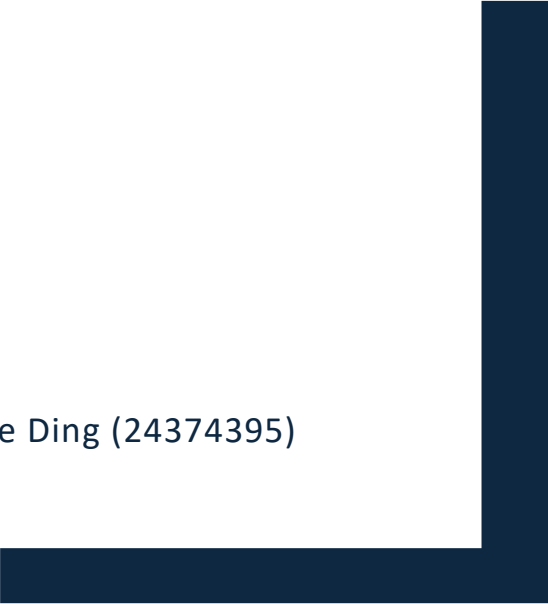




CRASH FATALITIES DATAWAREHOUSE PROJECT

Vinayak Gupta (24066272), Zeke Ding (24374395)



Contents

Executive Summary	3
Business Understanding.....	3
Business Process	3
Grain Determination	4
Design and Implementation	4
Choosing the Dimensions	4
StarNet and Concept Hierarchies.....	5
Choosing the Measure.....	6
Keys	6
Measures	7
Database Implementation.....	7
ETL Process	9
Extract	9
Transform.....	9
Data Cleaning.....	9
Load	10
Visualisation.....	11
Association Rule Mining	18
What Algorithm Was Used?.....	18
Generating Association Rules	18
Interpretation of top rules and suggestions	19
By Confidence (From association_rules_confidence.csv):.....	19
By Lift (From association_rules_confidence.csv):	21
Top 3 recommendations to the government (From strong_association_rules.csv).....	22
Recommendation 1: Enhanced Safety for Pedestrians at Moderate Speed Zones (41-80 km/h).....	22
Recommendation 2: Protection Measures for Elderly Pedestrians (Age ≥ 75)	23
Recommendation 3: Specific Interventions for Motorcyclists in Multiple-Vehicle Crashes During the Day.....	24
References	24

Appendix.....	25
Codes for creating tables and populating the database	25
Codes for SQL queries	26
Q-1	26
Q-2	26
Q-3	27
Q-4	27
Q-5	28
Visuals.....	28

Executive Summary

This project focuses on designing and implementing a data warehouse to analyse road crash fatalities in Australia, using datasets provided by the Bureau of Infrastructure and Transport Research Economics (BITRE). The aim is to provide government authorities with actionable insights into when, where, how, and to whom fatal crashes occur. The key objectives include identifying patterns related to time of day, crash types, road user roles, and vehicle involvement, while also examining demographic factors like age and gender. Additionally, the project incorporates dwelling data to normalize fatality counts and highlight high-risk regions. The analysis revealed several critical findings, such as a higher number of fatal crashes involving young pedestrians during nighttime in urban areas, increased crash severity in remote locations involving heavy vehicles, and noticeable spikes in fatalities during holiday periods like Christmas and Easter. The project also leveraged association rule mining to uncover strong relationships between crash severity, time factors, and vehicle types, offering valuable direction for data-driven policymaking and infrastructure planning.

Business Understanding

Kimball's 4-step approach was used to design and implement the data warehouse.

Business Process

This project is guided by a set of business questions designed to uncover patterns in fatal road crashes and assist government agencies in making informed decisions to improve road safety. Each question addresses a specific dimension of crash analysis, focusing not just on “what happened,” but also on “why” and “how it can be prevented.”

The business process being modelled is the **occurrence of fatal road crashes in Australia**, with each row in the data warehouse representing **one person killed** in such a crash. This process includes dimensions such as time, location, crash circumstances, vehicle involvement, and victim demographics, enabling a detailed analysis of how, when, and where fatal crashes occur, and who is most affected.

1. How many fatalities occurred in crashes involving buses, heavy rigid trucks, and articulated trucks, and what was the average speed limit associated with each vehicle type?

Crashes involving heavy vehicles—such as buses, heavy rigid trucks, and articulated trucks account for a significant number of road fatalities, including a notable portion in cases where the vehicle involvement was marked as ‘Unknown’.

2. Are older drivers more at risk of fatal crashes? Are younger pedestrians more vulnerable?

This question uncovers patterns in fatalities by role and age group, helping authorities tailor campaigns (e.g., elderly driver assessments, school zone safety, etc.).

3. When did crashes happen? Specifically, how many fatalities occurred during holiday periods, broken down by time of day (hourly)?

This gives rich temporal insight that can help in targeted patrolling hours during holidays.

4. Which Local Government Areas (LGAs) have the highest fatality rates per 100,000 people, and how do they compare in terms of dwelling density?

It ties in density (avg. people per dwelling), giving context — are these deaths happening in crowded cities or under-resourced rural areas?

5. Which road types account for the highest number of fatalities in the top 5 most-affected states?

Reveals whether specific road infrastructures (e.g., highways, local roads) are consistently more dangerous.

Grain Determination

The grain of the fact table is: **One row per person was killed in a fatal road crash in Australia.**

Each row in the fact_fatality table represents one fatality (i.e., one person). If multiple people died in one crash, there will be multiple rows, one for each person.

This level of granularity allows us to study fatalities at an individual level, including demographic, temporal, crash-specific, and geographic attributes. This granularity also enables normalization techniques (e.g., fatalities per 100,000 people) and more flexible roll-up and drill-down capabilities.

Design and Implementation

Choosing the Dimensions

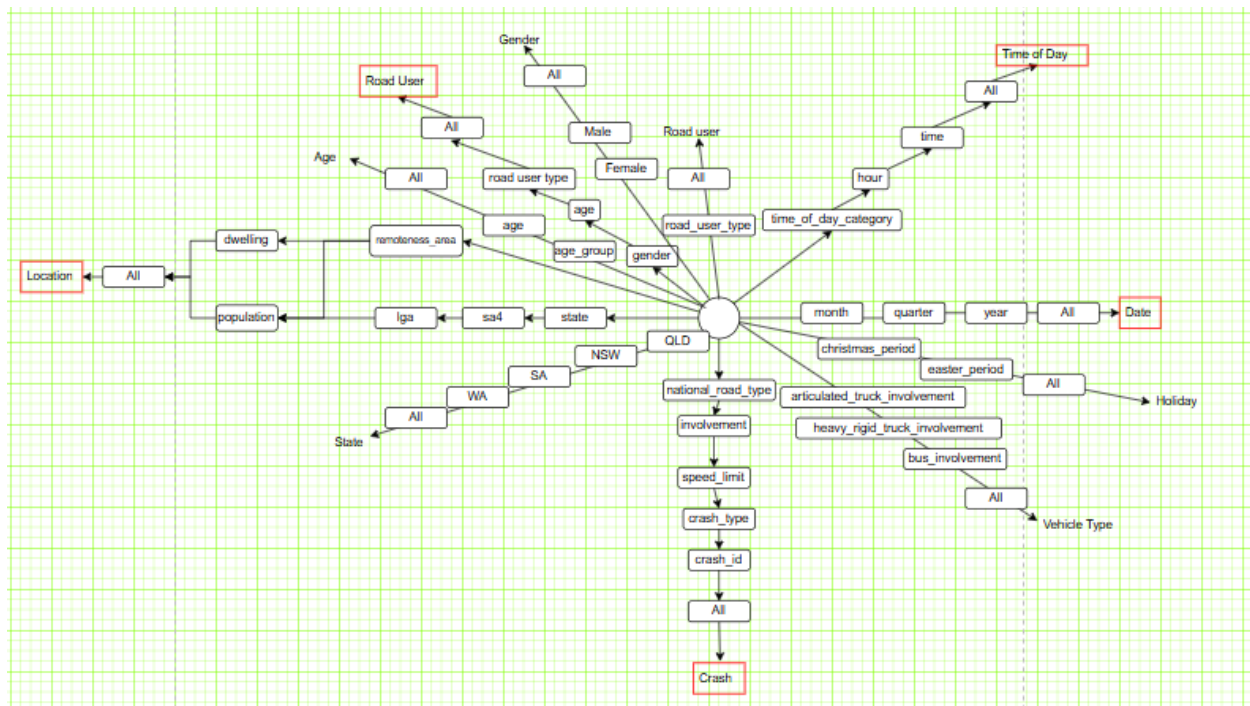
We have chosen **dim_date**, **dim_time_of_the_day**, **dim_road_user**, **dim_crash**, **dim_location**, **dim_dwelling**, **dim_population_lga**, **dim_population_remoteness** as dimension tables. The reason for choosing these dimensions is based on the type of queries we are answering. For Example, in the first question, we are answering a query related to vehicles used in crashes and

their average speed limit. Hence, we combined all those into one dimension table. In a similar way, we have made other dimension tables which are being used to answer our business questions or to make geometry.

The detailed structure of our schema is shown in the form of a StarNet Diagram and Concept hierarchies.

StarNet and Concept Hierarchies

This diagram was used as a blueprint to make an ERD diagram, which was the foundation of our data warehouse. It was made using **draw.io**.



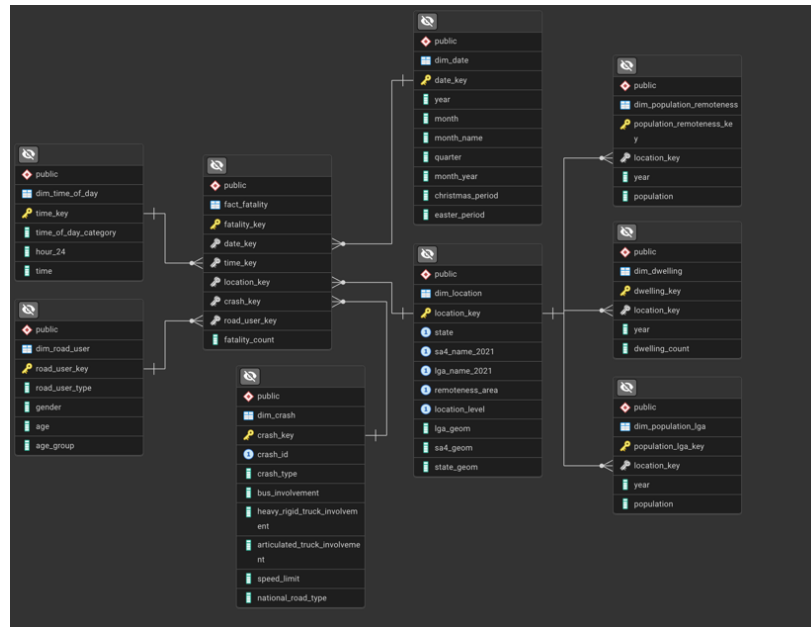
A concept hierarchy is a logical structure that organizes data at increasing levels of abstraction or granularity. It helps to organize dimensions smartly, support multi-level analysis, and drive meaningful drill-down capabilities for decision-makers.

Given the numerous dimension tables and columns in our data warehouse, describing every possible hierarchy would be excessive. Instead, we have focused only on those columns and dimensions that were used in addressing our business questions.

The diagram illustrates multiple concept hierarchies, where red boxes represent our main dimension tables, and the branches represent various levels or categories derived from them. These sub-branches are not separate dimension tables or hierarchies, but logical groupings created to reflect the query footprints and analytical needs.

For example, in Q-1, both vehicle type and speed limit come from the same table, `dim_crash`. Since we use them in separate analytical contexts, they are shown as distinct conceptual branches in the hierarchy. This approach helps maintain clarity when mapping StarNet diagrams or visualizing query footprints.

Choosing the Measure



Data Warehouse Design

We designed our data warehouse using a **Snowflake schema**, which means we've organized the data in a clean and structured way. At the centre, there's one main table called **fact_fatality** where each row represents one person who died in a road crash. Around this table, we've connected other tables with details about the crash (like time, location, vehicle type, and person involved). These connected tables are linked using special ID columns, which helps keep the data tidy and avoids repeating the same information over and over. We chose the snowflake model over the simpler star design because it lets us handle complex data, especially for location-based analysis using things like LGA and state boundaries.

This setup lets us answer a wide range of important questions. For example, we can find out what time of day most crashes happen, which areas have the highest fatality rates, or whether certain age groups or vehicle types are more involved in crashes. We also brought in map data (GeoJSON) so we can create detailed visuals in Power BI, like heatmaps showing where fatalities are highest.

Keys

Primary Keys – Auto-generated surrogate keys for each dimension (e.g., `date_key`, `crash_key`, etc, `time_key`, etc.).

Foreign Keys - These link the fact_fatalities table to its dimensions and are critical for building a snowflake schema.

- **date_key**: Connects each fatality to a specific date
- **time_key**: Associates the fatality with a specific hour
- **road_user_key**: Describes role, age, gender of person involved
- **crash_key**: Links to vehicle type, crash type, speed limit, road type
- **location_key**: Provides spatial context: state, SA4, LGA, remoteness, and geometry

Measures

fatality_count - in fact table as the primary measure

Database Implementation

Due to the long table building statement, the complete SQL script will be provided in the attachment. The code contains complete comments, and the body of the code creates the dimension table and the fact table in sequential order.

```
1 CREATE TABLE dim_location (
2     location_key BIGSERIAL PRIMARY KEY,      -- Surrogate key
3     state VARCHAR(7) NOT NULL,               -- e.g., 'NSW', 'VIC'
4     sa4_name_2021 VARCHAR(50),               -- Statistical Area Level 4 Name
5     lga_name_2021 VARCHAR(50),               -- Local Government Area Name
6     remoteness_area VARCHAR(50),            -- National Remoteness Area classification
7     location_level VARCHAR(20) NOT NULL,     -- Indicates the level this key represents ('LGA', 'SA4', 'State')
8
9     -- Geometry columns using PostGIS geography type
10    -- Use appropriate SRID for Australia (7844 for GDA2020)
11    -- Storing geometry for each level separately as requested. Nullable if a key represents a higher level.
12    lga_geom GEOMETRY(MultiPolygon, 4283),   -- Geometry for the LGA boundary
13    sa4_geom GEOMETRY(MultiPolygon, 7844),   -- Geometry for the SA4 boundary
14    state_geom GEOMETRY(MultiPolygon, 7844), -- Geometry for the State boundary
15
16    -- Unique constraint ensures we don't have duplicate definitions at the same level
17    -- Nulls in SA4/LGA names are expected for higher levels (State)
18    UNIQUE (state, sa4_name_2021, lga_name_2021, remoteness_area, location_level)
19 );
```



```

1 CREATE TABLE fact_fatality (
2     fatality_key BIGSERIAL PRIMARY KEY, -- Surrogate key for the fatality event itself (optional, but can be useful)
3     date_key BIGINT NOT NULL, -- Foreign key to dim_date
4     time_key BIGINT NOT NULL, -- Foreign key to dim_time_of_day
5     location_key BIGINT NOT NULL, -- Foreign key to dim_location
6     crash_key BIGINT NOT NULL, -- Foreign key to dim_crash
7     road_user_key BIGINT NOT NULL, -- Foreign key to dim_road_user
8
9     -- Measures
10    fatality_count INTEGER NOT NULL DEFAULT 1, -- Measure, always 1 per row representing one fatality
11
12    -- Foreign Key Constraints
13    CONSTRAINT fk_fact_fatality_dim_date FOREIGN KEY (date_key) REFERENCES dim_date (date_key),
14    CONSTRAINT fk_fact_fatality_dim_time_of_day FOREIGN KEY (time_key) REFERENCES dim_time_of_day (time_key),
15    CONSTRAINT fk_fact_fatality_dim_location FOREIGN KEY (location_key) REFERENCES dim_location (location_key),
16    CONSTRAINT fk_fact_fatality_dim_crash FOREIGN KEY (crash_key) REFERENCES dim_crash (crash_key),
17    CONSTRAINT fk_fact_fatality_dim_road_user FOREIGN KEY (road_user_key) REFERENCES dim_road_user (road_user_key)
18 );

```

The only thing need to keep in mind is that I used postgis **GEOMETRY(MultiPolygon, 4283)** and **GEOMETRY(MultiPolygon, 7844)** to store different geographic information according to the specification in the Australian Data Dictionary, but due to visualisation requirements in tableau, they need to be **converted to WGS84 (SRID 4326)**, so I created a new dedicated view for the visualisation.

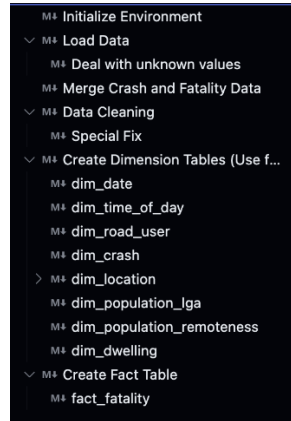
```

1 CREATE VIEW dim_location_wgs84 AS
2 SELECT
3     location_key,
4     state,
5     sa4_name_2021,
6     lga_name_2021,
7     remoteness_area,
8     location_level,
9     ST_Transform(lga_geom, 4326) AS lga_geom,
10    ST_Transform(sa4_geom, 4326) AS sa4_geom,
11    ST_Transform(state_geom, 4326) AS state_geom
12 FROM dim_location;

```

ETL Process

Pandas was used for ETL in Jupyter Notebook. The full script is provided in the appendix and as a separate file as well. Commands for Creating Tables in SQL are also provided below.



Extract

These files are used during the ETL process:

- **bitre_fatal_crashes_dec2024.xlsx** – Provided detailed crash-level data such as crash type, vehicle involvement, and road types.
- **bitre_fatalities_dec2024.xlsx** – Contained victim-level information including age, gender, and road user role.
- **Population and dwelling datasets** – Downloaded from the Australian Bureau of Statistics (ABS), giving us LGA-level population figures and dwelling counts for use in normalization and density analysis.
- **GeoJSON files** – Included spatial boundary data for Local Government Areas (LGAs), Statistical Areas (SA4s), and Australian states. These were crucial for map-based visualizations and spatial joins.

Transform

This stage was the most intensive and involved multiple sub-processes to clean, enrich, and normalize the data before it could be loaded into the warehouse.

Data Cleaning

Dropped rows with missing or malformed Crash IDs or timestamps, as they would break relational integrity or analysis timelines.

```

# Identify rows to be deleted (where Crash ID length ≠ 8)
rows_to_delete = fact_df[fact_df["Crash ID"].astype(str).str.len() != 8]

# Print only the Crash IDs that will be deleted
print("Crash IDs to be deleted (length ≠ 8):")
print(rows_to_delete["Crash ID"].to_string(index=False))

print(f"\nTotal rows to be deleted: {len(rows_to_delete)}")

# Execute deletion (keep only rows where Crash ID length equals 8)
fact_df = fact_df[fact_df["Crash ID"].astype(str).str.len() == 8]

# Verification
print(f"\nRemaining rows: {len(fact_df)}")
print("Crash ID length distribution in remaining data:")
print(fact_df["Crash ID"].astype(str).str.len().value_counts())

Crash IDs to be deleted (length ≠ 8):
201850049
201850039
201850059
201750029
201750019
201650079
201650019
201550039
201550099
201550059

Total rows to be deleted: 10

Remaining rows: 56864
Crash ID length distribution in remaining data:
Crash ID
8      56864
Name: count, dtype: int64

```

Standardized all column names and formats (e.g., converted “Bus\nInvolvement” to bus_involvement, unified date/time formats, etc.).

```

Special Fix

# Set missing age_group
fact_df.loc[fact_df["Crash ID"] == 20225110, "Age Group"] = "17_to_25"

# Fix missing Time of day
fact_df.loc[fact_df["Crash ID"] == 20225066, "Time of day_victim"] = "Day"

```

Dimensional Structuring

- Created surrogate keys (e.g., date_key, crash_key) for each dimension table.
- Parsed timestamps into hour, time category (day/night), and date-based features (month, quarter, etc.).
- Categorized ages into groups (e.g., 0–16, 17–25, etc.) for better demographic analysis.
- Transformed natural keys (like Crash ID) into foreign key links.
- Trimmed and standardized LGA and SA4 names to enable joining with spatial and population datasets.
- Loaded GeoJSON files and simplified their geometries to reduce rendering complexity.

Load

Finally, the cleaned and enriched datasets were written to CSV files and bulk-loaded into a PostgreSQL database using COPY commands.

Each dimension table (e.g., dim_date, dim_crash, dim_road_user) was assigned a primary key.

The fact table (fact_fatality) used foreign keys to link with dimensions.

This schema ensured referential integrity, enabling smooth joins, better query performance, and multi-dimensional slicing.

We only imported by table fields, and let the database generate the primary key itself.

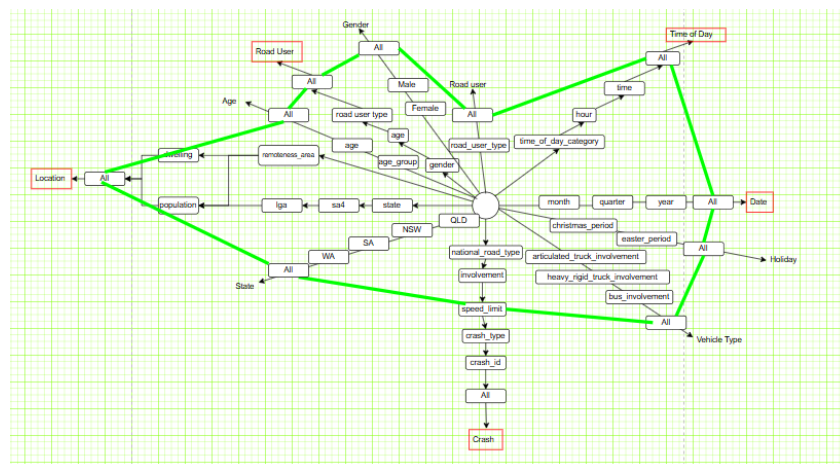
```

1 COPY dim_date (year, month, month_name, quarter, month_year, christmas_period, easter_period)
2 FROM '/tmp/dim_date.csv' DELIMITER ',' CSV HEADER;
3
4 COPY dim_time_of_day (time_of_day_category, hour_24, time)
5 FROM '/tmp/dim_timeofday.csv' DELIMITER ',' CSV HEADER;
6
7 COPY dim_road_user (road_user_type, gender, age, age_group)
8 FROM '/tmp/dim_roaduser.csv' DELIMITER ',' CSV HEADER;
9
10 COPY dim_crash (crash_id, crash_type, bus_involvement, heavy_rigid_truck_involvement, articulated_truck_involvement, speed_limit, national_road_type)
11 FROM '/tmp/dim_crash.csv' DELIMITER ',' CSV HEADER;
12
13 COPY dim_location (state, sa4_name_2021, lga_name_2021, remoteness_area, location_level, lga_geom, sa4_geom, state_geom)
14 FROM '/tmp/dim_location.csv' DELIMITER ',' CSV HEADER;
15
16 COPY dim_population_lga (location_key, year, population)
17 FROM '/tmp/dim_population_lga.csv' DELIMITER ',' CSV HEADER;
18
19 COPY dim_population_remoteness (location_key, year, population)
20 FROM '/tmp/dim_population_remoteness.csv' DELIMITER ',' CSV HEADER;
21
22 COPY dim_dwelling21 (location_key, year, dwelling_count)
23 FROM '/tmp/dim_dwelling.csv' DELIMITER ',' CSV HEADER;
24
25 COPY fact_fatality (date_key, time_key, location_key, crash_key, road_user_key, fatality_count)
26 FROM '/tmp/fact_fatality.csv' DELIMITER ',' CSV HEADER;

```

Visualisation

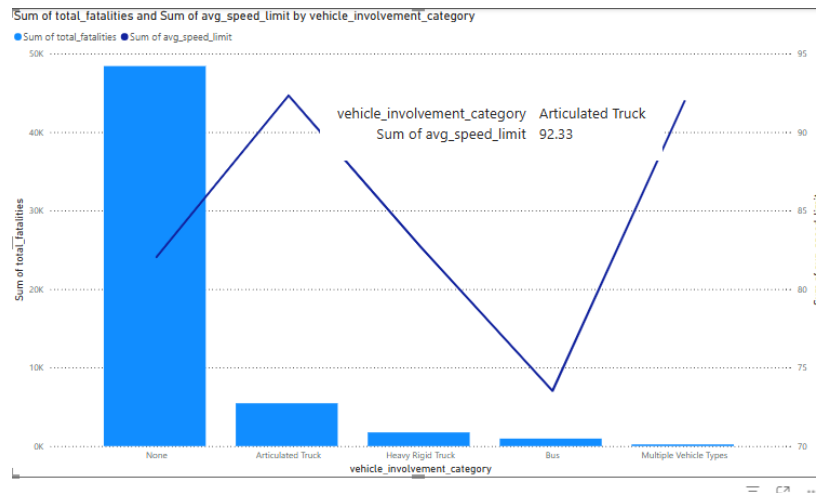
- 1) How many fatalities occurred in crashes involving buses, heavy rigid trucks, and articulated trucks, and what was the average speed limit associated with each vehicle type?



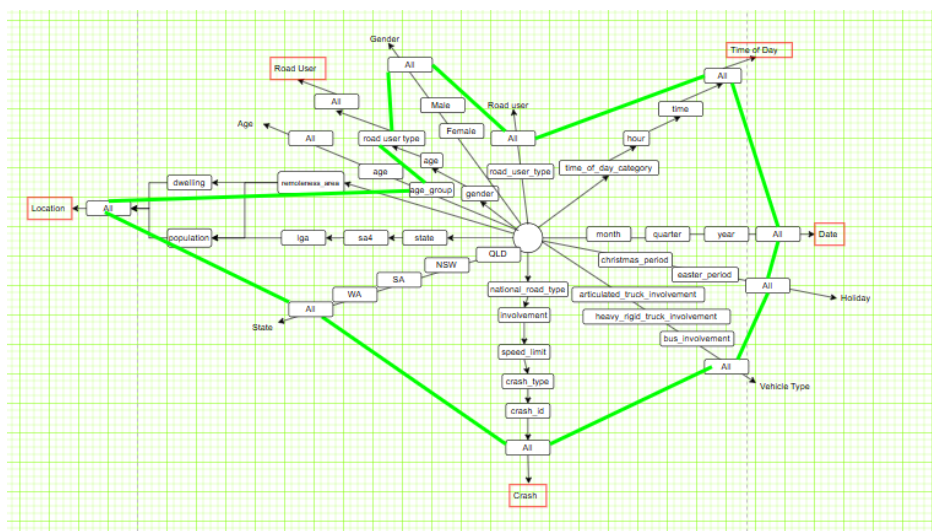
Here, we observe that most crashes are because of other vehicles with an average speed limit above 80 km/hr. In addition, the second most crashes are because of articulated trucks with an average speed limit of over 92 km/hr.

Recommendation

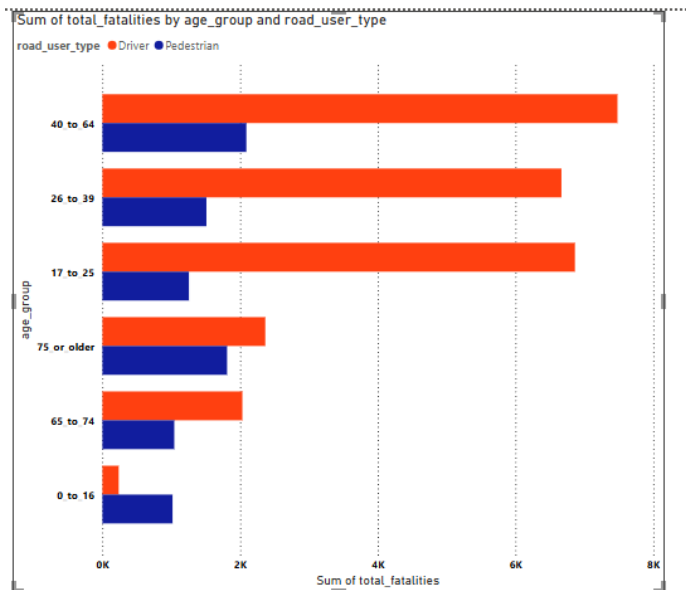
Based on the analysis, it is recommended that road safety authorities strengthen speed enforcement and review highway design in these areas. Additionally, targeted interventions—such as improving road signage and implementing vehicle-specific safety checks—should be prioritized, particularly where articulated trucks are frequently involved, to mitigate these high-risk scenarios.



2) Are older drivers more at risk of fatal crashes? Are younger pedestrians more vulnerable?



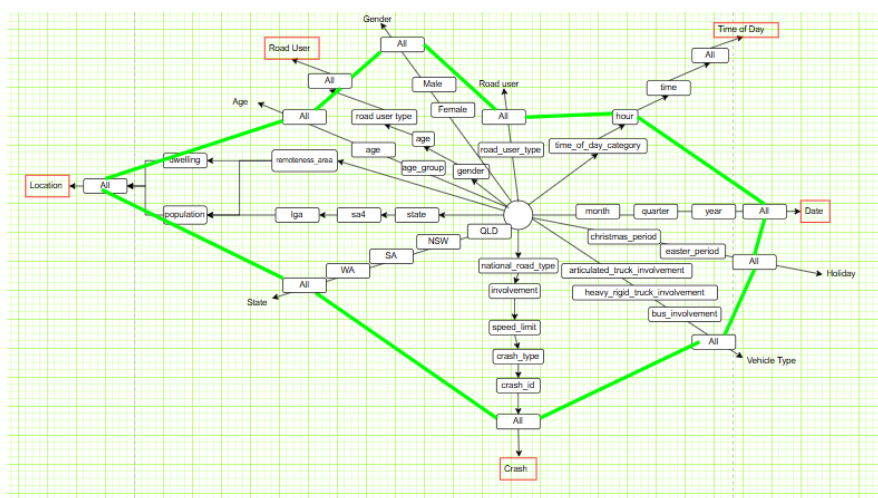
For simplicity, we only used two road_user_types, pedestrian and driver and filtered out unknown values and gender. We observed that most fatalities are due to drivers and mainly from the age group 40-64 or 17-25, with a count of more than 6000. Pedestrians who perished were mainly 75 or older and 40-64, with a count of up to 2000.



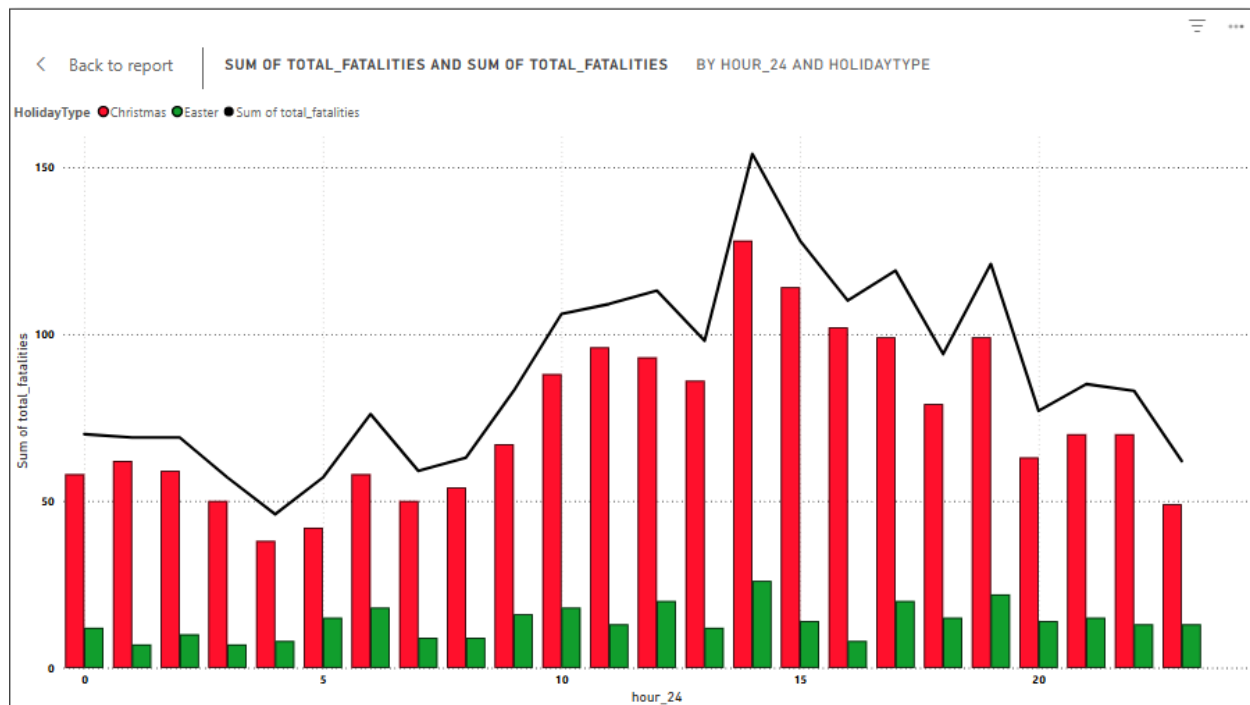
Recommendations

These insights highlight the need for targeted awareness campaigns for older pedestrians and interventions addressing younger and middle-aged drivers, such as defensive driving education or stricter enforcement during high-risk age brackets. Pedestrian risks are concentrated in older age groups, potentially due to slower reflexes and reduced mobility. Infrastructure improvements like safer pedestrian crossings, better lighting, and public campaigns tailored to high-risk age groups can help mitigate fatalities.

6. When did crashes happen? Specifically, how many fatalities occur during holiday periods, broken down by time of day (hourly)?



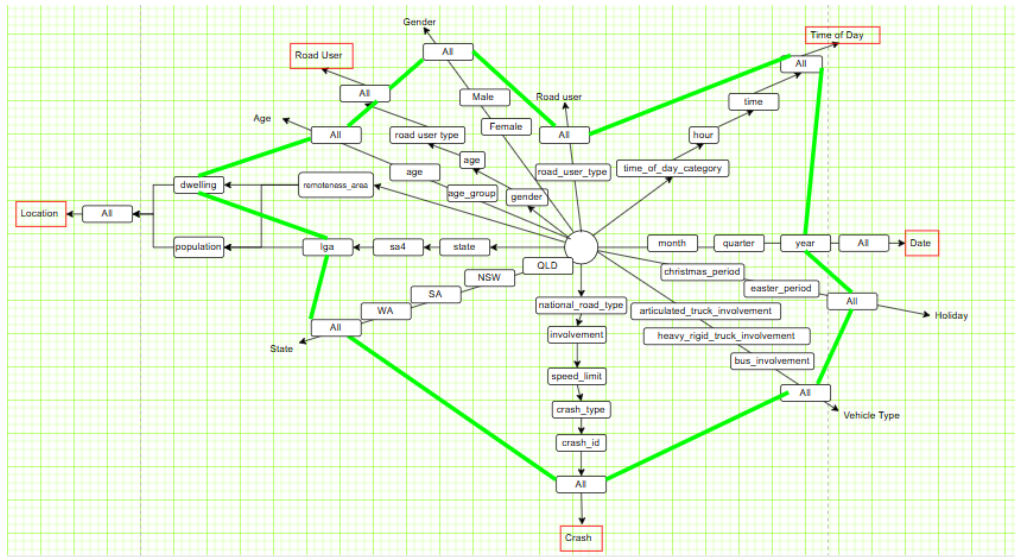
We can observe that most of the fatalities happened during Christmas as compared to Easter. Most accidents happened between 14 to 19 hours, i.e. between 2 – 7 pm for Christmas. Easter follows the constant pattern throughout, having the highest fatality at 14 (2 pm).



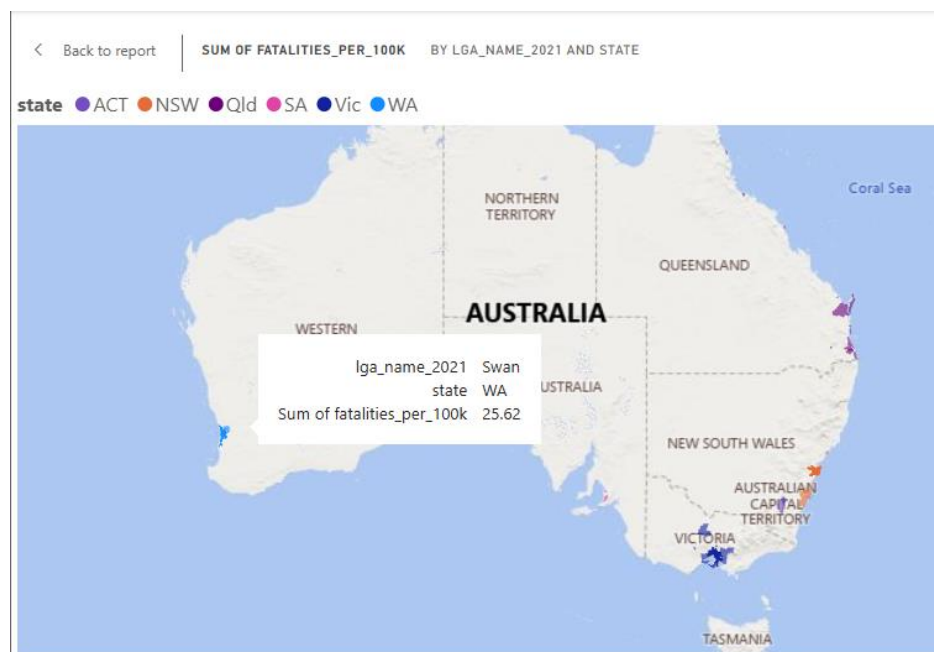
Recommendation

To mitigate the increased risk of road fatalities during holiday periods - especially between 2 PM and 7 PM when most incidents occur - targeted interventions are essential. Authorities should increase police presence and road monitoring during these peak hours to discourage speeding and reckless behaviour. Simultaneously, public awareness campaigns can educate drivers on the heightened risks of holiday travel, emphasizing the importance of alertness and sober driving. Local councils and law enforcement could introduce temporary sobriety checkpoints near event hubs or highways. Additionally, promoting alternatives like ridesharing services and public transport, especially through incentives or partnerships, can reduce congestion and alcohol-related incidents during festive times.

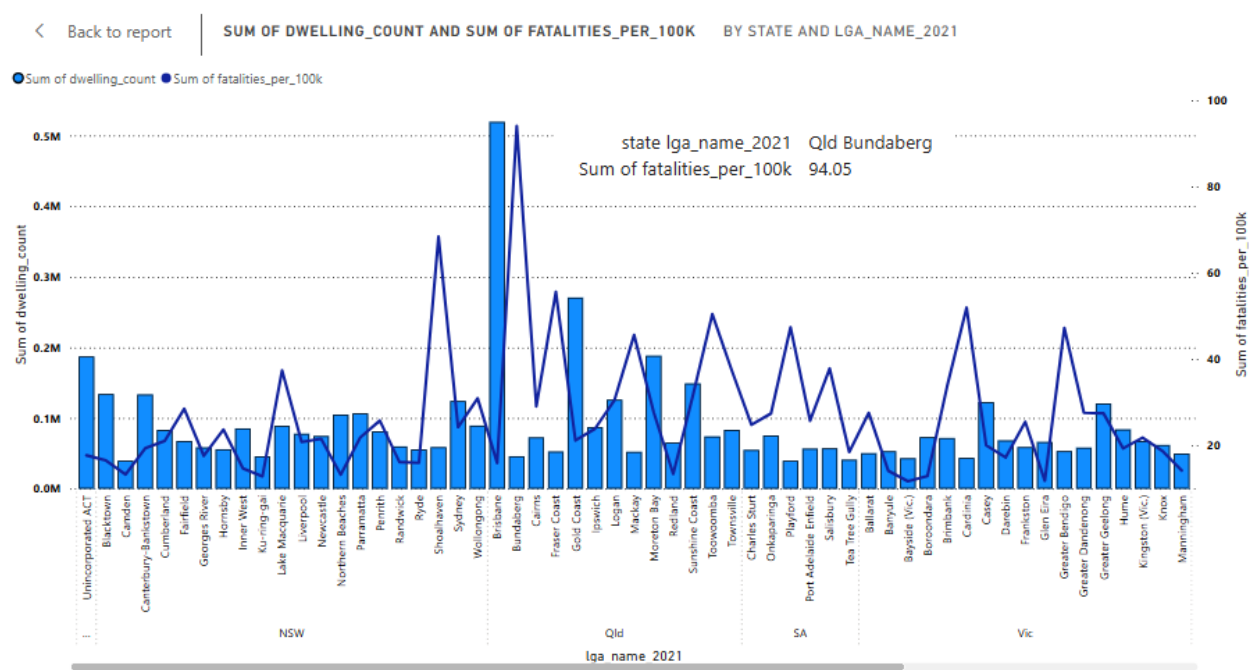
3) Which Local Government Areas (LGAs) have the highest fatality rates per 100,000 people, and how do they compare in terms of dwelling density?



We made two graphs. One was using a GeoJSON file, and the other graph for deeper insights. We observed that some areas have low dwelling count but high fatality rates, such as Bundaberg in QLD, Shoalhaven in NSW, Playford in SA, Cardinia in Vic, Armadale in WA. The Highest fatality is in Bundaberg with 95 people.



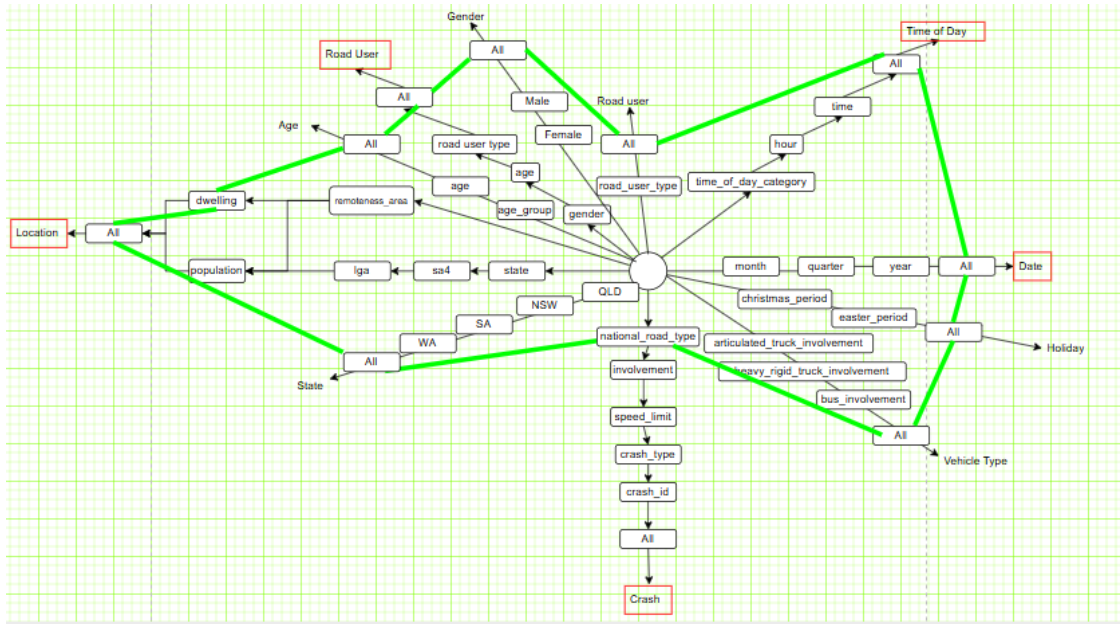
(Was made using GeoJSON files in Power BI)



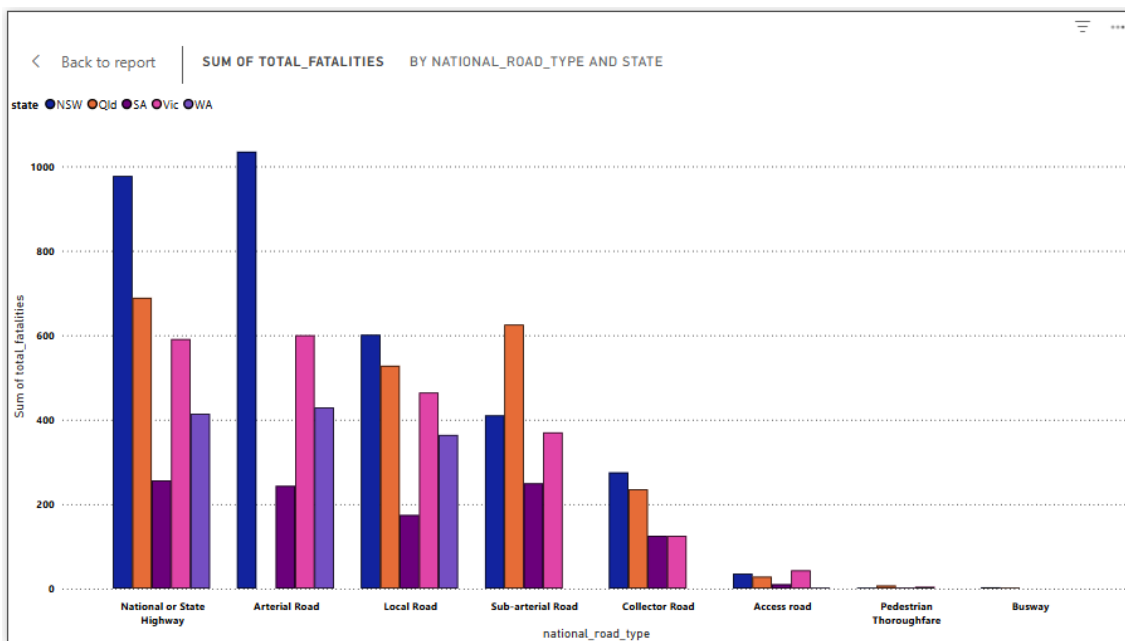
Recommendations

Authorities should implement micro-level traffic safety interventions in high-risk LGAs such as Bundaberg, Shoalhaven, and Armadale. This includes deploying speed cameras in known black spots, upgrading road geometry and visibility, and enhancing pedestrian infrastructure near residential zones. Additionally, launch community-specific awareness campaigns focusing on local crash patterns and collaborate with local councils to introduce data-driven enforcement schedules. These targeted, evidence-based actions can help reduce fatalities effectively without overextending statewide resources.

4) Which road types account for the highest number of fatalities in the top 5 most-affected states?



From the graph, we observed that NSW has the highest fatality rate in almost every type of road. QLD has the highest fatalities on National/State Highways and Sub-Arterial Roads. SA has almost the same fatality in every road. Victoria has the highest fatality rate in National/State and Arterial Roads. WA has high rates on the National Highways, Arterial and Local Road.



Recommendations

State transport authorities should adopt road-type-specific safety strategies. For example, in NSW where fatalities span all road types, a comprehensive statewide policy focusing on driver behaviour, enforcement, and signage is essential. In QLD and WA, where highways dominate

fatality counts, focus should be on speed control measures, highway patrol visibility, and safe overtaking lanes. For SA and Victoria, authorities should prioritize intersection redesign and pedestrian safety across various road types. Tailoring interventions based on road type will ensure more effective allocation of resources and lead to measurable reductions in road trauma.

Association Rule Mining

What Algorithm Was Used?

We applied the **FP-Growth (Frequent Pattern Growth) algorithm** — a fast and memory-efficient method to discover frequent itemsets without candidate generation. It works well for large datasets and is implemented using the `mlxtend` library in Python. The script was adapted from `mining.ipynb`, where we one-hot encoded our cleaned data, dropped columns with low variance, and then applied the FP-Growth algorithm with a minimum support of 2%.

```

FP-Growth Algorithm

min_support_threshold = 0.02
frequent_itemsets = fpgrowth(
    ohe_df, min_support=min_support_threshold, use_colnames=True
)
print(f"--- Frequent Itemsets (min_support={min_support_threshold}) ---")
print(frequent_itemsets)

[14]
...
--- Frequent Itemsets (min_support=0.02) ---
      support      itemsets
0    0.717886    (Gender=Male)
1    0.555868    (Crash Type=Single)
2    0.451523    (Road User=Driver)
3    0.430070    (Time of day=Night)
4    0.363510    (Speed Limit=Speed81-100)
...
2286 0.024762    (Gender=Male, Month=1, Crash Type=Multiple)
2287 0.021418    (Month=1, Road User=Driver, Time of day=Day)
2288 0.026874    (Gender=Male, Month=1, Road User=Driver)
2289 0.021911    (Gender=Male, Month=1, Speed Limit=Speed81-100)
2290 0.020239    (Month=1, Speed Limit=Speed81-100, Time of day=...

[2291 rows x 2 columns]

```

```

ohe_df = ohe_df[[col for col in ohe_df.columns if not col.endswith('No')]]
ohe_df = ohe_df[[col for col in ohe_df.columns if not col.endswith('Undetermined')]]
ohe_df = ohe_df[[col for col in ohe_df.columns if not col.endswith('Unknown')]]
print(ohe_df.sum().sort_values(ascending=False))

Gender=Male                40791
Time of day=Day             32384
Crash Type=Single           31585
Speed Limit=Speed41-80      27543
Road User=Driver            25656
...
Easter Period=Yes           335
National Road Type=Access road  157
Road User=Other/-9          120
National Road Type=Pedestrian Thoroughfare  14
National Road Type=Busway    4
Length: 69, dtype: int64

```

Generating Association Rules

Then we filtered out the rules in the right-hand column that contain "road user", sorted them by confidence and lift respectively, and exported them as CSV files for analysis.

```

Generate Association Rules

# Generate rules from the frequent itemsets
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.65)
result = rules[["antecedents", "consequents", "support", "confidence", "lift"]]

# Filter rules to include only those with "Road User" in the consequents
result = result[result["consequents"].apply(lambda x: "Road User" in str(x))]

# Sort the rules by confidence
result = result.sort_values(by="confidence", ascending=False)
result.to_csv("output/mining/association_rules_confidence.csv", index=False)

[1]

rules_lift = association_rules(frequent_itemsets, metric="lift", min_threshold=4.5)
result = rules_lift[["antecedents", "consequents", "support", "confidence", "lift"]]

# Filter rules to include only those with "Road User" in the consequents
result = result[result["consequents"].apply(lambda x: "Road User" in str(x))]

# Sort the rules by lift
result.sort_values(by="lift", ascending=False)
result.to_csv("output/mining/association_rules_lift.csv", index=False)

[1]

rules["antecedents_len"] = rules["antecedents"].apply(lambda x: len(x))
multi_condition_rules = rules[rules["antecedents_len"] >= 2]

strong_rules = rules[
    (rules["lift"] > 2) & (rules["confidence"] > 0.6) & (rules["support"] > 0.03)
]

strong_rules.to_csv("output/mining/strong_association_rules.csv", index=False)

[37]

```

Interpretation of top rules and suggestions

By Confidence (From association_rules_confidence.csv):

7. Elderly pedestrians involved in single crashes:

- Antecedents: Speed limit between 41-80 km/h, age 75 or older, single vehicle crash
- Consequence: Pedestrian involvement
- Confidence: Approximately 75.6%
- Lift: Very high (4.91), indicating a significantly increased likelihood of pedestrian involvement under these conditions.

8. Male drivers with articulated trucks in high-speed zones (81-100 km/h):

- Antecedents: Male, articulated truck involvement, speed limit 81-100 km/h
- Consequence: Driver involvement
- Confidence: Around 74.6%
- Lift: Moderate (1.65), showing a notable but less dramatic increase in likelihood of driver involvement.

1. Male drivers with articulated trucks in multiple-vehicle crashes:

- Antecedents: Male, articulated truck, speed limit 81-100 km/h, multiple crash type
- Consequence: Driver involvement
- Confidence: 73.7%, Lift: moderate (1.63).

2. Daytime elderly pedestrian incidents (age ≥ 75):

- Antecedents: Speed limit 41-80 km/h, age ≥ 75 , daytime, single crash
- Consequence: Pedestrian involvement
- Confidence: 73.6%, Lift: very high (4.78).

3. Middle-aged drivers (40-64 years) in articulated truck accidents:

- Antecedents: Articulated truck involvement, age group 40-64
- Consequence: Driver involvement
- Confidence: 70.6%, Lift: moderate (1.56).

4. Young female passengers (age ≤ 16):

- Antecedents: Female, age ≤ 16
- Consequence: Passenger involvement
- Confidence: 69.9%, Lift: high (3.08), suggesting high likelihood of being passengers in accidents.

5. General articulated truck involvement (male drivers):

- Antecedents: Male, articulated truck (varied additional contexts, e.g., time of day)
- Consequence: Driver involvement
- Confidence ranges from 67.0% to 68.6%, with moderate lift values (1.48–1.52), consistently indicating increased driver involvement likelihood.

Key Insights:

- **Pedestrian** risk is significantly higher for **elderly individuals** involved in **single-vehicle** incidents, especially at **moderate speeds**.
- **Driver involvement** is notably higher in scenarios **involving articulated trucks**, particularly **males at higher speed limits** and in **multiple-vehicle incidents**.

- **Young females** are **predominantly passengers** rather than drivers or pedestrians in crash situations.

By Lift (From `association_rules_confidence.csv`):

9. Elderly pedestrians (≥ 75 years) involved in single-vehicle daytime crashes at moderate speeds (41-80 km/h):

- Antecedents: Speed limit 41-80 km/h, Age ≥ 75 , Single crash type
- Consequents: Pedestrian involvement during daytime
- Lift: **7.78** (Very strong association suggesting high pedestrian vulnerability under these conditions.)

10. Elderly pedestrians involved in single-vehicle crashes (≥ 75 years) predominantly during the day:

- Antecedents: Age ≥ 75 , Single crash
- Consequents: Pedestrian involvement during daytime
- Lift: **5.99** (Strongly indicating elderly pedestrians' increased risk during daytime single-vehicle incidents.)

11. Moderate speed (41-80 km/h) single-vehicle daytime crashes associated strongly with elderly pedestrian involvement:

- Antecedents: Speed limit 41-80 km/h, Daytime, Single crash
- Consequents: Pedestrian aged ≥ 75
- Lift: **5.59** (Clearly indicates elderly pedestrian involvement is exceptionally high under these conditions.)

12. High-speed (> 100 km/h) single-vehicle crashes strongly associated with drivers in Western Australia (WA):

- Antecedents: Speed > 100 km/h, Single crash
- Consequents: Driver involvement in WA
- Lift: **5.13** (Very high association indicating WA drivers significantly involved in high-speed single crashes.)

13.High-speed (>100 km/h) crashes strongly associated with single-vehicle driver involvement in WA:

- Antecedent: Speed >100 km/h
- Consequents: Driver, Single crash, State=WA
- Lift: **4.92** (Strongly suggests a pattern of WA drivers being involved in single-vehicle, high-speed crashes.)

14.Elderly pedestrians (≥ 75) at moderate speeds (41-80 km/h), daytime single crashes:

- Antecedents: Age ≥ 75 , Single crash
- Consequents: Pedestrian, Speed 41-80 km/h, Daytime
- Lift: **6.44** (Strongly indicates elderly pedestrian vulnerability specifically at moderate speeds during daytime single crashes.)

Key Insights:

- **Elderly pedestrians (75+)** have an extremely high association with pedestrian crashes during **daytime at moderate speeds (41-80 km/h)**.
- Crashes involving **high speeds (>100 km/h)** have a notable pattern of **single-vehicle driver involvement, specifically in Western Australia**.

Top 3 recommendations to the government (From strong_association_rules.csv)

Recommendation 1: Enhanced Safety for Pedestrians at Moderate Speed Zones (41-80 km/h)

Rationale:

The rules strongly indicate that pedestrian-related crashes frequently occur at moderate speeds (41-80 km/h), especially involving single-vehicle incidents. Confidence levels exceed **70%**, and lift values consistently surpass **2.7**, signifying strong associations. The issue is even more critical during the **daytime** and especially pronounced for elderly and female pedestrians.

Recommended Actions:

- **Infrastructure Improvements:**

- Clearly marked pedestrian crossings, speed bumps, raised pedestrian pathways, and refuge islands at pedestrian-heavy locations.
- Traffic calming measures such as narrowing roads at pedestrian crossing points to naturally slow vehicles down.
- **Speed Regulation Enforcement:**
 - Targeted enforcement campaigns (speed cameras, regular patrols) at 41-80 km/h zones near shopping centres, hospitals, aged-care homes, and schools.
- **Awareness and Education Campaigns:**
 - Public awareness campaigns targeted at drivers emphasizing pedestrian safety in moderate-speed urban/suburban areas, specifically highlighting vulnerable pedestrian groups such as elderly and female pedestrians.

Recommendation 2: Protection Measures for Elderly Pedestrians (Age ≥ 75)

Rationale:

Multiple high-confidence ($>70\%$) rules show that elderly pedestrians are particularly vulnerable, especially in single-vehicle crashes during daytime. Lift values are notably high (ranging from **4.6 to 7.8**), strongly associating elderly pedestrians with crash scenarios.

Recommended Actions:

- **Elderly-Friendly Infrastructure:**
 - Longer pedestrian crossing signals at intersections frequented by elderly populations.
 - Improved pedestrian visibility, including better lighting, clearer signage, and anti-slip surfaces in walking areas.
- **Driver Awareness Programs:**
 - Public education and advertising campaigns to sensitize drivers about the vulnerability of elderly pedestrians, emphasizing cautious driving near retirement homes, community centres, and healthcare facilities.
- **Community Engagement:**
 - Initiatives promoting pedestrian safety training workshops for elderly people to encourage safe walking behaviours and increase situational awareness.

Recommendation 3: Specific Interventions for Motorcyclists in Multiple-Vehicle Crashes During the Day

Rationale:

The rule highlighting motorcyclists involved in multiple-vehicle crashes during daytime with male involvement exhibits substantial lift (over **2.1**) and high confidence (approx. **66.5%**). Even though slightly below 80%, it remains significant and indicates a clear risk area requiring targeted intervention.

Recommended Actions:

- **Motorcycle Awareness Campaigns:**
 - Educational programs directed toward both motorcyclists and drivers, promoting safe driving behaviours and mutual awareness on shared roads.
- **Infrastructure Improvements:**
 - Design or adjust road layouts to reduce conflict points, especially in intersections or merging lanes frequently used by motorcycles.
 - Increase signage and visible road markings to alert drivers about high motorcycle activity areas.
- **Enhanced Enforcement and Safety Checks:**
 - Periodic roadside checks for motorcyclists focusing on safety gear, vehicle condition, and riding practices, especially in urban and suburban environments with heavy traffic.

References

- https://www.sacramentoduiinformation.com/drunk-driving-is-a-major-issue-between-christmas-and-new-year-s-eve?utm_source=chatgpt.com
- Lecture Slides week 1,2,3,4
- Google Gemini 2.5 Pro and ChatGPT 4o have been used to generate the ETL script with thorough functional testing to ensure a complete understanding of how the code works. Also, assist in summarizing practical suggestions from the mined CSV rules.
- <https://rivory.io/data-learning-center/snowflake-schema-in-data-warehousing/>

Appendix

Codes for creating tables and populating the database

They are also given in a separate file.

```
1 CREATE EXTENSION IF NOT EXISTS postgres_fdw;
2
3 -- Create the foreign schema
4 CREATE SCHEMA FOREIGN_SCHEMA;
5
6 -- Create the foreign tables
7 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_date (
8     year integer,
9     month integer,
10    month_name text,
11    quarter integer,
12    month_year integer,
13    christmas_period boolean,
14    easter_period boolean
15) SERVER postgres_fdw OPTIONS (schema 'dim_date');
16
17 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_time_of_day (
18     time_of_day_category text,
19     hour_24 integer,
20     time text
21) SERVER postgres_fdw OPTIONS (schema 'dim_time_of_day');
22
23 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_road_user (
24     road_user_type text,
25     gender text,
26     age_group text
27) SERVER postgres_fdw OPTIONS (schema 'dim_road_user');
28
29 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_crash (
30     crash_id integer,
31     crash_type text,
32     bus_involvement boolean,
33     heavy_rigid_truck_involvement boolean,
34     articulated_truck_involvement boolean,
35     speed_limit integer,
36     national_road_type text
37) SERVER postgres_fdw OPTIONS (schema 'dim_crash');
38
39 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_location (
40     state text,
41     saf_name_2021 text,
42     lga_name_2021 text,
43     remoteness_area text,
44     location_level text,
45     lga_geom geometry,
46     saf_geom geometry,
47     state_geom geometry
48) SERVER postgres_fdw OPTIONS (schema 'dim_location');
49
50 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_population_lga (
51     location_key integer,
52     year integer,
53     population integer
54) SERVER postgres_fdw OPTIONS (schema 'dim_population_lga');
55
56 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_population_remoteness (
57     location_key integer,
58     year integer,
59     population integer
60) SERVER postgres_fdw OPTIONS (schema 'dim_population_remoteness');
61
62 CREATE FOREIGN TABLE FOREIGN_SCHEMA.dim_dwelling2 (
63     location_key integer,
64     year integer,
65     dwelling_count integer
66) SERVER postgres_fdw OPTIONS (schema 'dim_dwelling2');
67
68 CREATE FOREIGN TABLE FOREIGN_SCHEMA.fact_fatality (
69     date_key integer,
70     time_key integer,
71     location_key integer,
72     crash_key integer,
73     road_user_key integer,
74     fatality_count integer
75) SERVER postgres_fdw OPTIONS (schema 'fact_fatality');
```

```
1 COPY dim_date (year, month, month_name, quarter, month_year, christmas_period, easter_period)
2 FROM '/tmp/dim_date.csv' DELIMITER ',' CSV HEADER;
3
4 COPY dim_time_of_day (time_of_day_category, hour_24, time)
5 FROM '/tmp/dim_timeofday.csv' DELIMITER ',' CSV HEADER;
6
7 COPY dim_road_user (road_user_type, gender, age_group)
8 FROM '/tmp/dim_roaduser.csv' DELIMITER ',' CSV HEADER;
9
10 COPY dim_crash (crash_id, crash_type, bus_involvement, heavy_rigid_truck_involvement, articulated_truck_involvement, speed_limit, national_road_type)
11 FROM '/tmp/dim_crash.csv' DELIMITER ',' CSV HEADER;
12
13 COPY dim_location (state, saf_name_2021, lga_name_2021, remoteness_area, location_level, lga_geom, saf_geom, state_geom)
14 FROM '/tmp/dim_location.csv' DELIMITER ',' CSV HEADER;
15
16 COPY dim_population_lga (location_key, year, population)
17 FROM '/tmp/dim_population_lga.csv' DELIMITER ',' CSV HEADER;
18
19 COPY dim_population_remoteness (location_key, year, population)
20 FROM '/tmp/dim_population_remoteness.csv' DELIMITER ',' CSV HEADER;
21
22 COPY dim_dwelling2 (location_key, year, dwelling_count)
23 FROM '/tmp/dim_dwelling.csv' DELIMITER ',' CSV HEADER;
24
25 COPY fact_fatality (date_key, time_key, location_key, crash_key, road_user_key, fatality_count)
26 FROM '/tmp/fact_fatality.csv' DELIMITER ',' CSV HEADER;
```

Codes for SQL queries

Q-1

```
SELECT
-- Categorize each crash into a vehicle involvement type
CASE
-- If more than one heavy vehicle type is involved
WHEN
    (c.bus_involvement = 'Yes')::int +
    (c.heavy_rigid_truck_involvement = 'Yes')::int +
    (c.articulated_truck_involvement = 'Yes')::int > 1
    THEN 'Multiple Vehicle Types'
-- If only a bus is involved
WHEN c.bus_involvement = 'Yes' THEN 'Bus'
-- If only a heavy rigid truck is involved
WHEN c.heavy_rigid_truck_involvement = 'Yes' THEN 'Heavy Rigid Truck'
-- If only an articulated truck is involved
WHEN c.articulated_truck_involvement = 'Yes' THEN 'Articulated Truck'
-- If no specific heavy vehicle type is involved
ELSE 'None'
END AS vehicle_involvement_category,

-- Count the number of fatalities for each vehicle involvement category
COUNT(*) AS total_fatalities,

-- Calculate the average speed limit for each vehicle involvement category
ROUND(AVG(
    CASE
        -- Include only numeric speed limit values
        WHEN c.speed_limit ~ '^\d+$' THEN c.speed_limit::numeric
        ELSE NULL -- Skip non-numeric or invalid entries
    END
), 2) AS avg_speed_limit

-- Join the fact table with crash dimension to get vehicle and speed data
FROM fact_fatality f
JOIN dim_crash c ON f.crash_key = c.crash_key
-- Group the results by the custom vehicle involvement category
GROUP BY vehicle_involvement_category
-- Sort the output by the number of fatalities (descending)
ORDER BY total_fatalities DESC;
```

Q-2

Q-2

```
SELECT
-- Select the type of road user (e.g., Driver, Pedestrian)
dr.road_user_type,

-- Select the age group of the road user
dr.age_group,

-- Select the gender of the road user
dr.gender,

-- Count the number of fatalities for each combination of road user type, age group, and gender
COUNT(*) AS total_fatalities

-- Join the fact table with the road user dimension to get demographic details
FROM fact_fatality ff
JOIN dim_road_user dr ON ff.road_user_key = dr.road_user_key

-- Filter the data to include only Drivers and Pedestrians (excluding passengers and unknowns)
WHERE dr.road_user_type IN ('Driver', 'Pedestrian')

-- Group the result by road user type, age group, and gender to count fatalities for each subgroup
GROUP BY dr.road_user_type, dr.age_group, dr.gender

-- Sort the result so we can compare fatalities within each user type
ORDER BY dr.road_user_type, total_fatalities DESC;
```

Q-3

```
Q-3
SELECT
    -- Check if the crash occurred during the Christmas holiday period
    dd.christmas_period,

    -- Check if the crash occurred during the Easter holiday period
    dd.easter_period,

    -- Extract the hour of the day (in 24-hour format) when the crash occurred
    td.hour_24,

    -- Time of day category (e.g., Morning, Afternoon, Night)
    td.time_of_day_category,

    -- Count the number of fatalities that occurred under each combination of holiday and hour
    COUNT(*) AS total_fatalities

-- Join the fact table with the date dimension to get holiday info
FROM fact_fatality ff
JOIN dim_date dd ON ff.date_key = dd.date_key

-- Join with the time dimension to extract hour and time of day
JOIN dim_time_of_day td ON ff.time_key = td.time_key

-- Filter to include only fatalities that occurred during either Christmas or Easter
WHERE dd.christmas_period = TRUE OR dd.easter_period = TRUE

-- Group by holiday type and hour to analyze time-based patterns of holiday crashes
GROUP BY
    dd.christmas_period,
    dd.easter_period,
    td.hour_24,
    td.time_of_day_category

-- Sort the output by hour of day so trends are visually sequential (midnight to 11 PM)
ORDER BY td.hour_24;
```

Q-4

```
Q-4
SELECT
    -- Select the name of the state where the crash occurred
    l.state,

    -- Select the Local Government Area (LGA) name , year and fatalities
    p.lga_name_2021,
    p.year,

    SUM(f.fatality_count) AS total_fatalities,

    -- Get the population of the LGA and dwelling count
    p.population,
    d.dwelling_count,

    -- Calculate fatalities per 100,000 population (normalized metric)
    ROUND((SUM(f.fatality_count)::numeric / p.population) * 100000, 2) AS fatalities_per_100k,

    -- Calculate average number of people per dwelling
    ROUND((p.population::numeric / d.dwelling_count), 2) AS avg_people_per_dwelling

-- Join the fact table with dim_date to ensure proper year filtering
FROM fact_fatality f
JOIN dim_date dt ON f.date_key = dt.date_key

-- Join with dim_location to get state and LGA names
JOIN dim_location l ON f.location_key = l.location_key

-- Join with population table on LGA name to bring in population info
JOIN dim_population_lga p ON l.lga_name_2021 = p.lga_name_2021

-- Join with dwelling data on LGA name to calculate people per dwelling
JOIN dim_dwelling d ON l.lga_name_2021 = d.lga_name_2021
```

```

-- Filter to include only recent population data (2022) and sizable LGAs
WHERE p.year = 2022
AND p.population > 100000

-- Group results by geographic and population info for aggregation
GROUP BY
    l.state,
    p.lga_name_2021,
    p.year,
    p.population,
    d.dwelling_count

-- Order by fatalities per 100k to find highest-risk areas
ORDER BY fatalities_per_100k DESC;

```

Q-5

Q-5

```

-- First, create a Common Table Expression (CTE) to find the top 5 states with the most fatalities
WITH state_totals AS (
    SELECT
        l.state,
        COUNT(*) AS total_fatalities
    FROM fact_fatality f
    JOIN dim_location l ON f.location_key = l.location_key
    GROUP BY l.state
    ORDER BY total_fatalities DESC
    LIMIT 5
)

-- Now, for those top 5 states, break down fatalities by road type using ROLLUP
SELECT
    l.state,
    c.national_road_type,
    COUNT(*) AS total_fatalities
FROM fact_fatality f
JOIN dim_location l ON f.location_key = l.location_key
JOIN dim_crash c ON f.crash_key = c.crash_key
JOIN state_totals s ON l.state = s.state
GROUP BY ROLLUP (l.state, c.national_road_type)

-- Use ROLLUP to also include subtotals per state and a grand total
GROUP BY ROLLUP (l.state, c.national_road_type)

-- Order results for easy interpretation
ORDER BY l.state, c.national_road_type;

```

Visuals

