274         config = yaml.load(file, Loader=yaml.FullLoader)
273 275
274 276     wrfinput_file = config['Input']['wrfinput_file']
```

**0 comments on commit** `7a4849d`

Lock conversation

- Un programa en python con git

- Un programa en python con git
- Se “subió” a github

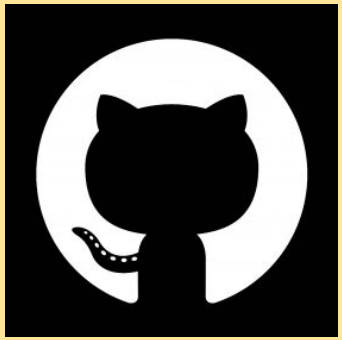
- Un programa en python con git
- Se “subió” a github
- Colegas pueden evaluar el método, siguiendo el README

- Un programa en python con git
- Se “subió” a github
- Colegas pueden replicar el método, siguiendo el README
- Pueden ver las correcciones

- Un programa en python con git
- Se “subió” a github
- Colegas pueden replicar el método, siguiendo el README
- Pueden ver las correcciones
- **Pueden crear copias, examinar, corregir y mejorar el programa.**

# Se replica, colabora y mejora :)

# (Último paréntesis personal)



>

Google Drive

>

# Resumiendo...

**Preferimos lenguajes científicos de programación.**

**Son gratis y versátiles. Hay una curva de aprendizaje, pero hay mucha ayuda online!**

# Importante en la reproducibilidad de la ciencia.

# Palabras finales

# Por qué estoy aquí?

**Invitarles a probar R o python, latex y git.**
**Con paciencia y optimismo.**


Dude, sucking at something is the first step
towards being sort of good at something

# Está presentación se ayudó de:


- Lowndes, J.S.S. et al. Our path to better science in less time using open data science tools. Nat. Ecol. Evol. 1, 0160 (2017).
- Introduction to Open Reproducible Science Workflows, Earth Lab Course, in: https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/
- **Scopatz, A., & Huff, K. D. (2015). Effective computation in physics.**
- Bartlett, Alice. 2016. “Git for Humans.” Financial Times, London; Talk at UX Brighton. (https://speakerdeck.com/alicebartlett/git-for-humans)
- Y varios memes del internet

# Lleve lleve casero!

- R (Junto con Rstudio)

  https://cran.r-project.org/bin/windows/base/

- Python (Junto con Spyder o Jupyter)

  https://docs.conda.io/en/latest/miniconda.html

- Latex

  https://www.overleaf.com

# gracias!

mario.calderon@iag.usp.br