**Consequences of the Covid-19 pandemic**

**World map of GDP changes and corona-related deaths in R studio**

Mie Arnau Martinez (MAM) (Student ID: 201806701)

School of Communication and Cognition, University of Aarhus,

Jens Chr. Skous Vej 2, 8000 Aarhus, Denmark

**Abstract**

In the light of the Covid-19 crisis, countries and states have instituted lockdowns and reduced production worldwide. However, not all countries approached the problem in the same way and therefore countries were not similarly affected. This paper aims to investigate the consequences of the pandemic on the individual countries. Which countries came out unscathed and which where most affected by the fast spread of the virus? For this, I examined the number of Covid-19 related deaths as well as how the pandemic affected the countries GDPs. The differences are visualized on world maps to ease interpretability and give a clear and fast overview.

**Keywords:** *Covid-19, lockdown, pandemic, GDP, R studio*

**Introduction**

The Covid-19 crisis has arguably been the most characterizing feature of the year 2020. It has brought numerous press-conferences outlining new approaches for controlling the spread of the disease. The most effective method for damage-control have seemed to be lockdowns of entire cities and countries. However, this comes at the cost of owners and employees of shops and restaurants and it severely affected the economy. Similarly, every way of approaching the crisis has had benefits and consequences. This project aims at enlightening how each country was affected by the spread of Covid-19. This is an important topic to examine as prior knowledge is the best tool to uncover how to best approach a similar problem the next time it occurs. To investigate this, I will use digital tools in R studio to display the data on a world map. This will provide a data presentation that is clear and easy to interpret.

**Problem and background**

Coronavirus disease 2019 (Covid-19) is an infectious disease caused by a single strand RNA virus. The virus is believed to have originated in a food market in Wuhan, China (Wu et al, 2020). Since then, the virus has been detected in numerous countries across the continents (Mas-Coma et al., 2020). One might argue that this is the result of having a globalized world in which people are travelling more than ever before as it allows for contagious diseases to transmit across borders (REF). However, this also means that pandemics could be a recurring problem. Therefore, it is important to examine the different approaches of stopping the spread of the virus. It is important to know what worked and what didn’t work so that we can be more prepared for the next virus. So, what is the most effective damage-control during a pandemic? To investigate this, I will look at all countries to see how they were affected by the virus. Specifically, I will examine the change in gross domestic product (GDP) in 2019 and 2020 to see how the Covid-19 influenced the economy. Moreover, I will look at the number of people that died from Covid-19. These two parameters are chosen due to a believed negative correlation between number of people who died from the virus and the state of the economy.

**Software framework**

To undertake my project, I used a 6-months old Lenovo IdeaPad S340-14IIL, 8 Gb RAM, which runs the Windows 10 operating system. Furthermore, the data preparation, analysis and data visualizations were produced in RStudio (1.3.959) with the desktop version of R (4.0.0) (R Core Team, 2020). Moreover, various libraries were loaded from the Comprehensive R Archive Network (CRAN) package repository. These include the **rgeos**, **ggplot2**, **sf**, **rnaturalearth** and **ggspatial** packages which were used for displaying the data on a world map. Additionally, world data was acquired through the package **rnaturalearthdata**. The project consists of two scripts. The first script is for data acquisition and preprocessing. The second script is for data visualization and correlation analysis. All data and code can be found on my Github repository (see Table 1, S2).

**Data Acquisition and Processing**

For the project, web scraping was used to obtain data on number of people who died of Covid-19 in each country. The data was last updated the 16th of December 2020 and retrieved from statista.com (See *Table 2, D1*). Statista is a provider of market and consumer data and is increasingly cited in media articles (Statista, 2020). After retrieval, the data was in the format of a single string. Therefore, preprocessing was needed to transform it into a data frame. Mainly, this consisted of substitution with regular expressions. For this, I used regex101.com to facilitate the search of appropriate string patterns to transform the data. Regex101 is a web page that allows one to try out different regular expressions and makes it easy to see what is captured by the different expressions.

The data for changes in gross domestic product in 2019 and 2020 was downloaded as an Excel file from the International Monetary Fund (imf.org – See *Table 2, D2*). Minimal preprocessing was needed for this data set.

For more details on data acquisition and preprocessing see the code in “Web\_scraping\_and\_data\_preprocessing.Rmd” under “script” folder in my\_final\_project (Link to Github repository can be seen in Table 1, S2). Lastly, data of all countries and their vector maps etc. was acquired through the CRAN packages **rnaturalearth** and **rnaturalearthdata** (South, 2017).

**Implementation and empirical results**

To make a simple presentation that shows how Covid-19 have had a different impact on each country, I decided to make world maps. This is an approach to express information of many variables and incorporating them in a way that is fast and easy to interpret. Furthermore, making bar plots would be difficult as it would have to display over 190 bars per plot. Besides the visualizations, I also conducted a correlation test of GDP changes in 2020 and corona-related deaths. The code and all plots can be found on Github (see Table 1, S2).

The first step of visualization was to merge the acquired data with the data frame obtained from the **rnaturalearthdata** package (South, 2017). During the merge, many rows were dropped due to differences in spelling. As I wanted as much data as possible, I made a for loop that could check and print out which countries were deleted in the two data frames. Countries were ordered alphabetically and this made it easy to compare the naming in the two data frames. Names were then changed to match and the data frames were merged again.

Making the world data frame, the function ne\_countries() from the **rnaturalearth** package (South, 2017) was used to get the polygons for all countries. Furthermore, the function returns a data frame containing the names of countries and other information like estimated population and whether the country is in a high or low income group. The data frame is also classified as a simple feature object from the **sf** package (Pebesma, 2018). This allows the program to know what column in the data frame refers to the spatial geometry of the polygons. These can then be plotted using **ggplot2** (Wickham, 2016).

I also created another data frame that holds the coordinates of a point on the surface of each polygon. This is used for printing the country names on the map to give a more clear overview. Again, the data frame is also a simple feature object with a geometry column.

Then the visualizations were made using **ggplot2** (Wickham, 2016) and adding layers. As mentioned, the first layer defines that the data is a world map and should be plotted as multipolygons. In this line of code, I added that the color of the fill should be equal to the parameter of interest (i.e., GDP change or number of corona-related deaths). Furthermore, I define the color scheme to be “plasma” as this facilitates viewing for people who are colorblind. For the visualization, a guide by Moreno and Basille (2018) were used. This explains and exhibits how the packages can be used to draw beautiful maps.

The first map (*Figure 1*) shows Covid-19 casualties by country in 2020. From this, it is apparent that the United States of America had the highest number of deaths of approx. 300.000. However, this does not account for population differences. Therefore, a second map was created (*Figure 2*). Here, the number of casualties are divided by population. Furthermore, the numbers are logarithmically transformed. This transformation had the purpose of drawing outliers closer to the mean. Thereby, differences between the countries are more prominent than they would have been without the transformation. In fact, without the log-transformation the map appeared to show no differences across countries. From this, one can gather that the western world along South America had most corona-related deaths per capita whereas most countries in Africa and China had the lowest number of casualties.

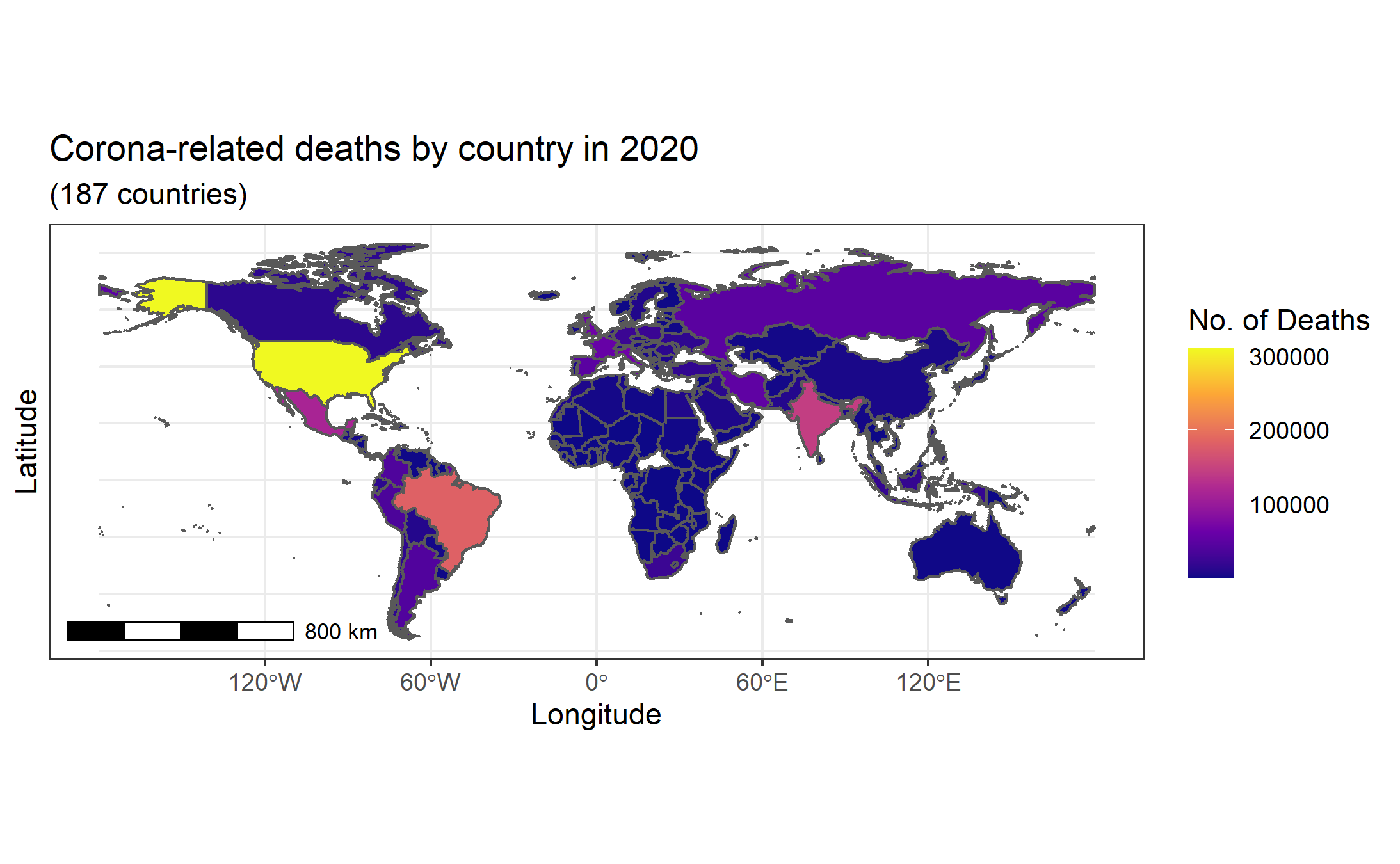


Figure 1 - World map displaying the raw numbers of Covid-19 casualties in 2020.

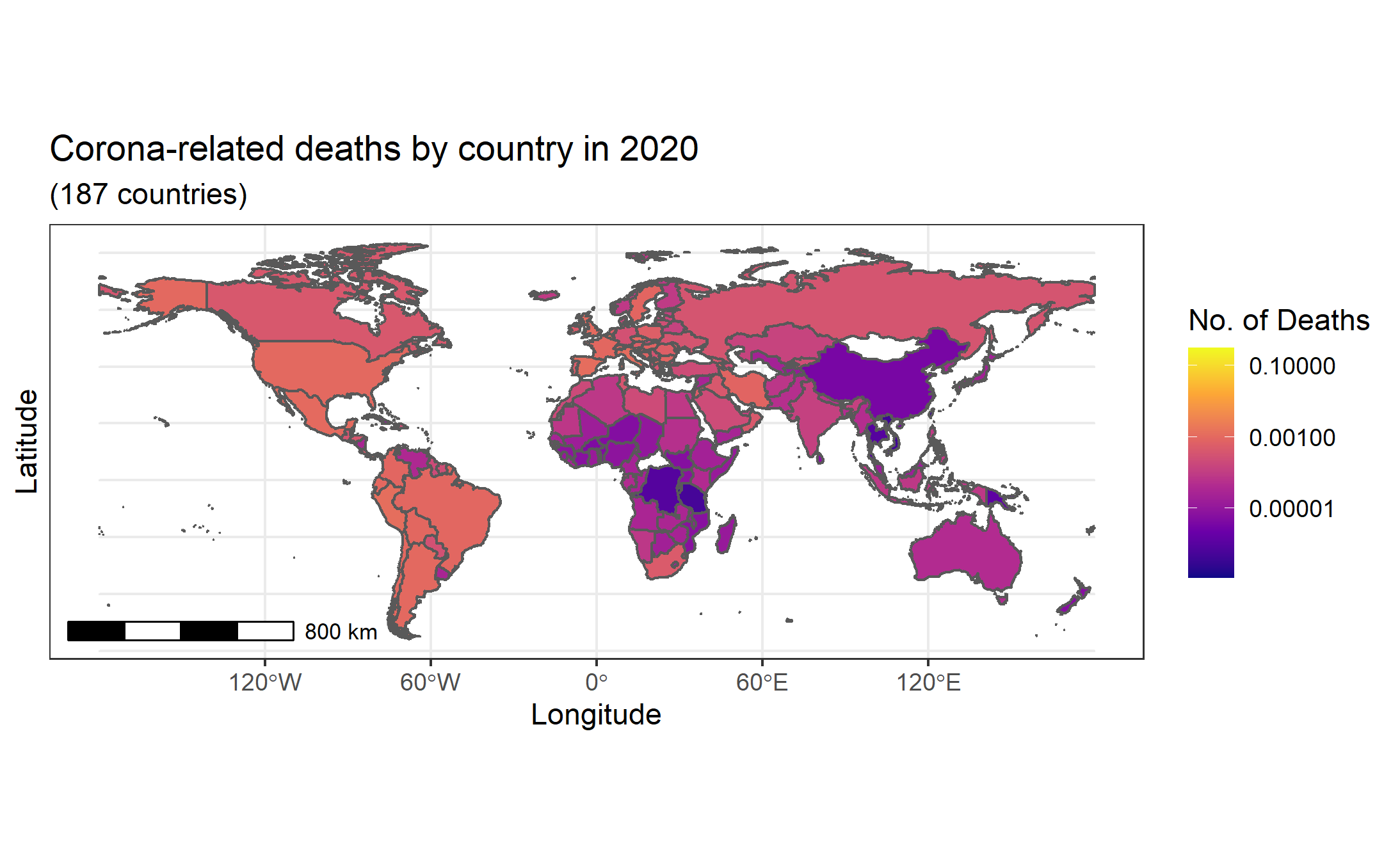


Figure 2 - World map of the Covid-19 casualties per capita. The data has been log-transformed to enhance the across-country differences.

These maps (*Figure 1 and 2*) show a great overview of how fatal the disease was for the world. However, it can be hard to see exactly how each country was affected. Therefore, I used the “coord\_sf()” function to zoom in on Europe (*Figure 3*). Moreover, I added labels for each country to facilitate identification. The greatest problem encountered was the placement of the countries’ names. First, I tried using the function “st\_centroid” recommended by the r-spatial guide (Moreno and Basille, 2018). However, this resulted in incorrect placing of country-labels. For instance, the “France” label was placed in the polygon that represents Spain. For this reason, I looked into the **sf** package (Pebesma, 2018) and found another operation that improved the location of the labels, i.e. “st\_point\_on\_surface()”. So, instead of calculating the centroid as the point in which to put the label, the latter function simply places the label within the surface of the polygon. The result can be seen in *Figure 3*. Here we can see that the countries with least number of deaths included Norway, Finland, Iceland and Belarus. Likewise, the countries with highest number of Covid-19-related deaths includes Spain, France, Italy, Belgium, the UK and Sweden.

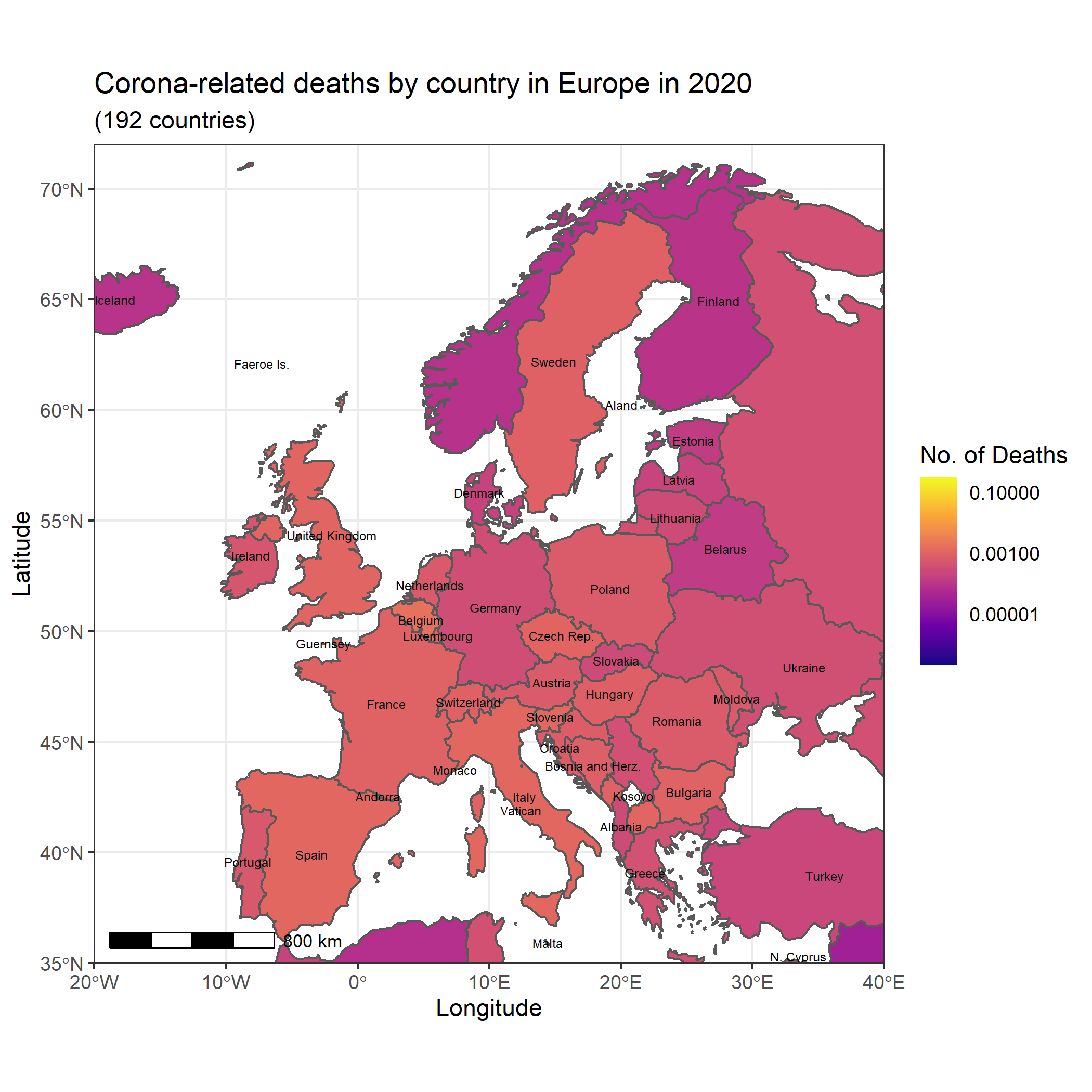


Figure 1 - Map of Europe showing Covid-19 casualties per capita.

The second parameter that I examined was the change in gross domestic product (GDP) in both 2019 (*Figure 4*) and 2020 (*Figure 5*). The reason to include both years is that they are linked. For instance, if a country has a great negative change of GDP in 2019 the change of GDP might not be as affected by Covid-19. Similarly, if a country was flourishing in 2019 the Covid-19 crises might have hit more severely. As an example of the latter, Libya had a very positive development in their GDP in 2019 (see *Figure 4*). However, they had a very negative change in GDP in 2020 (see *Figure 5*). Notably, they were also one of the African countries with highest number of deaths per capita (see *Figure 2*).

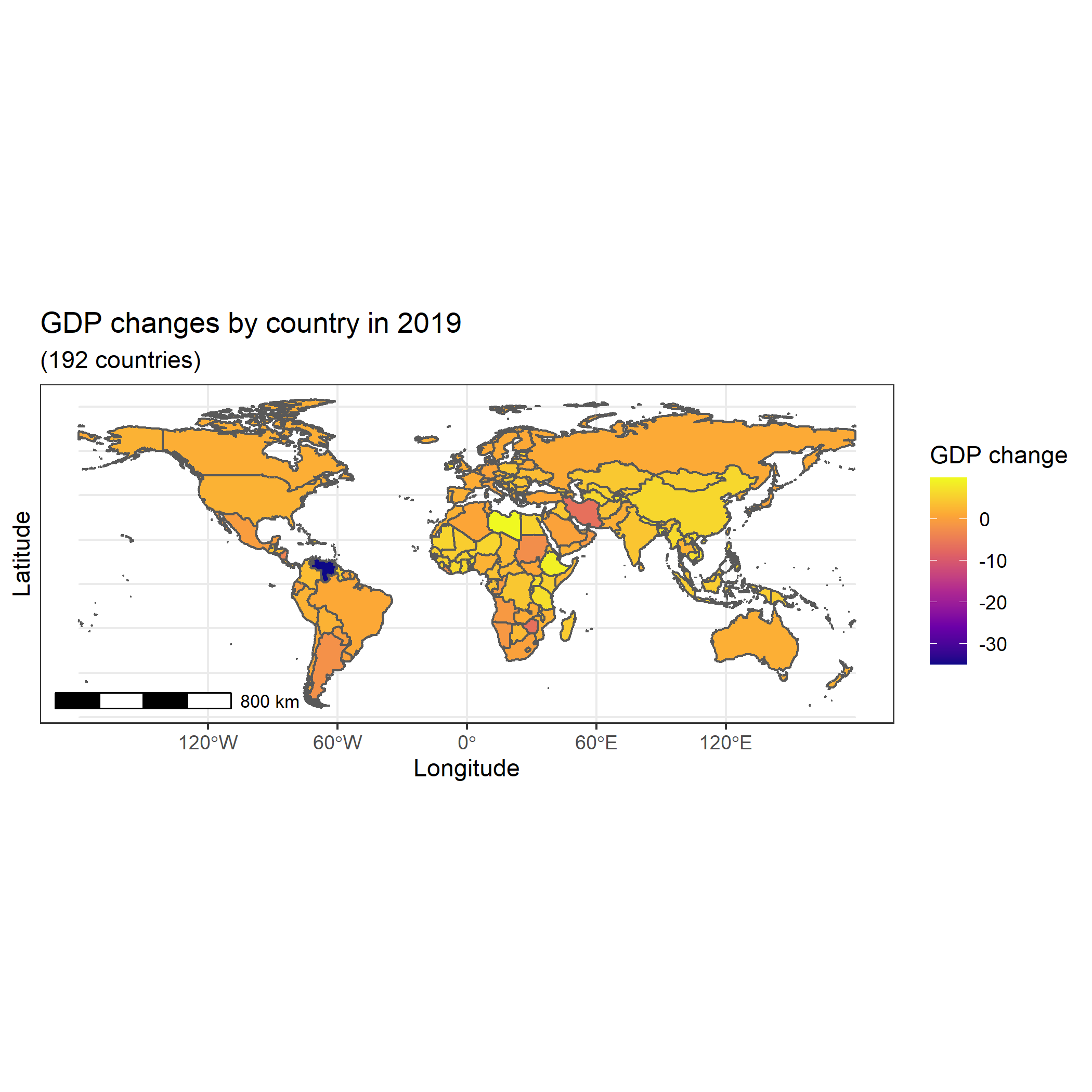


Figure 2 - Gross domestic product (GDP) changes by country in 2019.

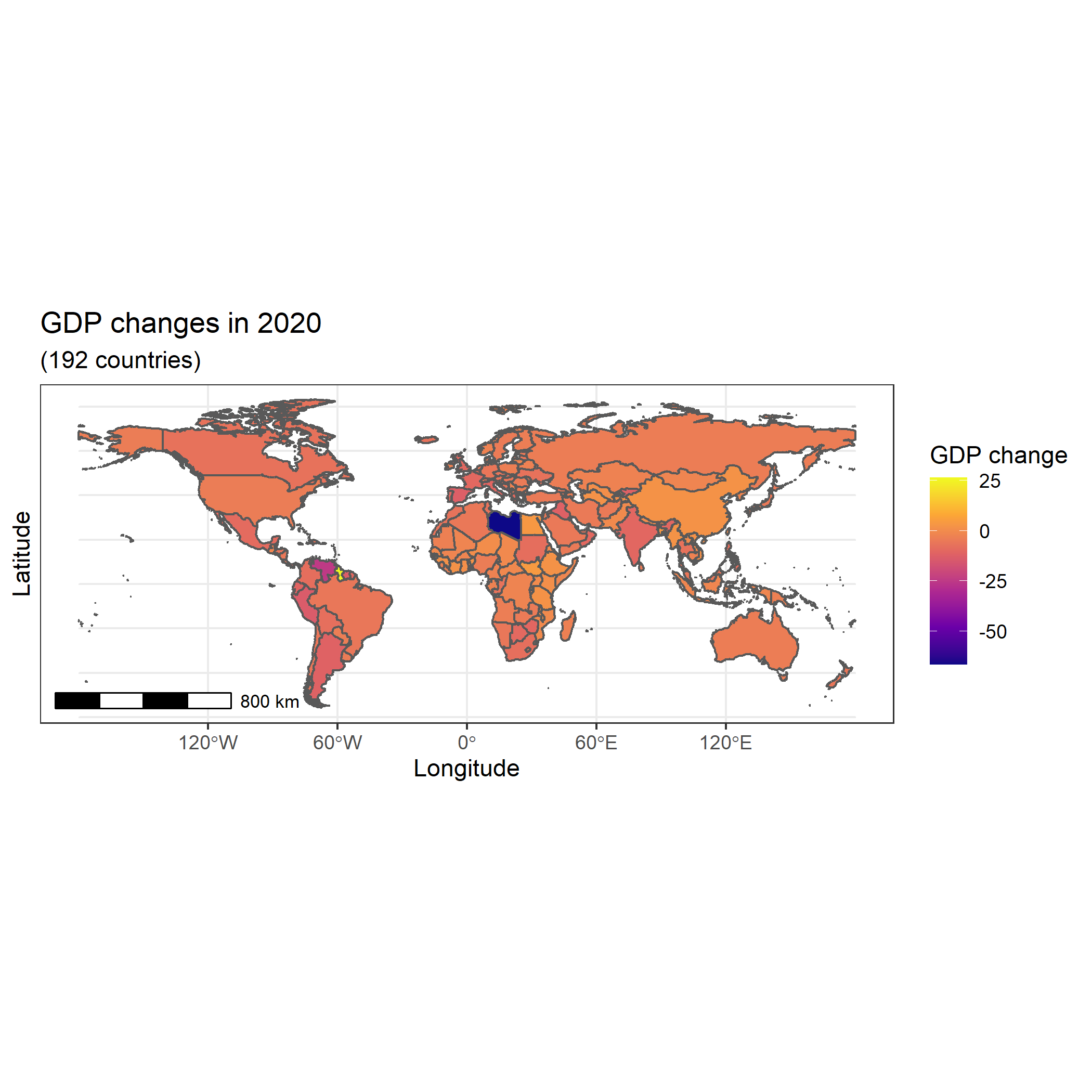


Figure 3 - Gross domestic product (GDP) changes by country in 2020.

It should be noted that the scales of the gdp color scheme changed between the two maps of GDP changes by country in 2019 and 2020 as the difference between countries’ gdp changes got smaller in 2020. Yet, overall the maps show that most countries had a positive development in 2019 and a negative development in 2020.

Figure 4 - Gross domestic product (GDP) changes in Europe in 2020

*Figure 6* displays the same GDP changes as *Figure 5*. However, as with *Figure 3*, I used the coord\_sf() function to zoom in on Europe to more clearly show the inter-country differences. Here, it is apparent that the countries that was most severely affected economically included Spain, Italy, France, Greece and UK.

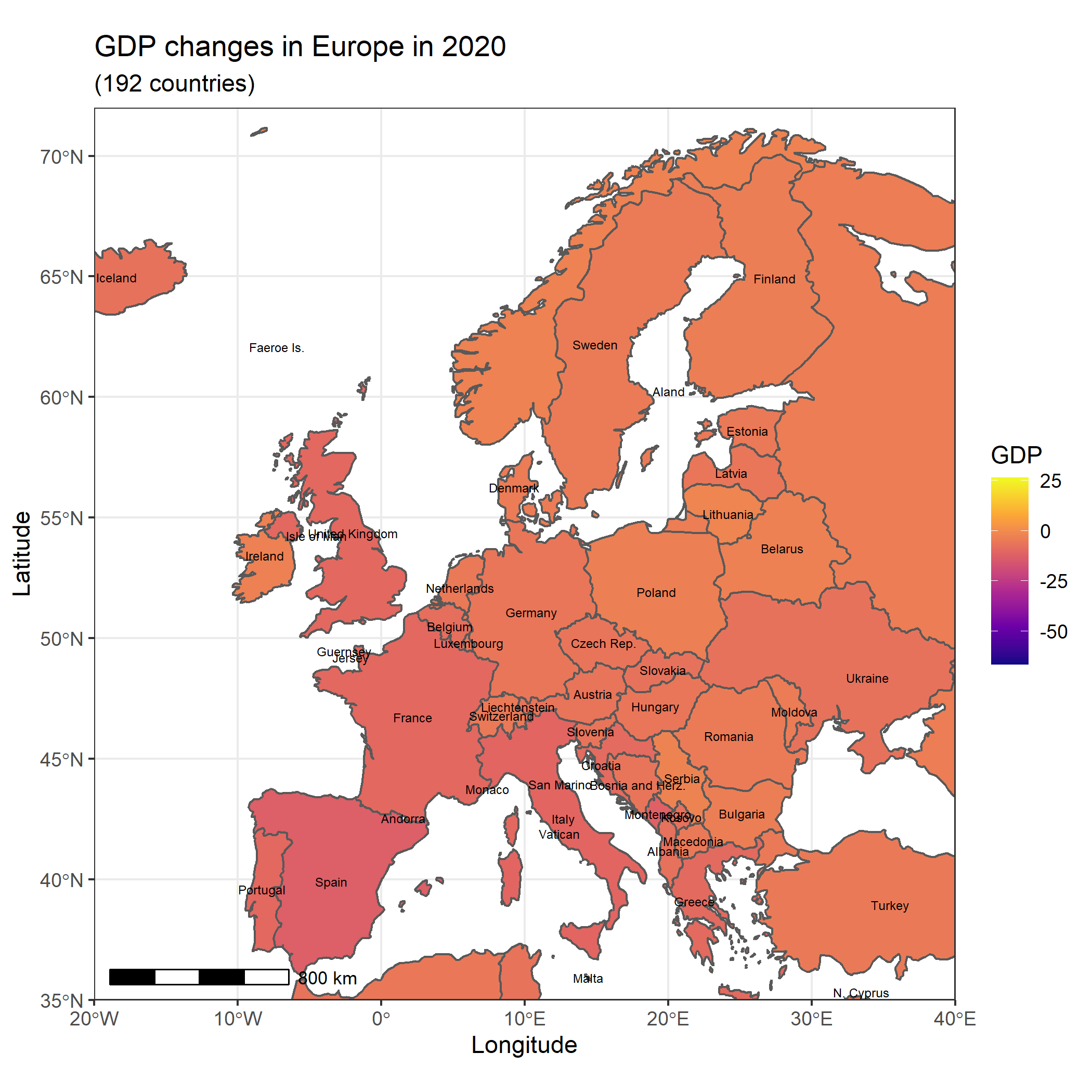


Figure 6 - Map of Europe showing the gross domestic product (GDP) changes by country in 2020.

So, it is save to say that the world economy was negatively affected in 2020. However, a correlation test is needed to be able to have causal link between the Covid-19 crisis and the GDP changes. This was performed using a Pearson’s correlation test and the results suggest that there is a moderate negative correlation between the Covid-19 crisis and the world economy; .

Difficulties in transcribing the code: Biggest challenge was to make the maps of Europe with the country names printed within each country.

Conclusion on results

**Critical Evaluation**

Reliability of sources

Evaluation of digital tools

**Conclusion and Acknowledgement**

**A - Free text**

*Description of your final project plus script or software in 2000-3000 words – suggested format follows;*

***1 Introduction / Goal***

* *Introduce the motivation for developing your research project, and explain why it is important and what its cultural relevance is (please use language accessible to non-specialized but intelligent audience)*

***2 Problems and Background***

* *Give the formulations of (cultural, historical, social, technical, etc.) problems to be solved by the project and role of the different digital tools in achieving the aims of the project*
* *Introduce the background and related work in literature (cite or list relevant literature, algorithms used, other scripts and software etc.)*

***3 Software Framework***

* *Give a short overview of the overall software architecture, dependencies and prerequisites if relevant, including the operating system of your machine, main software packages and their versions, and any cloud-based solutions necessary for a successful reproduction of the project results.*

***4 Data Acquisition and Processing***

* *List and cite all sources of data used in this paper*
* *Details of data extraction, filtering and preparation. Attach or link to processing scripts where relevant.*

***5 Implementation and Empirical Results***

* *Here you can provide either 1) a write up which presents in broad strokes the main elements of your digital workflow, pointing to specifics in the script, and highlighting decision-making bottlenecks and/or functions/tricks you found useful and wish to promote or credit. Alternatively, you can provide 2) the full annotated script, such as rmarkdown, demonstrating and documenting all major functions and decisions behind them.*
* *Empirical Results (product of your script ~slides, map, outline, timeline…)*

***6 Critical evaluation***

* *Critical assessment of the data sources (representativeness, reliability, etc.)*
* *Evaluation of the digital tools, the learning process, time on task, vis-à-vis the final product.*
* *For 'technical pipeline' projects: Provide a comparison with other state-of-the-art software if any exists for the same task* ***(****kindly cite relevant work, scripts, etc.****)***

***7 Conclusions***

* *Set out the conclusion of the project, summarize the achieved goals and highlight the most important lessons learnt while working on the project.*

***\*Acknowledgements***

* *Optionally thank people and institutes you need to acknowledge*

***References***

* *At least 5 are required, both domain-based literature as well as references to digital tutorials or internet resources consulted.*

**B- Required Metadata**

*Please fill in the right column column with the correct information about your digital resources, and leave the left columns as they are*

*Table 1 – Software metadata*

|  |  |  |
| --- | --- | --- |
| **Nr** | **Software metadata description** | ***Please fill in this column*** |
| S1 | Current software version | *R 4.0.0, RStudio 1.3.959.* |
| S2 | Permanent link to executables of this version | [*https://github.com/Digital-Methods-HASS/au616353\_martinez\_mie/tree/master/my\_final\_project*](https://github.com/Digital-Methods-HASS/au616353_martinez_mie/tree/master/my_final_project) |
| S3 | Legal Software License | *List one of the approved licenses, e.g. Creative Commons 4.0; see Week 6 lecture recordings for more* |
| S4 | Computing platform / Operating System | *Microsoft Windows 10* |
| S5 | Installation requirements & dependencies for software not used in class |  |
| S6 | If available Link to software documentation for special software | *Example http://mozart.github.io/documentation/* |
| S6 | Support email for questions | *miearnaumartinez@gmail.com* |

*Table 2 – Data metadata (use the template below or create your own metadata table)*

|  |  |  |
| --- | --- | --- |
| **Nr** | **Metadata description** | ***Please fill in this column*** |
| D1 | Link for data of corona-related deaths | [*https://www.statista.com/statistics/1093256/novel-coronavirus-2019ncov-deaths-worldwide-by-country/*](https://www.statista.com/statistics/1093256/novel-coronavirus-2019ncov-deaths-worldwide-by-country/) |
| D2 | Link to Excel file of gdp changes in 2019 and 2020 | <https://www.imf.org/external/datamapper/NGDP_RPCH@WEO/OEMDC/ADVEC/WEOWORLD> |

**References:**

Wu, Y.-C., Chen, C.-S., & Chan, Y.-J. (2020). The outbreak of COVID-19: An overview. Journal of the Chinese Medical Association, 83(3), 217–220. <https://doi.org/10.1097/JCMA.0000000000000270>

Mas-Coma, S., Jones, M. K., & Marty, A. M. (2020). COVID-19 and globalization. *One Health*, *9*, 100132. <https://doi.org/10.1016/j.onehlt.2020.100132>

Moreno, M., and Basille, M. (2018). Drawing beautiful maps programmatically with R, sf and ggplot2*—Part 1: Basics*. Retrieved December 31, 2020, from <https://www.r-spatial.org/r/2018/10/25/ggplot2-sf.html>

Statista (2020).Trustworthy data from reliable resources. *Statista.* Retrieved January 3, 2021, from <https://www.statista.com/aboutus/trust>

**Software:**

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

**Packages:**

Bivand, R., and Rundel, C. (2020). rgeos: Interface to Geometry Engine - Open Source ('GEOS'). R package version 0.5-5. <https://CRAN.R-project.org/package=rgeos>

Dunnington, D. (2020). ggspatial: Spatial Data Framework for ggplot2. R package version 1.1.4. <https://CRAN.R-project.org/package=ggspatial>

Feinerer, I., and Hornik, K. (2020). tm: Text Mining Package. R package version 0.7-8. <https://CRAN.R-project.org/package=tm>

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, <https://doi.org/10.32614/RJ-2018-009>

Rinker, T. W. and Kurkiewicz, D. (2017). pacman: Package Management for R. version 0.5.0. Buffalo, New York. <http://github.com/trinker/pacman>

Slowikowski, K. (2020). ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'. R package version 0.8.2. https://CRAN.R-project.org/package=ggrepel

South, A. (2017). rnaturalearth: World Map Data from Natural Earth. R package version 0.1.0. <https://CRAN.R-project.org/package=rnaturalearth>

South, A. (2017). rnaturalearthdata: World Vector Map Data from Natural Earth Used in 'rnaturalearth'. R package version 0.1.0. <https://CRAN.R-project.org/package=rnaturalearthdata>

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>

Wickham, H. (2019). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.3.5. <https://CRAN.R-project.org/package=rvest>

Wickham, H. (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. <https://CRAN.R-project.org/package=stringr>

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

Wickham, H., Hester, J., and Ooms, J. (2020). xml2: Parse XML. R package version 1.3.2. <https://CRAN.R-project.org/package=xml2>