UNIVERSITÀ DI PISA

Laboratory for Data Science

Project

Elisa Pashku - 628386
Xhenis Jaku - 639180

Department of Informatics
Data Science and Business Informatics
2022-2023

Contents

1

Project Part I

The aim of this first part of the project is to create and populate a database starting from the following `.cvs` files:

- `"answerdatacorrect.csv"` contains a table with data about answers given by students to various multiple-choice questions and data related to the questions, the students, and the subject of the questions
- `"subject metadata.csv"` contains information about the subject of each question. The subject is given by a list of integers in the main data that can be used to index the `"answerdatacorrect.csv"` to retrieve the topic of the question.

.

1.1. Build the datawarehouse

1.1.1. Assignment 0

The first assignment is to design and create the database schema using SQL Server Management Studio. The schema is composed of a fact table (Answers) and five dimensions (as shown in Figure 1.1). Relationships between tables were built using `Primary keys` (we imposed a **NOT NULL** constraint) and `Foreign Keys` (we allowed **NULL** values).

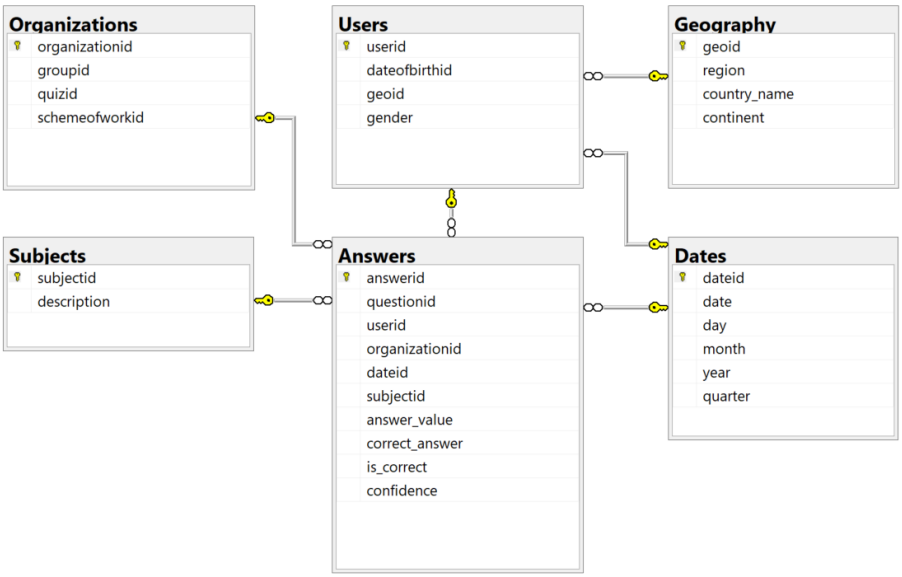


Figure 1.1: Database schema

1.1.2. Assignment 1

The second task consists in writing a Python program that splits the content of the given `.cvs` files into six separate tables: "Answers", "Organization", "Date", "Subject", "User", and "Geography". In the `main.py`, only the `csv` library is used. The program has one main function and other functions responsible for transforming and extracting the right data and writing them to the respective files.

There was no missing data apart from the two "ParentId" in `subject_metadata.csv` corresponding to the main topics of the subjects. The data read from the given `.csv` files, was assigned to lists where it was possible to enumerate in order to access the index and the respective values. In the following points, the workflow for producing the requested tables is described.

- An incrementing primary key (`organization_id`) was generated to produce the dimension table **Organizations**. Afterward, the columns "groupid", "quizid", "schemeofworkid" were extracted from the `answerdatacorrect.csv`.
- The **Dates** table contains the following attributes: "dateid", "date", "day", "month", "year",and "quarter". It was specified that it had to accommodate both dates of birth of users and dates of answers. In addition, the "DateAnswered" column from the `answerdatacorrect.csv` file was split and the first element, with index 0, was used (discarding hours and minutes). The index in this file is in the YYYYMMDD format. The variable quarter which has only 4 unique values $(1,2,3,4)$, was computed using the formula: (month +2)//3.
- To create the **Subjects** table, the files `answerdatacorrect.csv` and `subject_metadata.csv` were read. After stripping the "],[" and "−" from "subjectid", we used it as a guide to sort the "description" depending on the subject level order. Some "subjectid"-s had more than four levels (ParentId, Level1, Level2, Level3). The subject description has been kept as detailed as possible, therefore, keeping the same levels one after the other; for example Subject with ID 31 has the topic "Maths, Number, Decimals, Negative Numbers, Basic Arithmetic, Place Value, Ordering Negative Numbers, Ordering Decimals". In addition, we decided to generate an incrementing "subjectid".
- In order to generate the **Geography** table, the "Region" and "CountryCode" from the `answerdatacorrect.csv` have been used and then a data integration using countries.csv, obtained from the source `https://github.com /lukes/ISO-3166-Countries-with-Regional-Codes/blob/master/all/all.csv` was performed. This new dataset, provided additional information about the continent and the full country name. Before integrating the data, some preprocessing was required: converting "alpha-2" code to lowercase and substituting "GB" with "UK". The given "RegionId" was not very precise as the same id was used to identify all cities part of the same country. Consequently, an incrementing id was generated for each city.
- The **Users** table contains the already given "userid", "dateofbirthid", "geoid" and "gender". For the gender, we decided to alphabetically assign 1 with 'F' (for female) and 2 with 'M' (for male).
- In addition to the attributes already present in the `answerdatacorrect.csv`, **Answers** table requires the "is_correct" attribute. This information is obtained through a comparison of values of the columns "CorrectAnswer" with "AnswerValue"; if the values are equal, the answer is Correct (else Incorrect). The columns of Answers table are "answerid", "questionid", "userid", "organizationid", "dateid", "subjectid", "answer_value", "correct_answer", "is_correct" and "confidence".

1.1.3. Assignment 2

In this task the goal is to populate the database Group_5_DB with the data prepared during the previous steps. In order to do so the Python file "`loading_data.py`" was created. The file was connected to the Group_5_DB database through the pyodbc library by creating a cursor.

The `.csv` files were read using the `csv.reader()` method. A dictionary with keys being the name of tables and values being the reader of that table was used. An **Insert Into** query was created for each element of the dictionary, dynamically based on the number of elements present in the header of the corresponding file, executed with `cursor.execute()`. In order to speed up the process of inserting lines into the remote server running in Microsoft SQL server management the command `cursor.fast_executemany=True` has been applied. At the end of the code, files, cursor, and connection were closed.

In this process of populating the database, trial and errors were made. Therefore, a file for deleting the loaded data or specific tables was created too (`truncate.py`).

2

Project Part II

The second part of the project is about answering some business questions using the data warehouse created in Part I using only the SSIS tool in Visual Studio, with computation on the client side (i.e. without writing queries). To skip the redundancies in the report, it is important to note that for all the tasks, the initial and final steps are the **OLE DB Source** node, the **Lookup** node(s) (when needed) and the **File Flat Destination** node. Before getting into the solutions, it is worthwhile noting that every time a lookup and the conditional split node were used, the results verifying the condition were kept. Additionally, only the relevant attributes were kept in each step in order to reduce the computational cost.

The results were exported with the generic name `Part2_assignment(n)_output.csv`, where "n" refers to the ordinal number of the task. The solutions can be found in the `Group5_SSIS.zip` folder.

2.1. Exercises

2.1.1. Assignment 0: For every year, the users ordered by total number of answers

To obtain the results, the table Answers was accessed, getting "dateid", "userid" and "answerid" columns. Then the "year" information was retrieved from looking up on Dates table using the "dateid". The **Aggregate** node, allows to group by "year" and "userid" and count the number of answers (answerid) per user.

The flow of nodes required to complete such task is shown in Figure 2.1 .

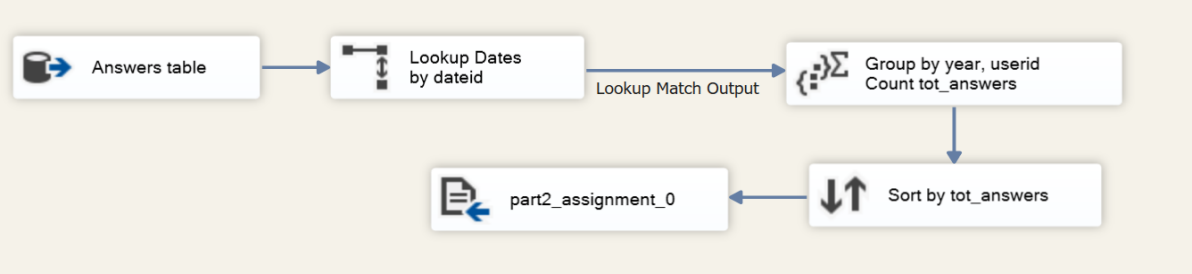


Figure 2.1: Flow of Assignment 0

2.1.2. Assignment 1: For every subject, the correctness confidence index is defined as the ratio between correct answers and wrong answers multiplied by the average confidence. Provide such index for every subject, considering only German and English students

In order to answer this question the information from the tables Answers, Users and Geography was used. The main source of data was Answers from which we got "answerid", "userid", "is_correct" and "confidence" columns. After the main fetch, "geoid" was retrieved by means of a **Lookup** node from Users, with the intention of looking up Geography table to get "country_name".

Once all columns were collected, the data was split (using **Conditional Split** node) in order to continue only with German and English students. Then, **Multicast** node was used to perform a **Group by** "subjectid" and to calculate the average confidence in one branch; in the other branch the data was split again into correct answers and incorrect answers. In these inner branches, a **Group by** "subjectid" was performed and the correct and incorrect answers were counted respectively. These branches were merged by means of a **Merge join** node on "subjectid", after being sorted on "subjectid". A final merge was performed resulting in having average confidence, number of correct answers and number of correct answers per subject. Using

the **Derivative Column** node, "correct answers" were divided by "incorrect answers" and multiplied with the "average confidence".

The division and the required Correctness Confidence Index were converted to numeric data type. Figure 3.7 shows the flow defined to solve this business question.

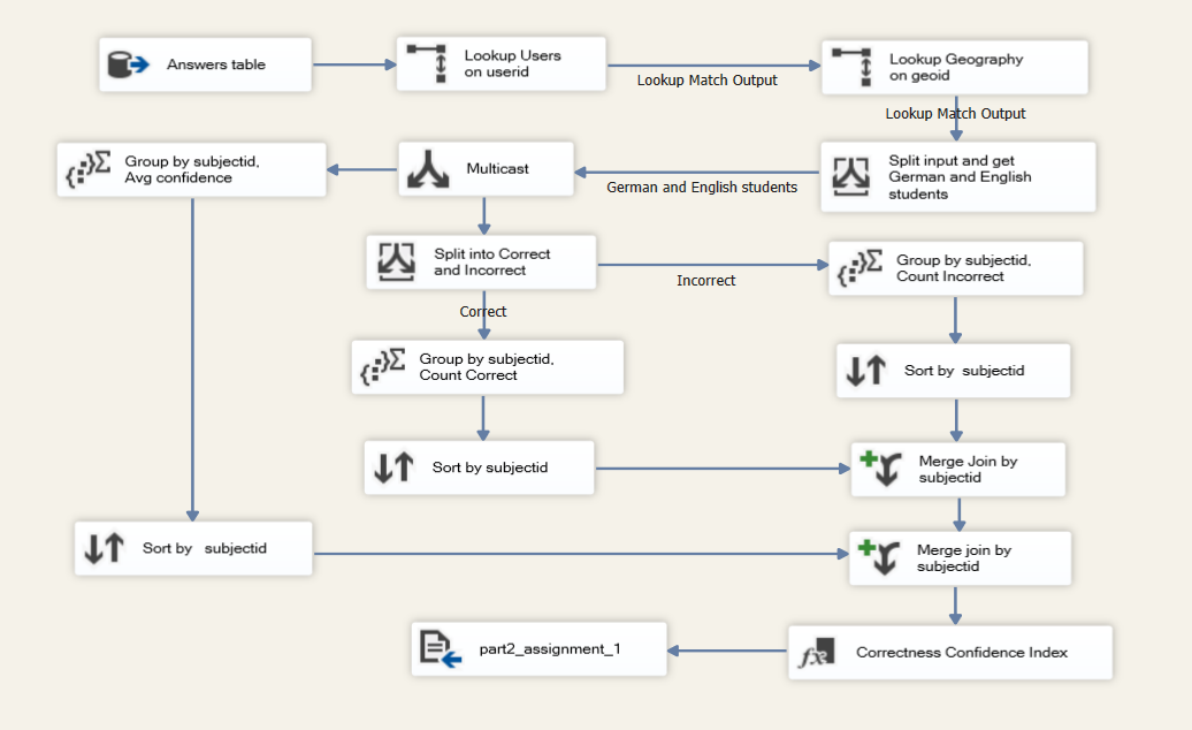


Figure 2.2: Flow of Assignment 1

2.1.3. Assignment 2 : For each region, the percentage of correct answers with respect to the country of origin.

Even for this assignment, firstly the Answers table was accessed getting "userid" and "is_correct". Through "userid", **Lookup** was applied on Users table to retrieve the "geoid". Then, **Lookup** was used again in Geography table to access "country_name" and "region". Exploiting a **Conditional Split** node, the correct answers were accessed and further **Group by** "region" and count the "correct_answers" per region were performed. Next, a **Multicast** node was used to group by "country_name" and "region" and sum the "correct_answers" on one branch and on the other one to do the same operations without grouping on region.

In order to merge the branches, data was sorted by "country_name". To calculate the percentage, first the correct answers per region and per country were converted to numeric type through a **Data Conversion Transformation** node. The Percentage of Correct Answers is a product of a **Derivative Columns** and calculated as correct answers per region divided by correct answers per country multiplied by 100. The flow of nodes to solve this assignment is shown below Figure 2.3 .

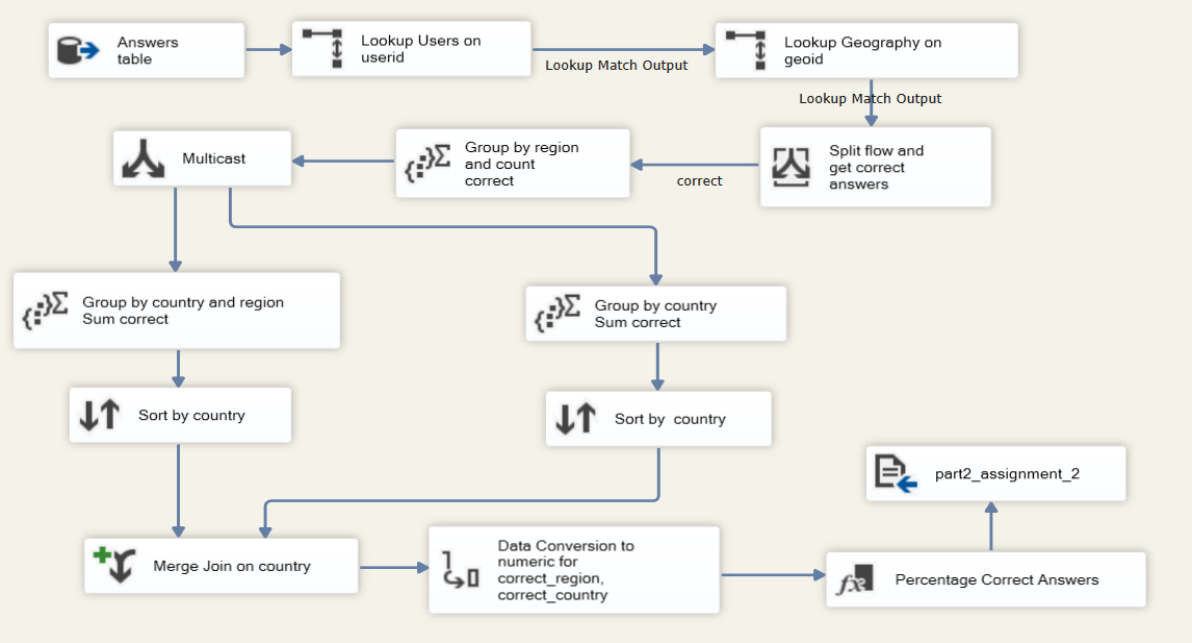


Figure 2.3: Flow of Assignment 2

3

Project Part III

In Part 3 of the project is required to answer some business questions on a datacube using MultiDimensional eXpressions (MDX) in SQL Management Studio. In this phase the SSAS package is used.

3.1. Building the Data-cube (Assignment 0)

After establishing a connection to the the database "Group_5 DB" with Visual Studio, a view of such schema was created, containing the following tables: Organizations, Users, Geography, Subjects, Answers and Dates. Starting from such view, an OLAP cube was later generated by selecting as measures **Answers Count** (the number of answers), Confidence (the sum), **Userid Distinct Count** (count of distinct participants), **Correct Ans** (count of correct answers), **Incorrect Ans** (count of incorrect answers) from Answers. Then dimensions "**Dates**", "**Users**", "**Subjects**" and "**Organizations**" were created. Also two distinct hierarchies were built: 'Geography' in the Users dimension and 'Time' in the Dates dimension. The hierarchy "Time", has the following relations: Year → The Quarter → The Month → Dateid. In order to guarantee the order between years, quarters and months, new columns were created "the_month" which contains the name of the months from 'January' to 'December' (instead of the number) and "the_quarter" in which a "Q" was added before the quarter digit. The new columns substituted respectively the attributes "month" and "quarter" which were used as a sorting key in the 'Advanced Options' menu for these new attributes.

Figure 3.1: Time hierarchy

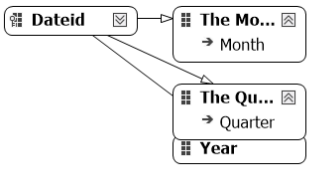


Figure 3.2: Attribute relationships in Dates dimension

The hierarchy "Geography" was created following the relation Continent → Country Name → Region → Geoid.

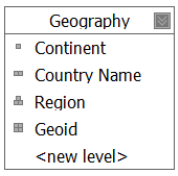


Figure 3.3: Geography hierarchy

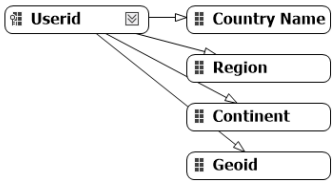


Figure 3.4: Attribute relationships in Users dimension

Once the dimensions with the hierarchies were defined, the data cube "Group 5 DB" was built, therefore it is possible to proceed with the solution of the business queries.

3.2. MDX Queries

3.2.1. Assignment 1: Show the total correct answers for each country and the grand total with respect to the continent

Within the "GT_continent" member, the total number of correct answers for each continent was calculated, starting from the current country in the row , via "currentmember" and accessing the parent via ".parent". In the results, each row shows the Country Name, Correct Answers per Country and Grand Total per Continent.

```
WITH MEMBER GT_continent AS ([Users].[Geography].currentmember.parent,[Measures].[Correct Ans])

select {[Measures].[Correct Ans], GT_continent} on columns,

nonempty([Users].[Geography].[Country Name]) on rows
from [GROUP 5 DB]
```

Figure 3.5: MDX Query 1

3.2.2. Assignment 2 : Show the total confidence for each year and the running yearly for European students

First of all, the run_con represents the running confidence which sums the confidence of that year with the confidence from the Time hierarchy year previous member (preceding year). The results are just 2 rows representing each year (2019, 2020), the sum of the confidence and the running confidence with the restriction on the Continent Europe, for the European players.

```
with member run_conf as

[Measures].[Confidence] + ([Dates].[Year].prevmember,run_conf)
select {[Measures].[Confidence],run_conf} on columns,
nonempty(([Dates].[Year].[Year])) on rows
from [Group 5 DB]
where [Users].[Continent].&[Europe];
```

Figure 3.6: MDX Query 2

3.2.3. Assignment 3: Show the ratio between the total correct answers of each year w.r.t the previous year

The member **ratio** is created, by first of all, defining an **iff** function for setting the null value where there is no previous year. This ratio is then created by dividing the count of correct answers over the count of correct answers of the previous year. The results are just 2 rows representing each year with the hierarchy Time (2019, 2020) and the ratio of correct answers.

```
with member ratio as
iif(([Dates].[Time].prevmember, [Measures].[Correct Ans]) = 0, null,
[Measures].[Correct Ans] / ([Dates].[Time].prevmember, [Measures].[Correct Ans]))
select {[Measures].[Correct Ans], ratio} on columns,
nonempty([Dates].[Time].[Year]) on rows
from [Group 5 DB];
```

Figure 3.7: MDX Query 3

3.3. Dashboards

3.3.1. Assignment 4

In this task it is required to create a dashboard that shows the geographical distribution of correct answers and incorrect answers. For this purpose the software **Power BI** was used. In order to retrieve the data from the cube "Group 5 DB" a connection with the server was established on the panel `get data` and option `Analysis Services`.

The plots were made using the "map"(shown in Figure 3.8 and Figure 3.9) and the "clustered column chart" visualizations (shown Figure 3.9)

In the map Figure 3.8 the measure selected is `Correct Ans` as a Bubble size, the hierarchy `Geography` is selected as location parameter with Continent- Country name. The same steps are applied also for the map Figure 3.9 but in this case `Incorrect Ans` is selected as a measure.

Figure 3.8: Map showing Correct answers by Country

Figure 3.9: Map showing Incorrect answers by Country

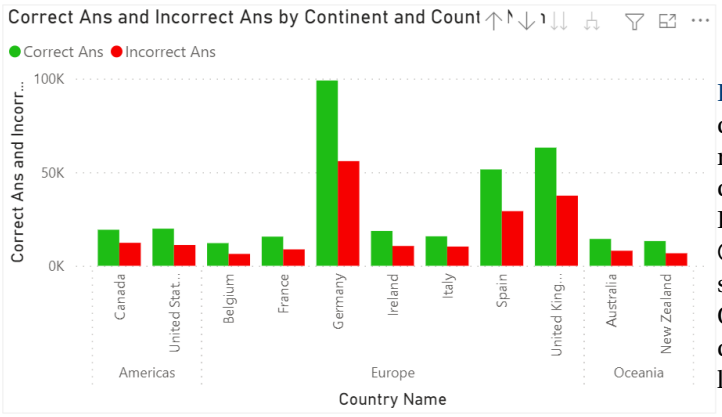


Figure 3.10

Figure 3.9 shows the clustered column chart representing the distribution of correct and as incorrect by continent and country.
In order to make this plot the measures `Correct Ans` and `Incorrect Ans` were selected and put on the y-axis of the plot. On the x-axis the hierarchy geography including continent-country name- was selected.

From these plots is clear that most of the multiple choice tests were made in Europe and Germany is the country where students performed best.

3.3.2. Assignment 5

After analyzing the geographical distribution of correct and incorrect answers by country, we thought it would be interesting to go deeper in this analysis and check the distribution of correct answers comparing by gender.

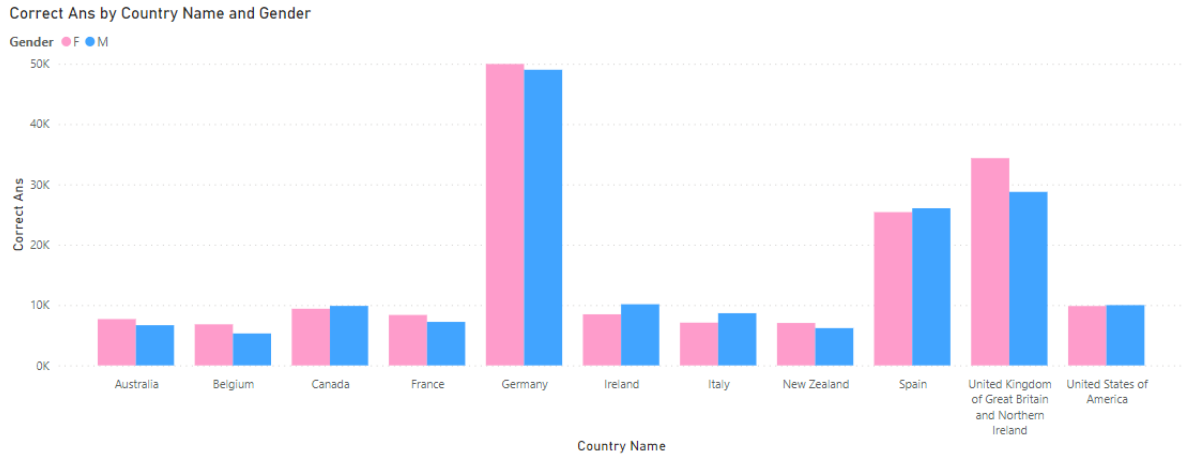


Figure 3.11: Clustered column chart showing Correct answers by Country

We can notice that the difference between correct answers by gender is minimal. In 6 countries out of 11 female students performed better than male students.

Furthermore, we wanted to analyze the correlation between confidence and number of correct answers by gender. From the scatter plot (Figure 3.12) we can see that there is a positive relationship between confidence and correct answers but again not much difference between students of different gender.

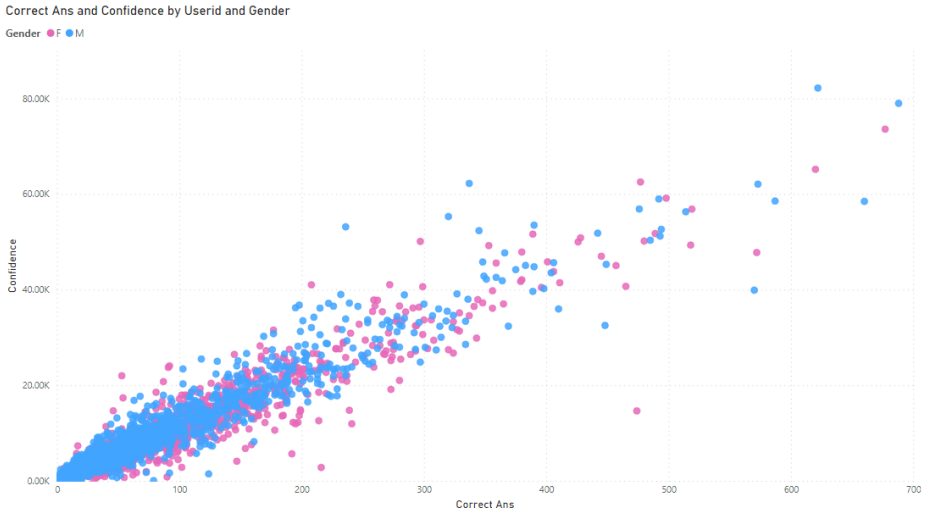
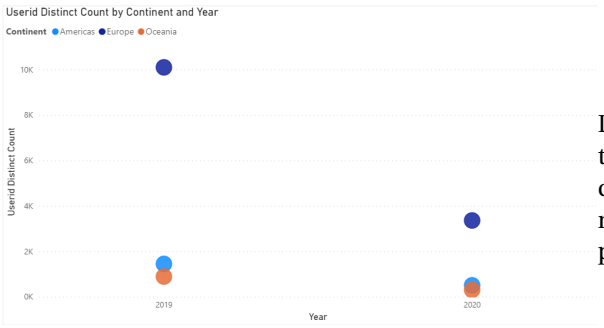


Figure 3.12: Correlation between confidence and correct answers by gender



Lastly, we analysed the number of distinct users over the years for Continent (Figure 3.13). We noticed a decrease of participants in year 2020. However, Europe remained the continent with the highest number of participants (specifically from Germany).

Figure 3.13: Number of distinct users over years per continent