Data visualization of O3 levels in Europe

5011CEM
Big Data Programming Project Module
24th of April of 2020

Project Management

For the management of the project a hybrid agile methodology was used. It implements both scrum methodology and Kanban, which depends on a Kanban board. The Kanban board was implemented with the help of trello. However, since this is an individual project, some processes were simplified, such as sprint reviews and daily scrums. Similarly, the scrum master and the team were represented by only one person.

User stories

User stories were the first step to manage successfully the project. The conventional method was applied, "As a [person] I [want to], [so that]" and each other given and id. Later, the sprint duration was set for one week, where each task would take around 3 hours to complete.

ID	User stories
1	As a user, I want to have the possibility to choose the colour of the data, so that it suits my needs (accessibility)
2	As a user, I want the possibility to visualize the graphs, so that I can analyse the data correctly
3	As a user, I want to visualize each data model values in a map, so that I can relate the data to its location
4	As a user, I want to visualize and compare the values of each model in window, so that I can compare and choose the best model to work with
5	As a user, I want a colorbar to display the map's data values, so that I know which colour represents each set of values
6	As a user, I want a "Quit All" button, so that all windows can be closed at the same time.
7	As a user, I want the graphs to be save in a folder, so that I can use later without having to run the program again.
8	As a user, I want fast processing of Big Data, so that processing time doesn't take too long to obtain results.

Sprint backlog

The sprint backlog is essentially a list of tasks that were broken down into smaller, more achievable parts from user stories. This list also incorporates a point system to help evaluate each task and its duration. In this point system, one point corresponds to one day where three hours are dedicated to that task. So, if a task has a punctuation of two, 6 hours were spent in that task over a period of two days.

User Stories	Tasks	Days(points)
As a user, I want to have the possibility to choose the colour of the data, so that it suits my needs (accessibility)	Code terminal-based inputs	1
	Change graphs colour	2
As a user, I want the possibility to visualize each model in graph, so that I can analyse the data correctly	Open data and extract models	1
	Create graphs axis	2
	Plot model's data in a graph	2
As a user, I want to visualize each data model values in a map, so that I can relate the data to its location	Extract coordinates and time	1
	Overlay map based on coordinates	2
As a user, I want to visualize and compare the values of each model in a window, so that I can compare and choose the best model to work with	Create a window per hour	1
	Create a graph per model in one window	3
	Positions the graphs in the window	1
	Add tittle of each model on top of the graphs	1
	Add windows title	1
As a user, I want a colorbar to display the map's data values, so that I know which colour represents each set of values	Create the colorbar	2
	Assign colour of the graphs to the colorbar	1
	Position colorbar on the window	1
	Set max ticks of colorbar	2
As a user, I want a "Quit All" button, so that all windows can be closed at the same time.	Create the button	1
	Set callback	1
	Position the button on the window	1
As a user, I want the graphs to be save in a folder, so that I can use later without having to run the program again.	Create folder on directory	1
	Save windows as png	1
	Rename image with the hours the models represent	1
As a user, I want fast processing of Big Data, so that processing time doesn't take too long to obtain results.	Implement parallel processing	4

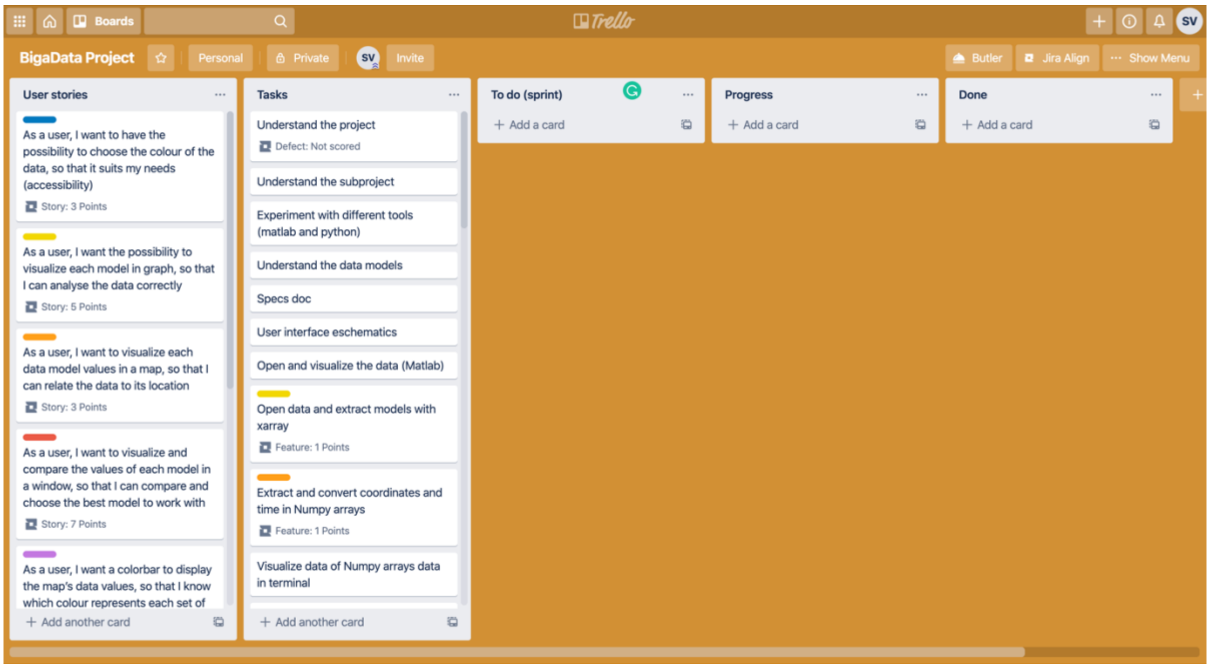
Sprint planning

The sprint planning consists of creating a plane with the tasks that are meant to be completed in a week having in mind the point system created in the sprint backlog.

Sprints	Tasks
1	Understand the project
	Understand the subproject
2	Experiment with different tools (MATLAB and python)
	Understand the data models
	Specs doc
3	User interface schematics
	Open and visualize the data (MATLAB)
4	Open data and extract models with xarray
	Extract and convert coordinates and time in Numpy arrays
	Visualize data of Numpy arrays data in terminal
	Visualize data of Numpy arrays data in terminal
	Experiment with Matplotlib
5	Create a window
	Plot model's data in a graph
	Create a graph per model in one window
	Positions the graphs in the window
	Overlay map based on coordinates
6	Create graphs axis
	Add tittle of each model on top of the graphs
	Create a window per hour
	Add windows title
	Create the colorbar
	Position colorbar on the window
	Assign colour of the graphs to the colorbar
	Set max ticks of colorbar
7	Create the button
	Set callback
	Position the button on the window
	Create folder on directory
	Save windows as png
	Rename image with the hours the models represent
8	Implement parallel processing
	Code terminal-based inputs
	Change graphs colour
9	Error handling
	Testing

Kanban board

Lastly, the Kanban board was used due to its ability to visualize the tasks, which consequently helps maximizing efficiency. When implementing the Kanban methodology, user stories were also added in a separate column next to it other columns for all tasks was implemented. The tasks were colour coded to reference them to their respective user story and their points also incorporated. Additionally, 3 more columns were created: sprint, where the tasks correspondent to that sprint were added at the beginning of the week; in progress and done.



User stories

- As a user, I want to have the possibility to choose the colour of the data, so that it suits my needs (accessibility) — Story: 3 Points
- As a user, I want the possibility to visualize each model in graph, so that I can analyse the data correctly — Story: 5 Points
- As a user, I want to visualize each data model values in a map, so that I can relate the data to its location — Story: 3 Points
- As a user, I want to visualize and compare the values of each model in a window, so that I can compare and choose the best model to work with — Story: 7 Points
- As a user, I want a colorbar to display the map's data values, so that I know which colour represents each set of
- + Add another card

Tasks

- Understand the project — Defect: Not scored
- Understand the subproject
- Experiment with different tools (matlab and python)
- Understand the data models
- Specs doc
- User interface eschematics
- Open and visualize the data (Matlab)
- Open data and extract models with xarray — Feature: 1 Points
- Extract and convert coordinates and time in Numpy arrays — Feature: 1 Points
- Visualize data of Numpy arrays data in terminal
- + Add another card

To do (sprint)

- + Add a card

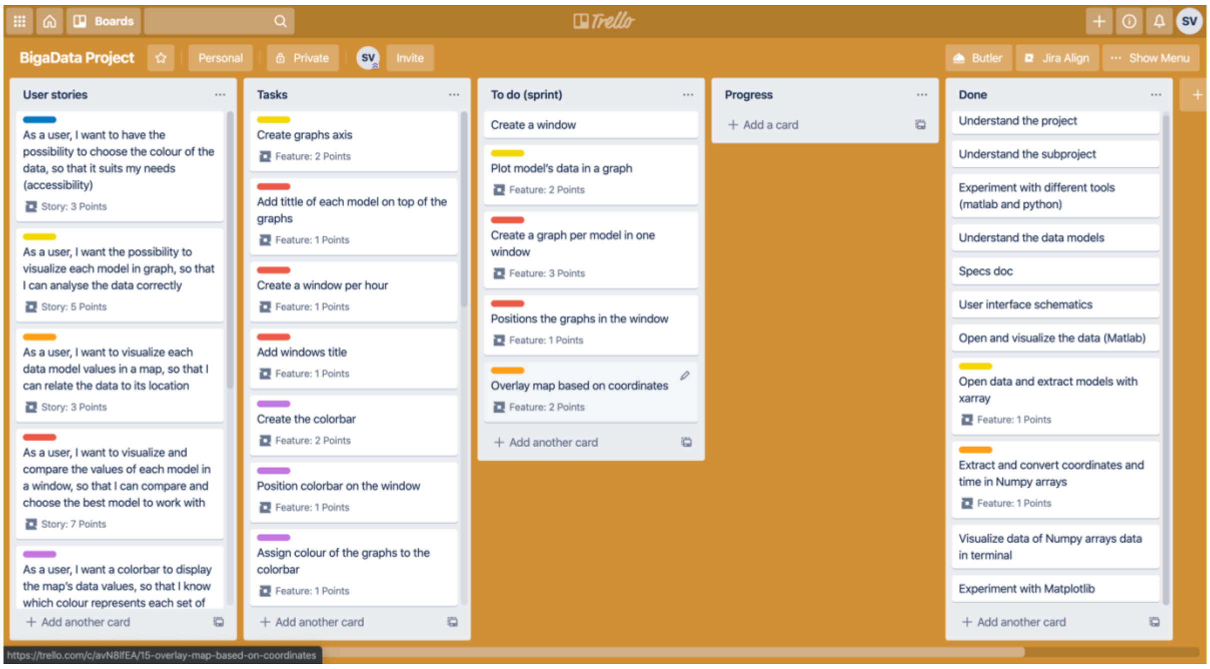
Progress

- + Add a card

Done

- + Add a card

Figure 1 - begging of the project (kanban board)



User stories

- As a user, I want to have the possibility to choose the colour of the data, so that it suits my needs (accessibility) — Story: 3 Points
- As a user, I want the possibility to visualize each model in graph, so that I can analyse the data correctly — Story: 5 Points
- As a user, I want to visualize each data model values in a map, so that I can relate the data to its location — Story: 3 Points
- As a user, I want to visualize and compare the values of each model in a window, so that I can compare and choose the best model to work with — Story: 7 Points
- As a user, I want a colorbar to display the map's data values, so that I know which colour represents each set of
- + Add another card

Tasks

- Create graphs axis — Feature: 2 Points
- Add tittle of each model on top of the graphs — Feature: 1 Points
- Create a window per hour — Feature: 1 Points
- Add windows title — Feature: 1 Points
- Create the colorbar — Feature: 2 Points
- Position colorbar on the window — Feature: 1 Points
- Assign colour of the graphs to the colorbar — Feature: 1 Points
- + Add another card

To do (sprint)

- Create a window
- Plot model's data in a graph — Feature: 2 Points
- Create a graph per model in one window — Feature: 3 Points
- Positions the graphs in the window — Feature: 1 Points
- Overlay map based on coordinates — Feature: 2 Points
- + Add another card

Progress

- + Add a card

Done

- Understand the project
- Understand the subproject
- Experiment with different tools (matlab and python)
- Understand the data models
- Specs doc
- User interface schematics
- Open and visualize the data (Matlab)
- Open data and extract models with xarray — Feature: 1 Points
- Extract and convert coordinates and time in Numpy arrays — Feature: 1 Points
- Visualize data of Numpy arrays data in terminal
- Experiment with Matplotlib
- + Add another card

https://trello.com/c/avN8lfEA/15-overlay-map-based-on-coordinates

Figure 2 - sprint 3 of the project (kanban board)

Time Management

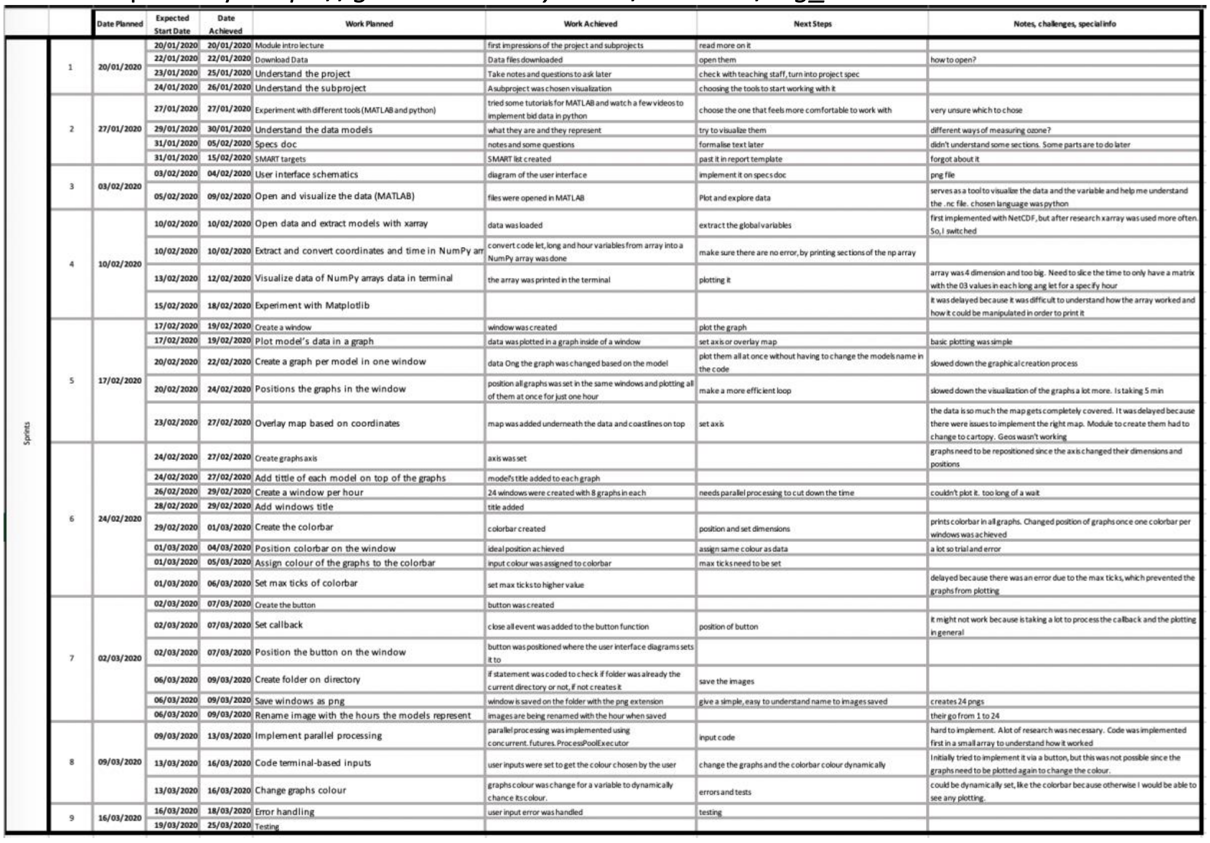
Time management is essential when developing a project to ensure its delivery in the expected time period. Agile methodology not only ensures the management of the project through tasks, but also its time management.

Both in scrum and Kanban, tasks are measure by their difficult, which therefore, reflects on the time spent one each task. With the backlog sprints tasks are analysed and a time measurement implemented as a point system; sprints planning groups the tasks into achievable goals for a period of time, which in this case was a week. These two lists help plane the project, but Kanban helps implementing it by forcing a team member to focus on a specific task until is completed.

Version Control

Version control is the management of file changes in an environment where several versions of the same files are store over time. Version control is vital not only to protect the files from accidental loses or undesired changes. This is especially relevant when work in collaborative projects, where several team members might be altering the same files and mistake are more prone to be made. This facilitates the correction of those issues a lot more easily. One of the tools used for this is GitHub. However, for this project a logbook was also created instead.

GitHub repository: *https://github.coventry.ac.uk/valente3/big_data*

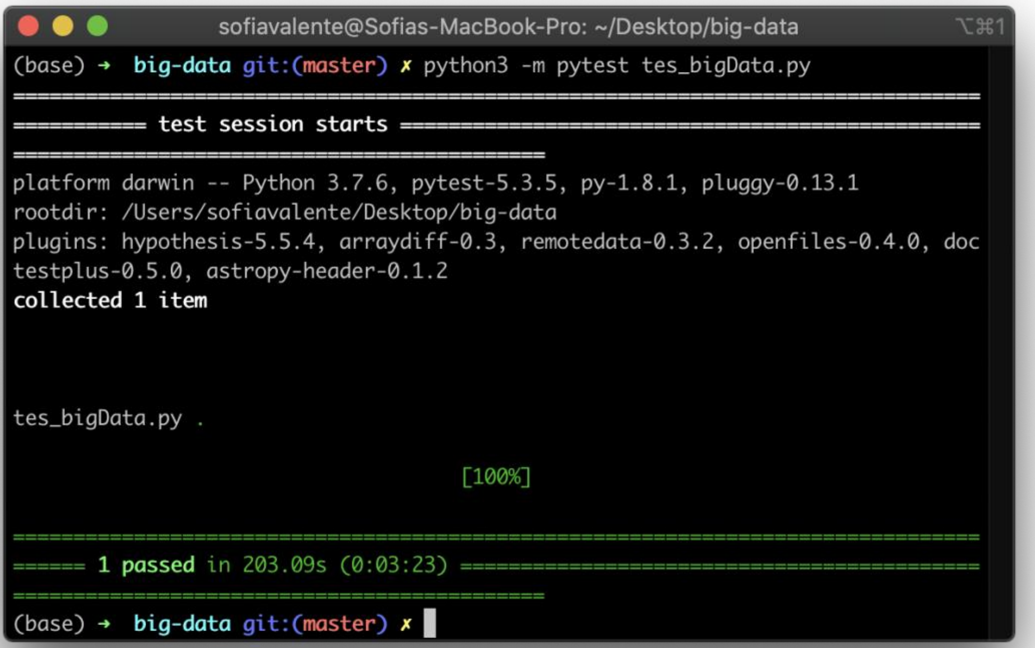


		Date Planned	Expected Start Date	Date Achieved	Work Planned	Work Achieved	Next Steps	Notes, challenges, special info
Sprints	1	20/01/2020	20/01/2020	20/01/2020	Module intro lecture	first impressions of the project and subprojects	read more on it	
			22/01/2020	22/01/2020	Download Data	Data files downloaded	open them	how to open?
			23/01/2020	25/01/2020	Understand the project	Take notes and questions to ask later	check with teaching staff, turn into project spec	
			24/01/2020	26/01/2020	Understand the subproject	A subproject was chosen visualization	choosing the tools to start working with it	
	2	27/01/2020	27/01/2020	27/01/2020	Experiment with different tools (MATLAB and python)	tried some tutorials for MATLAB and watch a few videos to implement bid data in python	choose the one that feels more comfortable to work with	very unsure which to chose
			29/01/2020	30/01/2020	Understand the data models	what they are and they represent	try to visualize them	different ways of measuring ozone?
			31/01/2020	05/02/2020	Specs doc	notes and some questions	formalise text later	didn't understand some sections. Some parts are to do later
			31/01/2020	15/02/2020	SMART targets	SMART list created	past it in report template	forgot about it
	3	03/02/2020	03/02/2020	04/02/2020	User interface schematics	diagram of the user interface	implement it on specs doc	png file
			05/02/2020	09/02/2020	Open and visualize the data (MATLAB)	files were opened in MATLAB	Plot and explore data	serves as a tool to visualize the data and the variable and help me understand the .nc file. chosen language was python
	4	10/02/2020	10/02/2020	10/02/2020	Open data and extract models with xarray	data was loaded	extract the global variables	first implemented with NetCDF, but after research xarray was used more often. So, I switched
			10/02/2020	10/02/2020	Extract and convert coordinates and time in NumPy arr	convert code let, long and hour variables from array into a NumPy array was done	make sure there are no error, by printing sections of the np array	
			13/02/2020	12/02/2020	Visualize data of NumPy arrays data in terminal	the array was printed in the terminal	plotting it	array was 4 dimension and too big. Need to slice the time to only have a matrix with the 03 values in each long ang let for a specify hour
			15/02/2020	18/02/2020	Experiment with Matplotlib			it was delayed because it was difficult to understand how the array worked and how it could be manipulated in order to print it
	5	17/02/2020	17/02/2020	19/02/2020	Create a window	window was created	plot the graph	
			17/02/2020	19/02/2020	Plot model's data in a graph	data was plotted in a graph inside of a window	set axis or overlay map	basic plotting was simple
			20/02/2020	22/02/2020	Create a graph per model in one window	data Ong the graph was changed based on the model	plot them all at once without having to change the models name in the code	slowed down the graphical creation process
			20/02/2020	24/02/2020	Positions the graphs in the window	position all graphs was set in the same windows and plotting all of them at once for just one hour	make a more efficient loop	slowed down the visualization of the graphs a lot more. Is taking 5 min
			23/02/2020	27/02/2020	Overlay map based on coordinates	map was added underneath the data and coastlines on top	set axis	the data is so much the map gets completely covered. It was delayed because there were issues to implement the right map. Module to create them had to change to cartopy. Geos wasn't working
	6	24/02/2020	24/02/2020	27/02/2020	Create graphs axis	axis was set		graphs need to be repositioned since the axis changed their dimensions and positions
			24/02/2020	27/02/2020	Add tittle of each model on top of the graphs	model's title added to each graph		
			26/02/2020	29/02/2020	Create a window per hour	24 windows were created with 8 graphs in each	needs parallel processing to cut down the time	couldn't plot it. too long of a wait
			28/02/2020	29/02/2020	Add windows title	title added		
			29/02/2020	01/03/2020	Create the colorbar	colorbar created	position and set dimensions	prints colorbar in all graphs. Changed position of graphs once one colorbar per windows was achieved
			01/03/2020	04/03/2020	Position colorbar on the window	ideal position achieved	assign same colour as data	a lot so trial and error
			01/03/2020	05/03/2020	Assign colour of the graphs to the colorbar	input colour was assigned to colorbar	max ticks need to be set	
			01/03/2020	06/03/2020	Set max ticks of colorbar	set max ticks to higher value		delayed because there was an error due to the max ticks, which prevented the graphs from plotting
	7	02/03/2020	02/03/2020	07/03/2020	Create the button	button was created		
			02/03/2020	07/03/2020	Set callback	close all event was added to the button function	position of button	it might not work because is taking a lot to process the callback and the plotting in general
			02/03/2020	07/03/2020	Position the button on the window	button was positioned where the user interface diagrams sets it to		
			06/03/2020	09/03/2020	Create folder on directory	if statement was coded to check if folder was already the current directory or not, if not creates it	save the images	
			06/03/2020	09/03/2020	Save windows as png	window is saved on the folder with the png extension	give a simple, easy to understand name to images saved	creates 24 pngs
			06/03/2020	09/03/2020	Rename image with the hours the models represent	images are being renamed with the hour when saved		their go from 1 to 24
	8	09/03/2020	09/03/2020	13/03/2020	Implement parallel processing	parallel processing was implemented using concurrent.futures.ProcessPoolExecutor	input code	hard to implement. A lot of research was necessary. Code was implemented first in a small array to understand how it worked
			13/03/2020	16/03/2020	Code terminal-based inputs	user inputs were set to get the colour chosen by the user	change the graphs and the colorbar colour dynamically	Initially tried to implement it via a button, but this was not possible since the graphs need to be plotted again to change the colour.
			13/03/2020	16/03/2020	Change graphs colour	graphs colour was change for a variable to dynamically chance its colour.	errors and tests	could be dynamically set, like the colorbar because otherwise I would be able to see any plotting.
	9	16/03/2020	16/03/2020	18/03/2020	Error handling	user input error was handled	testing	
			19/03/2020	25/03/2020	Testing			

Figure 3 - logbook

Automated Testing

Testing is an essential tool to guaranty product quality. However, automated tests not only evaluate software, but also reduce the time wasted in the testing process. nevertheless, automated testing can be quite challenging, especially when any kind of visualization is required for the software development. In this project, several methods were experimented with. The chosen module was Pytest, as it was the recommended one from matplotlib. This creates a reference plot and compares it to the obtained one. However, this is only successful for simple graphs. When using Cartopy and other modules together, Pytest is not fully effective, only comparing the dimensions of the both windows.



```
sofiavalente@Sofias-MacBook-Pro: ~/Desktop/big-data
(base) ➜  big-data git:(master) ✗ python3 -m pytest tes_bigData.py
==========================================================================================
========== test session starts ============================================================
==========================================
platform darwin -- Python 3.7.6, pytest-5.3.5, py-1.8.1, pluggy-0.13.1
rootdir: /Users/sofiavalente/Desktop/big-data
plugins: hypothesis-5.5.4, arraydiff-0.3, remotedata-0.3.2, openfiles-0.4.0, doc
testplus-0.5.0, astropy-header-0.1.2
collected 1 item



tes_bigData.py .

                                    [100%]

==========================================================================================
===== 1 passed in 203.09s (0:03:23) ======================================================
==========================================
(base) ➜  big-data git:(master) ✗
```

Figure 4 - testing output