ETL Project report – Group 3

Impact of tourism and economy on the “PUEBLOS MÁGICOS” program

19/10/2020

Adriana Avalos Vargas, Edwin Rosales & Jimi Jenneskens



Table of Contents

[Introduction 2](#_Toc54027746)

[1. Extract 3](#_Toc54027747)

[1.1 Magic Towns 3](#_Toc54027748)

[1.2 The gross domestic product 3](#_Toc54027749)

[1.3 Weather 4](#_Toc54027750)

[2. Transform 5](#_Toc54027751)

[2.1 Magic Towns table and State ids table 5](#_Toc54027752)

[2.2 The gross domestic product table 6](#_Toc54027753)

[2.3 Weather table 7](#_Toc54027754)

[3. Load 9](#_Toc54027755)

[3.1 Relationship diagram 9](#_Toc54027756)

[3.2 Load data into database 10](#_Toc54027757)

[4. Queries 11](#_Toc54027758)

[4.1 Magic Towns per state 11](#_Toc54027759)

[4.2 Average State GDP per Magic Town before and after entering as Magic Town 12](#_Toc54027760)

[4.3 Average weather per Magic Town 13](#_Toc54027761)

# Introduction

The “Pueblos Mágicos” program has the objective to transform Mexico, around the 2030, in a leading country in tourism activity. To achieve this, the government proposed as a strategy to recognize tourism as a key element in Mexico's economic development. Diversify tourism products and develop new markets. Promote tourism companies to be competitive nationally and internationally. Develop tourism while respecting natural, cultural and social environments, in this context the program “Pueblos Mágicos” was created.

According to the **SECTUR**, a magical town (“pueblo mágico”) is a locality that has symbolic attributes, legends, history, transcendent facts, everyday life, etc. In short, they possess magic that appears in each one of their socio cultural manifestations, and today they can be perceived as a great opportunity for tourist use. The Magical Towns Program contributes to revalue a group of populations in the country that have always been in the collective imagination of the nation as a whole and that represent fresh and different alternatives for national and foreign visitors.

**Proposal of an ETL project**

Since the main objective of this program is to convert Mexico in leading country on tourism, then it is necessary to create a unique database that captures how does this program has impacted the different localities that has access to this program.

To do so several data sources has to be consulted and aggregated.

The actual list of localities considered as Magic town.

Its contribution to the gross domestic product associated with the Tourism activity at the state and national level since the creation of the first magic town (this info is available yearly).

Also, it could be important to characterize the weather in each Magic town, in order to do a further analysis about the impact of this variable on the success of the program en each Magic Town.

# Extract

Three different sources have been used to compile the database. The first source is a dataset with the Magic Towns and the date it has entered as a magic town. The second source is a dataset with the yearly gross domestic product per state based on the influence of tourism, which is useful to determine the influence on the gross domestic product before and after entering as a magic town. The last source is a dataset with the average weather per year per state, which can be used to investigate if the weather influences the popularity of the Magic Town. The sources are explained in the following chapters.

## Magic Towns

The first source is a CSV file, which contains three columns with “State”, “Town” and “Year entered” per magic town. An example of the CSV file can be found in figure 1.

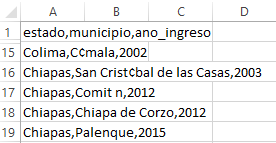


Figure : Screenshot magic towns CSV

The source can be found here: <https://datos.gob.mx/busca/dataset/localidades-que-cuentan-con-el-nombramiento-de-pueblo-magico-dggd>.

## The gross domestic product

The second source is a CSV file with the yearly gross domestic product based on tourism from 2003 till 2018, which contains the “State”, “Gross domestic product” and the “Year”. This information is only available on state level, so unfortunately it cannot be used on Magic Town level. An example of the CSV file can be found in figure 2.

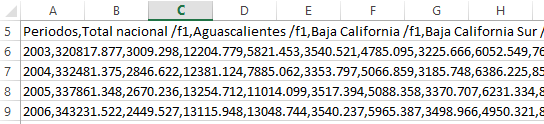


Figure : Screenshot gross domestic product CSV

The source can be found here: <https://www.inegi.org.mx/sistemas/bie/>

## Weather

The third source are twenty separated excel files per topic with tables of the yearly average rain, minimum temperature, maximum temperature and the average temperature per state. This information is only available on state level, so unfortunately it cannot be used on Magic Town level. An example of an excel file can be found in figure 3.

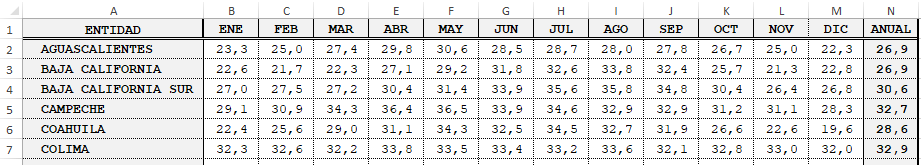


Figure : Screenshot weather PDF

The sources can be found here: <https://datos.gob.mx/busca/dataset/precipitacion>

<https://datos.gob.mx/busca/dataset/temperatura-promedio-excel>

<https://datos.gob.mx/busca/dataset/temperatura-minima-excel>

<https://datos.gob.mx/busca/dataset/temperatura-maxima-excel>

# Transform

To be able use the data for our database the data has to be cleaned and transformed, so that it can be used correctly for the purpose of the database. The three sources have been transformed into correctly formatted tables and an extra table named ‘States\_id’ has been created to be able to make a relationship between all the tables. This part is explained per table in the following chapters.

## Magic Towns table and State ids table

As shown in figure 1, the original CSV file contains spelling mistakes in the states and town, because the CSV file cannot contain special characters. Therefore, first the names of the states and towns have been corrected manually to without special characters.

Another problem is that the states are spelled differently throughout the sources and cannot have a relationship in the database via this way. Therefore, an extra table has been created named ‘States\_ids’, to be able to create a relationship between the different sources/tables. All the unique states have been stored in a dataframe and via a ‘for loop’ unique ids have been added. In figure 4 is an example of this table shown. This table has been stored as a CSV file.

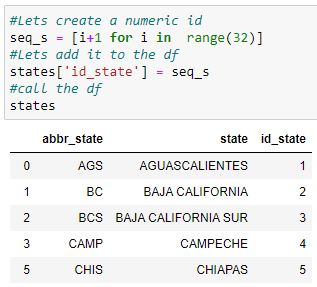
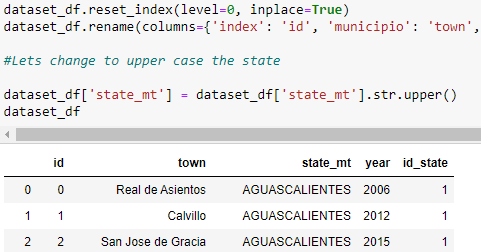


Figure 5: Magic towns table

Figure : States\_ids table

The id\_state has been added to the Magic Towns manually. The headers in the original CSV file, as shown in figure 1, are in Spanish and therefore, have been changed to English headers. Also the values in the states column have been changed to capital letters only, because it is like this in the State\_ids table as well. Also the index has been reset and the new column has been named id, so it creates unique ids per town. In figure 5 is an example of the Magic Towns table shown. This table has been stored as a CSV file.

## The gross domestic product table

The dataframe of the original CSV is not ready to be used yet. This one has to be modified in order to be able to use it as a table for a database. In figure 6 is an example shown of the dataframe of the original dataframe.

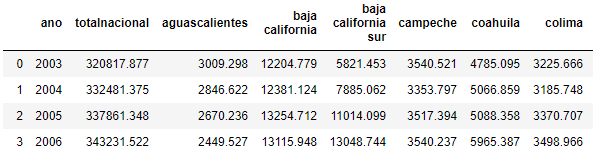


Figure 6: Dataframe of the original gross domestic product CSV

First index has been set to ‘ano’ and then the axes of the index and the columns have been swapped, so that the states are on the rows and the index and the years in the column. Then the states have been sorted on values, so it has the same order as the State\_ids table and an extra column named ‘id\_state’ has been added with counting from id 1 to 32, so that it can have a relationship with the other tables in the database. In figure 7 is a visible of the new dataframe visible.

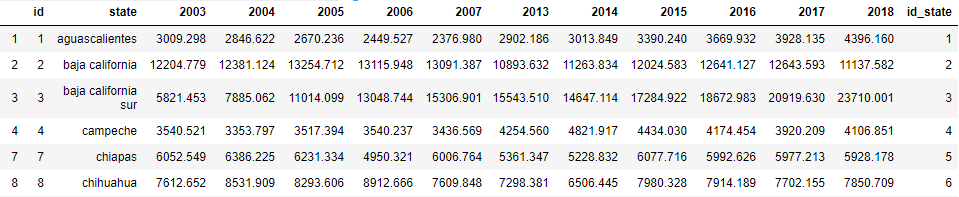


Figure 7: Dataframe with changed axes and id\_state column

Then the values in the column state have been changed to capital letters only, as in the other tables. A list of year 2003 till 2018 has been created, in order to create a new dataframe with a ‘for loop’. A ‘for loop’ has been created to create a new dataframe based on 'id\_state','states', 'tourism\_gdp' and 'year'.

The ‘for loop’ can be found in figure 8. The new and final gross domestic product table can be found in figure 9 on the next page. This table has been stored as a CSV file.

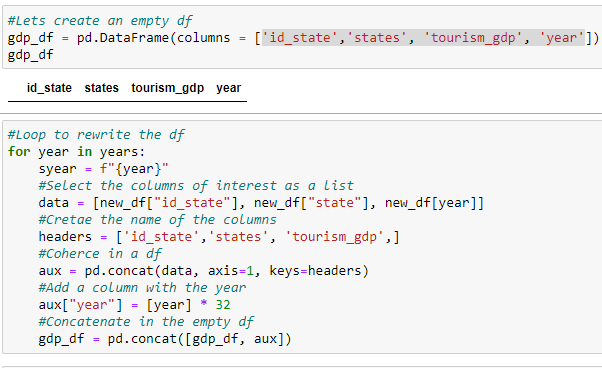


Figure 8: The ‘for loop’ to create the new dataframe

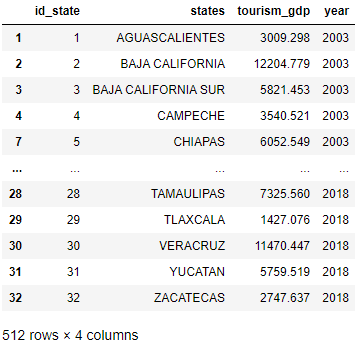


Figure 9: The gross domestic product table

## Weather table

For the topics yearly rain, minimum temperature, maximum temperature and average temperature there where 20 separated yearly excel files per topic. These files have been concatenated into a CSV file per topic via a ‘for loop’ and the id\_state column has been added. An example of this loop can be found in figure 10 on the next page. An example of the result, which has been stored as a CSV file, can be found in figure 11 on the next page.

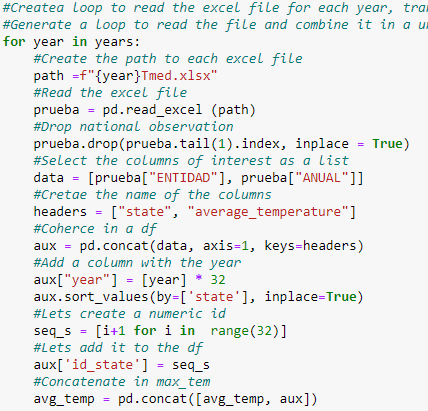
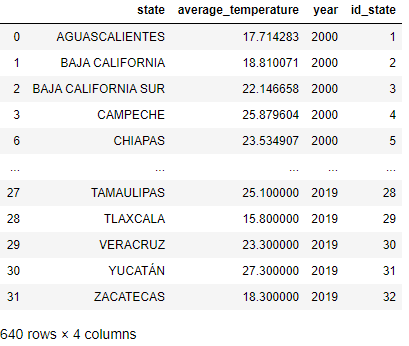


Figure 11: Dataframe of the result, which is stored as a CSV file

Figure 10: For loop to concatenate the excel files

These four separated CSV files have been loaded into a jupyter notebook and have been merged by using pandas. An example of the merge is visible in figure 12. The final dataframe/table for weather is visible in figure 13. This table has been stored as a CSV file.

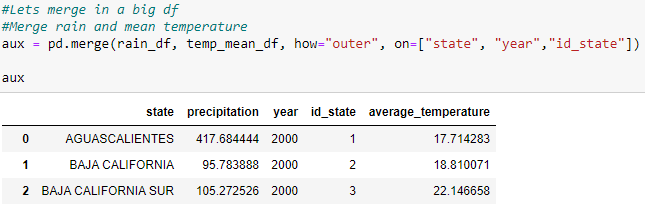


Figure 12: Example of the merge of the separated dataframes

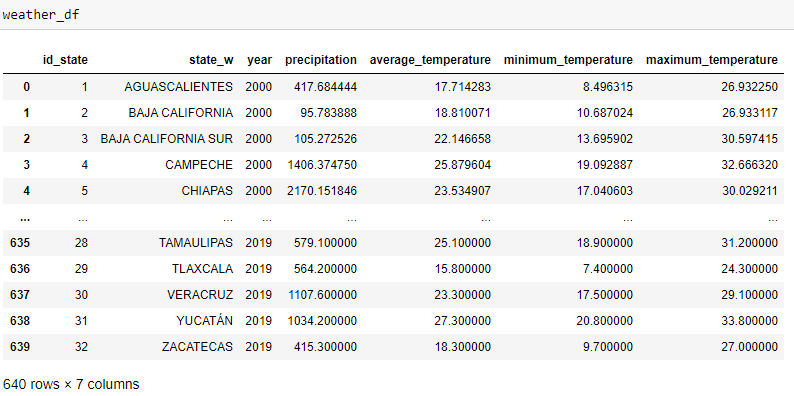


Figure 13: Final weather table

# Load

In order to visualize how the data is going to be related a data base diagram is made by using QuickDB. To load all the data retrieved a SQL database is built by using PostgreSQL. This will be explained in this chapter.

## Relationship diagram

After cleaning the databases a relationship diagram is created by using QuickD. How all the tables and columns are related is visible in the diagram of figure 14.

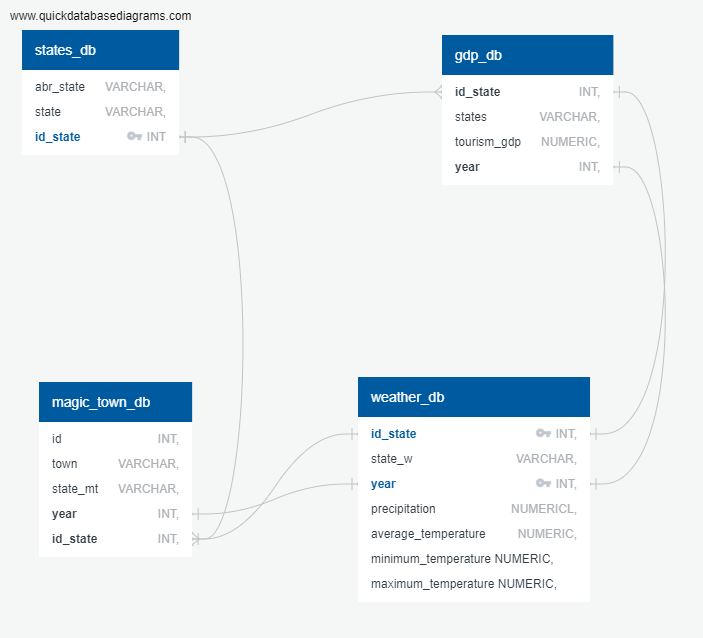


Figure 14: Relationship diagram of the database

It has to be noted that this tool does not allow to create composed primary keys. Therefore, the PostgreSQL has to be done as usual, and not through the import tool in QuickDB.

## Load data into database

Once relationship diagram has been visualized a PostgreSQL database has been created. This had to be done manually, because composed primary keys have been used. In figure 15 is a part of this code shown.

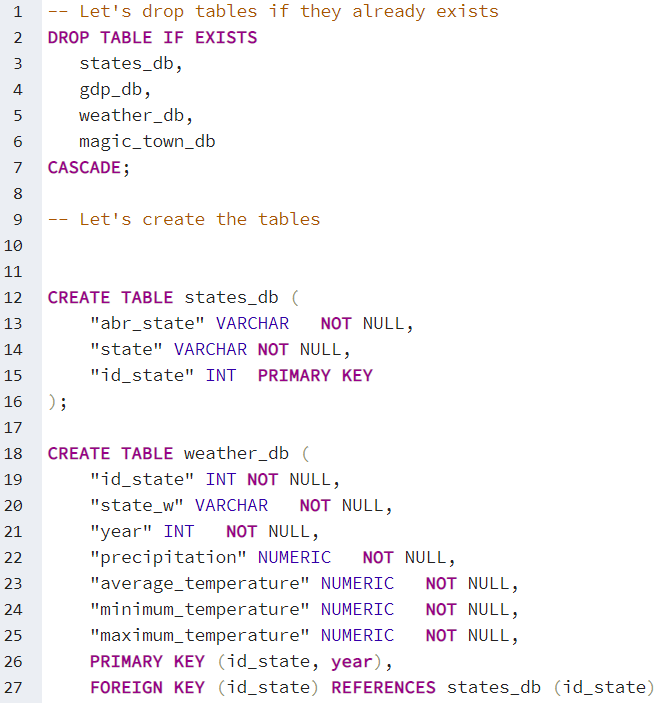
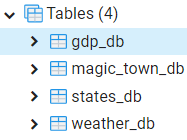


Figure 16: Tables in SQL database

Figure 15: Creating tables into the SQL database

After running this code all four tables have been created. In figure 16 an overview of the names of the tables in the SQL database can be found.

After creating the tables the data has to be loaded into the database. This has been done by importing the generated CSV files of the Transform part of this project.

# Queries

Three queries have been created to analyze the data in the created database. First a query to identify the magic towns per state has been created. The second query identifies Average State GDP per Magic Town before and after entering as Magic Town and the last query identifies Average weather per Magic Town. First the tables of the SQL database have been loaded into dataframes through queries. A screenshot of this step can be found in figure 16. The queries are explained in the following chapters.

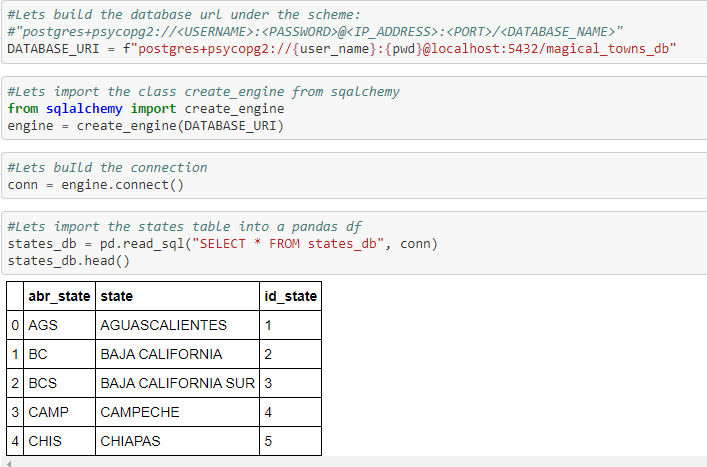


Figure 16: Loading the tables from database into dataframes in notebook

## Magic Towns per state

An easy query has been made to identify the amount of magic towns per state. This can be found in figure 17.

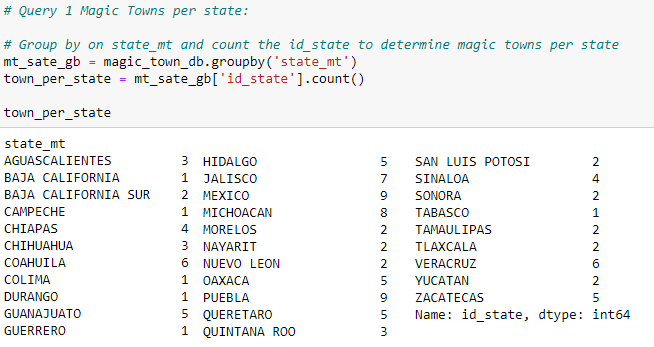


Figure 17: Magic Towns per state

## Average State GDP per Magic Town before and after entering as Magic Town

The second query compares the average yearly state GDP per Magic Town before and entering as a Magic Town. Unfortunately, there is only data available on state level and therefore, the data on state level has been used to compare the differences.

A ‘for loop’ has been created to calculate the yearly average of the GDP per state before and after entering as a Magic Town. For every row (= every magic town) in the magic\_town table the town, year of entering and id\_state is stored in a variable and has been used to calculate the yearly average of the gdp of before and after as entering as a Magical Town. The ‘for loop’ and the results are visible in figure 18.

It is visible in the results that for almost every Magic Town the yearly average state GDP has been higher after entering as a Magical Town. As it is on state level, it is difficult to confirm that the magical town directly contributes on this, but as there is a positive difference for almost every town, we can assume that the Magical Town program does contribute to its economy.

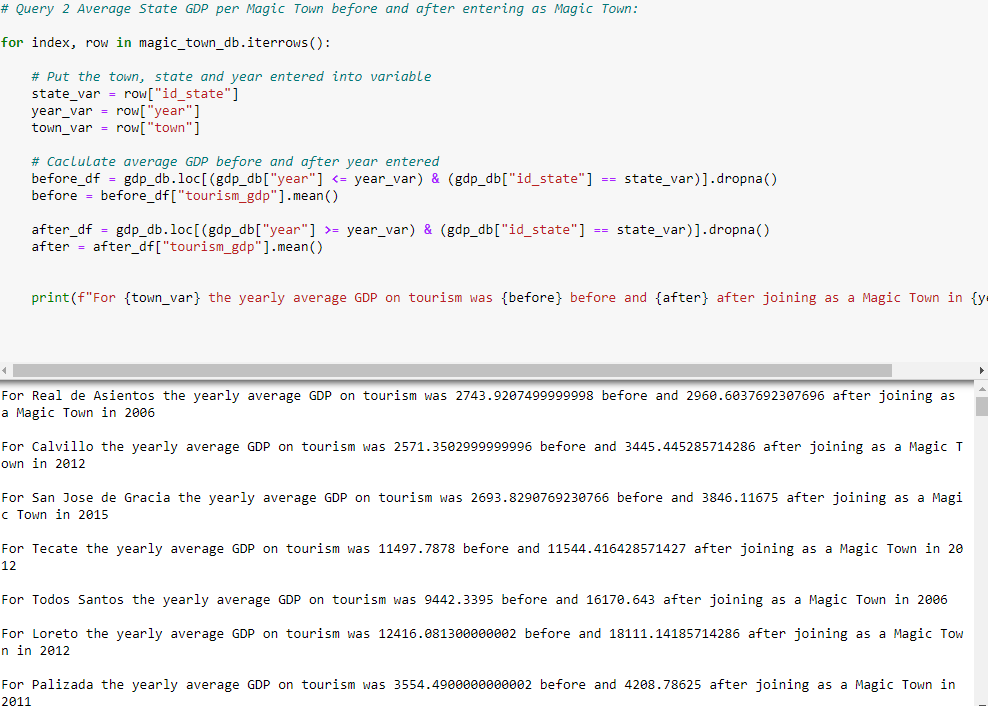
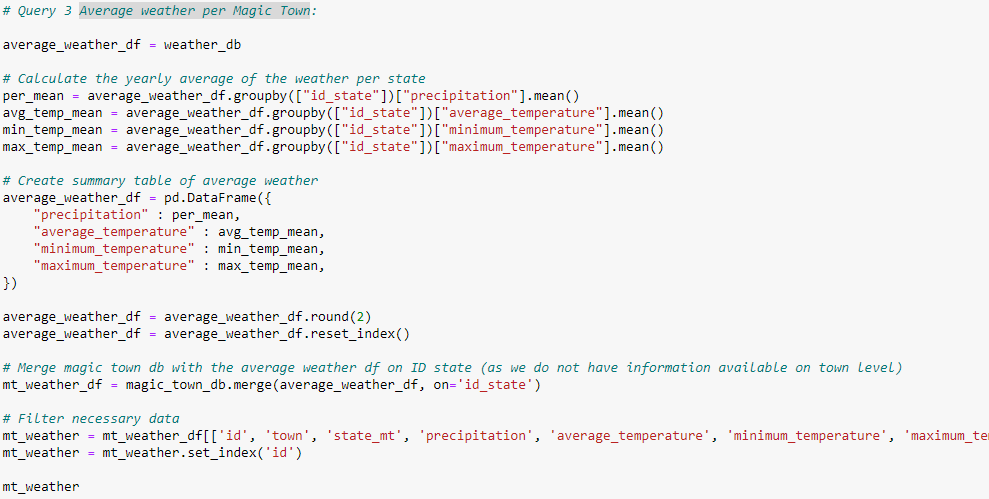


Figure 18: For loop to determine difference before and after joining and the results per Magic Town

## Average weather per Magic Town

An average weather per magic town dataframe has been created, which can be used to investigate if the weather might influence the popularity of the Magic Town. Unfortunately, the data was only available on state level, so this had to be executed on state level per Magic Town.

First the data of the Weather table has been grouped by on ‘id\_state’ and the yearly averages of the precipitation, average temperature, minimum temperature and maximum temperature have been stored in new variables. These variables have been stored in a new dataframe. The index has been reset, so the ‘id\_state’ becomes a column again. Then the Magic Town table has been merged with the average weather dataframe on id\_state, to get an overview of the Magic Towns and the local weather averages per year. The code and the generated dataframe is visible in figure 19.



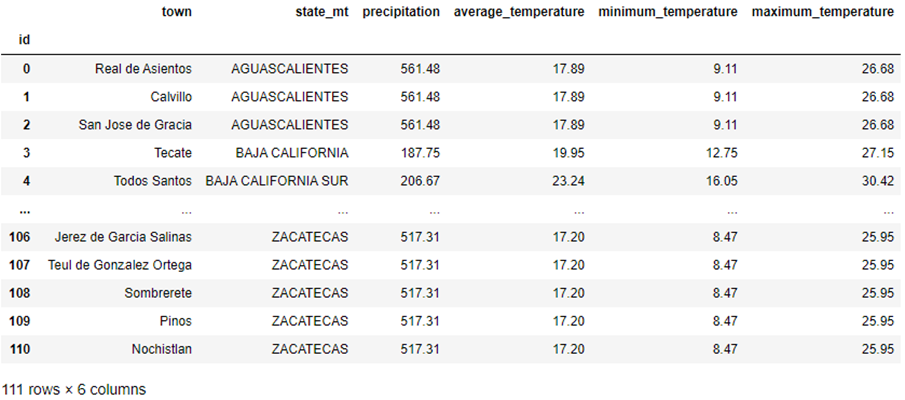


Figure 19: Construction of weather per magic town dataframe and the result