2022 - MSc - Data Analytics for Immersive Environments - CA4 - RDBMS & Linear Regression Project

Database design, SQL, and Linear Regression

Niall McGuinness

December 2022

Contents

Learning Outcomes

To practice the following:

- Design of a logical data model for a database.
- Understand and apply database normalization techniques (1NF to 3NF).
- Formulation and execution of SQL queries.
- Interpretation, manipulation, summarization and analysis of a data set.
- Generation of inferential statistics using linear regression.
- Analysis and visualization of linear regression data.
- Report generation in R Markdown and R Notebook.
- Use of R language, SQL, and RStudio.

Part A - Database Design & SQL Querying [75 Marks]

Overview

"Relational database design (RDD) models information and data into a set of tables with rows and columns. Each row of a relation/table represents a record, and each column represents an attribute of data. The Structured Query Language (SQL) is used to manipulate relational databases. The design of a relational database is composed of four stages, where the data are modeled into a set of related tables. The stages are: define relations/attributes, define primary keys, define relationships, and normalization" [1]

Database Design [40 Marks]

Apon graduation, you decide to form a small company specialising in the production of independent video game, or computer animation, projects. For this assignment, you are required to model a **simplified**

production management tool which represents a subset of the entities related to the development of a single video game, or computer animation, project.

All video game, or computer animation, projects require the following **core** entities:

- **Team Member**: describes an individual on the team (e.g. tester, lead artist, manager, 3D modeller).
- **Work Item**: describes smallest parcel of work resulting in a testable output (e.g., name, status, assigned to).
- **Asset**: describes a single electronic resource produced by one or more work items (e.g., type,format,created by).
- **Library**: describes a collection of available assets.
- **Project**: describes project being produced, timeline, and team involved (e.g. name, members, output, delivery date).

For this assignment you are required to model a database using a relational database modelling technique. This will involve the use of a third-party tool (i.e., Vertabelo) to generate a physical data model of the required database. A physical data model represents relational data objects (e.g., tables, fields, primary and foreign keys), their relationships, and the cardinality of these relationships.[2] Once database modelling is complete, you will then generate an SQL script which will be used to create an SQLite database in R Studio. Once the database has been created in R Studio, you will then populate the database with data, and execute a number of SQL queries.

During the database modelling phase you will need to decide on the table attributes, data types (i.e., type, length, and precision), and the cardinality (e.g., one-to-one, one-to-many) of the relationship between tables. Your design should consist of no fewer than **five** tables and no more than **ten** tables. Note, not all tables will have relationships and may be **stand-alone** tables with no connection(s) to any other table. A total of **30 marks** will be awarded for the proper modelling of the database tables.

All tables must be normalized to satisfy **Third Normal Form**. Links have been provided in the Recommended Reading section on applying 1NF to 3NF. A total of **10 marks** will be awarded for the proper normalization of the database tables.

In thinking about the core entities, described above, you should add whatever attributes you feel are necessary to reasonably describe the entity. An exact structure for each entity is not prescribed because you are required to think through what is reasonably required, which includes the structure and relationship of any tables.

Note, your design is **not** intended to exactly describe the software development lifecycle. Rather, you may think of your design as describing **who**, is involved in what **project**, to produce what **asset**, under whose **supervision**, and for what **deadline** and **project**. You may have encountered software similar to Adobe ShotGrid [3] and reviewing the product overview video for this product may be useful.

Database Setup [10 Marks]

Once you have finalised the physical data model of your database you will then generate the SQL script using the recommended third-party tool (i.e., Vertabelo). Next, use this script in an R Notebook, along with SQL chunks to generate an SQLite database. This is a local database file on your computer's hard drive. The file should be clearly and logically named (e.g. *daie_ca4_data.sqlite*).

Database Querying [25 Marks]

Once the database has been created and populated with data, you should then create a second R Notebook to be used to query the data in the database using SQL queries. You are required to demonstrate the following SQL concepts using no fewer than **five** distinct SQL statements:

1. SELECT with WHERE, LIKE, and OR
2. SELECT with DISTINCT and ORDER BY
3. Inner Join
4. Subquery with SELECT
5. SELECT across a date range

Part A - Deliverable Requirements

These steps will result in the generation of the following files:

1. An SQL Script (Vertabelo)
2. An E-R diagram image (Vertabelo) of the logical data model
3. An SQLite file (R Studio)
4. Two R Notebook files (e.g., *ca4_generate.rmd*, *ca4_query.rmd*) containing the R and SQL code to generate and populate, and query the database, respectively.

Each R Notebook will have the following structure:

1. Title (see YAML header)
2. A 1-2 line subtitle of file contents (see YAML header)
3. An embedded E-R diagram (where appropriate)
4. SQL & R Code
    - A 1-3 line description of each SQL code chunk
    - SQL code chunk
    - R code chunk showing SQL query output

Part B - Linear Regression Analysis [20 Marks]

Overview

"Simple linear regression is used to estimate the relationship between two quantitative variables... Linear regression models use a straight line, while logistic and nonlinear regression models use a curved line. Regression allows you to estimate how a dependent variable changes as the independent variable(s) change." [4]

Linear Regression Requirement [15 marks]

A survey of gamer preferences was conducted across 250 undergraduate students. The participants were asked to provide the following information:

- **gender**
- **age**
- **ethnicity**
- **top_reason_gaming** - reason provided why they game
- **gaming_platform** - platform used when gaming
- **favourite_game** - preferred game genre
- **avg_monthly_hrs_gaming** - estimated hours/month spent gaming
- **avg_years_playing_games** - average years playing computer games
- **avg_monthly_expenditure_dlc** - average monthly expenditure on downloadable content
- **play_roblox** - familiarity with Roblox platform
- **use_steam** - use of Steam platform to download games

Using the survey data provided, you are required to perform linear regression on the data from any pair of appropriate columns. The CSV file contains 250 records and you are required to perform linear analysis on a **random sample (without replacement)** of **200 records**. The *sample* function in R can be used for the purpose of random sampling. The data has already been cleaned and so no erroneous values are present. The data are available in the CSV file under the CA link in Moodle.

In the R Notebook file, you should clearly identify the **variables** chosen, any **assumptions** made, and any **tests/justifications** for those assumptions. You should then report the statistical **results** of the linear regression and briefly **discuss** the strength of any positive/negative association. You must include an **interpretation** of both R and R^2 for the analysis conducted and results obtained. Finally, discuss the possible effect of any outliers (i.e., *influential* or *high-leverage*) in the data set. This discussion should be limited to a maximum of **200 words**.

Visualization Requirements [5 marks]

A plot of the linear regression data must be generated using GGPLOT2. This plot should show the data points and the regression line. It should include clear labelling of the x-y axis, a title, and a legend, where necessary.

Part B - Deliverable Requirements

These steps will result in the generation of a single R Notebook file with the following structure:

1. Title (see YAML header)
2. A 1-2 line subtitle of file contents (see YAML header)
3. Statement of Assumptions (clearly **state** any assumptions made with respect to the variables chosen for analysis)
4. Testing of Assumptions (provide measures, plots and/or justification for your assumptions)
5. Discuss and interpret the analysis conducted and results obtained.
6. R Code
    - A 1-3 line description of each R code chunk
    - R code chunk to load and randomly sample data
    - R code code calculating linear regression for data
    - R code chunk using GGPLOT2 to generate linear regression plot(s)

Structure, Presentation & Quality [5 Marks]

Your R Notebooks should output both the code and any data/plot/table to the rendered document (e.g. HTML, PDF, Word).

A total of **2.5 marks** will be awarded for the clear commenting of any (non-trivial) line of R and SQL code and a clear description of any novel processing and/or user-defined functions.

Next, **2.5 marks** will be awarded if all included graphs/tables are formatted to maximise readability (i.e. main heading, labs, tick spacing and frequency, plot character, and caption).

Version Control Requirements

You must use a recognised online code repository (e.g., GitHub) and make regular well-named commits to your private repository. A link to your code repo must be included in a README as part of the final submission and you must add your lecturer as a developer to the repository. The repository must be named *2022_DAIE_CA4_StudentInitials1*.

Your grade for this component will depend directly upon the **regularity** of your commits. A development project of this size should consist of a minimum of **5+ distinct commit messages** spread over the lifetime of the development. Committing all your code in one commit, before the deadline, will be interpreted negatively.

Any submission made which does not include a repository link will not be graded.

Submission Requirements

1. A single **ZIP file** containing your SQL Script (Vertabelo), E-R diagram image (Vertabelo), SQLite database file (R Studio), and a **README** file containing a link to your R source code repository. You may also add these files (i.e., SQL, E-R, and SQLIte) to your repository for convenience.
2. Ensure that **no** changes are made to the repository following the submission deadline. You should create a separate fork for this submission and leave it unchanged after the deadline. Ask your lecturer for details on fork creation.
3. The assignment must be entirely the work of each student. Students are **not** permitted to share any pseudocode or source code from their solution with any other student in the class.
4. Students may **not** distribute the source code of their solution to any other student, in any format (i.e., electronic, verbal, or hardcopy transmission).

5. Plagiarised assignments will receive a mark of **zero**. This also applies to the individual/group allowing their work to be plagiarised. Any plagiarism will be reported to the Head of Department and a report will be added to your permanent academic record.
6. Late assignments will **only** be accepted if accompanied by the appropriate medical note. This documentation must be received within 10 working days of the project deadline. The Institute standard penalties for late submission will apply.
7. Each student **must** complete and **sign** a single assignment cover sheet. Please submit the signed cover sheet, via email, before 5 pm on the Friday of the week of the deadline.

Recommended Reading

- Recommended Text - OpenIntro Statistics Videos & Slides
- Normalization in Relational Databases: First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF)
- Learn Database Normalization - 1NF, 2NF, 3NF, 4NF, 5NF
- A Comprehensive Introduction to Working with Databases using R
- Simple Linear Regression | An Easy Introduction & Examples
- Linear regression using R programming
- Add Regression Line to ggplot2 Plot in R
- Fitting and visualizing linear regression models with the ggplot2 R package (CC237)
- ggplot for plots and graphs. An introduction to data visualization using R programming

Additional Resources

The resources below are provided as a mixture of **optional background reading**, technical reference, and useful tools.

- YouTube - Riffomonas Project - R Language Tutorials
- Working With SQL Databases From R: Querying SQL Databases Using dplyr
- Ian Cook | Bridging the Gap between SQL and R | RStudio (2020)
- R Markdown Cookbook
- R Markdown Cheatsheet
- Understanding YAML headers
- R Markdown Themes
- Harvard referencing quick guide: Citing and referencing material

References

1. Technopedia (2017) Relational Database Design Available at: https://www.techopedia.com/definition/25113/relational-database-design-rdd (Accessed: 20 December 2022)]
2. IBM (2021) Physical data models Available at: https://www.ibm.com/docs/en/ida/9.1.1?topic=modeling-physical-data-models (Accessed: 20 December 2022)]
3. Adobe ShotGrid (2022) What is ShotGrid? Available at: https://www.autodesk.com/products/shotgrid/overview (Accessed: 22 December 2022)]
4. Rebecca Bevans (2020) Simple Linear Regression | An Easy Introduction & Examples Available at: https://www.scribbr.com/statistics/simple-linear-regression/ (Accessed: 23 December 2022)]