



Master's Degree in Data Science and Management

Chair: Big Data and Smart Data Analytics

Analytics on Luiss Teaching Staff

Irene Finocchi

SUPERVISOR

Giuseppe F. Italiano

CO-SUPERVISOR

Anastasia Picchia

CANDIDATE

Academic Year 2022/2023

Table of contents

Introduction.....	3
1. Dataset Analysis.....	6
1.1 The first data source: Website “Cerca Università”	6
1.1.1 Introducing the dataset.....	7
1.1.2 Exploring the dataset with Python.....	8
1.2 The second data source: Docenti Luiss.....	13
1.2.1 Introducing Docenti Luiss.....	13
1.2.2 Exploring Docenti Luiss with SQL.....	14
2. Retrieving data from Google Scholar	16
2.1 Short description of Google Scholar.....	16
2.2 Data extraction from Websites	17
2.2.1 Step 1. Preliminary cleaning.....	17
2.2.2 Step 2. Profiles retrieve.....	20
2.2.3 Step 3. Cleaning of homonymies.....	20
2.2.4 Step 4. Retrieve of articles, citations and metrics	20
3. Data processing	22
3.1 Data check/integration	22
3.2 The String similarity problem	23
3.3 Data normalization	29
3.3.1 First step: creation of the LU_TEACHERS table.....	29
3.3.2 Second step: creation of the historical CU_ASSIGNMENTS table.....	30
4. Data Analytics	32
4.1 Data import and Data model setup.....	32
4.2 Data Visualization.....	32
Conclusions and future directions.....	43
Bibliography.....	45
Sitography	46
Appendix 1: the Python code	47
1. Consolidation of Cerca Università extractions in one dataset	47
2. Google Scholar Profile extraction.....	55
3. Google Scholar Citations, Metrics and Articles extraction.....	57
Appendix 2: the SQL database definition	59
Appendix 3: the SQL data model.....	64
Appendix 4: the Levenshtein Distance implementation as SQL function.....	65
Appendix 5: the SQL store procedure for data processing	67

Introduction

This thesis provides a *demographic analysis* of the Luiss teaching staff over the years with a specific focus on the information extracted from Google Scholar regarding the articles published by each teacher and the related citations.

Demographic analyses are used in many fields and contexts, both at a country level and at a more local level (by companies and organizations), to study the change and evolution in the population with respect to certain demographic characteristics (age, gender, ethnicity, income...).

The literature about this topic has identified many cases of application of demographic analyses especially in the academic field: El Refae et al. (2021) investigated the impact of demographic characteristics, in this case gender, college, and status of the students, on academic performance in face-to-face (F2F) learning and distance learning (DL); Sadeghi et al. (2012) conducted a study to verify if demographic characteristics, specifically gender, level of education, academic rank and age, could determine some differences in job satisfaction among academic staff, considering as population three Malaysian Research Universities; Starting from the evidence of the underrepresentation of women among science and engineering faculties and for high-ranking positions, Thomas et al (2015) studied how demographic data could influence these gender gaps specifically in faculty recruitment, retention, and promotion.

This thesis considers the Luiss teaching staff from 2000 until today as a population and a series of information (personal and on employment status), taken from multiple sources, as demographic characteristics, with the aim of showing the evolution over the years teaching staff through various analytical perspectives.

The major contribution of the thesis is therefore to guide the University, with these analyses, in the decision-making process and create an increasingly data-driven culture.

In detail, the work is structured as follows: the **first chapter** describes the two data sources supporting the final model: Cerca Università, i.e. a site that allows to obtain information on Universities (degree courses, funding, teachers, university students, tenders...), and an Excel file from the Faculty, later called “Docenti Luiss”. The first data source provides 23 Excel files that contains mainly the information about the employment status of the Luiss teaching staff from 2000. It was therefore necessary a consolidation of these files in one structured dataset in order to develop the subsequent analyses. The latter made it possible to obtain some initial insights (mainly relating to the evolution of the teaching staff between departments over the years) which were shown in a more structured ways in the final dashboards.

Both the operations (extraction and analysis) have been performed in Python. In this step the main criticalities are related to the necessity of various data cleaning.

The second data source, even if with a limited history, integrates the previous one with more information about the teaching staff especially from a personal point of view. In this case, the problems that emerged related, in addition to the need to clean the data, to the identification of incorrect situations reported (as the file was managed manually).

The **second chapter** illustrates the procedure for extracting in Python the teachers' data who have a profile on Google Scholar, a search engine whose functions are briefly described at the beginning of the paragraph. In particular, the information extracted concerns the publications and the citations of each professor, with the relative indexes, and their interests. Also in this case the difficulties encountered were those concerning the cleaning prior to the extraction process: the way Google Scholar searches for the information entered is not entirely clear and sometimes it did not return the expected results.

The **third chapter** is dedicated to the integration of the three data sources and the subsequent creation of a data model on which to base the final analyses and visualizations. Both operations were executed using SQL scripts. Here the critical issues encountered mostly related to the complexity to integrate heterogenous data sources not having a common key: the only field able to connect the various sources was the teacher's name leading to the necessity to explore the string similarity theory. All other inconsistencies between the various data sources have been identified in this step.

The **fourth chapter** explains the import of the SQL data into a business intelligence tool (Microsoft Power BI) where an extensive set of dashboards has been created to show all the analytics. The first series of graphs shows the evolution of the number of professors over time by department, role and gender. The following dashboards show the average age of the teachers, with reference, also in this case, to the dimensions just mentioned. Then some interactive maps illustrate the distribution by country, first with reference to 2000 and then to 2023, showing a steady increase of the number of teachers from the European Union. Finally, the historical trend of new entrants and retired/terminated teachers is displayed in a specific dashboard.

The second set of dashboards show the results obtained from the Google Scholar data extraction. The most interesting ones are those related to the average number of articles and citations by year and department. Finally, two wordclouds are used to highlight the most frequent interests among the professors' profiles.

Of course the print screens attached do not render the power of interactive visualization/navigation of this tool (e.g. filtering/selecting an item causes the immediate refresh of all the graphical objects).

The thesis ends with some final conclusions also regarding the possible future development of this work.

From a technical point of view this thesis has required various skillset:

- The ability to write Python code to manage dataset and to extract data from the Web
- The ability to create a relational data model with the development of SQL scripts (stored procedures and functions) to process the data
- The ability to create interactive dashboards with one of the most commonly used Business Intelligence tool

1. Dataset Analysis

1.1 The first data source: Website “Cerca Università”

The first step of the thesis involved extracting data from the “Cerca Università” Website.



Ricerca avanzata

Ruolo:
Tutti i ruoli

Tutti Confermati Non confermati

Cognome: Nome:

Cognome esatto Parte del cognome Inizio del cognome

Genere: Tutti Maschile Femminile

Ateneo:
Luiss Guido Carli

Servizio in altro ateneo al 100% (convenzioni Art.6, comma 11, Legge 240/10): 100%

Facoltà:
Tutte le Facoltà

S.S.D.:
Tutti i settori

Area:
Tutte le aree

Macrosettore:
Tutti i macrosettori

Settore concorsuale:
Il menù a tendina presenta i settori concorsuali determinati dal Decreto Ministeriale n. 855 30 ottobre 2015, GU n. 271 del 20-11-2015 - Suppl. Ordinario n.63. Per ricercare docenti inquadrati in settori concorsuali soppressi è necessario selezionare la voce 'Tutti'
Tutti i settori concorsuali

Situazione:
ad oggi

The site allows to see the data “as of today” or the previous year-end (back until year 2000).

vai alla ricerca avanzata Trovati: **127 nominativi** , visualizzati da **1 a 20**

Risultati della ricerca

Hai cercato:

CONFERMATI E NON CONFERMATI

Cognome: Tutti

Nome: Tutti

Genere: Tutti

Servizio in altro ateneo: Tutti

Ateneo: **Luiss Guido Carli**

Facoltà: Tutte

Settore: Tutti

Area: Tutte

Macrosettore: Tutti

Settore concorsuale: Tutti

Qualifica: Tutte

Situazione al: ad oggi (20/5/2023)

trovati: **127 nominativi**, visualizzati da **1 a 20**

Salva il risultato

File Excel

[Elenco settori scientifico disciplinari](#)

[Elenco settori concorsuali](#)

Fascia	Cognome e Nome	Genere	Facoltà	S.S.D.	S.C.	Struttura di appartenenza	Servizio prestato in altro ateneo
■ Associato	BELLACOSA Maurizio	M		IUS/17	12/G1	GIURISPRUDENZA	
■ Ordinario	BENIGNO Pierpaolo	M		SECS-P/01	13/A1	ECONOMIA E FINANZA	
■ Associato	BIAGINI Sara	F		SECS-S/06	13/D4	ECONOMIA E FINANZA	
■ Ordinario	BIFULCO Raffaele	M		IUS/08	12/C1	GIURISPRUDENZA	
■ Ordinario	BOCCARDELLI Paolo	M		SECS-P/08	13/B2	IMPRESA E MANAGEMENT	
■ Ricercatore a t.d. - t.pieno (art. 24 c.3-a L. 240/10)	BONTADINI Filippo	M		SECS-P/06	13/A4	IMPRESA E MANAGEMENT	
■ Ricercatore a t.d. - t.pieno (art. 24 c.3-b L. 240/10)	BORRI Nicola	M		SECS-P/11	13/B4	ECONOMIA E FINANZA	
■ Ordinario	BOZZOLAN Saverio	M		SECS-P/07	13/B1	IMPRESA E MANAGEMENT	
■ Associato	BRUNETTA Federica	F		SECS-P/08	13/B2	IMPRESA E MANAGEMENT	
■ Ricercatore a t.d. - t.pieno (art. 24 c.3-a L. 240/10)	BRUNI Elena	F		SECS-P/10	13/B3	IMPRESA E MANAGEMENT	
■ Ordinario	BUSCO Cristiano	M		SECS-P/07	13/B1	IMPRESA E MANAGEMENT	
■ Ordinario	CAROLI Matteo Giuliano	M		SECS-P/08	13/B2	IMPRESA E MANAGEMENT	
■ Associato	CAVALLARO Maria Elena	F		SPS/06	14/B2	SCIENZE POLITICHE	

The list of teachers in charge to a specific date can be extracted as an Excel file.

I have extracted all the available data into 23 Excel files containing the “pictures” of the teaching staff on December 31st of the years ranging from 2000 to 2022 plus that relating to 28 February 2023 (the day the extraction was made). The Excel files have been consistently named including the period of extraction as “RICERCADOCENTI_YYYY_MM.XLSX”

1.1.1 Introducing the dataset

The 23 Excel files have the same structure: each row corresponds to a specific professor and consequently they have a variable number of rows depending on the number of professors present at Luiss that year.

The columns of the Excel files are the following:

- “Fascia”: role played by the teacher
- “Cognome e Nome”: Professors’ name
- “Genere”: M (male) or F (female)
- “Facoltà”: Department (until 2010 and then not used anymore)
- “S.S.D.”: Settore Scientifico Disciplinare
- “S.C.”: Settore Concorsuale
- “Struttura di appartenenza”: Department
- “Servizio prestato in altro ateneo” – column introduced since 2023

1.1.2 Exploring the dataset with Python

The consolidation of the 23 Excel files and subsequent data analyses have been performed using Python. The first step was to merge the 23 files into one single dataset adding an additional column containing the reference date. The consolidated file was at the end saved as “consolidato.xlsx”.

The initial phase of Explanatory Data Analysis (EDA) usually aims to get a detailed understanding of the dataset: checking for duplicate records, presence of nulls, domain of the categorical columns ... are just some of the main activities performed in this step.

This first overview showed that the dataset has a total of 2.360 records (each professor can be present in several years): the unique number of professors is 255.

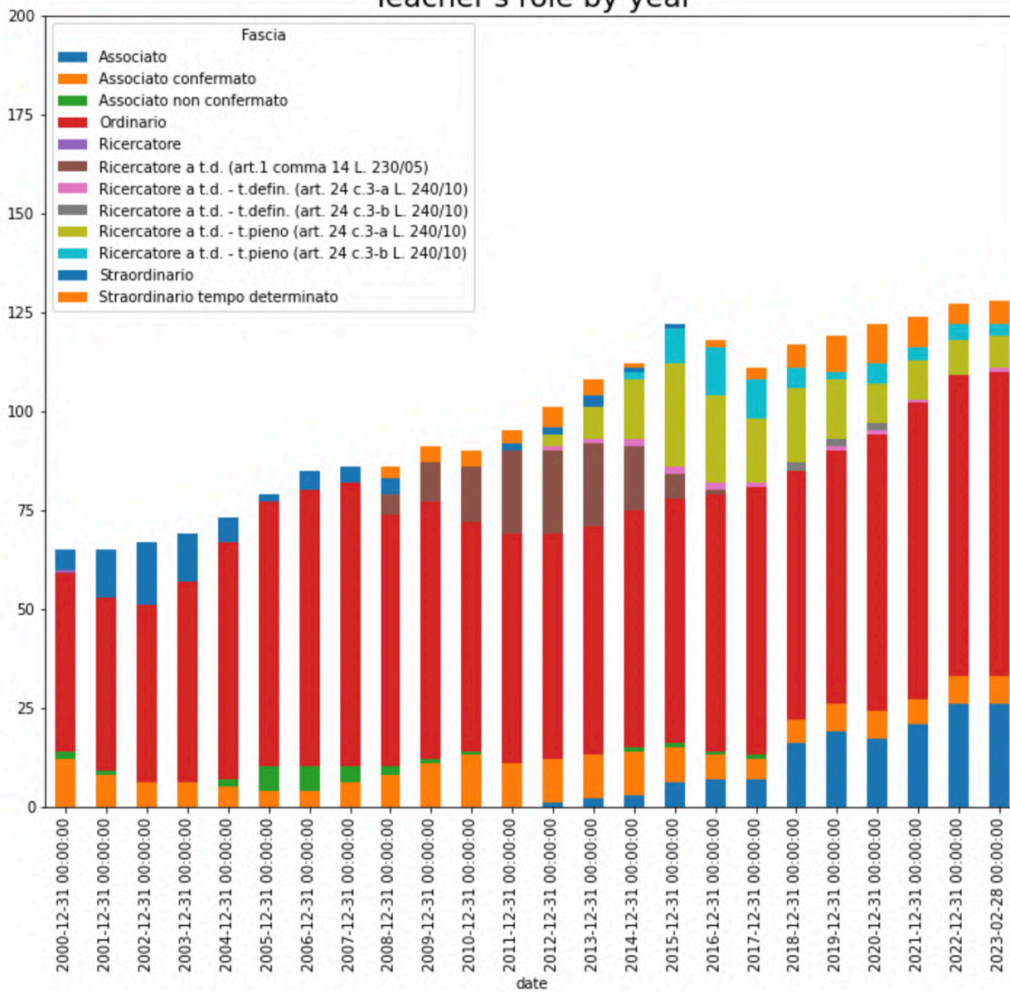
Considering as unique key the couple <Date>/<Cognome e nome> there are no duplicated values.

Null values are only present in the following columns:

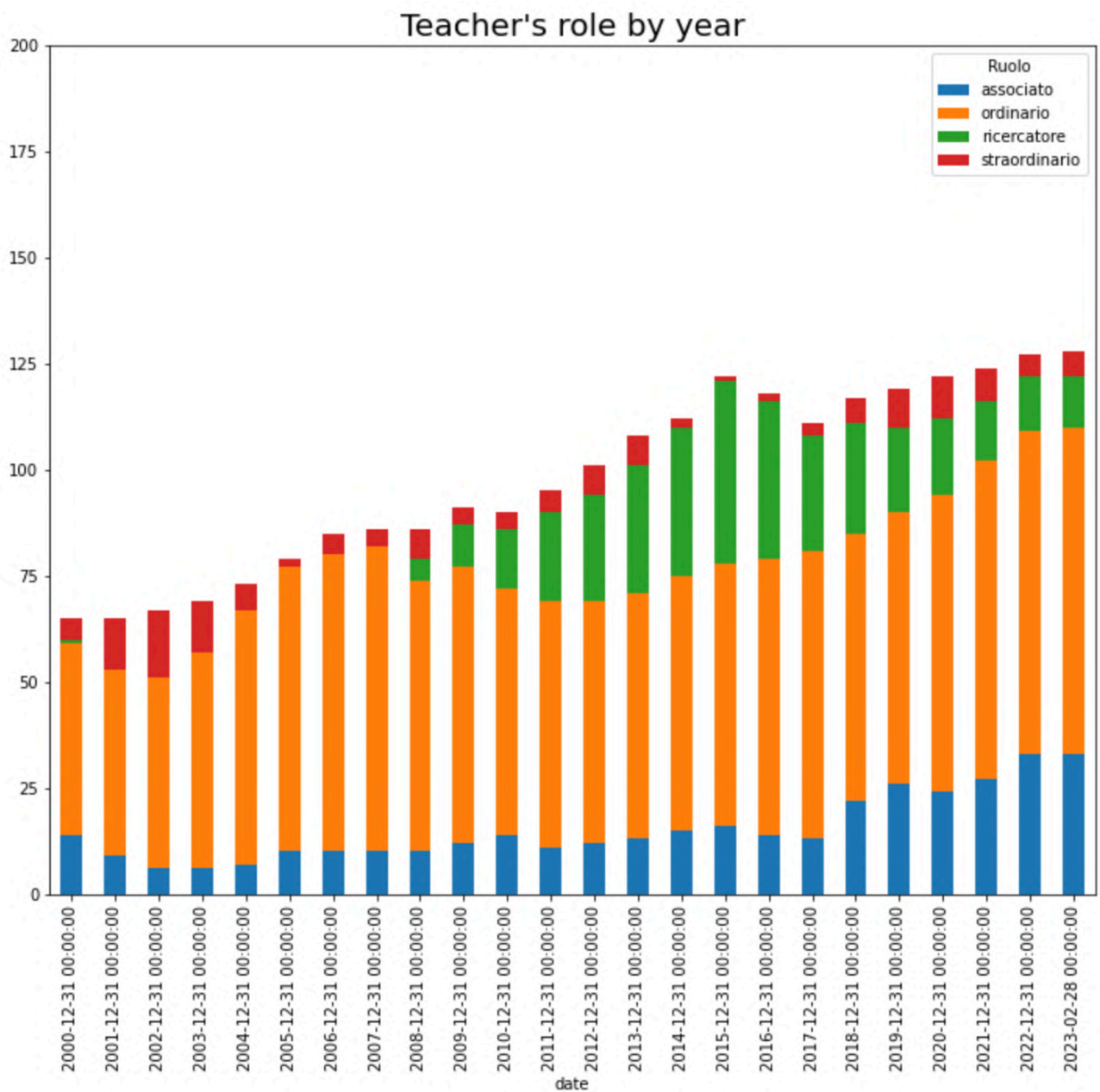
- “Facoltà”: in 2008, 2009 and 2010 a few nulls start to appear and since 2011 on the column is always empty
- “Settore Scientifico Disciplinare”: the column is empty only in year 2000 and since 2001 onwards is always valued
- “Settore Concorsuale”: a lot of nulls until 2010 but from 2011 only rare exceptions
- “Servizio prestato in altro ateneo”: always null

At this point let’s analyze the most significant categorical variables in detail (this phase is often given the name of “univariate analysis”). For this purpose, a series of visualizations were carried out. The first graph is a stacked barplot that represents the teacher’s role by year: in the period considered most of the professors cover the role of ordinario and that this evidence remains true even in the face of the trend of growth assumed by the number of professors until 2023.

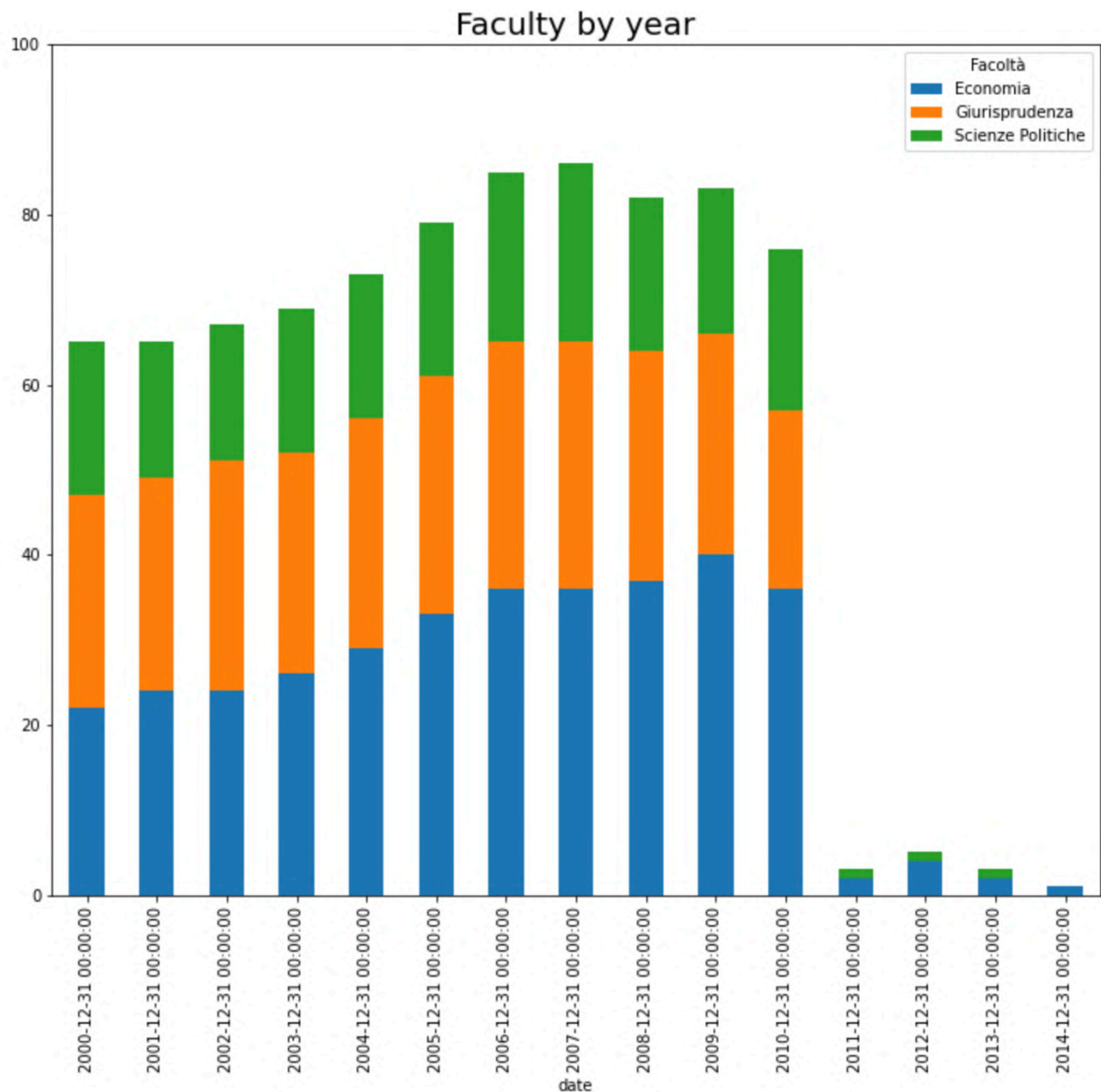
Teacher's role by year



However, the data is quite fragmented into the 12 categories assumed by the “Fascia” variable and in order to have a better visualization and understanding of the graph, similar roles have been aggregated by adding a new column to the dataset containing exclusively 4 roles: associato, ordinario, ricercatore and straordinario. Also in this case the role of ordinario prevails over the others.

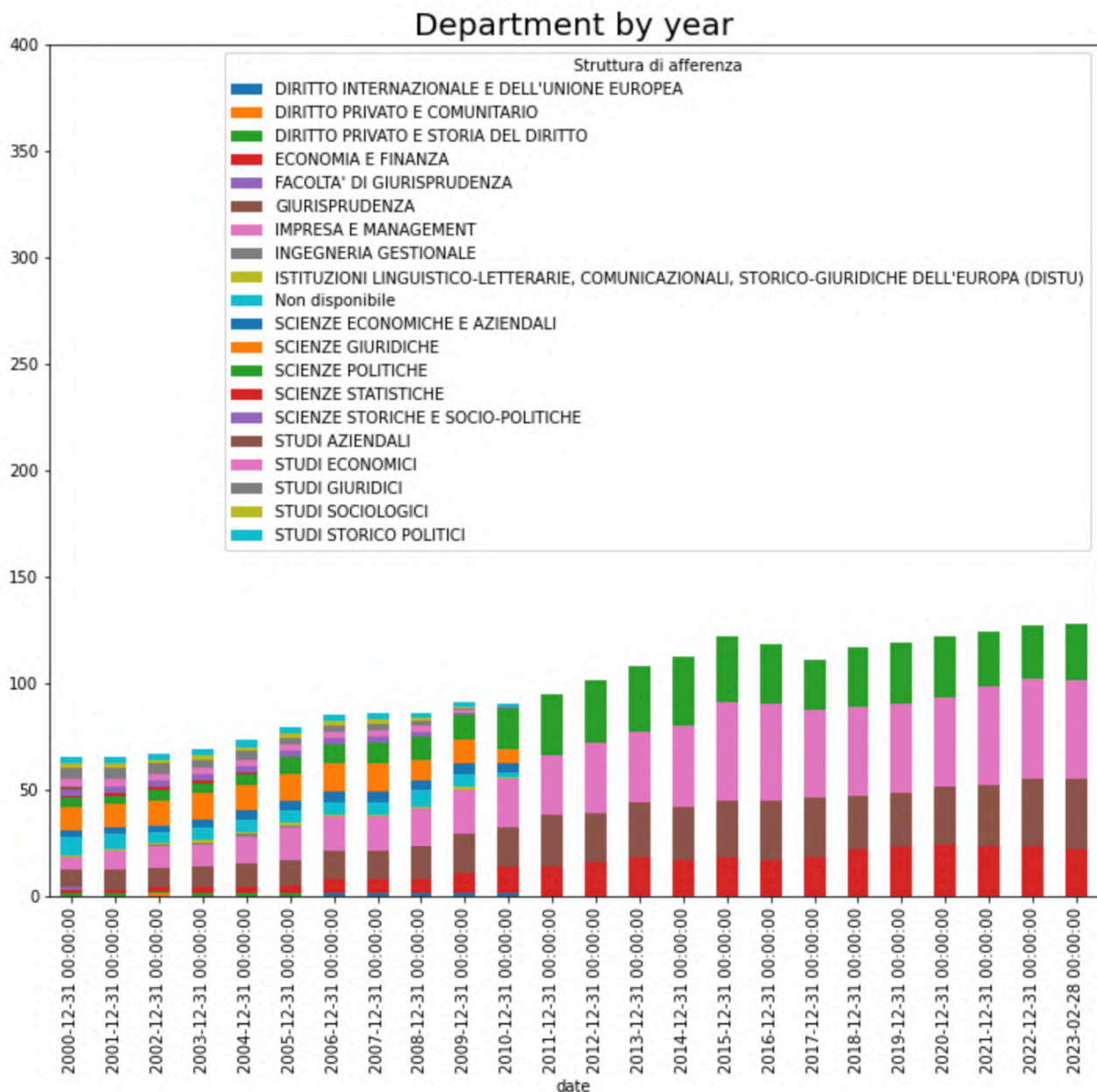


The following stacked barplot represents the “Facoltà” variable by year. Unfortunately, the information is valued only up to 2010 and in subsequent years it is essentially no longer present.



From the stacked barplot relating to the “Struttura di afferenza” it can be seen that since 2011 the various departments have been reduced to 4:

- Economia e Finanza
- Impresa e Management
- Giurisprudenza
- Scienze Politiche



For analytical purposes it would have been important to homogenize the years on the 4 previous categories and for this reason an attempt was made to compare in Excel the data of the “Facoltà” variable with those of the “Struttura di afferenza” variable to understand how to aggregate the years from 2000 to 2010. However, it was not possible to aggregate the information as the data are not reconcilable (moreover, there is also a “non disponibile” category).

In the final step the dataset with the consolidated data from Cerca Università has been imported into a relational database (Microsoft SQL Server) for an easier integration/processing when new data sources are added to the model.

1.2 The second data source: Docenti Luiss

Cerca Università has certainly provided a solid starting point for the analysis which, however, cannot be limited exclusively to this source: on the one hand, the data provide partial information on the history of the professors (in particular, personal data are missing), on the other hand the site does not record all those professors who have held the position of “Assistant Professor” and “Lecturer” at Luiss. It was therefore necessary to integrate Cerca Università with a second data source, i.e. a partially anonymized selection of data from an Excel Faculty sheet which, hereinafter, will be referred to as “Docenti Luiss”

1.2.1 Introducing Docenti Luiss

The structure of the Docenti Luiss broadly follows that of Cerca Università: each row is a professor and each column is a variable, categorical or numeric, which provides some information on the subject. However, it is possible to find the same teacher on several rows as, at this level, there is a coloring of the cells with the following meaning:

- White background --> current situation (updated in April 2023)
- Orange background --> finished positions (when a professor changes qualification the old record turns orange and a new white one is added)
- Green background --> expected future positions (in some cases there is a real name and a hire date in 2022)
- Heavenly background --> Rectorate requests (there is never a real name)
- Yellow background --> uncertain forecast on the expiry of the term.

Docenti Luiss has the following 10 columns:

- “Nominativo”
- “Dipartimento”
- “Qualifica”
- “Qualifica_aggregata”
- “Settore”
- “Settore_concorsuale”
- “Data_di_nascita”
- “Nazionalità”
- “UE_ExtraUE_ITA”
- “Genere”
- “Colore” - B, A or V (this column has been added to codify the meaning of the records)

Unfortunately, it has a quite limited history covering substantially the last 2-3 years (e.g. a professor that left Luiss in 2015 is not present in the file).

Also in this case I have imported Docenti Luiss into an SQL table for a more in-depth analysis: however not all rows from the Excel file have been imported but only the white ones (current situation), the orange ones (finished positions) and the green ones (expected future positions) having name valued.

1.2.2 Exploring Docenti Luiss with SQL

As in the case of Cerca Università, after importing Docenti Luiss into SQL, the first phase was that of understanding the general framework, which brought to light the following characteristics: in total there are 269 records (professors) which correspond to a number of distinct teachers equal to 224. Furthermore, the current teachers (with white background) are 170. This last information was obtained thanks to the previous color coding of the Excel file in an additional column having 3 different values: “B” (white), “V” (green), “A” (orange).

```

select count(*) as teachers
from DOCENTI_LUISS

select count(distinct Nominativo) as distinct_teachers
from DOCENTI_LUISS

select Colore, count(*) as teachers
from DOCENTI_LUISS
group by colore
order by 1
    
```

teachers	
1	269

distinct_teachers	
1	224

	Colore	teachers
1	A	88
2	B	170
3	V	11

An analysis of the “Dipartimento” column then highlighted that in addition to the 4 expected values (Impresa e Management, Economia e Finanza, Scienze Politiche, Giurisprudenza) there are also the teachers of the Business School who are not present in Cerca Università: I did not consider these records in the analysis.

```

select *
FROM GS_ARTICLES
where author_id = 'QBuy5B0AAAAJ' AND title = 'Search this site'

SELECT Dipartimento, count(*) AS Num_records
FROM DOCENTI_LUISS
GROUP BY Dipartimento
    
```

Dipartimento	Num_records	
1	NULL	3
2	Economia e Finanza	44
3	Giurisprudenza	53
4	Impresa e Management	99
5	Luiss Business School	14
6	School of European Political Economy	1
7	School of Government	1
8	Scienze Politiche	54

```

select Nominativo, Dipartimento, Qualifica, Qualifica_aggregata, Settore, Settore_concorsuale, Data_di_nascita, Nazionalita, UE_ExtraUE_ITA, Genere, Colore
from DOCENTI_LUISS
WHERE Dipartimento in ('Luiss Business School', 'School of Government', 'School of European Political Economy')

```

	Nominativo	Dipartimento	Qualifica	Qualifica_aggregata	Settore	Settore_concorsuale	Data_di_nascita	Nazionalita	UE_ExtraUE_ITA	Genere	Colore
1	ROSSI Salvatore	School of European Political Economy	Professor of practice	Professor of practice	NULL	NULL	1949-01-06	Italiana	Italiani	M	B
2	GIORGINO Francesco	School of Government	Professor of practice	Professor of practice	NULL	NULL	1967-08-08	Italiana	Italiani	M	B
3	MAGLIONE Roberto	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1957-08-20	Italiana	Italiani	M	B
4	MAZZU' Marco Francesco	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1972-08-24	Italiana	Italiani	M	B
5	MONTEFUSCO Andrea	Luiss Business School	Lecturer	Lecturer	NULL	NULL	1963-03-10	Italiana	Italiani	M	B
6	ZANARDI CAPPON Anna	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1964-07-26	Italiana	Italiani	F	B
7	MASTROGIACOMI Francesca	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1973-12-21	Italiana	Italiani	F	B
8	MAGNI Luca	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1964-05-09	Italiana	Italiani	M	B
9	FEI Carlo	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1961-03-30	Italiana	Italiani	M	B
10	PARDOSSI Valeria	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1970-07-20	Italiana	Italiani	F	B
11	TORRISI Alfio	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1947-10-10	Italiana	Italiani	M	B
12	STERK John	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	NULL	Olandese	UE	M	B
13	SILARD Tony Glen	Luiss Business School	Associate Professor LBS	Full professor LBS	NULL	NULL	NULL	Statunitense	Extra UE	M	B
14	MUFF Katrin	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1969-09-06	Svizzera	Extra UE	F	B
15	VIGNE Samuel Alexandre	Luiss Business School	Full Professor LBS	Full professor LBS	NULL	NULL	NULL	Tedesca	UE	M	B
16	FALKENSTEIN Mathias	Luiss Business School	Professor of practice	Professor of practice	NULL	NULL	1970-02-11	NULL	NULL	M	B

Being the source a manually managed Excel file, the problems are various: in addition to the fact that NULLs are found on various columns (sector, competition sector, date of birth) there are probably incorrect situations.

For example, in the two cases shown in the figure below, the records with a white background (being finished positions) should have been orange as well as the green records should have been white (representing the current situation). Probably the Excel file is not timely and consistently updated.

	A	B	C	D	E	F	G	H	I
1	Nominativo	Dipartimento	Qualifica	Qualifica aggregata	Data di nascita	Nazionalità	UE_ExtraUE_ITA	Genere	Colore
2	POZHARLIEV Rumen Ivaylov	Impresa e Management	Ricercatore Junior lett. a)	Ricercatore a)	07/01/83	Bulgara	UE	M	B
3	POZHARLIEV Rumen Ivaylov	Impresa e Management	Assistant Professor (Research)	Assistant Professor (Research)	07/01/83	Bulgara	UE	M	V
4									
5									
6	Nominativo	Dipartimento	Qualifica	Qualifica aggregata	Data di nascita	Nazionalità	UE_ExtraUE_ITA	Genere	Colore
7	TARANTINO Emanuele	Economia e Finanza	Professore straordinario ex art. 1 co. 12 L. 230/05	Ordinario	30/07/79	Italiana	Italiani	M	B
8	TARANTINO Emanuele	Economia e Finanza	Ordinario	Ordinario	30/07/79	Italiana	Italiani	M	V

2. Retrieving data from Google Scholar

2.1 Short description of Google Scholar

Google Scholar is a comprehensive academic search engine that offers researchers and scholars a wide range of functionalities to access scholarly literature and perform in-depth research. With its vast database of academic publications, Google Scholar serves as a powerful tool for locating and exploring scholarly articles, books, theses, conference papers, and more. The main functionalities of Google Scholar are the following:

1. **Extensive Database:** Google Scholar boasts a vast collection of scholarly literature from a wide range of disciplines and sources. It indexes academic content from publishers, universities, repositories, and other scholarly Websites, making it a comprehensive resource for researchers.
2. **Search Capabilities:** The search feature of Google Scholar allows users to enter specific keywords, phrases, or author names to retrieve relevant academic publications. Advanced search options enable users to refine their queries by specifying publication dates, authors, journals, and more.
3. **Article Citations and Metrics:** Google Scholar provides citation information, including the number of times an article has been cited by other scholarly works. This feature helps researchers gauge the impact and influence of a particular publication within the academic community.
4. **Related Articles and Recommendations:** Google Scholar provides related article suggestions based on the content of the currently viewed publication. This feature assists researchers in discovering additional relevant works and expanding their knowledge in a particular subject area.
5. **Researcher Profiles:** Authors can create and manage their researcher profiles on Google Scholar, allowing them to showcase their publications, citation metrics, and h-index—a measure of their research productivity and impact. These profiles serve as a hub for researchers to share their work and connect with peers.
6. **Metrics and Rankings:** Google Scholar provides several metrics and rankings, including the h-index and i10-index, which assess the productivity and impact of individual researchers and research groups. These metrics help evaluate scholarly output and facilitate collaborations.
7. **Advanced Search Options:** Google Scholar offers advanced search options to refine queries, such as restricting the search to specific publications, authors, or date ranges. Users can also search within specific disciplines or specify the type of content, such as patents or legal opinions.

Overall, Google Scholar offers researchers a powerful and user-friendly platform to discover scholarly literature, access full-text articles, track citations, and stay informed about the latest developments in their

respective fields. Its extensive functionalities and comprehensive database make it an invaluable tool for academic research and exploration.

2.2 Data extraction from Websites

The process of extracting data from Websites automatically using a computer program or script is usually referred to as Web Scraping. It involves retrieving structured or unstructured data from Web pages, which can then be analyzed, stored, or utilized for various purposes. Web scraping enables users to collect large amounts of data from multiple sources quickly and efficiently.

It is important to note that when doing Web scraping, one should be mindful of legal and ethical considerations. Websites may have terms of service or usage policies that prohibit or limit scraping. It's advisable to review and respect these guidelines, seek permission if necessary, and ensure that the scraping process does not cause harm or disruption to the target Website's servers.

Web scraping has numerous applications, including market research, price monitoring, sentiment analysis, content aggregation, data journalism, and academic research. It provides a valuable means of automating data collection from the Web, saving time and effort in retrieving and analyzing information.

In this thesis I have written the Python code and used a library called SerpApi to search for the teacher's profile and, when found, to extract summary information like the number of articles that have been published and the few metrics available (e.g. h-index, i10 index ...).

The limited number of professors that were extracted (less than 200) and the limited amount of data that has been collected, joined to the non-commercial usage of this information, should strongly mitigate any ethical concern.


2.2.1 Step 1. Preliminary cleaning


Before starting data scraping from Google Scholar, a preliminary data cleaning activity was necessary for professors with **double names** or **double surnames**: in such cases, the search with the names extracted from Cerca Università often does not produce the desired results.

For example, trying to search for "ITALIANO Giuseppe Francesco" (the name available in Cerca Università) only two profiles are retrieved and none of them is the profile I was looking for.

Google Scholar ITALIANO Giuseppe Francesco

Profili


 **Giuseppe Francesco Rigano** Citato da 159
 Istituto **Italiano** di Tecnologia
 Email verificata su iit.it
 Robotics Software


 **Francesco Giuseppe Biondo** Citato da 123
 ... UOC di Chirurgia Generale e Specialistica, Ospedale Fatebenefratelli, Benevento
Italia
 Email verificata su virgilio.it
 Chirurgia Generale - Chirurgia ...


Vice versa, changing the name into “ITALIANO Giuseppe”, a series of profiles are extracted, among which the desired one appears in second place.


Google Scholar ITALIANO Giuseppe


Profili


 **Giuseppe Filardo**
 Rizzoli Orthopaedic Institute, Ente Ospedaliero Cantonale, Università della Svizzera
Italiana
 Email verificata su gfilardo.com


 **Giuseppe F. Italiano**
 Professor of Computer Science, LUISS University, Rome, Italy
 Email verificata su luiss.it
 Algorithms Data Structures Graph Algorithms Networks

 **Manuel Giuseppe Catalano**
 Researcher, Istituto **Italiano** di Tecnologia, Soft Robotics for Human Cooperation and ...
 Email verificata su iit.it
 Robotics Rehabilitation Prosthetics Humanoids Mechatronics

 **Giuseppe Vicidomini**
 Researcher, Istituto **Italiano** di Tecnologia, Molecular Microscopy and Spectroscopy
 Email verificata su iit.it
 Biophysics Optics Microscopy Super-Resolution Microscopy

 **Giuseppe Bardi**
 Ricercatore, Istituto **Italiano** di Tecnologia, Genova
 Email verificata su iit.it
 nanobiotechnology

 **Spartaco Gippoliti**
 Società **Italian** per la Storia della Fauna "**Giuseppe** Altobello"
 zoologia biologia della conservazione

 **Giuseppe Bianco**
 Agenzia Spaziale **Italiana**
 Email verificata su asi.it
 space geodesy quantum communications astronomy

To overcome this problem, a new column named “Docente” was added to the Cerca Università dataset, initially valued with the starting “Cognome e Nome” field from Cerca Università: subsequently, in all cases with a double name or surname, the name to be entered was checked manually in order to obtain the expected result and this name has therefore been overwritten in the “Docente” column.

The matching criteria of the search functionality of Google Scholar remains quite unclear: for sure Google Scholar searches the words entered in the search bar not only in the field dedicated to the name but also in the next line dedicated to the position held by the professor. Again, with reference to the search for “ITALIANO Giuseppe Francesco”, the third name is extracted due to the presence of the word “Italiano” in the job position.

Moreover, even the matching of the words seems quite loose: the first name is extracted because the word “Italiana” appears in the position held even if “Italiano” was searched for.

Another example about the strange behavior of the search functionality of Google Scholar is here below, where the full name “DI CAGNO Daniela Teresa” from Cerca Università is not found.


 

La ricerca su **DI CAGNO Daniela Teresa** non corrisponde ad alcun autore.

Suggerimenti:

- Assicurarsi che tutte le parole siano state digitate correttamente.
- Provare con parole chiave diverse.
- Provare con parole chiave più generiche.
- Provare con un numero minore di parole chiave.
- [Prova la query su tutto Scholar.](#)

Changing the double name into simply “Daniela” allows to find the desired profile.

Daniela Di Cagno

LUISS Guido Carli
Email verificata su luiss.it

[Microeconomia](#) [Economia sperimentale](#) [Microeconomics](#) [Experimental Economics](#)

Once again this confirms the necessity to implement a preliminary data cleaning in case of double surnames or double first names.

2.2.2 Step 2. Profiles retrieve

The first part of the Python code starts from the list of distinct teachers taken from Cerca Università and for each of them executes a profile search. The result can retrieve zero, one or more than one profile.

In case of multiple profiles are retrieved, I have coded a filter taking only those having the teacher's surname in the first line (tag «name»). The outcome of this step is that of the initial 255 teachers only 204 profiles were found.



Giuseppe F. Italiano

Professor of Computer Science, LUISS University, Rome, Italy

Email verificata su luiss.it

Algorithms Data Structures Graph Algorithms Networks

Citato da 8510

For each profile the 5 fields visible at screen were extracted plus some data not visible like “author_id” and the url of the teacher’s profile.

1	name	link	author_id	affiliations	email	cited_by	teacher	interests_new
2	Feray Adiguzel	https://scholar.google.com/citations?hl=en&user=m8K0YewAAAAJ	m8K0YewAAAAJ	Professor of Marketing	Verified email at ntu.ac.uk	721	ADIGUZZEL-Feray	Marketing, Statistics
3	Andrej Angelovski	https://scholar.google.com/citations?hl=en&user=DAEiEQ0iAAAAJ	DAEiEQ0iAAAAJ	Assistant Professor, Middlesex University	Verified email at mdx.ac.uk	62	ANGELOVSKI-Andrej	Experimental Economics, Behavioral Economics
4	Vincenzo Antonelli	https://scholar.google.com/citations?hl=en&user=juQzP1sAAAAJ	juQzP1sAAAAJ	Research Associate, Polito	Verified email at polito.it	34	ANTONELLI-Vincenzo	Algebraic Geometry
5	Antonio Baldassarre	https://scholar.google.com/citations?hl=en&user=4uZfmmQAAAAJ	4uZfmmQAAAAJ	University degli Studi di Firenze	Verified email at unimore.it	824	BALDASSARRE-Antonio	Medicina del Lavoro
6	Massimo BALDINI	https://scholar.google.com/citations?hl=en&user=Cj8kqgAAAAJ	Cj8kqgAAAAJ	Full professor of Political Economy, Universit� di Modena e Reggio Emilia	Verified email at unimore.it	1738	BALDINI-Massimo	Income tax, inequality, poverty, tax-benefit systems, welfare state
7	Gala Barone	https://scholar.google.com/citations?hl=en&user=xCTzAwAAAAJ	xCTzAwAAAAJ	Assistant Professor, Economics and Finance, National College of Ireland	Verified email at ncri.ie	48	BARONE-Gala	Finance, Credit Risk, Derivatives
8	Pierpaolo Benigno	https://scholar.google.com/citations?hl=en&user=lgrp-cQAAAAJ	lgrp-cQAAAAJ	Professor of Economics, University of Bern	Verified email at wvli.unibe.ch	6352	BENIGNO-Pierpaolo	Monetary Economics, International Economics

This profile data is finally saved to a csv file for the subsequent processing in step 3.

The “interest” of the profiles (e.g. in the example above “Algorithms”, “Data Structures”, “Graph Algorithms” and “Networks”) are extracted in a text field where individual interests are separated by comma. To simplify the subsequent processing I have created a new dataset with 2 columns: author_id and interest (in the example above there will be 4 records).

2.2.3 Step 3. Cleaning of homonymies

The extracted data requires a second manual data cleaning to resolve the case of homonyms on the Surname. In some cases the correct profile was easily identifiable (email, interests ...) while in other cases it was necessary to search on the Internet (for example Linkedin ...) especially for former teachers where the extraction from Google Scholar shows the new job with no longer reference to the Luiss University.

After this step the number of professors with a Scholar profile has furtherly reduced to 153.

2.2.4 Step 4. Retrieve of articles, citations and metrics

For each of the 153 profiles filtered in the previous step, the Python code extracts the 3 sections on the profile detail page:

1. three summary KPIs: number of citations, H-index and i10-index. For the 3 metrics I got both the overall value and the value calculated on the last 5 years
2. the trend of citations by year (unlike the graph, the data extracted start from 2000)
3. the details of all the articles published by the teacher. Here unfortunately the co-authors are listed only with one single char for the first name followed by the surname (e.g. R Baldoni): not having the author_id of the co-authors makes difficult to create a graph structure representing collaboration

The h-index, abbreviation for Hirsch index, was introduced by Jorge E. Hirsch in 2005 to measure the productivity and citation impact of an author’s publications. It is defined as the maximum value of h such that the given author has published h papers that have each been cited at least h number of times. For example, Irene Finocchi has an h-index of 27 which means that she has published 27 articles that have each earned at least 27 citations.

Introduced 6 years later, the i10-index calculates the total number of published articles of an author with at least 10 citations. According to Google Scholar Irene Finocchi has an i10-index score of 50.

The screenshot shows the Google Scholar profile of Irene Finocchi. It includes a profile picture, name, affiliation (Professor of Computer Science, LUISS University, Rome, Italy), and research interests (Algorithms, Big data analytics, Program analysis). A table lists her publications with columns for title, citations, and year. A summary table shows total citations (2731), H-index (27), and i10-index (50), along with values for the last 5 years (1101, 12, 18). A bar chart shows the citation trend from 2016 to 2023. Three purple circles with numbers 1, 2, and 3 are overlaid on the image, pointing to the summary table, the bar chart, and the publications table respectively.

Citata da	Tutte	Dal 2018
Citazioni	2731	1101
Indice H	27	12
i10-index	50	18

TITOLO	CITATA DA	ANNO
A survey of symbolic execution techniques R Baldoni, E Coppa, DC D'elia, C Demetrescu, I Finocchi ACM Computing Surveys (CSUR) 51 (3), 1-39	582	2018
Handbook of data structures and applications DP Mehta, S Sahni Chapman and Hall/CRC	323	2004
Input-sensitive profiling E Coppa, C Demetrescu, I Finocchi ACM SIGPLAN Notices 47 (6), 89-98	118	2012
Trading off space for passes in graph streaming problems C Demetrescu, I Finocchi, A Ribichini ACM Transactions on Algorithms (TALG) 6 (1), 1-17	118	2009

The outcome of the data extraction from Google Scholar has been to get 5 datasets:

1. Profiles (unique key: author_id)
2. Interests (unique key: author_id, interest)
3. Indexes (unique key: author_id)
4. Citations by Year (unique key: author_id, year)
5. Articles (unique key: citation_id)

Author_id is the information present in all the tables, making it possible to logically connect all of them. On the Profiles dataset I have added an additional column “teacher” containing the cleaned name (from Cerca Università) used to search for the profile.

3. Data processing

3.1 Data check/integration

As already anticipated, to make this step more efficient I have created an SQL database (called Docenti_Luiss) and imported all the datasets of the project using the import wizard of MS SQL Server (in appendix 2 I have attached the script of all tables in the database).

In the previous chapter I have analyzed the 2 main datasets (Cerca_Università and Docenti Luiss) individually. In this step I called “Data Integration” I want to compare the 2 datasets to find any potential inconsistency. Unfortunately, there is no unique key on both tables to identify a teacher (e.g. registration number, tax code ...) and I have therefore found myself faced with a “record linkage” problem. The only possibility was to join the two tables based on the column “Nominativo” of Docenti Luiss and “Cognome e Nome” of Cerca Università.

First of all, I wanted to identify all the teachers present in Docenti Luiss but not present in Cerca Università: it can be easily done using a left join query between the two tables.

```
-- TEACHERS PRESENT IN DOCENT_LUISS NOT PRESENT IN CERCA_UNIVESITATA -> FOUND 45 RECORDS
select Nominativo, Dipartimento, Qualifica, Qualifica_aggregata, DOCENTI_LUISS.Settore, DOCENTI_LUISS.Settore_concorsuale, Data_di_nascita, Nazionalita, UE_ExtraUE_ITA
from DOCENTI_LUISS LEFT JOIN CERCA_UNIVERSITA ON Cognome_Nome = Nominativo
WHERE Dipartimento NOT IN ( 'Luiss Business School', 'School of Government', 'School of European Political Economy') AND date IS NULL
```

Nominativo	Dipartimento	Qualifica	Qualifica_aggregata	Settore	Settore_concorsuale	Data_di_na
1 ALAIMO Cristina	Scienze Politiche	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/10 - Organizzazione aziendale	NULL	1979-05-30
2 ALTIERI Michela	Impresa e Management	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/09 - Finanza aziendale	NULL	1979-12-11
3 ANGELUCCI Davide	Scienze Politiche	Lecturer	Lecturer	NULL	NULL	NULL
4 BALACHANDRAN NAIR Lakshmi	Impresa e Management	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/10 - Organizzazione aziendale	NULL	1985-04-00
5 BERKOVITCH Jonathan	Impresa e Management	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/07 - Economia aziendale	NULL	NULL
6 BRUNI Domenico Maria	Scienze Politiche	Lecturer	Lecturer	M-STO/04 - Storia contemporanea	NULL	1971-03-11
7 CALLUSO Cinzia	Impresa e Management	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/10 - Organizzazione aziendale	NULL	1988-11-21
8 CANOFARI Paolo	Economia e Finanza	Ricercatore Junior lett. a)	Ricercatore a)	SECS-P/01 - Economia politica	13/A1 Economia politica	1977-12-00
9 CARI INI Federico	Economia e Finanza	Assistant Professor (Research)	Assistant Professor (Research)	SECS-P/05 - Economia	NULL	1985-06-20

The result is that there are 45 professors on Docenti Luiss who are not present in Cerca Università: from the “Qualifica” column is evident that most of the professors not present in Cerca Università (31 out of 45) have the roles of Assistant Professor or Lecturer that, as expected, are not present in Cerca Università.

```
select qualifica, count(*)
from DOCENTI_LUISS LEFT JOIN CERCA_UNIVERSITA ON Cognome_Nome = Nominativo
WHERE Dipartimento not in ( 'Luiss Business School', 'School of Government', 'School of European Political Economy') and date IS NULL
group by qualifica
order by 2 desc
```

qualifica	(Nessun nome di colonna)
1 Assistant Professor (Research)	16
2 Lecturer	15
3 Ricercatore Junior lett. a)	3
4 Associato distaccato	3
5 Associate professor (Research)	2
6 Associato	2
7 Ordinario	2
8 Ordinario distaccato	1
9 Adjunct professor	1

However, the 14 remaining records relating to qualifications (full professor, researcher ...) which, theoretically, should have been present in Cerca Università, remained to be explained.

The two qualifications of Assistant Professor and Lecturer were therefore excluded from the join in order to better understand who are the professors not present in the Cerca Università table.

```
-- TEACHERS PRESENT IN DOCENT_LUISS NOT PRESENT IN CERCA_UNIVESITATA EXCLUDING 2 QUALIFICA -> FOUND 14 RECORDS
select Nominativo, Dipartimento, Qualifica, Qualifica_aggregata, DOCENTI_LUISS.Settore, DOCENTI_LUISS.Settore_concorsuale, Data_di_nascita, Nazionalita, UE_ExtraUE_ITA
from DOCENTI_LUISS LEFT JOIN CERCA_UNIVERSITA ON Cognome_Nome = Nominativo
WHERE Dipartimento NOT IN ('Luiss Business School', 'School of Government', 'School of European Political Economy') AND date IS NULL
AND Qualifica NOT IN ('Assistant Professor (Research)', 'Lecturer')
```

	Nominativo	Dipartimento	Qualifica	Qualifica_aggregata	Settore	Settore_concorsuale
1	CANOFARI Paolo	Economia e Finanza	Ricercatore Junior lett. a)	Ricercatore a)	SECS-P/01 - Economia politica	13/A1 Economia politica
2	DI DONATO Francesca	Impresa e Management	Associato distaccato	Associato	SECS-P/07 - Economia aziendale	13/B1 Economia aziendale
3	FERNANDEZ DA SILVA RANCHORDAS Sofia Hina	Giurisprudenza	Associate Professor (Research)	Assistant Professor (Research)	IUS/09 - Istituzioni di diritto pubblico	12/D1 Diritto amministrativo
4	FERNANDEZ DA SILVA RANCHORDAS Sofia Hina	Giurisprudenza	Associato	Associato	IUS/09 - Istituzioni di diritto pubblico	12/D1 Diritto amministrativo
5	HOMBERG Fabian Karl	Impresa e Management	Ordinario	Ordinario	SECS-P/10 - Organizzazione aziendale	13/B3 Organizzazione aziendale
6	MARE' Mauro	Impresa e Management	Ordinario distaccato	Ordinario	SECS-P/03 - Scienza delle finanze	13/A3 Scienza delle finanze
7	POZZI Cesare	Giurisprudenza	Associato distaccato	Associato	SECS-P/06 - Economia applicata	13/A4 Economia applicata
8	SACCHI Stefano	Scienze Politiche	Associato distaccato	Associato	SPS/04 - Scienza politica	14/A2 Scienza politica
9	SAMMARCO Fabrizio	Impresa e Management	Adjunct professor	Adjunct professor	NULL	NULL
10	TEE Richard	Impresa e Management	Ricercatore Junior lett. a)	Ricercatore a)	SECS-P/10 - Organizzazione aziendale	13/B3 Organizzazione aziendale
11	TURNER Karynne Lenore	Impresa e Management	Ricercatore Junior lett. a)	Ricercatore a)	SECS-P/08 - Economia e gestione delle imprese	13/B2 Economia e gestione delle imprese
12	VENEL Xavier	Economia e Finanza	Associato	Associato	SECS-S/06 - Metodi matematici dell'economia e d...	13/D4 Metodi matematici dell'economia e
13	VENEL Xavier	Economia e Finanza	Associate professor (Research)	Assistant Professor (Research)	SECS-S/06 - Metodi matematici dell'economia e d...	13/D4 Metodi matematici dell'economia e
14	VINUALES Jorge Enrique	Giurisprudenza	Ordinario	Ordinario	IUS/13 - Diritto internazionale	12/E1 Diritto internazionale

I have randomly taken some of these names (e.g. VINUALES) to verify that they were really not present in Cerca Università.

```
select *
from CERCA_UNIVERSITA
where Cognome_Nome like '%VINUALES%'
```

	date	Cognome_Nome	Fascia	Genere	Facolta	Settore	Settore_concorsuale	Struttura_di_afferenza	Servizio_prestito_altro_ateneo	Ruolo
1	2022-12-31	VINUALES Jorge Enrique	Ordinario	M	NULL	IUS/13	12/E1	GIURISPRUDENZA	NULL	ordinario
2	2023-02-28	VINUALES Jorge Enrique	Ordinario	M	NULL	IUS/13	12/E1	GIURISPRUDENZA	NULL	ordinario

Surprisingly it was found there too. The reason for the missing join is that the name is not exactly the same in the 2 tables (Enrique <> Enrique): that could be probably related to a typing error in the Excel file.

3.2 The String similarity problem

To automatically identify these situations, was used the Levenshtein Distance (or Edit Distance), one of the most implemented algorithms to date to deal with the problem of “String Similarity”. Developed in 1965 by the Russian scientist Vladimir Levenshtein from which it takes its name, this algorithm falls into the broader category of the so-called Edit distance based algorithms which calculate the similarity between two strings in terms of the number of *operations* necessary to transform one string into another. In this specific case the allowed operations or “edits” are the following:

- *Insertion* of a character
- *Deletion* of a character
- *Substitution* of a character

In other words, the Levenshtein Distance between two words is the smallest number of single character transformations (insertions, deletions, or substitutions) required to transform one word into the other.

The logic is simple: the higher the number of transformations/edits/operations, that is, the greater the Levenshtein Distance, the more the strings differ.

In the present case the Levenshtein Distance is 4 because only one edit is needed to transform the word Enrique into Erique (or vice versa):

Enrique → Erique (deletion of “n”)

In mathematical terms, this metric is defined as follows:

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

where:

- a = string #1
- b = string #2
- i = the terminal character position of string #1
- j = the terminal character position of string #2
- $\text{lev}_{a,b}(i,j)$ is the distance between the first i characters of a and the first j characters of b
- $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i \neq b_j$ and equal to 1 otherwise
- the first element in the minimum corresponds to deletion (from a to b)
- the second to insertion
- the third to match or mismatch, depending on whether the respective symbols are the same

The common way of calculating the Levenshtein Distance is by using a dynamic programming algorithm.

In order to understand how it works, it will be used the ENRIQUE – ERIQUE example.

The algorithm builds a (m+1, n+1) matrix, where m is the length of the first string “ENRIQUE” and n is the length of the second string “ERIQUE”. The matrix is initialized with values representing the number of edits required to transform one string of length zero into the other.

	#	E	R	I	Q	U	E
#	0	1	2	3	4	5	6
E	1						
N	2						
R	3						
I	4						
Q	5						
U	6						
E	7						

Since 0 represents the null string, this means that, for example, only one edit will be required to transform the first letter “E” of both strings into the null string, 2 edits will be required to transform the first two letters (“ER” and “EN”) of both strings into the null string, and so on and so forth.

The matrix is filled from the upper left to the lower right corner that, in the end, gives the Levenshtein Distance. At each position (i, j), where i represents the row and j represents the column, the algorithm compares the characters at the corresponding positions in the two strings and calculates the Levenshtein Distance. For example, to convert the letter E of the first word ENRIQUE into the substring ERIQ of the second word ERIQUE it will be necessary 3 edits (the insertion of R, I and Q) so the Levenshtein Distance between E and ERIQ is 3 and the corresponding cell will be filled with this value.

	#	E	R	I	Q	U	E
#	0	1	2	3	4	5	6
E	1	0	1	2	3	4	5
N	2	1	1	2	3	4	5
R	3	2	1	1	2	3	4
I	4	3	2	1	1	2	3
Q	5	4	3	2	1	1	2
U	6	5	4	3	2	1	1
E	7	6	5	4	3	2	1

In Appendix 4 I have reported an SQL implementation of the Levenshtein Distance (as SQL function) I found on the Internet. After having checked with various tests that it worked properly, I applied it to the problem.

To highlight the cases in which there are small differences in the name I have changed the left join condition: instead of the equality between Cognome_nome and Nominativo the new condition foresees that the Edit Distance has to be lower than a certain threshold (8 in this case). In the image below the 7 cases in which they are the same person but with names written differently are highlighted in red (for example for

CANOFARI there are 2 blanks between the surname and name, in other cases there is a letter of difference or a name is missing).

```
-- docenti presenti in DOCENTI_LUISS non presenti in CERCA_UNIVERSITA -> trovati 62 record univoci
select distinct Nominativo
INTO #TEMP
from DOCENTI_LUISS LEFT JOIN CERCA_UNIVERSITA ON Cognome_Nome = Nominativo
WHERE date IS NULL

SELECT *, dbo.Levenshtein(Nominativo, Cognome_Nome, 100) AS DISTANCE
FROM #TEMP LEFT JOIN (SELECT DISTINCT Cognome_Nome
FROM CERCA_UNIVERSITA
) AS CU ON dbo.Levenshtein(Nominativo, Cognome_Nome, 100) < 8
WHERE Cognome_Nome IS NOT NULL AND LEFT(Nominativo,1)=LEFT(Cognome_Nome,1)
ORDER BY 3
```

	Nominativo	Cognome Nome	DISTANCE
1	FERNANDEZ DA SILVA RANCHORDAS Sofia Hina	FERNANDES DA SILVA RANCHORDAS Sofia Hina	1
2	CANOFARI Paolo	CANOFARI Paolo	1
3	VINUALES Jorge Enrique	VINUALES Jorge Enrique	1
4	DONATO Camela	DECARO Camela	3
5	MARTINO Alessio	MARTINO Antonio	4
6	MARTINO Alessio	MARINO Alessandro	5
7	MARZIONI Stefano	MANZOCCHI Stefano	5
8	SANTELLA Rosella	SABIA Rossella	6
9	VILLARROEL ORDENES Francisco	VILLARROEL ORDENES Francisco Javier	6
10	DI DONATO Francesca	DI CIOMMO Francesco	6
11	PORCHIA Paolo	PEVERINI Paolo	6
12	GOLIA Angelo	GALLO Daniele	6
13	IZZO Federica	IZZO Maria Federica	6
14	HOMBERG Fabian Karl	HOMBERG Fabian Kurt Falk	6
15	LUO Jianchuan	LUPO Nicola	7
16	MAGNI Luca	MARENGO Luigi	7
17	FARACE Stefania	FARANO Alessia	7
18	MONTEFUSCO Andrea	MANZELLA Andrea	7
19	TURNER Karynne Lenore	TURNER Karynne	7
20	PORCHIA Paolo	POLETTI Arlo	7
21	FEI Carlo	FERSINI Paola	7
22	LATELLA Maria	LAZAR Marc	7
23	PARDOSSI Valeria	PARDOLESI Roberto	7
24	DI DONATO Francesca	DI SABATO Franco	7
25	MONTEFUSCO Andrea	MIGLIONICO Andrea	7

Starting again from the list of the 14 names not present in Cerca Università, in 9 cases it is a matter of different coding of the name. Five cases remain to be explained (SAMMARCO, DI DONATO, MARE', POZZI, SACCHI) 4 of which are teachers temporarily assigned to other universities (that could explain why they are not present in Cerca Università). The only remaining unclarified case would be SAMMARCO.

```
-- TEACHERS PRESENT IN DOCENTI_LUISS NOT PRESENT IN CERCA_UNIVERSITATA EXCLUDING 2 QUALIFICA -> FOUND 14 RECORDS
select Nominativo, Dipartimento, Qualifica, Qualifica_aggregata, DOCENTI_LUISS.Settore, DOCENTI_LUISS.Settore_concorsuale, Data_di_nascita, Nazionalita, UE_ExtraUE_ITA
from DOCENTI_LUISS LEFT JOIN CERCA_UNIVERSITA ON Cognome_Nome = Nominativo
WHERE Dipartimento NOT IN ('Luiss Business School', 'School of Government', 'School of European Political Economy') AND date IS NULL
AND Qualifica NOT IN ('Assistant Professor (Research)', 'Lecturer')
```

	Nominativo	Dipartimento	Qualifica	Qualifica_aggregata	Settore	Settore_concorsuale
1	CANOFARI Paolo	Economia e Finanza	Ricercatore Junior (lett. a)	Ricercatore a)	SECS-P/01 - Economia politica	13/A1 Economia politica
2	DI DONATO Francesca	Impresa e Management	Associato distaccato	Associato	SECS-P/07 - Economia aziendale	13/B1 Economia aziendale
3	FERNANDEZ DA SILVA RANCHORDAS Sofia Hina	Giurisprudenza	Associate Professor (Research)	Assistant Professor (Research)	IUS/09 - Istituzioni di diritto pubblico	12/D1 Diritto amministrativo
4	FERNANDEZ DA SILVA RANCHORDAS Sofia Hina	Giurisprudenza	Associato	Associato	IUS/09 - Istituzioni di diritto pubblico	12/D1 Diritto amministrativo
5	HOMBERG Fabian Karl	Impresa e Management	Ordinario	Ordinario	SECS-P/10 - Organizzazione aziendale	13/B3 Organizzazione aziendale
6	MARE' Mauro	Impresa e Management	Ordinario distaccato	Ordinario	SECS-P/03 - Scienza delle finanze	13/A3 Scienza delle finanze
7	POZZI Cesare	Giurisprudenza	Associato distaccato	Associato	SECS-P/06 - Economia applicata	13/A4 Economia applicata
8	SACCHI Stefano	Scienze Politiche	Associato distaccato	Associato	SPS/04 - Scienza politica	14/A2 Scienza politica
9	SAMMARCO Fabrizio	Impresa e Management	Adjunct professor	Adjunct professor	NULL	NULL
10	TEE Richard	Impresa e Management	Ricercatore Junior (lett. a)	Ricercatore a)	SECS-P/10 - Organizzazione aziendale	13/B3 Organizzazione aziendale
11	TURNER Karynne Lenore	Impresa e Management	Ricercatore Junior (lett. a)	Ricercatore a)	SECS-P/08 - Economia e gestione delle imprese	13/B2 Economia e gestione delle imprese
12	VENEL Xavier	Economia e Finanza	Associato	Associato	SECS-S/06 - Metodi matematici dell'economia e d...	13/D4 Metodi matematici dell'economia e d...
13	VENEL Xavier	Economia e Finanza	Associate professor (Research)	Assistant Professor (Research)	SECS-S/06 - Metodi matematici dell'economia e d...	13/D4 Metodi matematici dell'economia e d...
14	VINUALES Jorge Enrique	Giurisprudenza	Ordinario	Ordinario	IUS/13 - Diritto internazionale	12/E1 Diritto internazionale

At this point, the opposite case to the previous one was analysed: professors in Cerca Università not present in Docenti Luiss.

In the left join, the most recent date on which they are present in Cerca Università was extracted for each professor: most of the 93 extracted records relate to professors who are no longer present in Luiss (Max_date in 2019 and earlier): I expected this situation knowing that the Excel file has a limited history. All that remains is to investigate the block of teachers with the max_date highlighted in red.

```
-- docenti presenti in CERCA_UNIVERSITA non presenti in DOCENTI_LUISS -> trovati 93 docenti distinct
select Cognome_Nome, MAX(date) as Max_date
from CERCA_UNIVERSITA LEFT JOIN DOCENTI_LUISS ON Cognome_Nome = Nominativo
WHERE Nominativo IS NULL
GROUP BY Cognome_Nome
order by 2 DESC
```

	Cognome Nome	Max date
1	BONTADINI Filippo	2023-02-28
2	FERNANDES DA SILVA RANCHORDAS Sofia Hina	2023-02-28
3	LAZAR Marc	2023-02-28
4	SABIA Rossella	2023-02-28
5	STOECKL Kristina	2023-02-28
6	TATI' Elisabetta	2023-02-28
7	VENEL Xavier Mathieu Raymond	2023-02-28
8	VILLARROEL ORDENES Francisco Javier	2023-02-28
9	VINUALES Jorge Enrique	2023-02-28
10	TURNER Karynne	2019-12-31
11	TEE Richard Liong Gie	2018-12-31
12	CANOFARI Paolo	2018-12-31
13	DI TARANTO Giuseppe	2016-12-31
14	IZZO Maria Federica	2016-12-31
15	MORLINO Leonardo	2016-12-31
16	RAGUSA Giuseppe	2016-12-31
17	DIMAC T...	2016-12-31

Also in this case there are the same differences in some names seen previously (for which 4 of the 9 are actually present in both data sources but written differently). The real names not present in Docenti Luiss file are BONTADINI, LAZAR, SABIA, STOECKL and TATI'.

For the teachers in common between the two tables, it was verified that the last photograph in the Cerca Università (as of 2/28/2023) coincided with Docenti Luiss. The “Dipartimento”, “Settore” and “Genere” fields are always aligned while for “Settore Concorsuale” there is a difference on the teacher IAIONE Fernando Christian (12/C1 <> 12/D1).

```

-- CHECK CONSISTENZA TRA DIPARTIMENTO E STRUTTURA_DI_AFFERENZA
select Nominativo, Struttura_di_afferenza, Dipartimento
from CERCA_UNIVERSITA JOIN DOCENTI_LUISS ON Cognome_Nome = Nominativo
WHERE DATE = '2023-02-28' AND Colore <> 'A' AND
      CERCA_UNIVERSITA.Struttura_di_afferenza <> DOCENTI_LUISS.Dipartimento

-- CHECK CONSISTENZA DEL SETTORE
select Nominativo, CERCA_UNIVERSITA.Settore, DOCENTI_LUISS.Settore
from CERCA_UNIVERSITA JOIN DOCENTI_LUISS ON Cognome_Nome = Nominativo
WHERE DATE = '2023-02-28' AND Colore <> 'A' AND
      CERCA_UNIVERSITA.Settore <> LEFT(DOCENTI_LUISS.Settore, CHARINDEX(' ', DOCENTI_LUISS.Settore)-1)

-- CHECK CONSISTENZA DEL SETTORE_CONCORSUALE
select Nominativo, CERCA_UNIVERSITA.Settore_concorsuale, DOCENTI_LUISS.Settore_concorsuale
from CERCA_UNIVERSITA JOIN DOCENTI_LUISS ON Cognome_Nome = Nominativo
WHERE DATE = '2023-02-28' AND Colore <> 'A' AND
      CERCA_UNIVERSITA.Settore_concorsuale <> LEFT(DOCENTI_LUISS.Settore_concorsuale, CHARINDEX(' ', DOCENTI_LUISS.Settore_concorsuale)-1)

-- CHECK CONSISTENZA GENERE
select Nominativo, CERCA_UNIVERSITA.Genere, DOCENTI_LUISS.Genere
from CERCA_UNIVERSITA JOIN DOCENTI_LUISS ON Cognome_Nome = Nominativo
WHERE DATE = '2023-02-28' AND Colore <> 'A' AND
      CERCA_UNIVERSITA.Genere <> DOCENTI_LUISS.Genere

```

100 %

Risultati Messaggi

	Nominativo	Struttura_di_afferenza	Dipartimento
	Nominativo	Settore	Settore
	Nominativo	Settore_concorsuale	Settore_concorsuale
1	IAIONE Fernando Christian	12/C1	12/D1 Diritto amministrativo
	Nominativo	Genere	Genere

The “Qualifica” field is not easily reconcilable (the qualifications of Assistant Professor and Lecturer have been excluded as they are not present in Cerca Università).

From the query here below it emerges that:

- On Cerca Università there are 7 values in the “Fascia” field
- On Docenti Luiss there are 9 values in the “Qualifica” field
- On Docenti Luiss there are 7 values in the “Qualifica_aggregata” field

At the end I decided to keep the coding of Cerca_Università considered as more reliable and discard the fields coming from Docenti_Luiss.


```

-- CHECK CONSISTENZA QUALIFICA - le info non sono facilmente riconciliabili: 7 valori su CERCA_UNIVERSITA mentre in DOCENTI_LUISS ne abbiamo 9 (o 7 aggregate)
select FASCIA, COUNT(*)
from CERCA_UNIVERSITA
WHERE DATE = '2023-02-28'
GROUP BY Fascia
ORDER BY 1

select Qualifica, COUNT(*)
from DOCENTI_LUISS
WHERE Colore<>'A' and Qualifica not in ('Assistant Professor (Research)', 'Lecturer')
GROUP BY Qualifica
ORDER BY 1

select Qualifica_aggregata, COUNT(*)
from DOCENTI_LUISS
WHERE Colore<>'A' and Qualifica not in ('Assistant Professor (Research)', 'Lecturer')
GROUP BY Qualifica_aggregata
ORDER BY 1

select Qualifica, Qualifica_aggregata, COUNT(*)

```

Risultati		100 %
1	Associato	26
2	Associato confermato	7
3	Ordinario	77
4	Ricercatore a.t.d. -t.defin. (art. 24 c.3-a L. 240/10)	1
5	Ricercatore a.t.d. -t.pieno (art. 24 c.3-a L. 240/10)	8
6	Ricercatore a.t.d. -t.pieno (art. 24 c.3-b L. 240/10)	3
7	Straordinario tempo determinato	6

} CERCA_UNIVERSITA

Risultati		100 %
1	Qualifica	(Nessun nome di colonna)
2	Adjunct professor	4
3	Associate Professor LBS	1
4	Associato	32
5	Full Professor LBS	1
6	Ordinario	77
7	Professor of practice	13
8	Professore straordinario ex art. 1 co. 12 L. 230/05	8
9	Ricercatore Junior lett. a)	10
10	Ricercatore Senior lett. b)	1

} DOCENTI_LUISS

Risultati		100 %
1	Qualifica_aggregata	(Nessun nome di colonna)
2	Adjunct professor	4
3	Associato	32
4	Full professor LBS	2
5	Ordinario	85
6	Professor of practice	13
7	Ricercatore a)	10
8	Ricercatore b)	1

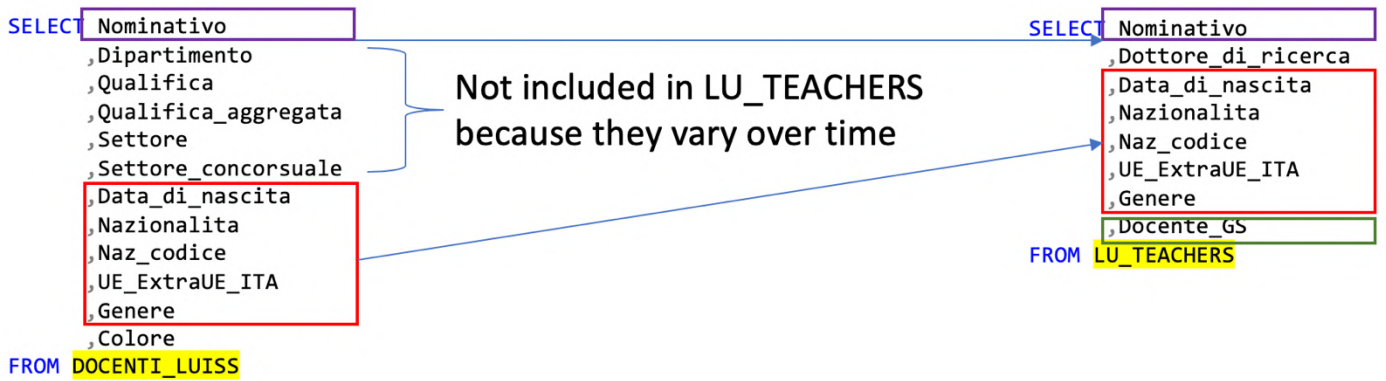
3.3 Data normalization

In order to obtain a data model on which to create reports with Power BI, the currently available information has been reorganized in a slightly different way with the aim of having in two separate tables the information on teachers that do not vary over time and those that potentially could.

3.3.1 First step: creation of the LU_TEACHERS table

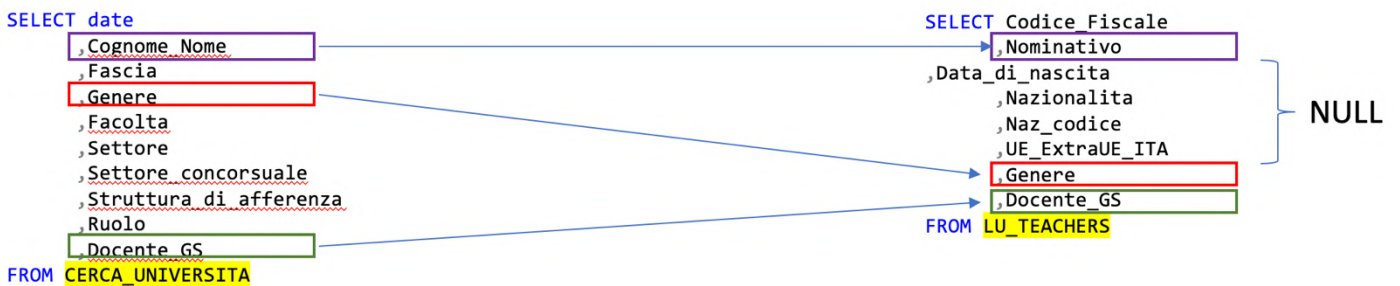
Starting from Docenti Luiss, was created a table called LU_TEACHERS with all the professors' data that do not vary over time (it is essentially a teacher masterfile) plus the “Docente GS” column (the name normalized through manual data cleaning in the case of double names/surnames) taken from Cerca Università.

With this first step, 153 teachers are transferred to LU_TEACHERS: I have not transferred the teachers of the BUSINESS SCHOOL as they are not present on the historical data from Cerca Università.



In this table were also added all the old teachers not present on Docenti Luiss by taking the data from Cerca Università. All unavailable fields (date of birth...) are set to NULL.

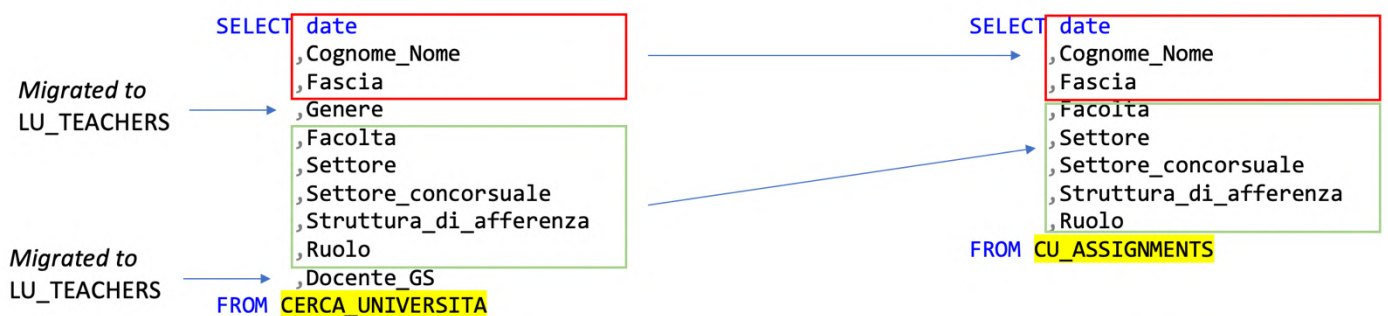
With this second step, 128 teachers are added (in total 282).



3.3.2 Second step: creation of the historical CU_ASSIGNMENTS table

The second step was to create an historical table called CU_ASSIGNMENTS starting from Cerca Università and eliminating the information (e.g. gender) that does not vary over time that had already been included in the teacher Masterfile (LU_TEACHERS).

There is therefore a 1 to N relationship between teachers and temporal positions (the same teacher at different dates can have different roles, different sectors...).



The columns “Settore” and “Settore Concorsuale” contain only the codes (e.g. IUS/04 or 13/A1).

For an easier understanding of this information I have created two decoding tables ST_SECTOR and ST_SECTOR_C having only 2 columns: code and description. In this way it is possible to see in the analytics the most meaningful descriptions (e.g. “Diritto Commerciale” instead of IUS/04 or “Economia Politica” instead of 13/A1).

In appendix 3 the SQL data model is reported. In summary there are 9 tables:

- The teacher Masterfile (LU_TEACHERS)
- The historical database of the staff in place at each year-end (CU_ASSIGNMENT)
- Five tables from Google Scholar (all starting with the prefix GS_)
- Two decoding tables (all starting with the prefix ST_)

To have a more structured and solid model I have also defined the foreign keys where necessary to ensure the referential integrity of the model (foreign keys are visible in the graphical data model as connection between tables).

In appendix 5 is attached the stored procedure that creates the tables LU_TEACHERS and CU_ASSIGNMENT starting from DOCENTI_LUISS and CERCA_UNIVERSITA. In the same script there are also some other small tasks (e.g. automated data cleaning of the inconsistencies discovered during the integration phase).

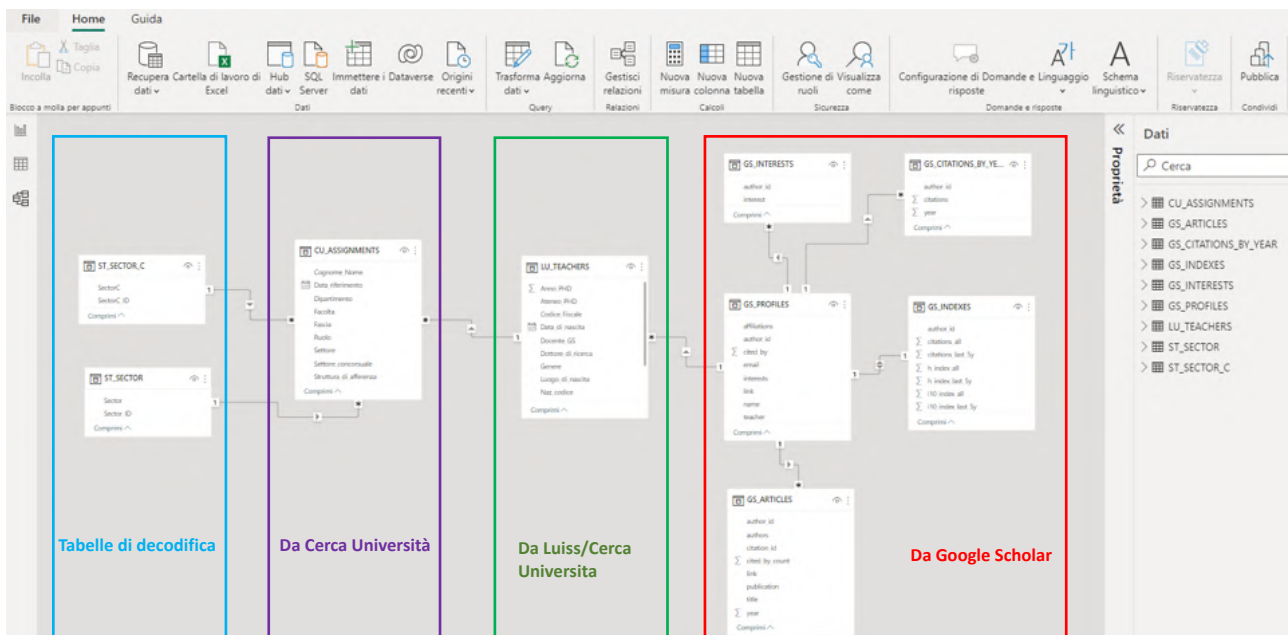
4. Data Analytics

The analytics have been developed using the Power BI Desktop application that is freely available on the Microsoft Website. This thesis has been the opportunity to learn Power BI, one of the most used tools for Data Visualization. While the usage and formatting options of the various graphical objects are quite easy to learn, the most difficult part remains the DAX language, that is necessary to create even the simplest measure.

4.1 Data import and Data model setup

Power BI allows to import data from a wide number of sources (csv, Excel, relational databases ...): in this case, having used Microsoft SQL Server, the integration is really native. It is sufficient to specify the server name and then choose which table has to be imported.

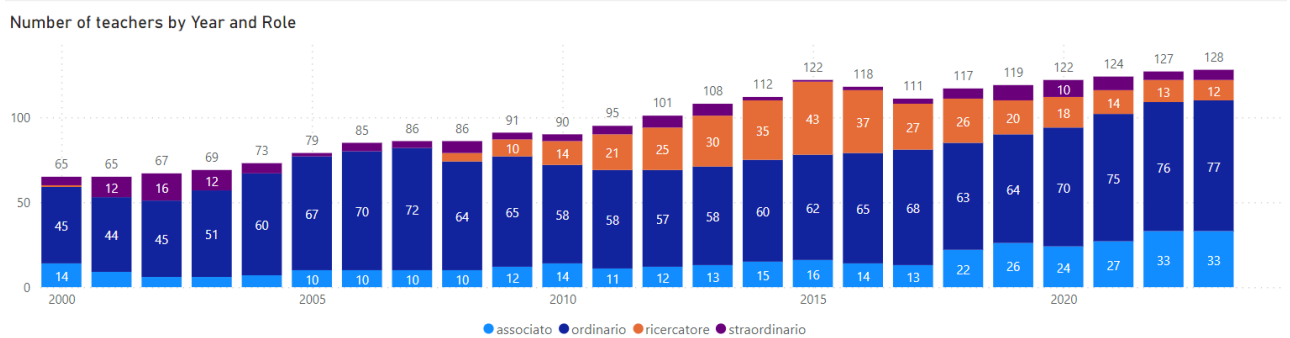
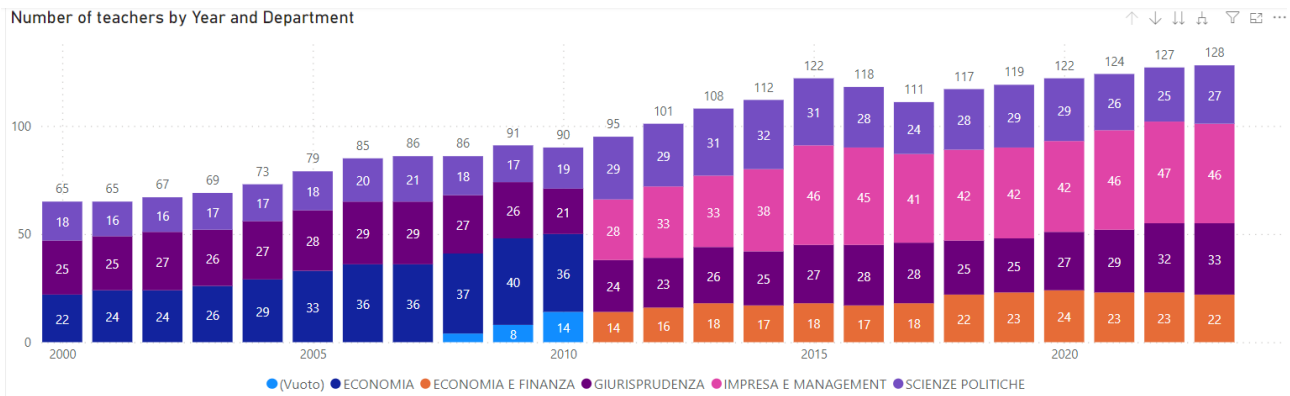
Power BI automatically creates the relationship between tables if foreign keys have been previously defined in SQL Server. Here below the data model after the importing of the 9 tables.



4.2 Data Visualization

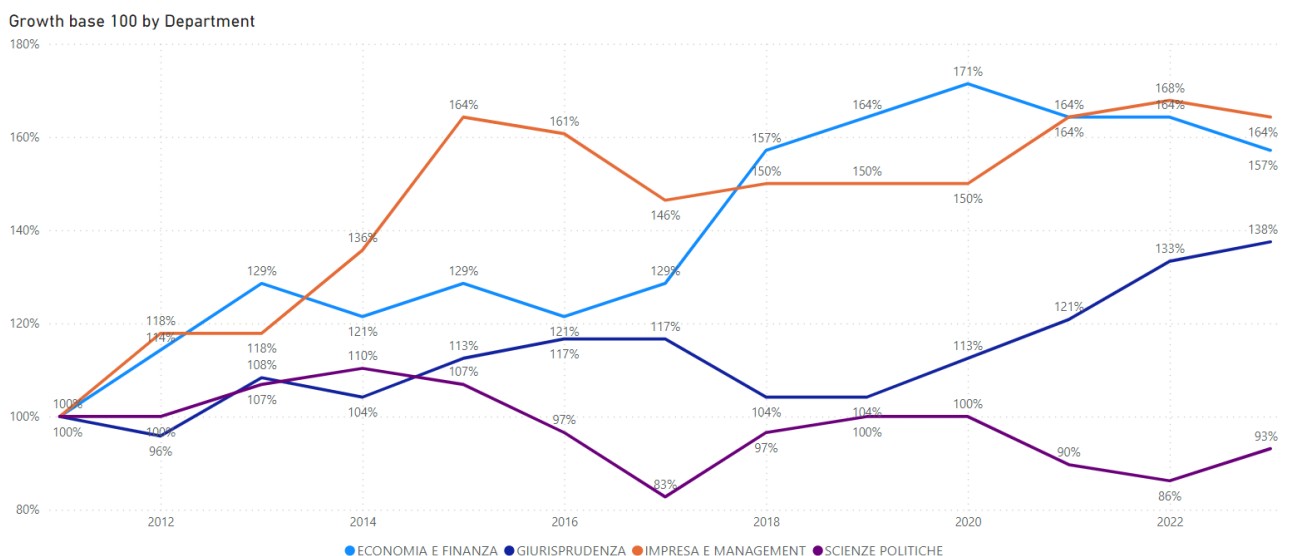
The first analysis I have developed is the trend of the teacher staff across the years: the first stacked bar chart shows the split by Department and the second by Role.

In the first stacked bar chart please note the discontinuity between 2010 and 2011 where “ECONOMIA” has been split into “ECONOMIA E FINANZA” and “IMPRESA E MANAGEMENT”.



The first bar chart does not allow to immediately realize which is the department that has grown most: in these cases a commonly used visualization is the Relative growth taking a specific year as baseline (sometimes known as “base 100 index”). In this case the year 2011 is the baseline.

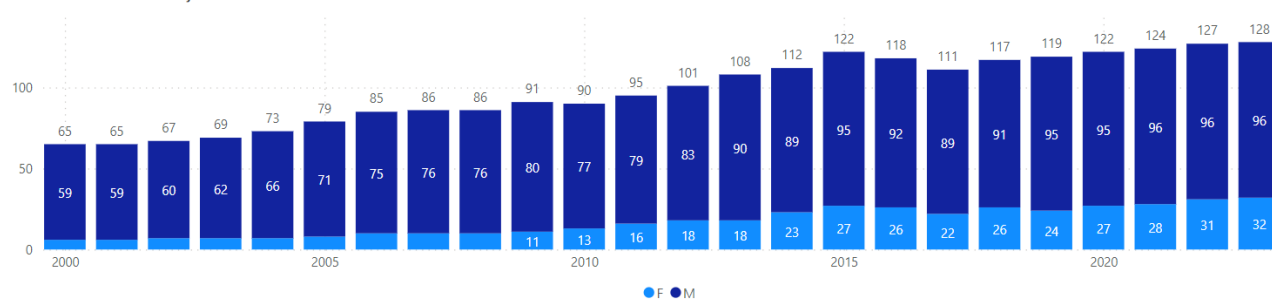
The second bar chart reflects the initial analysis in Python: the predominant category is the role of “ordinario” even in face of the trend of growth assumed by the number of professors over the years.



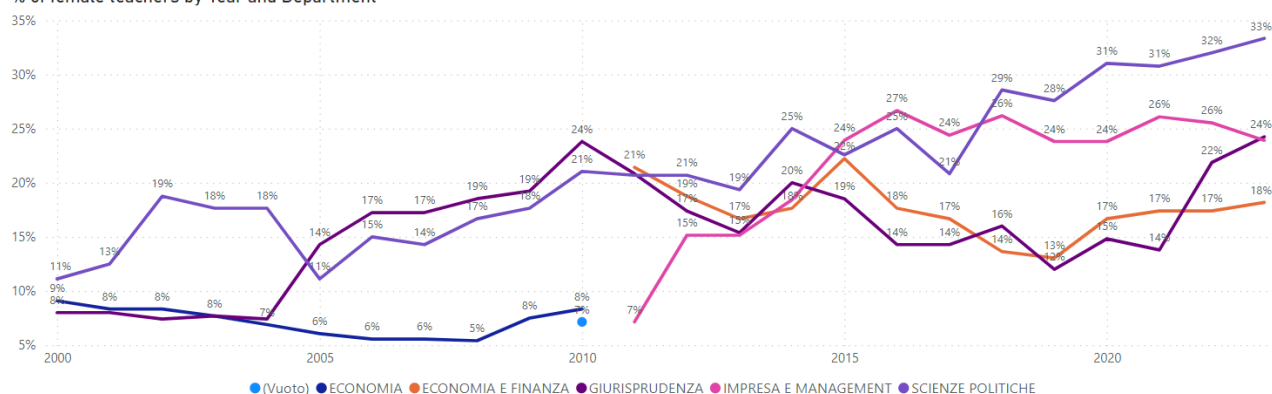
It is possible to notice that the teaching staff of SCIENZE POLITICHE has had a slight decrease while the number of the teachers in the other 3 departments has significantly increased in particular for the two economics departments (+57% and +64% respectively).

Another interesting analysis is the split across gender. The percentage of female teachers has steadily grown up to a 25% as of February 2023. Scienze politiche has the highest percentage at 33% while Economia e Finanza has the lowest at 18%

Number of teachers by Year and Gender

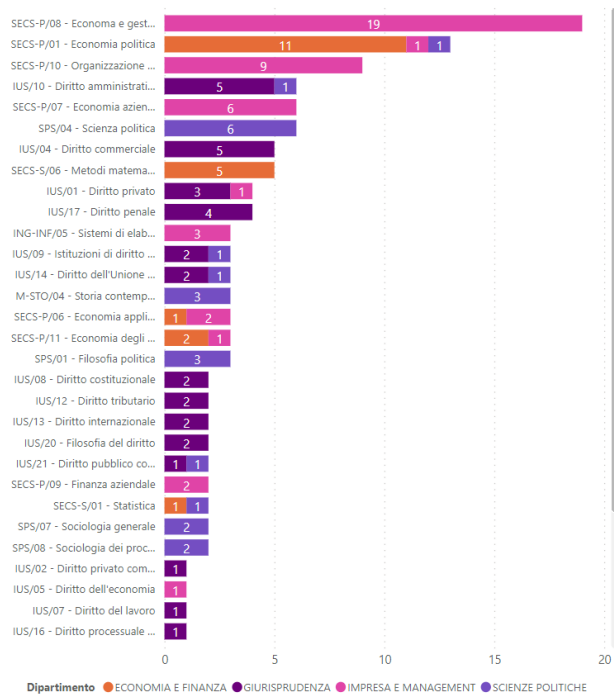


% of female teachers by Year and Department

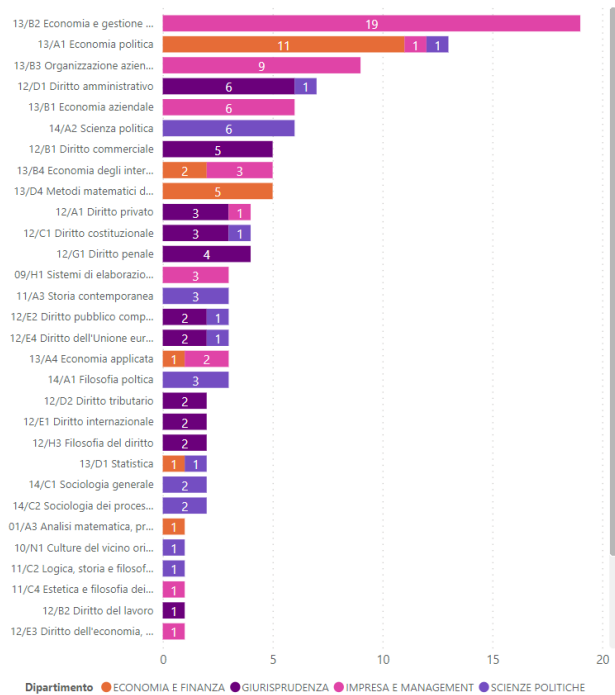


Even if not particularly interesting I have reported the split by “Settore” and “Settore Concorsuale”: here below the latest picture available (Feb 2023) is reported.

Distribution by Settore in 2023

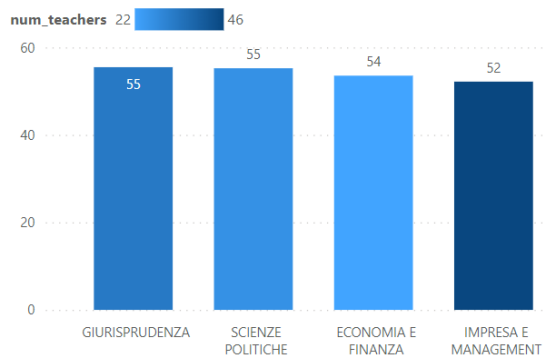


Distribution by Settore Concorsuale in 2023

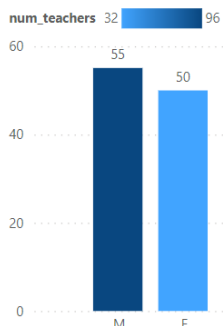


The information about the birth date, taken from Docenti Luiss file, was not available for many teachers: a manual cleaning has been performed searching the information on the Internet but in many cases the search was unsuccessful. Regarding the teachers in charge as of 2023 only on 7 cases (on 128) the age remains null. So even if is not possible to show the historical trend of the average age I could analyse the distribution across the most significant dimensions as of Feb 2023 in a quite reliable way. The barcharts below show the age distribution by Department, Gender and Role. I used the color to provide the additional information of how many teachers contribute to each bar (the darker, the more numerous): the color gradient with min and max is shown immediately below the graph title. Results are completely in line with the expectations (e.g. female are younger than male, average age increase progressively from ricercatore to straordinario ...).

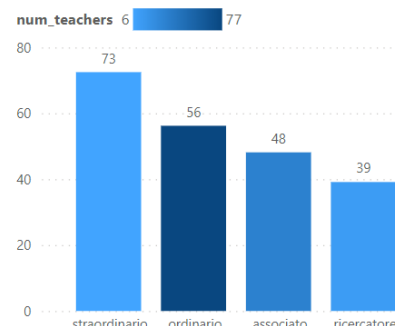
Avg age by Department



Avg age by Gender

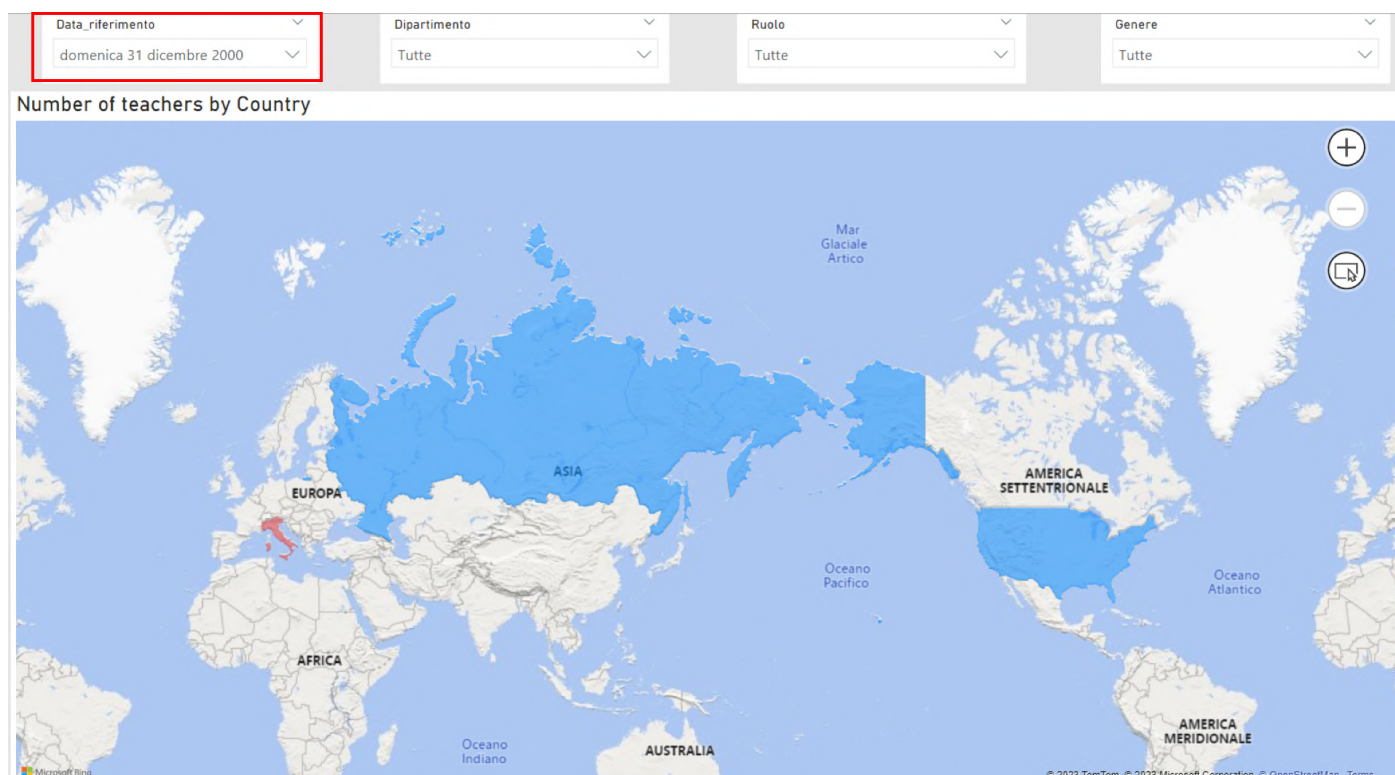


Avg age by Role

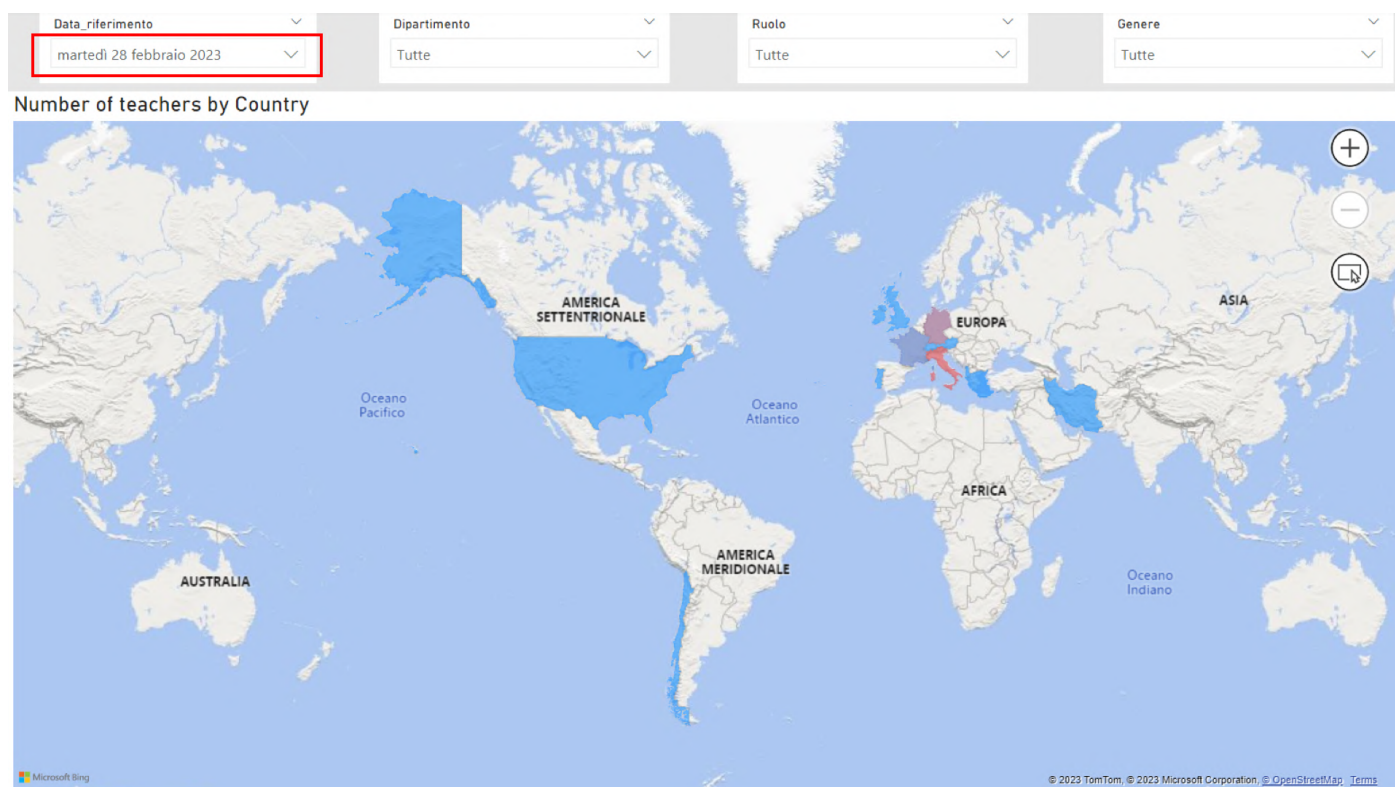


The Country of birth is another information that has been cleaned manually but differently from the date of birth I was able to clean the entire database of professors (I had to add a column with the country code).

Here below is the distribution by country as of 2000: at that time 63 teacher on 65 where Italian.

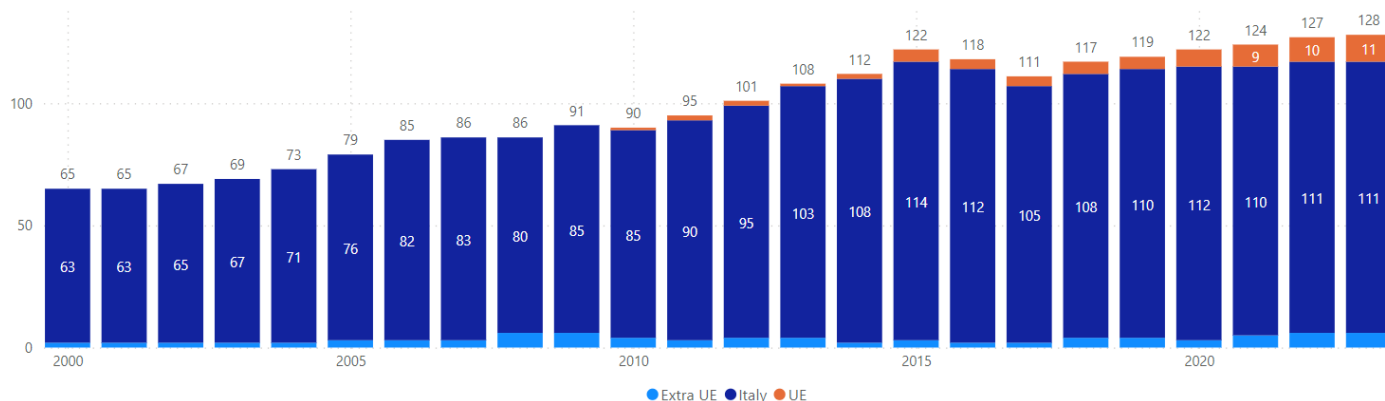


As of 2023 the country distribution shows a much more European footprint (of course Italy still dominant).



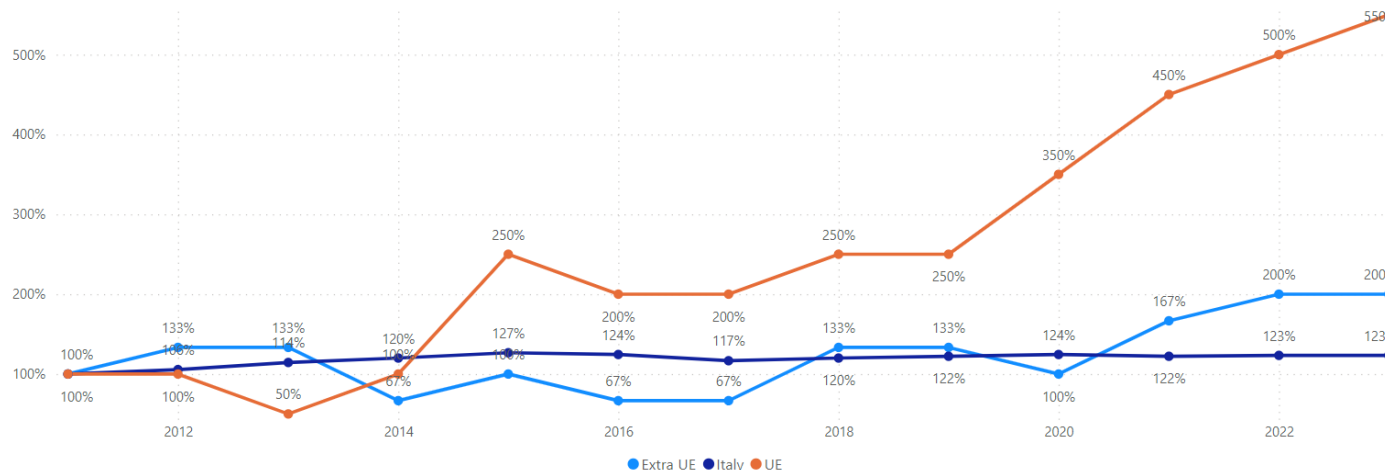
I could furtherly investigate the historical trend of the Nationality of the Luiss teaching staff, grouping the professors in 3 clusters: Italian, European Union and Extra-UE.

Number of teachers by Year and Nationality



Even if the majority of the professors are still Italian the number of teachers from the European Union is significantly increasing. Considering as baseline the year 2011, I could show the relative growth using the base-100 index. The Italian professors have increased of 123% in the last 13 years, while Extra-UE increased of 200% and UE increased of 550%.

Relative growth (index base 100) by UE-ExtraUE-ITA

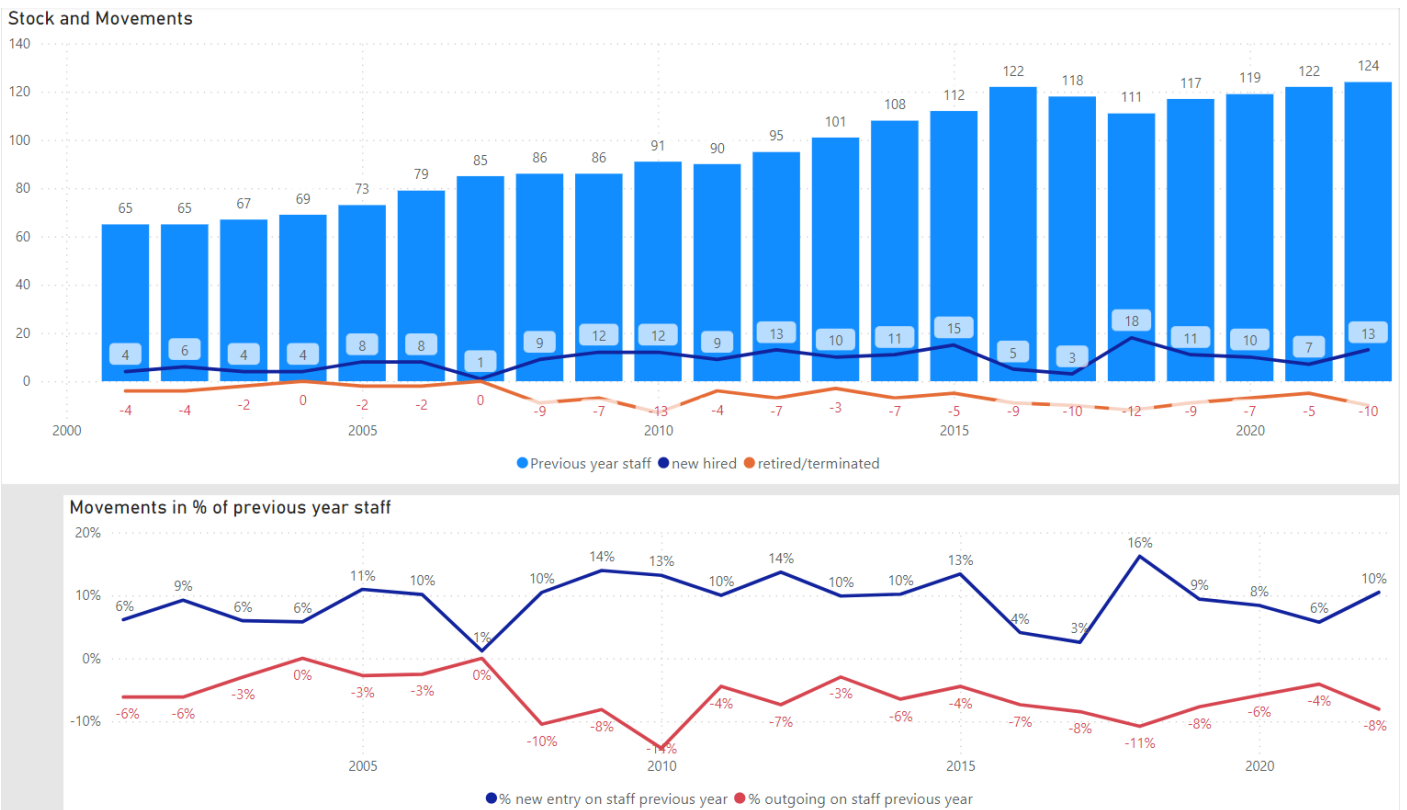


Another interesting insight related to the trend of the teaching staff consist in analyzing the movements between 2 consecutive years in terms of:

1. new hired teachers
2. retired/terminated teachers

In the following dashboard I have highlighted not only the absolute value of the 2 movements but also their percentage in comparison to the staff of the previous year.

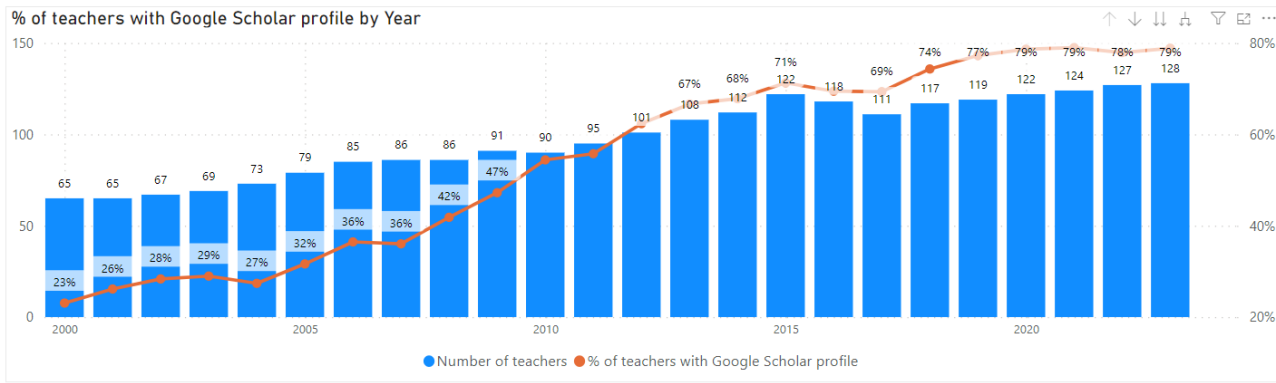
The analysis shows that every year there is a percentage of new entrants in average above 10% while the average of retirements/terminations is generally lower (around 6%) leading in the long term to the workforce increase seen before.



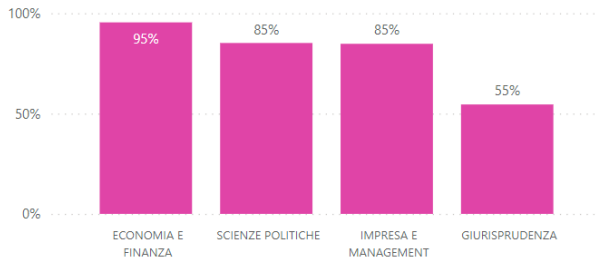
Having integrated the Google Scholar data the first analysis I wanted to perform is to check how many teachers have a profile on Google Scholar: it is possible to see here below that the percentage has increased over time, reaching a 79% in February 2023.

The two bar charts in purple show the breakdown by Department and Role as of February 2023.

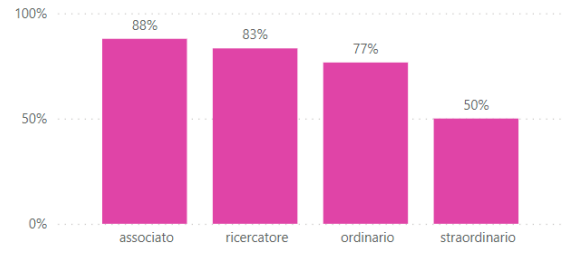
It is possible to notice that Giurisprudenza is by far the department where Google Scholar has the lower penetration (55%): the role “straordinario” has also the lowest penetration (50%) in his dimension.



% of teachers with Google Scholar profile in 2023 by Department

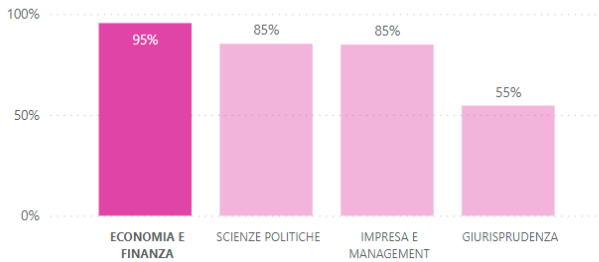


% of teachers with Google Scholar profile in 2023 by Role

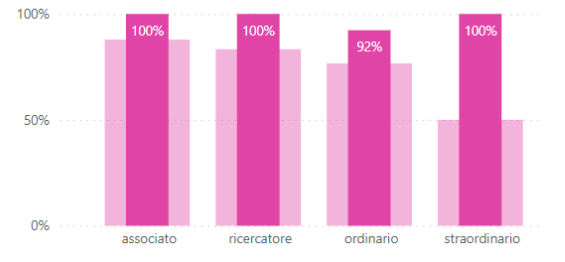


Power BI allows to easily navigate the data to perform additional investigations. Selecting Economia e Finanza it emerges that all roles stand at 100% with the exception of Ordinario at 92%. Selecting Giurisprudenza the percentages vary from 0% to 61%.

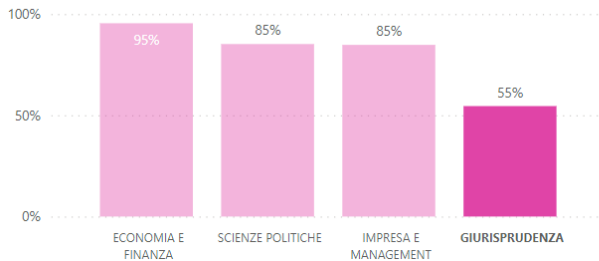
% of teachers with Google Scholar profile in 2023 by Department



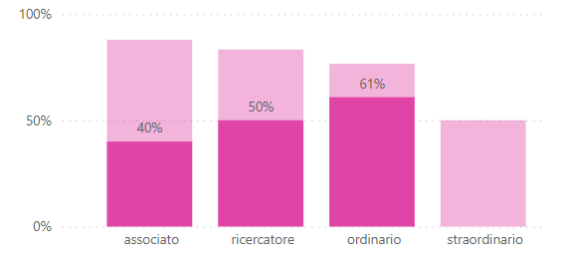
% of teachers with Google Scholar profile in 2023 by Role



% of teachers with Google Scholar profile in 2023 by Department

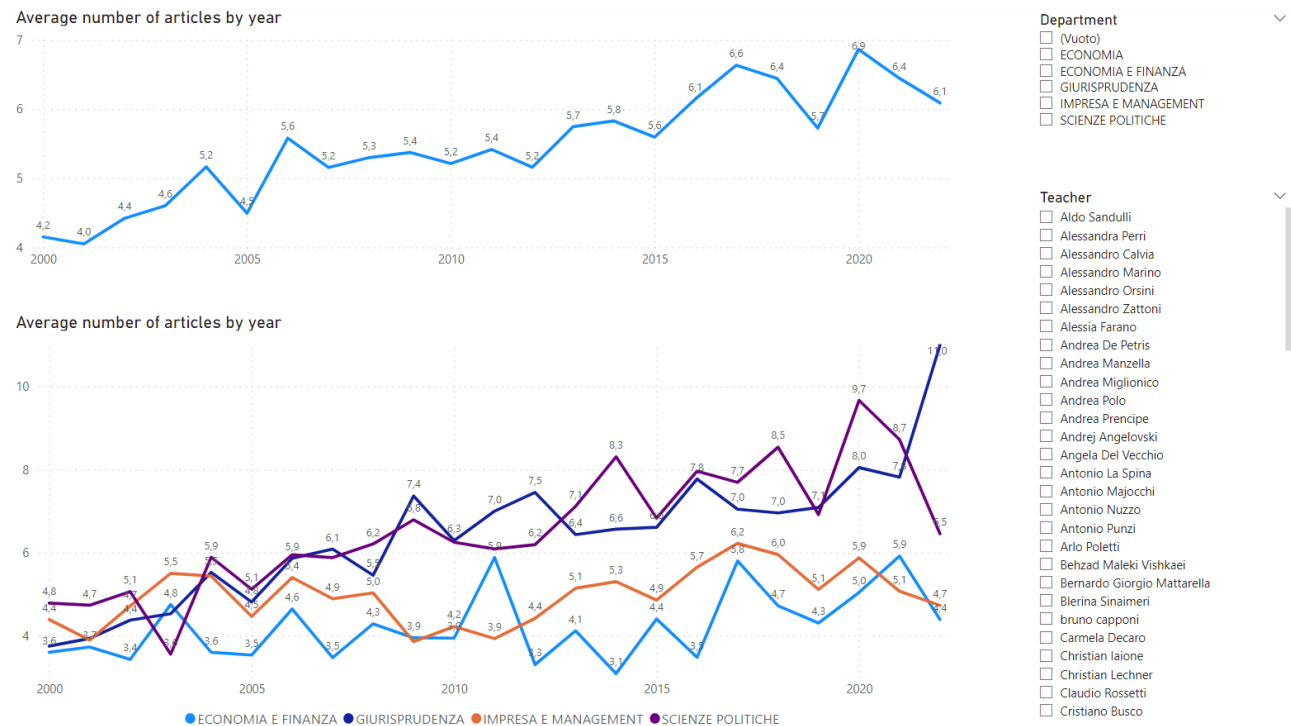


% of teachers with Google Scholar profile in 2023 by Role

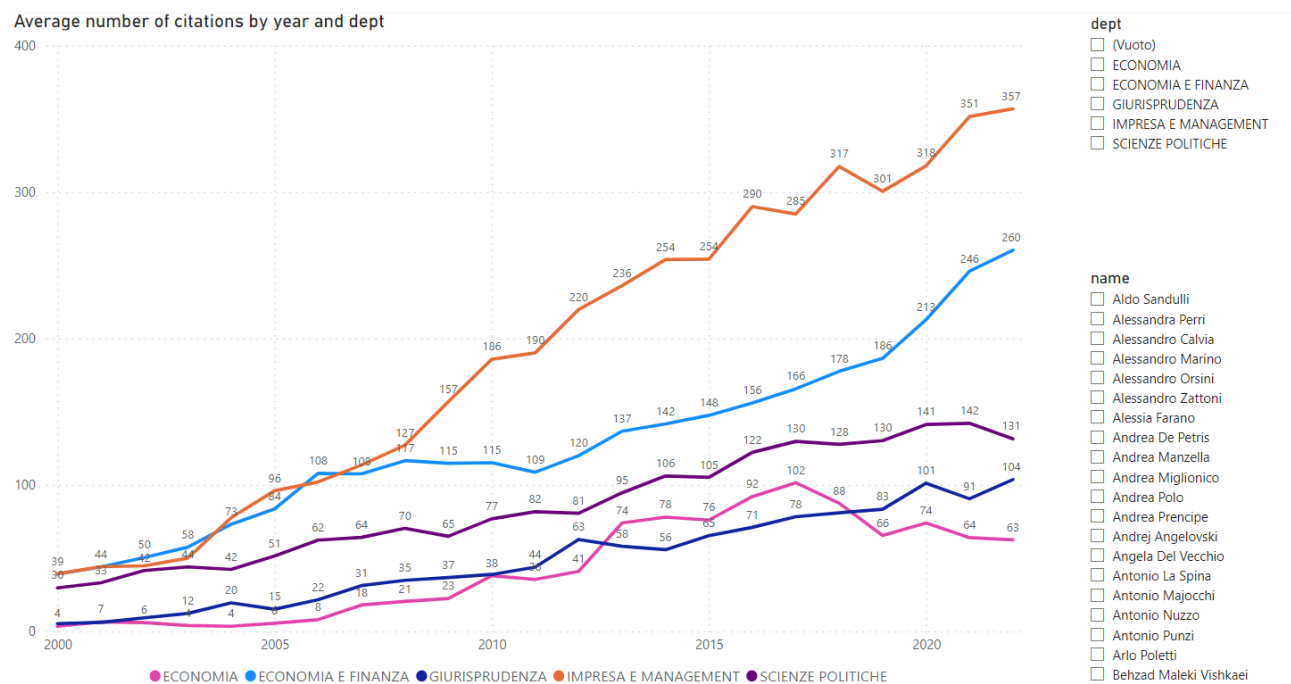


Google Scholar contains the information of the articles published, with the year of publishing. I have aggregated this information (of course for the only teachers having a profile in GS) to understand how many

articles in average per year are issued by a Luiss teacher. The second line chart shows the same information split at department level.

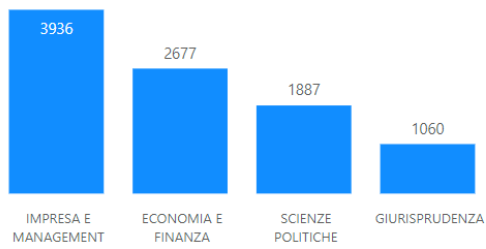


Another particularly interesting analysis shows the average number of citations.

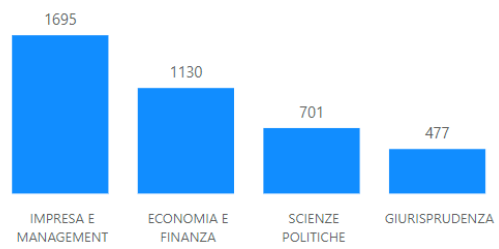


Having extracted from Google Scholar also some aggregated indexes I have created a dashboard showing the split by department as of February 2023. The graph on the right shows the index calculated on the last 5 years.

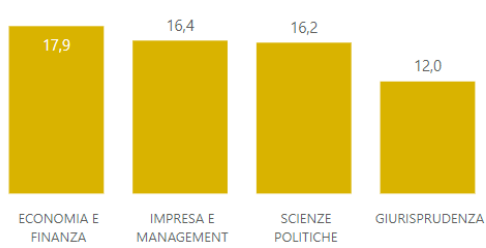
Average citations by Department



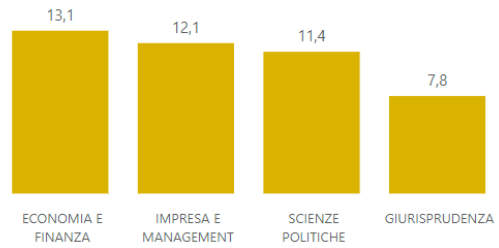
Average citations by Department - last 5 years



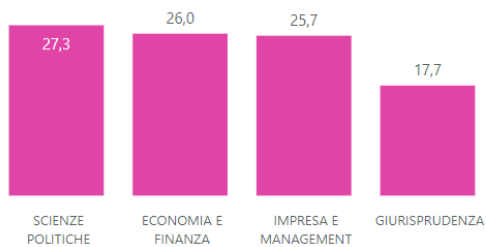
Average H index by Department



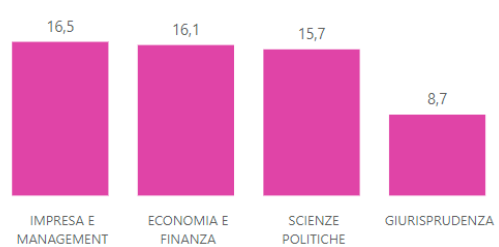
Average H index by Department - last 5 years



Average i10 index by Department



Average i10 index by Department - last 5 years



Having extracted from Google Scholar also the “interests” of each profile, I decided to create a visualization for this information.

I decided to use the Wordcloud chart (an additional tool of Power Bi that can be downloaded from the library of external components).

In this visualization all the interests are reported and the size of the text is proportional to number of times the interest is present in the database.

Conclusions and future directions

This thesis produced several interesting results.

From the data obtained from Cerca Università and Docenti Luiss the most outstanding are the following: an internal restructuring at the level of departments which, from 2011 onwards, were set to four, namely Economia e Finanza, Impresa e Management, Giurisprudenza, Scienze Politiche; a constant growth of professors over the years, especially in the two economics departments, accompanied by that of the number of women, who, however, are still much less than male professors. The number of foreign professors coming from the European Union is also increasing in the latest years.

From the information extracted from Google Scholar the thesis has shown a growth in the average number of publications in the departments of Giurisprudenza and Scienze Politiche which is accompanied, this time at a global level, by that of the average number of citations. Finally, concerning teachers' interests, the research areas in common to the four departments are "Strategy" and "Corporate Governance".

To the best of my knowledge this thesis is the first attempt to create some structured analytics about the teaching staff of the Luiss University integrating data from external data sources, such as Google Scholar and Cerca Università.

Several future development areas can be imagined starting from this work:

- The first enhancement is related to the possibility to collect additional information at teacher level: date of birth (currently not available for the teachers no longer in charge), date of first engagement at Luiss, date of resignation, date of first academic assignment (not necessarily at Luiss), presence of phd's, previous academic or professional experiences... Some of this information are certainly already available in the Luiss administration department while some other need to be discovered on the Internet. These new data could provide some additional insights on the teaching staff like the trend of the average age of the teachers, how long do they teach at Luiss, how long is their academic experience, which percentage has a phd ...
- A second interesting development would be to analyze the correlation between the number of enrolled students and the number of teachers. It implies to collect the historical data of enrolled students over the past years in the various departments. It will be interesting to understand to which extent the growth of students has been mirrored in the growth of teaching staff seen in this thesis.
- A third evolution is related to Cerca Università, the publicly available portal where the data of the Italian universities are stored. An automated script could extract the same historical information I got for Luiss, for a panel of universities to be considered as a benchmark. In this way it would be

possible to analyze which university has grown the most or to compare the percentage of female teachers. It could be also investigated if similar information is already available in a structured way for foreign universities, especially those considered as the most prestigious abroad.

- Another development is related to the research work of the Luiss professors. In this thesis I got the data from Google Scholar but other sites are also used to store and classify articles, papers ... (e.g. Reserchgate, SemanticScholar ...). Here the complexity will be not to duplicate works that are present on more than one site.
- Last but not least a graph model could be created to describe the cooperation between professors (co-authorship on articles, papers...). The creation of a “Luiss network” would allow to understand to what extent the Luiss teachers cooperate between themselves or with “external” authors. The graph theory allows to easily discover situations where two Luiss professors have never collaborated directly but both have cooperated with the same external author. All the graph metrics (e.g. distance, vertex degree...) would allow to better understand the Luiss Network: the number of years a professor has been teaching at Luiss should also be considered (a professor with a lower ageing is expected to be a more peripheral node in the Luiss network).

Bibliography

- El Refae, G. G. A., Kaba, A., & Eletter, S. (2021). The impact of demographic characteristics on academic performance: face-to-face learning versus distance learning implemented to prevent the spread of COVID-19. *The International Review of Research in Open and Distributed Learning*, 22(1), 91-110.
- Noruzi, A. (2016). Impact Factor, h-index, i10-index and i20-index of Webology. *Webology*, 13(1), 1-4.
- Sadeghi, A., Zaidatol, A. L. P., Habibah, E., & Foo, S. F. (2012). Demographic analysis on academic staff's job satisfaction in Malaysian research universities. *Pertanika Journal of Social Science and Humanities*, 20(SUPPL.), 1-20.
- Thomas, N. R., Poole, D. J., & Herbers, J. M. (2015). Gender in science and engineering faculties: Demographic inertia revisited. *PLoS One*, 10(10), e0139767.
- Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, 1-3.

Sitography

Biswal, A. (2023, April 13). What is Power BI?: Architecture, and Features Explained.
<https://www.simplilearn.com/tutorials/power-bi-tutorial/what-is-power-bi>

Cuelogic. (2017, January 25). The Levenshtein Algorithm
<https://www.cuelogic.com/blog/the-levenshtein-algorithm>

Mayank, M. (2019, February 2). String similarity — the basic know your algorithms guide!
<https://itnext.io/string-similarity-the-basic-know-your-algorithms-guide-3de3d7346227>

Seth, N. (2021, February 24). A Simple Guide to Metrics for Calculating String Similarity.
<https://www.analyticsvidhya.com/blog/2021/02/a-simple-guide-to-metrics-for-calculating-string-similarity/>

Appendix 1: the Python code

1. Consolidation of Cerca Università extractions in one dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from pandas.tseries.offsets import MonthEnd
import os

# os library gives a way of using operating system dependent functionalities,
# such as manipulating folder and file paths. We use this library to get all the Excel file names, i
```

```
cwd = os.path.abspath("/Users/anastasia/Library/Mobile Documents/com-apple-CloudDocs/Tesi_Luiss")
files = os.listdir(cwd)
# The variable cwd shows the path to the current working directory, and
# The variable files is a list of all the file names within the current working directory.
```

```
# first create an empty dataframe df for storing the consolidated data of the 23 excel files
df = pd.DataFrame()

# Then loop through all the files within the current working directory (only process the Excel files
# starts with "RICERCA" and ends with ".xlsx").
for file in files:
    if file.endswith(".xlsx") & file.startswith("RICERCA"):
        # get the date from the file name
        d = datetime.date(int(file[15:19]), int(file[20:22]), 1) + MonthEnd(0)
        # read the file
        temp = pd.read_excel(file)
        # add a column date
        temp["date"] = d
        df = df.append(temp, ignore_index=True)
df.head()
```

	Fascia	Cognome e Nome	Genere	Facoltà	S.S.D.	S.C.	Struttura di afferenza	date	Servizio prestato in altro ateneo
0	Ordinario	ANTISERI Dario	M	Scienze Politiche	M- FIL/02	NaN	SCIENZE STORICHE E SOCIO-POLITICHE	2001- 12-31	NaN
1	Ordinario	ARCELLI Mario	M	Economia	SECS- P/01	NaN	STUDI ECONOMICI	2001- 12-31	NaN
2	Ordinario	BALDASSARRE Antonio	M	Giurisprudenza	IUS/08	NaN	STUDI GIURIDICI	2001- 12-31	NaN
3	Ordinario	BALDINI Massimo	M	Scienze Politiche	M- FIL/05	NaN	SCIENZE STORICHE E SOCIO-POLITICHE	2001- 12-31	NaN
4	Ordinario	BORGIA Rosella	F	Giurisprudenza	IUS/01	NaN	GIURISPRUDENZA	2001- 12-31	NaN

```
df.shape
```

(2360, 9)

```
groupby_Fascia.sort_values(['date'], ascending=True, inplace=True)
```

```
groupby_Fascia
```

	Fascia	date	Cognome e Nome
73	Ricercatore	2000-12-31	1
49	Ordinario	2000-12-31	45
36	Associato non confermato	2000-12-31	2
12	Associato confermato	2000-12-31	12
118	Straordinario	2000-12-31	5
...
107	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	2023-02-28	8
11	Associato	2023-02-28	26
92	Ricercatore a t.d. - t.defin. (art. 24 c.3-a L...	2023-02-28	1
117	Ricercatore a t.d. - t.pieno (art. 24 c.3-b L....	2023-02-28	3
146	Straordinario tempo determinato	2023-02-28	6

147 rows x 3 columns

```
# We can use pd.pivot_table to transform a long format dataframe to a wide format dataframe.
```

```
pivot = pd.pivot_table(data=groupby_Fascia, index=['date'], columns=['Fascia'], values='Cognome e No
```

```
pivot
```

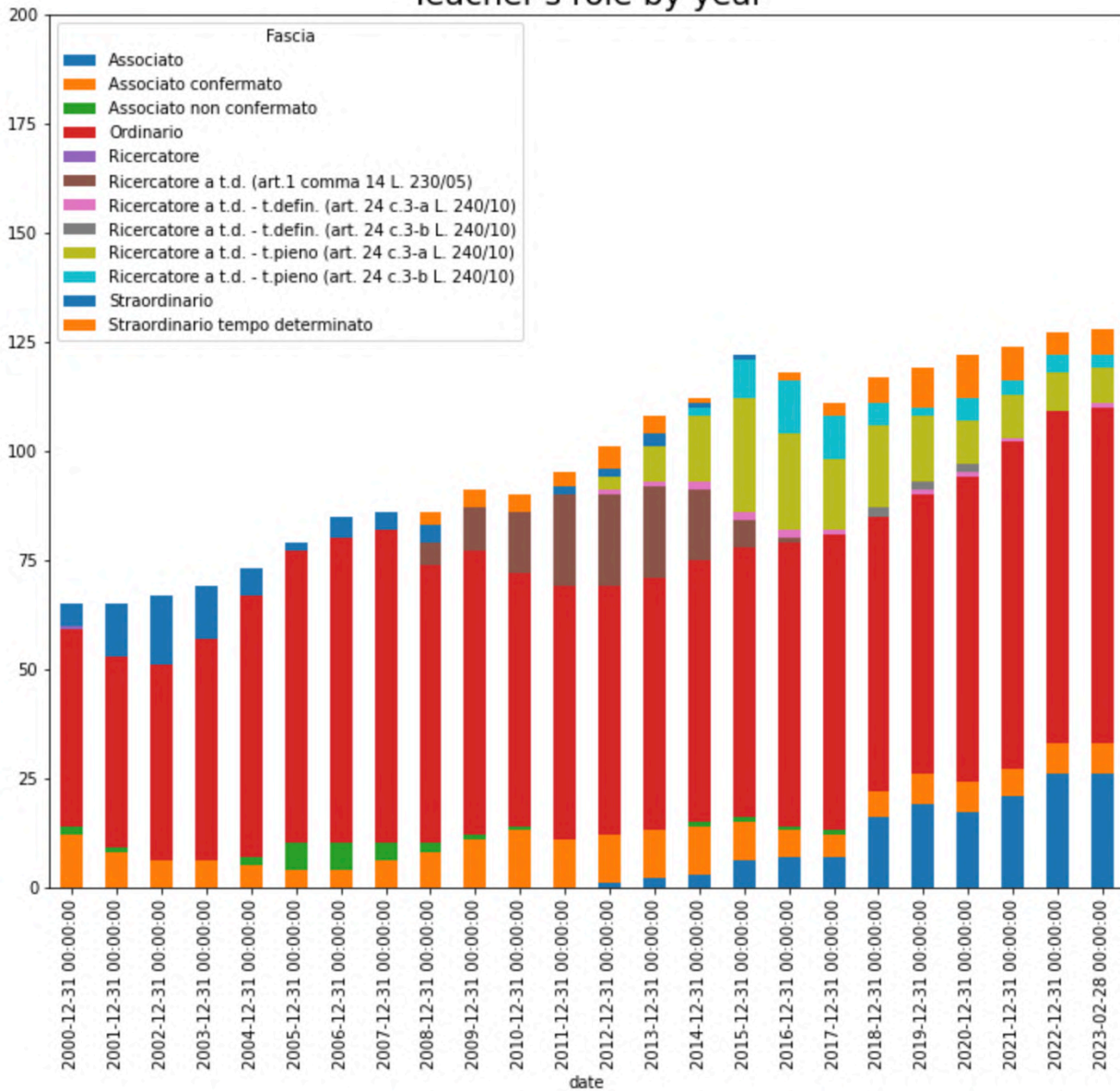
Fascia	Associato	Associato confermato	Associato non confermato	Ordinario	Ricercatore	Ricercatore a t.d. (art.1 comma 14 L. 230/05)	Ricercatore a t.d. - t.defin. (art. 24 c.3-a L. 240/10)	Ricercatore a t.d. - t.defin. (art. 24 c.3-b L. 240/10)	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L. 240/10)	Rice
2000-12-31	NaN	12.0	2.0	45.0	1.0	NaN	NaN	NaN	NaN	
2001-12-31	NaN	8.0	1.0	44.0	NaN	NaN	NaN	NaN	NaN	
2002-12-31	NaN	6.0	NaN	45.0	NaN	NaN	NaN	NaN	NaN	

```
# Let's create a stacked barplot.
```

```
ax = pivot.plot.bar(stacked=True, figsize=(12,10))
ax.set_title("Teacher's role by year", fontsize=20)
ax.set_ylim(0,200)
```

```
(0.0, 200.0)
```


Teacher's role by year



```
groupby_Struttura = df.groupby(["Struttura di afferenza", "date"], as_index=False)["Cognome e Nome"].count()
```

```
groupby_Struttura
```

	Struttura di afferenza	date	Cognome e Nome
0	DIRITTO INTERNAZIONALE E DELL'UNIONE EUROPEA	2006-12-31	1
1	DIRITTO INTERNAZIONALE E DELL'UNIONE EUROPEA	2007-12-31	1
2	DIRITTO INTERNAZIONALE E DELL'UNIONE EUROPEA	2008-12-31	1
3	DIRITTO INTERNAZIONALE E DELL'UNIONE EUROPEA	2009-12-31	1

```
groupby_Struttura.sort_values(['date'],ascending=True, inplace=True)
```

```
groupby_Struttura
```

	Struttura di afferenza	date	Cognome e Nome
133	SCIENZE POLITICHE	2000-12-31	4
204	STUDI STORICO POLITICI	2000-12-31	3
194	STUDI SOCIOLOGICI	2000-12-31	2
122	SCIENZE GIURIDICHE	2000-12-31	11
36	FACOLTA' DI GIURISPRUDENZA	2000-12-31	1
...

```
pivot = pd.pivot_table(data=groupby_Struttura, index=['date'], columns=['Struttura di afferenza'], v
```

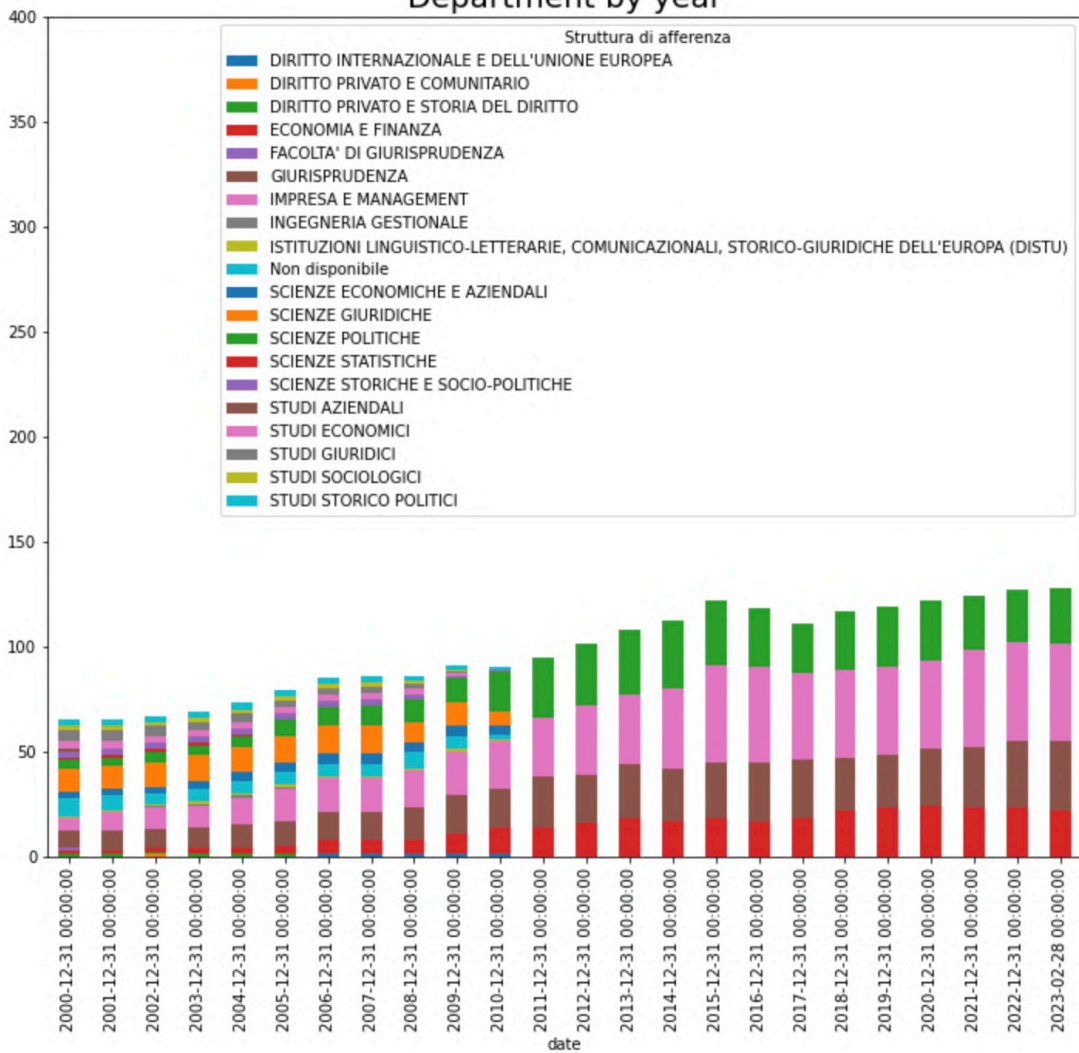
```
pivot
```

Struttura di afferenza	DIRITTO INTERNAZIONALE E DELL'UNIONE EUROPEA	DIRITTO PRIVATO E COMUNITARIO	DIRITTO PRIVATO E STORIA DEL DIRITTO	ECONOMIA E FINANZA	FACOLTA' DI GIURISPRUDENZA	GIURISPRUDENZA	IMPRESA E MANAGEMENT C
date							
2000-12-31	NaN	NaN	1.0	2.0	1.0	8.0	6.0
2001-12-31	NaN	NaN	1.0	2.0	NaN	9.0	9.0
2002-12-31	NaN	1.0	1.0	2.0	NaN	9.0	10.0
2003-12-31	NaN	NaN	1.0	3.0	NaN	10.0	10.0

```
out[146]: ax = pivot.plot.bar(stacked=True, figsize=(12,10))
ax.set_title("Department by year", fontsize=20)
ax.set_ylim(0,400)
```

```
(0.0, 400.0)
```

Department by year



```
groupby_Facoltà = df.groupby(["Facoltà", "date"], as_index=False) ["Cognome e Nome"].count()
```

Note that from 2011 there are only 4 main Departments: "Impresa e Management", "Scienze politiche", "Giurisprudenza", "Economia e Finanza"

```
groupby_Facoltà
```

	Facoltà	date	Cognome e Nome
0	Economia	2000-12-31	22
1	Economia	2001-12-31	24
2	Economia	2002-12-31	24

```
groupby_Facoltà.sort_values(['date'], ascending=True, inplace=True)
```

```
groupby_Facoltà
```

	Facoltà	date	Cognome e Nome
0	Economia	2000-12-31	22
26	Scienze Politiche	2000-12-31	18
15	Giurisprudenza	2000-12-31	25

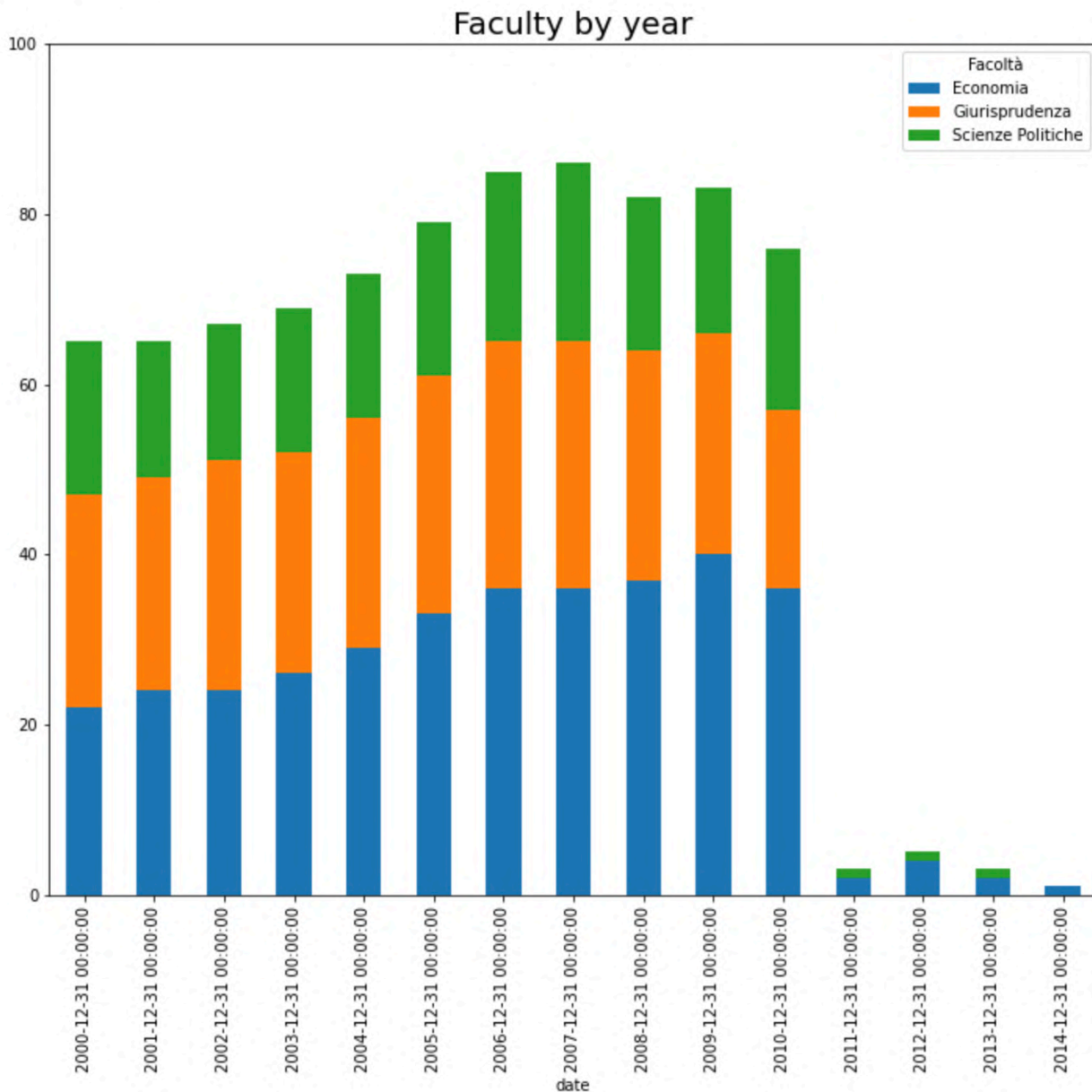
```
pivot = pd.pivot_table(data=groupby_Facoltà, index=['date'], columns=['Facoltà'], values='Cognome e I
```

```
pivot
3]:
```

Facoltà	Economia	Giurisprudenza	Scienze Politiche
date			
2000-12-31	22.0	25.0	18.0
2001-12-31	24.0	25.0	16.0
2002-12-31	24.0	27.0	16.0

```
ax = pivot.plot.bar(stacked=True, figsize=(12,10))
ax.set_title("Faculty by year", fontsize=20)
ax.set_ylim(0,100)
```

```
(0.0, 100.0)
```



```
conditions = [
    (df["Fascia"].str[0:3] == "Ric"),
    (df["Fascia"].str[0:3] == "Ord"),
    (df["Fascia"].str[0:3] == "Ass"),
    (df["Fascia"].str[0:3] == "Str")
]
```

```
values = ["ricercatore", "ordinario", "associato", "straordinario"]
```

```
df['Ruolo'] = np.select(conditions, values)
```

```
df.head()
```

	Fascia	Cognome e Nome	Genere	Facoltà	S.S.D.	S.C.	Struttura di afferenza	date	Servizio prestato in altro ateneo	Ruolo
0	Ordinario	ANTISERI Dario	M	Scienze Politiche	M- FIL/02	NaN	SCIENZE STORICHE E SOCIO-POLITICHE	2001- 12-31	NaN	ordinario
1	Ordinario	ARCELLI Mario	M	Economia	SECS- P/01	NaN	STUDI ECONOMICI	2001- 12-31	NaN	ordinario
2	Ordinario	BALDASSARRE Antonio	M	Giurisprudenza	IUS/08	NaN	STUDI GIURIDICI	2001- 12-31	NaN	ordinario

```
groupby_Ruolo = df.groupby(["Ruolo", "date"], as_index=False)["Cognome e Nome"].count()
```

```
groupby_Ruolo
```

	Ruolo	date	Cognome e Nome
0	associato	2000-12-31	14
1	associato	2001-12-31	9
2	associato	2002-12-31	6
3	associato	2003-12-31	6
4	associato	2004-12-31	7
...

```
groupby_Ruolo.sort_values(['date'], ascending=True, inplace=True)
```

```
groupby_Ruolo
```

	Ruolo	date	Cognome e Nome
0	associato	2000-12-31	14
24	ordinario	2000-12-31	45
48	ricercatore	2000-12-31	1
65	straordinario	2000-12-31	5
25	ordinario	2001-12-31	44
...

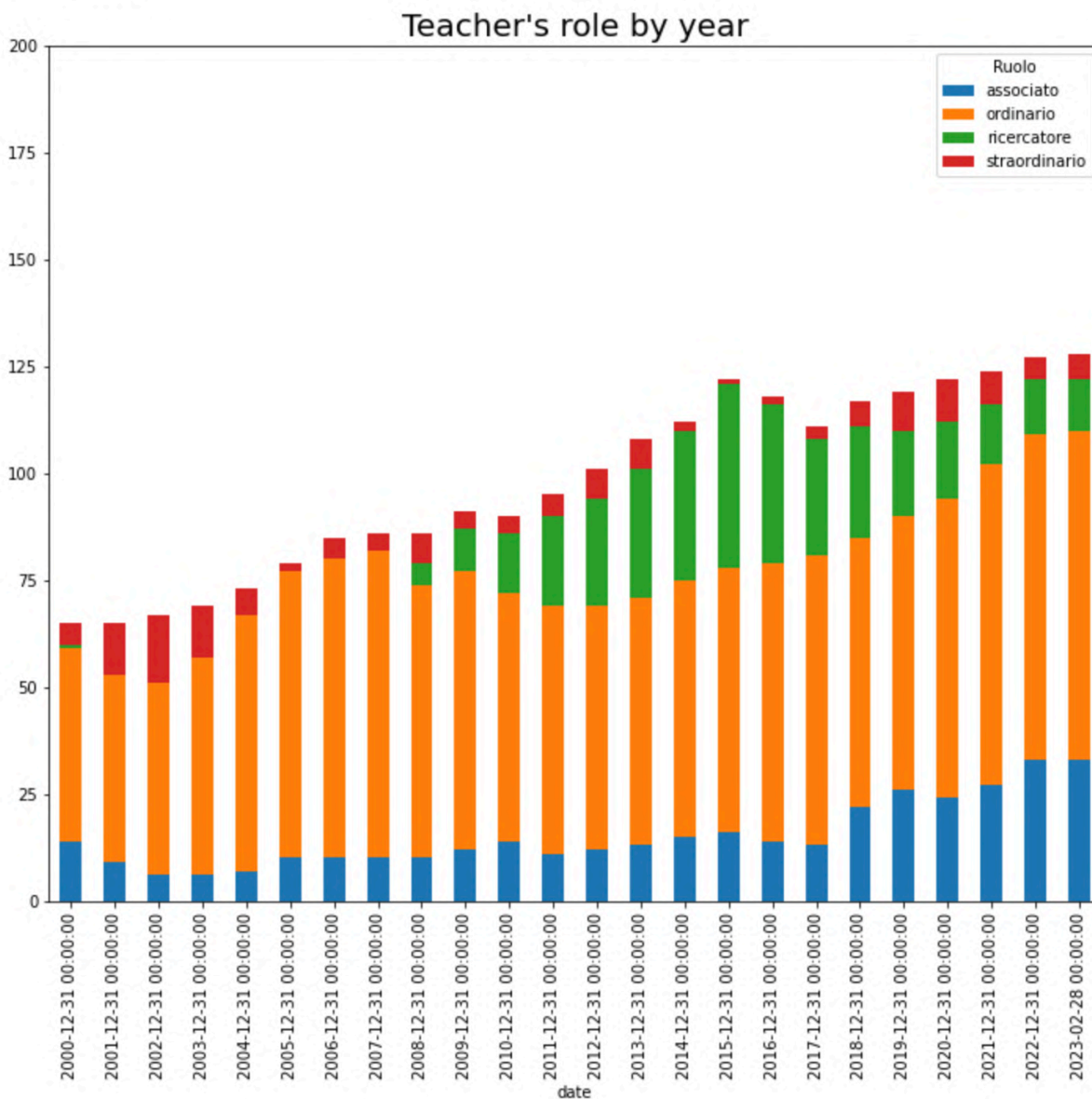

```
pivot = pd.pivot_table(data=groupby_Ruolo, index=['date'], columns=['Ruolo'], values='Cognome e Nome')
```

```
pivot
```

	associato	ordinario	ricercatore	straordinario
2000-12-31	14.0	45.0	1.0	5.0
2001-12-31	9.0	44.0	NaN	12.0
2002-12-31	6.0	45.0	NaN	16.0
2003-12-31	6.0	51.0	NaN	12.0

```
ax = pivot.plot.bar(stacked=True, figsize=(12,10))
ax.set_title("Teacher's role by year", fontsize=20)
ax.set_ylim(0,200)
```

```
(0.0, 200.0)
```



```
df.to_excel('Consolidato.xlsx', index=False)
```


2. Google Scholar Profile extraction

```
import pandas as pd
from serpapi import GoogleSearch
import time
from urllib.parse import urlsplit, parse_qs
import os, json
```

part 2: read consolidato.xlsx and extract profile data

```
docenti=pd.read_excel("/Users/anastasia/Library/Mobile Documents/com~apple~CloudDocs/Tesi_Luiss/consolidato.xlsx")
```

```
docenti
```

	Fascia	Cognome e Nome	Genere	Facoltà	S.S.D.	S.C.	Struttura di afferenza	date	Servizio prestatato in altro ateneo	Ruolo	Docente
0	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	ADIGUZEL Feray	F	NaN	SECS-P/08	13/B2	IMPRESA E MANAGEMENT	2018-12-31	NaN	ricercatore	ADIGUZEL Feray
1	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	ADIGUZEL Feray	F	NaN	SECS-P/08	13/B2	IMPRESA E MANAGEMENT	2019-12-31	NaN	ricercatore	ADIGUZEL Feray
2	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	ADIGUZEL Feray	F	NaN	SECS-P/08	13/B2	IMPRESA E MANAGEMENT	2016-12-31	NaN	ricercatore	ADIGUZEL Feray
3	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	ADIGUZEL Feray	F	NaN	SECS-P/08	13/B2	IMPRESA E MANAGEMENT	2017-12-31	NaN	ricercatore	ADIGUZEL Feray
4	Ricercatore a t.d. - t.pieno (art. 24 c.3-a L....	ADIGUZEL Feray	F	NaN	SECS-P/08	13/B2	IMPRESA E MANAGEMENT	2015-12-31	NaN	ricercatore	ADIGUZEL Feray
...

```
# let's check how many distinct teachers are present
vect_teachers = docenti['Docente'].unique()
```

```
len(vect_teachers)
```

```
255
```

```
for teach in vect_teachers:
    print(teach)
```

```
ADIGUZEL Feray
ANGELOVSKI Andrej
ANTISERI Dario
ANTONELLI Vincenzo
```

```
# let's define a function that extract the surname from a string knowing that the surname is uppercase
def get_surname(strg):
    l1 = strg.split()
    ret = ''
    for word in l1:
        if word.isupper():
            ret = ret + word + ' '
    return ret.rstrip()
```

```
all_teachers = []
```

```
# cycle on the list of distinct teachers and for each one get the profile from Google Scholar
for teach in vect_teachers:
    params = {
        "engine": "google_scholar_profiles",
        "mauthors": teach,
        "api_key": "a0d3108de435939f356c624aea7cb2170970944d8188dede582e2c83e4072375"
    }
    time.sleep(4)
    search = GoogleSearch(params)
    results = search.get_dict()
    if 'profiles' in results.keys():
        profiles = results["profiles"]
        for prof in profiles:
            # check that the surname we looked for is included in the 'name' field
            if get_surname(teach) in prof['name'].upper():
                prof['thumbnail'] = teach
                all_teachers.append(prof)
```

```
len(all_teachers)
```

```
204
```

```
# convert the extracted dictionary to a dataframe  
df = pd.DataFrame(all_teachers)
```

```
df
```

	name	link	serpapi_link	author_id	affiliations	email	cited_by	interests
0	Feray Adiguzel	https://scholar.google.com/citations?hl=en&use...	https://serpapi.com/search.json?author_id=m8K0...	m8K0YewAAAAJ	Professor of Marketing	Verified email at ntu.ac.uk	721.0	['title': 'Marketing', 'serpapi_link': 'https...']
1	Andrej Angelovski	https://scholar.google.com/citations?hl=en&use...	https://serpapi.com/search.json?author_id=DAnE...	DAnEQQkAAAAJ	Assistant Professor, Middlesex University	Verified email at mdx.ac.uk	62.0	['title': 'Experimental Economics', 'serpapi_...']
2	Vincenzo Antonelli	https://scholar.google.com/citations?hl=en&use...	https://serpapi.com/search.json?author_id=juQc...	juQcP1sAAAAJ	Research Associate, Polito	Verified email at polito.it	34.0	['title': 'Algebraic Geometry', 'serpapi_link...']

```
# drop a column that is not useful  
df = df.drop("serpapi_link", axis=1)  
# add a column that will contain the interests separated by comma  
df['interests_new'] = ''
```

```
df
```

	name	link	author_id	affiliations	email	cited_by	interests	thumbnail	interes
0	Feray Adiguzel	https://scholar.google.com/citations?hl=en&use...	m8K0YewAAAAJ	Professor of Marketing	Verified email at ntu.ac.uk	721.0	['title': 'Marketing', 'serpapi_link': 'https...']	ADIGUZEL Feray	
1	Andrej Angelovski	https://scholar.google.com/citations?hl=en&use...	DAnEQQkAAAAJ	Assistant Professor, Middlesex University	Verified email at mdx.ac.uk	62.0	['title': 'Experimental Economics', 'serpapi_...']	ANGELOVSKI Andrej	

```
# assign the new column interest_new  
s1 = []  
for index, row in df.iterrows():  
    s1 = df['interests'][index]  
    # attenzione: nel caso la colonna contiene nan il tipo di s1 diventa float (non più str)  
    if isinstance(s1, list):  
        df['interests_new'][index] = (','.join(str(x['title']) for x in s1))
```

```
# drop the old column interest being replaced by the new column interest_new  
df = df.drop("interests", axis=1)
```

```
df.rename(columns={'thumbnail': 'teacher'}, inplace=True)
```

```
df['name'] = df['name'].str.replace(',', ' ')
```

```
df
```

	name	link	author_id	affiliations	email	cited_by	teacher	interests_nu
0	Feray Adiguzel	https://scholar.google.com/citations?hl=en&use...	m8K0YewAAAAJ	Professor of Marketing	Verified email at ntu.ac.uk	721.0	ADIGUZEL Feray	Marketing, Statist
1	Andrej Angelovski	https://scholar.google.com/citations?hl=en&use...	DAnEQQkAAAAJ	Assistant Professor, Middlesex University	Verified email at mdx.ac.uk	62.0	ANGELOVSKI Andrej	Experimen Economics, Behavior Economi

```
df.to_csv('google_scholar_01.csv', index=False, header=True)
```

3. Google Scholar Citations, Metrics and Articles extraction

```
# let's start from the excel file google_scholar_02.xlsx that has been manually cleaned excluding all the
# profiles (homonimies) not related to Luiss Teachers

teach_verified = pd.read_excel("/Users/anastasia/Library/Mobile Documents/com~apple~CloudDocs/Tesi_Luiss/google_scho

# citations_by_year è la lista che conterrà tutte le citazioni per anno
all_citations_by_year = []

# indexes è la lista che conterrà gli indicatori di sintesi per chiascun teacher
all_indexes = []

# all_articles è la lista che conterrà gli articoli per chiascun teacher
all_articles = []

for author_id in teach_verified["author_id"]:

    params = {
        'engine' : "google_scholar_author",
        'author_id' : author_id,
        'api_key' : "a0d3108de435939f356c624aea7cb2170970944d8188dede582e2c83e4072375",
        'hl' : 'en',
        'start' : '0',
        'num' : '100'
    }

    search = GoogleSearch(params)

    first_search = True

    while True:

        results = search.get_dict()

        if first_search:
            cited_by = results["cited_by"]
            for year in cited_by["graph"]:
                y1 = year
                y1['author_id'] = author_id
                all_citations_by_year.append(y1)

            ind = {}
            ind['author_id'] = author_id
            ind['citations_all'] = cited_by["table"][0]['citations']['all']
            ind['citations_last_5y'] = cited_by["table"][0]['citations']['since_2018']
            ind['h_index_all'] = cited_by["table"][1]['h_index']['all']
            ind['h_index_last_5y'] = cited_by["table"][1]['h_index']['since_2018']
            ind['i10_index_all'] = cited_by["table"][2]['i10_index']['all']
            ind['i10_index_last_5y'] = cited_by["table"][2]['i10_index']['since_2018']
            all_indexes.append(ind)

            first_search = False

        for article in results['articles']:
            title = article.get('title')
            link = article.get('link')
            authors = article.get('authors')
            publication = article.get('publication')
            citation_id = article.get('citation_id')
            year = article.get('year')
            cited_by_count = article.get('cited_by').get('value')

            all_articles.append({
                'author_id': author_id,
                'title': title,
                'link': link,
                'authors': authors,
                'publication': publication,
                'citation_id': citation_id,
                'cited_by_count': cited_by_count,
                'year': year
            })
    })
```

```
# check if the next page is present in 'serpapi_pagination' dict key
if 'next' in results.get('serpapi_pagination', []):
    time.sleep(4)
    # split URL in parts as a dict() and update 'search' variable to a new page
    search.params_dict.update(dict(parse_qs(urlsplit(results['serpapi_pagination']['next']).query)))
else:
    break
```

```
df = pd.DataFrame(all_citations_by_year)
df.to_csv('all_citations_by_year.csv', index=False, header=True)
```

```
df = pd.DataFrame(all_indexes)
df.to_csv('all_indexes.csv', index=False, header=True)
```

```
df = pd.DataFrame(all_articles)
df.to_csv('all_articles.csv', index=False, header=True)
```

```
# let's create a dataset where interests are splitted in single rows
# (initially they are on the same string comma separated)
df_new_1 = df[['author_id', 'interests_new']]
```

```
df_new_1
```

	author_id	interests_new
0	dHjU_3cAAAAJ	Administrative Law,European Law,Public Law,Dir...
1	9rkhNooAAAAJ	International Business,Innovation,Strategy and...
2	V1CIdrQAAAAJ	Stochastic optimal control,Stochastic filterin...
3	9r9kP7YAAAAJ	Strategy
4	sUFtCR0AAAAJ	Qualitative sociology,ethnography,terrorism,po...
...

```
df_new_2 = df_new_1.set_index(['author_id']).apply(lambda x: x.str.split(',').explode()).reset_index()
```

```
df_new_2.head(20)
```

	author_id	interests_new
0	dHjU_3cAAAAJ	Administrative Law
1	dHjU_3cAAAAJ	European Law
2	dHjU_3cAAAAJ	Public Law
3	dHjU_3cAAAAJ	Diritto amministrativo
4	dHjU_3cAAAAJ	Diritto pubblico
5	9rkhNooAAAAJ	International Business
6	9rkhNooAAAAJ	Innovation

```
# Drop all rows having at least one null value
df_new_3 = df_new_2.dropna()
```

```
df_new_3.to_excel('Interests.xlsx')
```

Appendix 2: the SQL database definition

```
USE [Docenti_Luiss]
GO
/***** Object: Table [dbo].[CERCA_UNIVERSITA]    Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CERCA_UNIVERSITA](
    [date] [date] NOT NULL,
    [Cognome_Nome] [nvarchar](64) NOT NULL,
    [Fascia] [nvarchar](64) NOT NULL,
    [Genere] [char](1) NOT NULL,
    [Facolta] [nvarchar](32) NULL,
    [Settore] [nvarchar](16) NULL,
    [Settore_concorsuale] [nvarchar](16) NULL,
    [Struttura_di_afferenza] [nvarchar](255) NULL,
    [Ruolo] [nvarchar](16) NOT NULL,
    [Docente_GS] [nvarchar](64) NOT NULL,
    CONSTRAINT [PK_CERCA_UNIVERSITA] PRIMARY KEY CLUSTERED
(
    [date] ASC,
    [Cognome_Nome] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
/***** Object: Table [dbo].[CU_ASSIGNMENTS]    Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CU_ASSIGNMENTS](
    [Data_riferimento] [date] NOT NULL,
    [Cognome_Nome] [nvarchar](64) NOT NULL,
    [Fascia] [nvarchar](64) NOT NULL,
    [Facolta] [nvarchar](32) NULL,
    [Settore] [nvarchar](16) NULL,
    [Settore_concorsuale] [nvarchar](16) NULL,
    [Struttura_di_afferenza] [nvarchar](128) NULL,
    [Ruolo] [nvarchar](16) NOT NULL,
    [Dipartimento] [nvarchar](128) NULL,
    CONSTRAINT [PK_CU_ASSIGNMENTS] PRIMARY KEY CLUSTERED
(
    [Data_riferimento] ASC,
    [Cognome_Nome] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
/***** Object: Table [dbo].[DOCENTI_LUISS]    Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DOCENTI_LUISS](
    [Nominativo] [nvarchar](255) NULL,
    [Dipartimento] [nvarchar](255) NULL,
    [Qualifica] [nvarchar](255) NULL,
    [Qualifica_aggregata] [nvarchar](255) NULL,
```

```

[Settore] [nvarchar](255) NULL,
[Settore_concorsuale] [nvarchar](255) NULL,
[Data_di_nascita] [date] NULL,
[Nazionalita] [nvarchar](255) NULL,
[Naz_codice] [char](3) NULL,
[UE_ExtraUE_ITA] [nvarchar](255) NULL,
[Genere] [char](1) NOT NULL,
[Colore] [char](1) NOT NULL
) ON [PRIMARY]
GO

```

/* Object: Table [dbo].[GS_ARTICLES] Script Date: 22/05/2023 10:02:51 */

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GS_ARTICLES](
    [author_id] [varchar](16) NOT NULL,
    [title] [nvarchar](255) NOT NULL,
    [link] [nvarchar](255) NOT NULL,
    [authors] [nvarchar](255) NULL,
    [publication] [nvarchar](255) NULL,
    [citation_id] [nvarchar](255) NOT NULL,
    [cited_by_count] [int] NULL,
    [year] [int] NULL,
    CONSTRAINT [PK_GS_ARTICLES] PRIMARY KEY CLUSTERED
(
    [citation_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```

/* Object: Table [dbo].[GS_CITATIONS_BY_YEAR] Script Date: 22/05/2023 10:02:51 */

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GS_CITATIONS_BY_YEAR](
    [author_id] [varchar](16) NOT NULL,
    [year] [int] NOT NULL,
    [citations] [int] NULL,
    CONSTRAINT [PK_GS_CITATIONS_BY_YEAR] PRIMARY KEY CLUSTERED
(
    [author_id] ASC,
    [year] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```

/* Object: Table [dbo].[GS_INDEXES] Script Date: 22/05/2023 10:02:51 */

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GS_INDEXES](
    [author_id] [varchar](16) NOT NULL,
    [citations_all] [int] NULL,
    [citations_last_5y] [int] NULL,
    [h_index_all] [int] NULL,
    [h_index_last_5y] [int] NULL,
    [i10_index_all] [int] NULL,
    [i10_index_last_5y] [int] NULL,
    CONSTRAINT [PK_GS_INDEXES] PRIMARY KEY CLUSTERED
(
    [author_id] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
/****** Object: Table [dbo].[GS_INTERESTS] Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GS_INTERESTS](
    [author_id] [varchar](16) NOT NULL,
    [interest] [nvarchar](255) NOT NULL,
    CONSTRAINT [PK_GS_INTERESTS] PRIMARY KEY CLUSTERED
(
    [author_id] ASC,
    [interest] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
/****** Object: Table [dbo].[GS_PROFILES] Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GS_PROFILES](
    [name] [nvarchar](255) NOT NULL,
    [link] [nvarchar](255) NOT NULL,
    [author_id] [varchar](16) NOT NULL,
    [affiliations] [nvarchar](255) NULL,
    [email] [nvarchar](255) NULL,
    [cited_by] [int] NULL,
    [teacher] [nvarchar](128) NOT NULL,
    [interests] [nvarchar](255) NULL,
    [dept] [nvarchar](128) NULL,
    CONSTRAINT [PK_GS_PROFILES] PRIMARY KEY CLUSTERED
(
    [author_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
/****** Object: Table [dbo].[LU_TEACHERS] Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[LU_TEACHERS](
    [Nominativo] [nvarchar](64) NOT NULL,
    [Data_di_nascita] [date] NULL,
    [Nazionalita] [nvarchar](255) NULL,
    [Naz_codice] [char](3) NULL,
    [UE_ExtraUE_ITA] [nvarchar](255) NULL,
    [Genere] [char](1) NOT NULL,
    [Docente_GS] [nvarchar](128) NOT NULL,
    CONSTRAINT [PK_LU_TEACHERS] PRIMARY KEY CLUSTERED
(
    [Nominativo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```



```

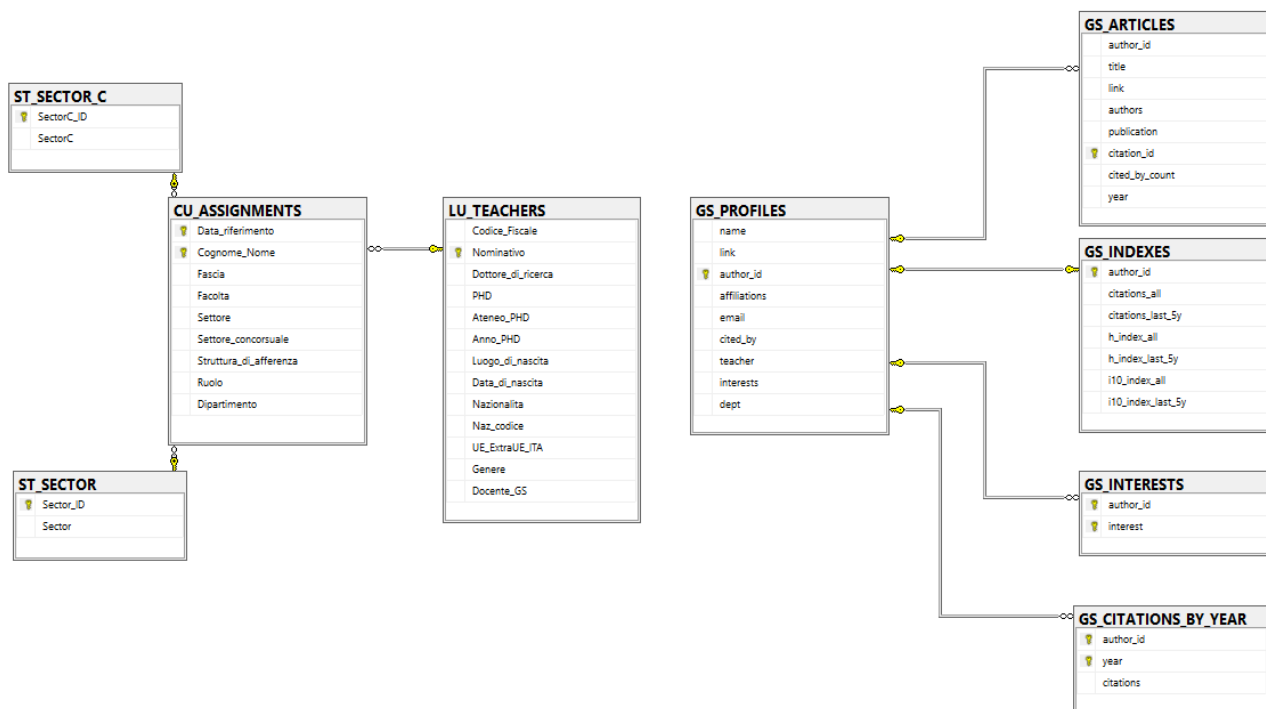
/***** Object: Table [dbo].[ST_SECTOR]    Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ST_SECTOR](
    [Sector_ID] [nvarchar](16) NOT NULL,
    [Sector] [nvarchar](255) NOT NULL,
    CONSTRAINT [PK_ST_SECTOR] PRIMARY KEY CLUSTERED
(
    [Sector_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[ST_SECTOR_C]    Script Date: 22/05/2023 10:02:51 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ST_SECTOR_C](
    [SectorC_ID] [nvarchar](16) NOT NULL,
    [SectorC] [nvarchar](255) NOT NULL,
    CONSTRAINT [PK_ST_SECTOR_C] PRIMARY KEY CLUSTERED
(
    [SectorC_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] WITH CHECK ADD CONSTRAINT [FK_CU_ASSIGNMENTS_LU_TEACHERS] FOREIGN
KEY([Cognome_Nome])
REFERENCES [dbo].[LU_TEACHERS] ([Nominativo])
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] CHECK CONSTRAINT [FK_CU_ASSIGNMENTS_LU_TEACHERS]
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] WITH CHECK ADD CONSTRAINT [FK_CU_ASSIGNMENTS_ST_SECTOR] FOREIGN
KEY([Settore])
REFERENCES [dbo].[ST_SECTOR] ([Sector_ID])
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] CHECK CONSTRAINT [FK_CU_ASSIGNMENTS_ST_SECTOR]
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] WITH CHECK ADD CONSTRAINT [FK_CU_ASSIGNMENTS_ST_SECTOR_C] FOREIGN
KEY([Settore_concorsuale])
REFERENCES [dbo].[ST_SECTOR_C] ([SectorC_ID])
GO
ALTER TABLE [dbo].[CU_ASSIGNMENTS] CHECK CONSTRAINT [FK_CU_ASSIGNMENTS_ST_SECTOR_C]
GO
ALTER TABLE [dbo].[GS_ARTICLES] WITH CHECK ADD CONSTRAINT [FK_GS_ARTICLES_GS_PROFILES] FOREIGN
KEY([author_id])
REFERENCES [dbo].[GS_PROFILES] ([author_id])
GO
ALTER TABLE [dbo].[GS_ARTICLES] CHECK CONSTRAINT [FK_GS_ARTICLES_GS_PROFILES]
GO
ALTER TABLE [dbo].[GS_CITATIONS_BY_YEAR] WITH CHECK ADD CONSTRAINT
[FK_GS_CITATIONS_BY_YEAR_GS_PROFILES] FOREIGN KEY([author_id])
REFERENCES [dbo].[GS_PROFILES] ([author_id])
GO
ALTER TABLE [dbo].[GS_CITATIONS_BY_YEAR] CHECK CONSTRAINT [FK_GS_CITATIONS_BY_YEAR_GS_PROFILES]
GO
ALTER TABLE [dbo].[GS_INDEXES] WITH CHECK ADD CONSTRAINT [FK_GS_INDEXES_GS_PROFILES] FOREIGN
KEY([author_id])
REFERENCES [dbo].[GS_PROFILES] ([author_id])
GO
ALTER TABLE [dbo].[GS_INDEXES] CHECK CONSTRAINT [FK_GS_INDEXES_GS_PROFILES]
GO

```

```
ALTER TABLE [dbo].[GS_INTERESTS] WITH CHECK ADD CONSTRAINT [FK_GS_INTERESTS_GS_PROFILES] FOREIGN
KEY([author_id])
REFERENCES [dbo].[GS_PROFILES] ([author_id])
GO
ALTER TABLE [dbo].[GS_INTERESTS] CHECK CONSTRAINT [FK_GS_INTERESTS_GS_PROFILES]
GO
```

Appendix 3: the SQL data model



Appendix 4: the Levenshtein Distance implementation as SQL function

```
USE [Docenti_Luiss]
GO
/***** Object: UserDefinedFunction [dbo].[Levenshtein]    Script Date: 22/05/2023 09:39:46 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Computes and returns the Levenshtein edit distance between two strings, i.e. the
-- number of insertion, deletion, and substitution edits required to transform one
-- string to the other, or NULL if @max is exceeded. Comparisons use the case-
-- sensitivity configured in SQL Server (case-insensitive by default).
--
-- Based on Sten Hjelmqvist's "Fast, memory efficient" algorithm, described
-- at http://www.codeproject.com/Articles/13525/Fast-memory-efficient-Levenshtein-algorithm,
-- with some additional optimizations.
-- =====
ALTER FUNCTION [dbo].[Levenshtein](
    @s nvarchar(4000)
    , @t nvarchar(4000)
    , @max int
)
RETURNS int
WITH SCHEMABINDING
AS
BEGIN
    DECLARE @distance int = 0 -- return variable
    , @v0 nvarchar(4000) -- running scratchpad for storing computed distances
    , @start int = 1 -- index (1 based) of first non-matching character between the two string
    , @i int, @j int -- loop counters: i for s string and j for t string
    , @diag int -- distance in cell diagonally above and left if we were using an m by n matrix
    , @left int -- distance in cell to the left if we were using an m by n matrix
    , @sChar nchar -- character at index i from s string
    , @thisJ int -- temporary storage of @j to allow SELECT combining
    , @jOffset int -- offset used to calculate starting value for j loop
    , @jEnd int -- ending value for j loop (stopping point for processing a column)
    -- get input string lengths including any trailing spaces (which SQL Server would otherwise ignore)
    , @sLen int = datalength(@s) / datalength(left(left(@s, 1) + '.', 1)) -- length of smaller string
    , @tLen int = datalength(@t) / datalength(left(left(@t, 1) + '.', 1)) -- length of larger string
    , @lenDiff int -- difference in length between the two strings
    -- if strings of different lengths, ensure shorter string is in s. This can result in a little
    -- faster speed by spending more time spinning just the inner loop during the main processing.
    IF (@sLen > @tLen) BEGIN
        SELECT @v0 = @s, @i = @sLen -- temporarily use v0 for swap
        SELECT @s = @t, @sLen = @tLen
        SELECT @t = @v0, @tLen = @i
    END
    SELECT @max = ISNULL(@max, @tLen)
    , @lenDiff = @tLen - @sLen
    IF @lenDiff > @max RETURN NULL

    -- suffix common to both strings can be ignored
    WHILE (@sLen > 0 AND SUBSTRING(@s, @sLen, 1) = SUBSTRING(@t, @tLen, 1))
        SELECT @sLen = @sLen - 1, @tLen = @tLen - 1

    IF (@sLen = 0) RETURN @tLen

    -- prefix common to both strings can be ignored
    WHILE (@start < @sLen AND SUBSTRING(@s, @start, 1) = SUBSTRING(@t, @start, 1))
        SELECT @start = @start + 1
    IF (@start > 1) BEGIN
        SELECT @sLen = @sLen - (@start - 1)
        , @tLen = @tLen - (@start - 1)

        -- if all of shorter string matches prefix and/or suffix of longer string, then
        -- edit distance is just the delete of additional characters present in longer string
        IF (@sLen <= 0) RETURN @tLen

        SELECT @s = SUBSTRING(@s, @start, @sLen)
        , @t = SUBSTRING(@t, @start, @tLen)
    END
END
```

```

-- initialize v0 array of distances
SELECT @v0 = '', @j = 1
WHILE (@j <= @tLen) BEGIN
    SELECT @v0 = @v0 + NCHAR(CASE WHEN @j > @max THEN @max ELSE @j END)
    SELECT @j = @j + 1
END

SELECT @jOffset = @max - @lenDiff
, @i = 1
WHILE (@i <= @sLen) BEGIN
    SELECT @distance = @i
    , @diag = @i - 1
    , @sChar = SUBSTRING(@s, @i, 1)
    -- no need to look beyond window of upper left diagonal (@i) + @max cells
    -- and the lower right diagonal (@i - @lenDiff) - @max cells
    , @j = CASE WHEN @i <= @jOffset THEN 1 ELSE @i - @jOffset END
    , @jEnd = CASE WHEN @i + @max >= @tLen THEN @tLen ELSE @i + @max END
    WHILE (@j <= @jEnd) BEGIN
        -- at this point, @distance holds the previous value (the cell above if we were using an m by n matrix)
        SELECT @left = UNICODE(SUBSTRING(@v0, @j, 1))
        , @thisJ = @j
        SELECT @distance =
            CASE WHEN (@sChar = SUBSTRING(@t, @j, 1)) THEN @diag --match, no change
                ELSE 1 + CASE WHEN @diag < @left AND @diag < @distance THEN @diag --substitution
                    WHEN @left < @distance THEN @left -- insertion
                    ELSE @distance -- deletion
                END
            END
        SELECT @v0 = STUFF(@v0, @thisJ, 1, NCHAR(@distance))
        , @diag = @left
        , @j = case when (@distance > @max) AND (@thisJ = @i + @lenDiff) then @jEnd + 2 else @thisJ + 1 end
    END
    SELECT @i = CASE WHEN @j > @jEnd + 1 THEN @sLen + 1 ELSE @i + 1 END
END
RETURN CASE WHEN @distance <= @max THEN @distance ELSE NULL END
END

```

Appendix 5: the SQL store procedure for data processing

```
USE [Docenti_Luiss]
GO
/***** Object: StoredProcedure [dbo].[PROCESS_TABLES]    Script Date: 22/05/2023 09:43:12 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Anastasia Picchia
-- Create date:    02/05/2023
-- Description:    Transform DOCENTI_LUISS      into LU_TEACHERS      and
--                CERCA_UNIVERSITA      into CU_ASSIGNMENT
-- =====
ALTER PROCEDURE [dbo].[PROCESS_TABLES]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- ALL RECORDS ARE DELETED FROM LU_TEACHERS
    TRUNCATE TABLE LU_TEACHERS

    -- FIRST ASSIGNMENT OF LU_TEACHERS STARTING FROM DOCENTI_LUISS
    INSERT INTO LU_TEACHERS
    SELECT distinct Nominativo, Data_di_nascita, Nazionalita, Naz_codice, UE_ExtraUE_ITA, Genere, '' as Docente_GS
    FROM DOCENTI_LUISS
    WHERE Dipartimento not in ('Luiss Business School', 'School of Government', 'School of European Political
    Economy') and Colore <> 'A'

    -- DATA CLEANING BASED ON COMPARISON BETWEEN CERCA_UNIVERSITA AND DOCENTI_LUISS

    UPDATE LU_TEACHERS
    SET Nominativo = 'FERNANDES DA SILVA RANCHORDAS Sofia Hina'
    WHERE Nominativo = 'FERNANDEZ DA SILVA RANCHORDAS Sofia Hina'

    UPDATE LU_TEACHERS
    SET Nominativo = 'CANOFARI Paolo'
    WHERE Nominativo = 'CANOFARI Paolo'

    UPDATE LU_TEACHERS
    SET Nominativo = 'VINUALES Jorge Enrique'
    WHERE Nominativo = 'VINUALES Jorge Erique'

    UPDATE LU_TEACHERS
    SET Nominativo = 'VILLARROEL ORDENES Francisco Javier'
    WHERE Nominativo = 'VILLARROEL ORDENES Francisco '

    UPDATE LU_TEACHERS
    SET Nominativo = 'IZZO Maria Federica'
    WHERE Nominativo = 'IZZO Federica'

    UPDATE LU_TEACHERS
    SET Nominativo = 'HOMBERG Fabian Kurt Falk'
    WHERE Nominativo = 'HOMBERG Fabian Karl'

    UPDATE LU_TEACHERS
    SET Nominativo = 'TURNER Karynne'
    WHERE Nominativo = 'TURNER Karynne Lenore'

    UPDATE LU_TEACHERS
    SET Nominativo = 'VENEL Xavier Mathieu Raymond'
    WHERE Nominativo = 'VENEL Xavier'

    UPDATE LU_TEACHERS
    SET Nominativo = 'TEE Richard Liong Gie'
    WHERE Nominativo = 'TEE Richard'

    -- column DOCENTE_GS is assigned starting from CERCA_UNIVERSITA
    UPDATE LU_TEACHERS
```

```

SET LU_TEACHERS.Docente_GS = CERCA_UNIVERSITA.Docente_GS
FROM LU_TEACHERS JOIN CERCA_UNIVERSITA ON LU_TEACHERS.Nominativo = CERCA_UNIVERSITA.Cognome_Nome

-- addition to LU_TEACHERS of all teachers in CERCA_UNIVERSITA not already present in LU_TEACHERS.
-- Columns not available in CERCA_UNIVERSITA are assigned with NULL
INSERT INTO LU_TEACHERS
SELECT DISTINCT
    Cognome_Nome AS Nominativo,
    NULL AS Data_di_nascita,
    NULL AS Nazionalita,
    NULL AS Naz_codice,
    NULL AS UE_ExtraUE_ITA,
    Genere,
    Docente_GS
FROM CERCA_UNIVERSITA
WHERE Cognome_Nome NOT IN (SELECT Nominativo FROM LU_TEACHERS)

-- initial delete all of CU_ASSIGNMENTS
TRUNCATE TABLE CU_ASSIGNMENTS

-- in comparison to CERCA_UNIVERSITA we loose the columns Genere and Docente_GS that have been moved into
-- LU_TEACHERS because not changing over time
INSERT INTO CU_ASSIGNMENTS
SELECT date as Data_riferimento, Cognome_Nome, Fascia, Facolta, Settore, Settore_concorsuale,
    Struttura_di_afferenza, Ruolo,
    case when year(date)<=2010 then Facolta else Struttura_di_afferenza end as dipartimento
FROM CERCA_UNIVERSITA

-- automatic data cleaning of settore: NULL are replaced by the oldest available data for each teacher
select CU_ASSIGNMENTS.Cognome_Nome, CU_ASSIGNMENTS.Settore
into #TEMP1
from CU_ASSIGNMENTS join
(
    select Cognome_Nome, min(data_riferimento) as min_data
    from CU_ASSIGNMENTS
    where Settore is not null
    group by Cognome_Nome
) as A
on CU_ASSIGNMENTS.Cognome_Nome = A.Cognome_Nome and
CU_ASSIGNMENTS.data_riferimento = a.min_data

update CU_ASSIGNMENTS
set CU_ASSIGNMENTS.Settore = #TEMP1.Settore
from CU_ASSIGNMENTS join #TEMP1 ON
CU_ASSIGNMENTS.Cognome_Nome = #TEMP1.Cognome_Nome
where CU_ASSIGNMENTS.SETTORE IS NULL

-- automatic data cleaning of settore concorsuale: NULL are replaced by the oldest data for each teacher
select CU_ASSIGNMENTS.Cognome_Nome, CU_ASSIGNMENTS.Settore_concorsuale
into #TEMP2
from CU_ASSIGNMENTS join
(
    select Cognome_Nome, min(data_riferimento) as min_data
    from CU_ASSIGNMENTS
    where Settore_concorsuale is not null
    group by Cognome_Nome
) as A
on CU_ASSIGNMENTS.Cognome_Nome = A.Cognome_Nome and
CU_ASSIGNMENTS.data_riferimento = a.min_data

update CU_ASSIGNMENTS
set CU_ASSIGNMENTS.Settore_concorsuale = #TEMP2.Settore_concorsuale
from CU_ASSIGNMENTS join #TEMP2 ON
CU_ASSIGNMENTS.Cognome_Nome = #TEMP2.Cognome_Nome
where CU_ASSIGNMENTS.Settore_concorsuale IS NULL

-- on the remaining NULL we assign a default
update CU_ASSIGNMENTS
set Settore = 'n.d.'
where SETTORE IS NULL

update CU_ASSIGNMENTS
set Settore_concorsuale = 'n.d.'
where Settore_concorsuale IS NULL

-- update of column GS_PROFILES.dept to make easier the reporting in Power BI

```



```

UPDATE GS_PROFILES
SET dept = ''

select A.Cognome_Nome, Dipartimento
into #T1
from CU_ASSIGNMENTS A join      ( -- for each teacher take the most recent department
select Cognome_Nome, max(Data_riferimento) as max_data
from CU_ASSIGNMENTS
group by Cognome_Nome
) as B on A.Cognome_Nome = B.Cognome_Nome and
Data_riferimento = B.max_data

select Docente_GS, Dipartimento
into #T2
from #T1 join LU_TEACHERS on Cognome_Nome = Nominativo

UPDATE GS_PROFILES
SET dept = Dipartimento
from GS_PROFILES join #T2 on teacher = Docente_GS

```

END