Road Safety Case Study

Prepared for:

Amadeus

Amadeus Software Limited
Email: info@amadeus.co.uk
Tel: +44 (0) 1993 848010
Web: www.amadeus.co.uk

The Old School Hall, 11 Wesley Walk,
Witney, Oxfordshire, OX28 6ZJ, England
Company Registration Number: 261 8399

Document History

Revision	Author	Date	Details
0.1	Edward Blow	16 SEP 2021	Initial Draft
0.2	Lucy Parkin	07 OCT 2021	Internal Review
1.0	Dave Evans	20 JUL 2022	Published
1.1	Lucy Parkin	13 JAN 2023	Amended for Python

Table of Contents

1 Introduction

This case study should be completed using Python.

A government agency has provided you with some data, containing information on road accidents that occurred during 2016. They have also provided a spreadsheet containing information about the columns within the data, this can be used for cleaning and formatting purposes, and to better understand the data provided.

Following further discussion with the agency, they have requested that analysis be carried out to better understand the following information:

- Does driver gender have an impact on accident fatality rate?
- The volume of accidents per highway authority in each country (per 1,000,000 inhabitants in the country).
- The days with the fewest and the most accidents.
- How the number of accidents changes over time, including the number of accidents on each day of the week and the volume in each season.
- The number of accidents recorded involving animals.

They also wish to have a profile of any missing data, so that they can find ways to improve their reporting to combat this in the future.

Population statistics were obtained from the ONS website (contains population by country and by local authority). Within this table, GREAT BRITAIN is defined as England, Scotland, and Wales, whilst UK also includes Northern Ireland.

1.1 Requirements

The following sections provide information about what is required from you during this case study

1.1.1 Programming Standards

Professional programs adhere to the following principles:

1. The program generates reproducible output:
    - The program is free from errors, warnings and any unnecessary notes that indicate potentially inaccurate results, unless suitably justified. (Any such justifications should be made within the report).
2. Auditable and Documented:
    - The program is appropriately documented with a header and comments for ease of readability.
3. The program is organised and efficient:
    - The program should be easily executed by another person, even if the initial input data is moved to a new storage location.
    - Data, reports, programs and other case study components should be stored in an organised folder structure.
    - The program code is clear, uses consistent naming conventions, and uses programming techniques readily understood by colleagues and peers.
    - The program should avoid unnecessary processing, combining multiple steps into fewer steps where clear and appropriate to do so.

These principles should be kept in mind while completing this assignment.

1.1.2 Summary of Deliverables

This case study is presented in three sections. All content created should be stored in a single root folder, including input data, your programs and the output you generate.

Files containing all the written code and results should also be saved in a format so that they may be run by another user with ease and without problems.

1.1.3 Case Study Data Sources

The following data is provided for this case study:

- **Accidents.csv** contains general information on each reported road accident that occurred.
- **Casualties.csv** contains information on every casualty that occurred due to a road accident. The information is specific to the individual.
- **Vehicles.csv** contains vehicle-specific information on the vehicles and drivers involved in each road accident. Note that there may be more than one vehicle involved in one accident and more than one casualty occurring due to each accident.
- **Variable_lookup.xlsx** is a spreadsheet provided to better understand the data where required. This spreadsheet does not need to be imported, but is just included for reference.
- **Population_statistics.csv** contains a record from the ONS of estimates for the population of each authority area and each country within the UK. All the data pertain to 2016.

(The data from the ONS does not contain population statistics for Heathrow. For this reason, you should use the average daily number of passengers for 2016, which is 206,800, taken from Heathrow's website as the "population" of Heathrow airport: https://www.heathrow.com/company/about-heathrow/performance/airport-operations/traffic-statistics)

2 Assignment

This assignment is split into three sections:

A. Importing Data
B. Transforming Data
C. Reporting

2.1 Section A: Importing Data

Import each of Accidents.csv, Casualties.csv, Vehicles.csv and Population_statistics.csv to create tables for use in this case study. Verify that the contents of each table accurately reflect the original data files, this can be done visually.

Note: If you encounter a problem with the size of the data, investigate programming options to bypass this by importing the data in smaller chunks, and concatenate the results together.

You should create a program containing the relevant code written for this section. The code must run without issues upon starting a new session. Any assumptions made, importing constraints, or issues discovered whilst importing the data should also be recorded suitably within the program containing the code. (For example, within comments in the code).

2.2 Section B: Transforming Data

In preparation for analysis and reporting, you are required to clean the tables, derive new columns, and apply business logic to the tables. (Do not overwrite the original tables).

2.2.1 Missing Data Analysis

To help the agency easily identify how much missing data there are, a csv file should be created named missing.csv. Missing values throughout the ACCIDENTS, CASUALTIES and VEHICLES tables have been inputted as -1. A partial view of the expected output table structure can be seen below.

Table	Column	Number_Missing
VEHICLES	propulsion_code	58260
ACCIDENTS	junction_control	56623
VEHICLES	impact	1265

It should contain three columns:

- Table: Name which table the column comes from,
- Column: Name all the columns in the original tables,
- Number_Missing: Provide the number of missing values for each column.

This csv file should be sorted by Number_Missing in descending order and contain all the columns from the tables ACCIDENTS, CASUALTIES and VEHICLES which contain missing values.

2.2.2 Data Preparation

- Any missing values identified in the previous section should be changed within the data to ensure that any further processing is easier to interpret as missing. Use your knowledge of how to handle missing values to decide on an appropriate technique.
- You should then perform exploratory data analysis to identify any data quality issues. Appropriate steps to take will include cleaning the columns within the tables to ensure that values remain consistent within columns.
- Using the ACCIDENTS table, you must create the following columns:
    - Country
    - Weekday
    - Season
- Using the CASUALTIES table, create the following column, which can be determined from a combination of the columns index and casualty_ref:
    - Number of casualties per accident
- Using the VEHICLES table, create a column which can be determined from a combination of the columns index and vehicle_ref:
    - Number of vehicles involved per accident

Within the program, comments should be included in a suitable location detailing what has taken place. Store all programs written with appropriate names within the PROGRAMS folder.

2.2.3 Preparing the Data for Profiling and Analytics

- In preparation for the analysis, all three tables (ACCIDENTS, CASUALTIES and VEHICLES) are needed to be combined into one table, named ROAD_ACCIDENTS. Merge the tables suitable to create a single source of information detailing each casualty

that was reporting due to a road accident in 2016. Ensure you find the appropriate columns to merge on.

- POPULATION_STATISTICS should also be merged to this table to provide the population of each area for reporting purposes. The columns in common between the tables are “code” in POPULATION_STATISTICS and “highway_authority” in ROAD_ACCIDENTS. Reasons as to which kind of joins were used should be provided in comments within the code.
- Another column denoting whether an animal was involved in the accident should be created for analysis. Animals being involved are reported under each of the following columns – a carriageway hazard, an object hit in the carriageway, or a type of vehicle.

  The variable_lookup spreadsheet can be used to identify all possible values indicating animal involvement within these columns. The tables will need to be merged before creating this column.

2.3 Section C: Reporting

Create an output csv file for each of the following bullet points. Using the output, provide an answer the questions as a comment in your program code:

- The number of accidents per weekday. Do any days have a significantly different number of accidents to others?
- A table showing the frequency of casualty severity by country per 1,000,000 inhabitants of each country.

Explore each of the following bullet points with meaningful graphs. Using the output, provide an answer the questions as a comment in your program code:

- The frequency of casualty severity by driver sex. Does driver sex affect likelihood of an accident?
- The number of accidents involving animals
- The number of accidents per day. What are the top 5 quietest and busiest days in terms of the number of accidents?
- The number of accidents per season
- How does the number of inhabitants in an area affect the likelihood of an accident occurring in that area?

2.4 Optional Extension:

Perform relevant hypothesis tests to help you answer the questions posed above. You should make sure you use appropriate statistical methods.

Note: this is not taught during the python training courses, but theory knowledge should have been gained the the other training attended. Use this, along with additional research into how to achieve this in Python to have a go at this section.

Create a new program for this section. Within the program, comments should be included in a suitable location detailing what assumptions where made and what analysis has taken place.

Store all programs written with appropriate names within the PROGRAMS folder.

SAS Software | SAS Consultancy | Data Science

SAS Training | SAS Managed Services | Graduate Placement

Amadeus Software Limited

Email: info@amadeus.co.uk | Tel: +44 (0) 1993 848010 | Web: www.amadeus.co.uk

Amadeus Software Limited, The Old School Hall, 11 Wesley Walk, Witney, Oxfordshire, OX28 6ZJ, England