Road Safety Data Warehouse and Analysis Project

Team Information

- Huixian Xu (24120279)
- Swapnil Gaikwad (24060283)

1. Introduction

This data warehousing project aims to support government and public understanding of road safety by analyzing historical data on fatal crashes. The primary objectives are to build a comprehensive data warehouse that stores road accident data, present key insights through visual dashboards, and apply data mining techniques to support decision-making around road safety policies. Through the development of a well-structured dimensional model following Kimball's methodology, this project seeks to identify patterns, trends, and risk factors in fatal road crashes that can inform targeted interventions to reduce traffic fatalities.

2. Data Warehouse Design

2.1 Process Identification

This data warehouse design models the process of fatal crashes in Australia, with each car crash accident being recorded as single transaction.

Based on the structure of several dataset, there are several business questions which data warehouse should be able to answer:

- How do fatal crash patterns vary across different geographic areas and road types, and which combinations present the highest risk?
- How do different types of road users experience different fatality patterns?
- How do fatality rates vary by month throughout the year?
- How does articulated truck involvement affect crash severity?
- How do different road types correlate with crash fatality rates?

2.2 Grain Determination

In this case, the most valuable grain unit would be crash events, which means each row will represent a single crash incident. The reason why we choose crash events as the grain is that it can maximize analytical flexibility - we can track multiple metrics per crash (such as fatality count, injury severity, and risk factors) while still enabling fatality-focused analysis through aggregation. This approach allows us to analyze data both at the individual crash level and to summarize fatality patterns across different dimensions.

2.3 Dimension Selection

For our crash analysis data warehouse, we have carefully selected the following dimensions to provide comprehensive analytical capabilities:

1. **Victim Dimension:** Helps us understand who is affected by crashes like age, gender group to create better safety programs for vulnerable groups.

2. **Crash Event Dimension:** Shows what kind of crashes are happening and why, helping us understand root causes.

3. **Date Dimension:** Reveals patterns by season, month, or holiday periods when crashes might increase.

4. **Time of Day Dimension:** Identifies dangerous times (rush hour, night) when certain safety measures might be needed.

5. **Location Dimension:** Shows where crashes happen most often so we can focus improvements in those areas.

6. **Remoteness Dimension:** Distinguishes between urban and rural crash patterns, which often have different causes and solutions.

7. **Road Characteristics Dimension:** Shows how road type, speed limit can affect the crashes.

8. **Vehicle Involvement Dimension:** Provides insights on which vehicle types are involved and how many types of vehicle contribute to crashes.

2.4 Measures Identification

Primary Measures

Fatalities: The core measure tracking the number of deaths per crash, essential for all safety analyses.

Derived Measures

1. **Normalized Fatality Rate per 100k Dwellings** - Allows for fair comparison between different areas by adjusting for population density, making it possible to identify locations with disproportionately high fatality rates regardless of their size.

2. **Risk Factor Score** - Combines multiple risk elements (time of day, speed limits, road type, weekend status) into a single score, helping prioritize interventions by identifying which combination of factors creates the highest risk situations.

2.5 Schema Design

2.5.1 StarNet diagram with concept hierarchies

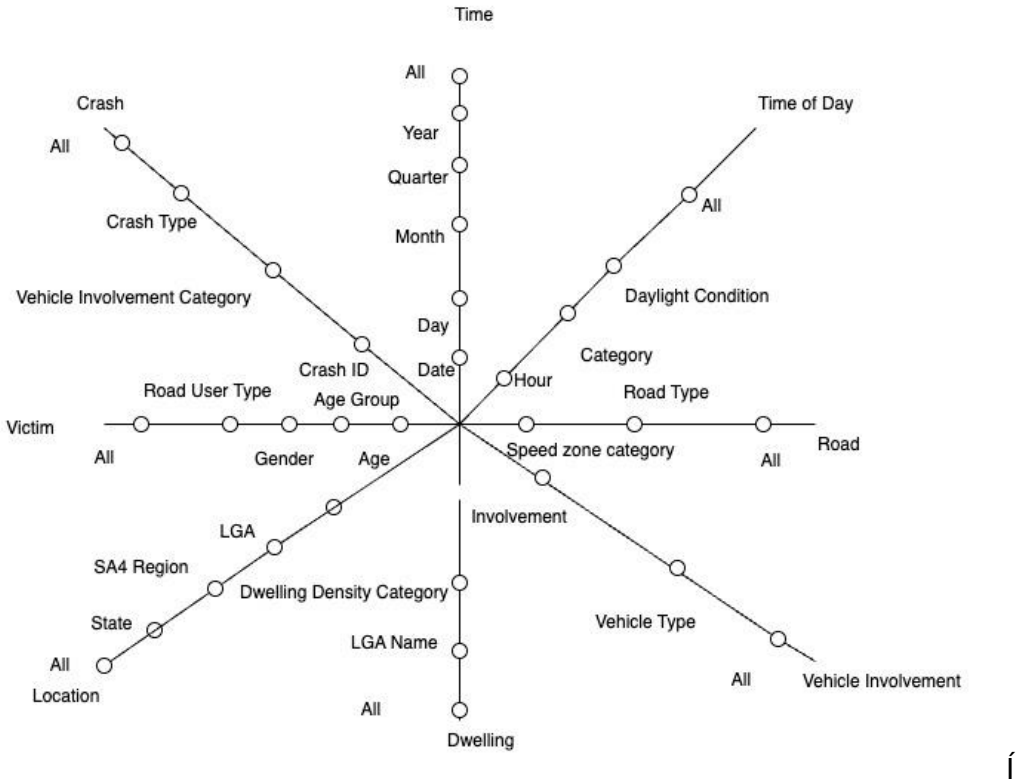

Figure 1: StarNet Diagram for Road Safety Data Warehouse

Diagram Annotations:

- The StarNet shows how each dimension connects to the central fact entity, allowing for multi-dimensional analysis of fatal crashes across various attributes.
- The central node represents fatal crash incidents, with each fact corresponding to an individual crash event. This grain selection enables detailed analysis at the crash level.

2.5.2 Star Schema Design for Road Safety Data Warehouse

The schema follows a star design with clear surrogate keys in each dimension and appropriate measures in the fact table. This design supports efficient querying across multiple dimensions while maintaining data integrity through proper foreign key relationships.

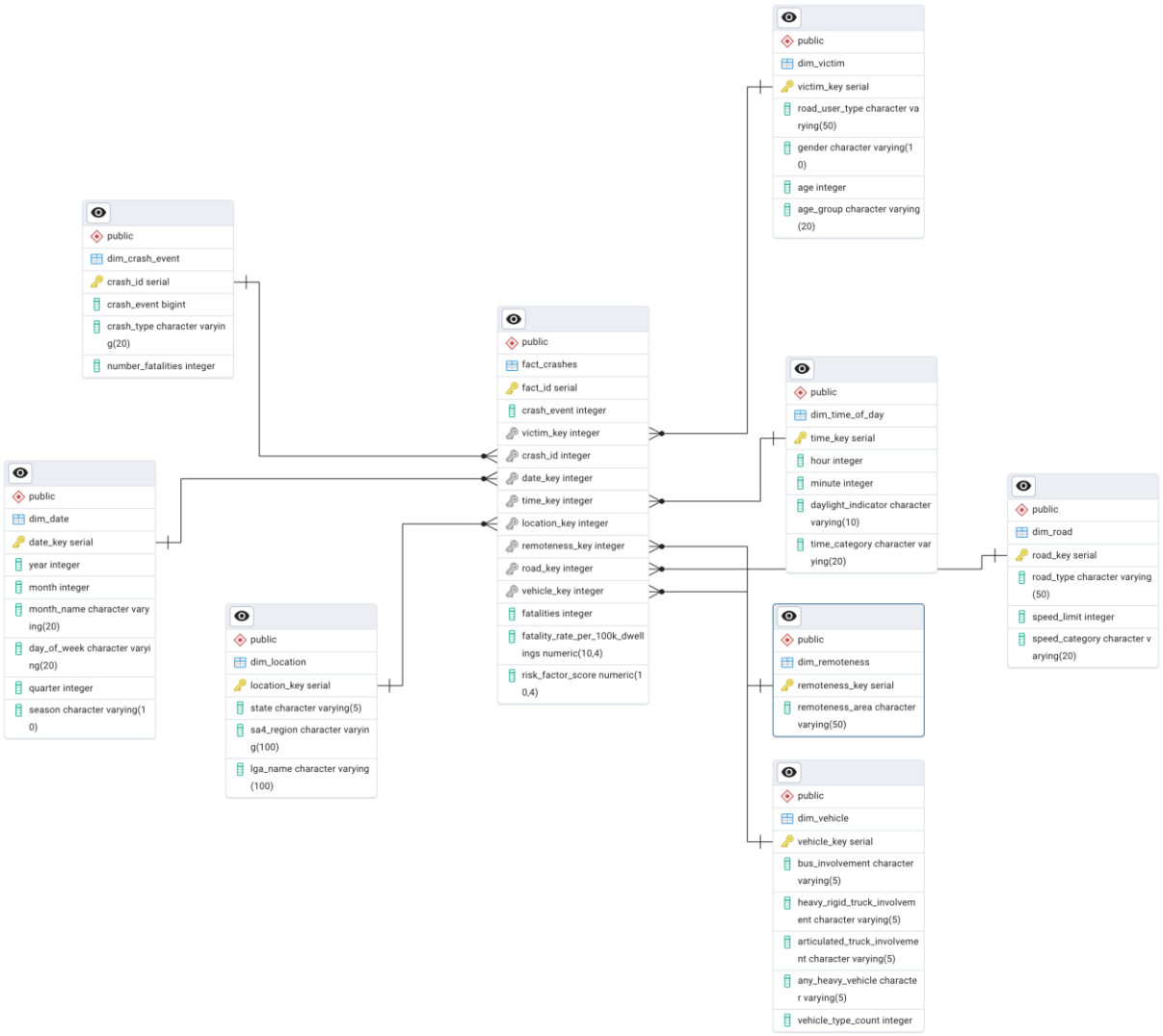

Figure 2: Star Schema Design for Road Safety Data Warehouse

3. Data Cleaning and ETL Process

3.1 Data Cleaning Strategy

3.1.1 Data Sources in Detail

Primary Data Sources:

1. BITRE Fatal Crash Data (bitre_fatal_crashes_dec2024.xlsx)

    a. Contains information about individual crash incidents
    b. Includes spatial, temporal, and circumstantial information
    c. Records extend from 1989 to 2024
    d. Data is organized at the crash level (one row per crash)
2. BITRE Fatality Data (bitre_fatalities_dec2024.xlsx)
    a. Contains information about individual fatalities
    b. Includes demographic information (age, gender)
    c. Contains road user type (driver, passenger, pedestrian, etc.)
    d. Data is organized at the person level (one row per fatality)

Supplementary Data Sources:

3. LGA Dwelling Count Data (LGA (count of dwellings).csv)
    a. Contains dwelling counts by Local Government Area (LGA)
    b. Used to provide context for crash rates relative to population density
4. Geographic Boundary Data (LGA_2021_AUST_GDA94.geojson)
    a. Contains geographic boundary information for Australian LGAs
    b. Used to derive centroid coordinates for spatial analysis
    c. Follows the 2021 Australian Bureau of Statistics boundaries

3.1.2 Dataset Structure and Quality Assessment

Fatal Crash Dataset Structure: The BITRE Fatal Crash dataset includes the following key column types:

- Identifiers: Crash ID
- Location Information: State, National Remoteness Areas, SA4 Name 2021, National LGA Name 2021
- Temporal Information: Month, Year, Dayweek, Time, Christmas Period, Easter Period
- Crash Characteristics: Crash Type, Number Fatalities, Speed Limit, Road Type
- Vehicle Involvement: Bus Involvement, Heavy Rigid Truck Involvement, Articulated Truck Involvement

The dataset contains 51,284 crashes spanning from 1989 to 2024, with specific data quality issues including:

- Missing values coded as -9
- Inconsistent categorization

- Temporal data stored across multiple columns

Fatality Dataset Structure: The BITRE Fatality dataset includes the same crash-level information as the crash dataset, plus:

- Demographic Information: Age, Gender
- Crash Role: Road User (driver, passenger, etc.)
- Age Grouping: Pre-defined age groups

This dataset is organized at the individual level, with each row representing a person who died in a crash, resulting in one or more rows per crash ID.

3.1.3 Handling Missing Values

Our data quality assessment revealed that missing values were inconsistently represented, with -9 often used as a code for missing data. We standardized all missing values by replacing these codes with proper NULL values to ensure consistent handling throughout the ETL process.

3.1.4 Outlier Detection and Management

We implemented statistical methods to identify outliers, particularly in numerical fields such as speed limits and age data. Outliers were either corrected based on domain knowledge (e.g., speed limit values outside normal ranges) or flagged for special handling during analysis.

3.1.5 Data Standardization Approach

Our standardization strategy involved:

1. Converting categorical columns to consistent formats
2. Standardizing temporal data into proper datetime format
3. Normalizing geographic naming conventions
4. Creating consistent value ranges for numerical fields

3.2 ETL Implementation

3.2.1 Detailed Steps of Extraction, Transformation, and Loading

3.2.1 Detailed Steps of Extraction, Transformation, and Loading

Phase 1: Initial Data Exploration and Loading The process begins with loading the datasets and understanding their structure:

python
```
df_crash = pd.read_excel(file_path,
sheet_name='BITRE_Fatal_Crash', header=4)
df_fatality = pd.read_excel(file_path_1,
sheet_name='BITRE_Fatality', header=4)
```

Column data types were assessed to guide the cleaning process, which revealed:

- 4 purely numerical columns (Crash ID, Month, Year, Number Fatalities)
- 16 categorical columns (State, Dayweek, Crash Type, etc.)
- Several columns with mixed content requiring special handling

Phase 2: Preprocessing and Data Cleaning

Standardizing Missing Values:

python
```
# Replace -9 (coding for missing) with pandas NA
df_crash.replace(-9, pd.NA, inplace=True)
df_fatality.replace(-9, pd.NA, inplace=True)
```

Categorical Data Processing: For categorical data, the approach involved:

- Converting to categorical data type to optimize memory usage and enable category-specific operations
- Adding "Unknown" as an explicit category to maintain data integrity
- Filling missing values with "Unknown" where appropriate

python
```
# Define categorical columns
categorical_cols = [
```

```
    'State', 'Dayweek', 'Crash Type', 'Bus \nInvolvement',
    'Heavy Rigid Truck Involvement', 'Articulated Truck
Involvement',
    'National Remoteness Areas', 'SA4 Name 2021', 'National LGA
Name 2021',
    'National Road Type', 'Christmas Period', 'Easter Period',
'Day of week', 'Time of day'
]

# Process each categorical column
for col in categorical_cols:
    df_crash[col] = df_crash[col].astype('category')
    if "Unknown" not in df_crash[col].cat.categories:
        df_crash[col] =
df_crash[col].cat.add_categories("Unknown")
    df_crash[col].fillna("Unknown", inplace=True)
```

Numerical Data Processing: For numerical columns, the approach was to:

- Extract numeric values using regular expressions
- Convert to appropriate numeric type
- Fill missing values with median values to maintain central tendency

python

```
def clean_numeric_column(df, column_name):
    # Convert to string to handle mixed types
    df[column_name] = df[column_name].astype(str)
    # Extract only numeric patterns
    df[column_name] = df[column_name].str.extract(r'(\d+)')
    # Convert to numeric, coercing errors to NaN
    df[column_name] = pd.to_numeric(df[column_name],
errors='coerce')

# Apply to specific columns like Speed Limit
clean_numeric_column(df_crash, 'Speed Limit')
clean_numeric_column(df_fatality, 'Speed Limit')
```

```
# Fill numeric missing values with median
for col in numerical_cols:
    df_crash[col].fillna(df_crash[col].median(), inplace=True)
```

Special handling was implemented for the Speed Limit column:

python

```
# Replace '<40' with 40 in the 'Speed Limit' column
df_fatality['Speed Limit'] = df_fatality['Speed
Limit'].replace({'<40': 40})
```

Phase 3: Temporal Data Enhancement The code adds significant value by deriving useful temporal fields from the existing date and time columns:

python

```
# Ensure Year and Month are integers
df_crash['Year'] = df_crash['Year'].astype(int)
df_crash['Month'] = df_crash['Month'].astype(int)

# Clean and format Time column
df_crash['Time'] = df_crash['Time'].astype(str).str.strip()
df_crash['Time'] = df_crash['Time'].apply(lambda x: x if ":" in x
else "00:00:00")
df_crash['Time'] = pd.to_datetime(df_crash['Time'],
format='%H:%M:%S', errors='coerce')

# Create `Crash_Timestamp` combining Year, Month, and Time
df_crash['Crash_Timestamp'] = pd.to_datetime(df_crash[['Year',
'Month']].assign(DAY=1))

# Create `Crash_Date` (only date, for daily trends)
df_crash['Crash_Date'] = df_crash['Crash_Timestamp'].dt.date

# Create `Crash_Hour` (only hour, for hourly trends)
df_crash['Crash_Hour'] = df_crash['Crash_Timestamp'].dt.hour
```

```python
# Drop redundant columns to avoid repetition
df_crash.drop(columns=['Year', 'Month', 'Time'], inplace=True)
```

This transformation creates a proper timestamp field, enabling more sophisticated temporal analysis including trends by hour, day, month, and year.

Phase 4: Data Integration The project integrates the fatality dataset with the crash dataset, enriching the information available for each crash:

python

```python
# Find columns in df_fatality that are not in df_crash
missing_in_crash = set(df_fatality.columns) -
set(df_crash.columns)

# Add these columns to df_crash
for col in missing_cols:
    df_crash[col] = df_fatality[col]
```

Later, dwelling count data is integrated to provide population context:

python

```python
merged_df = crash_df.merge(
    lga_df,
    left_on="National LGA Name 2021",
    right_on="LGA",
    how="left"
).drop(columns=["LGA"])
```

Phase 5: Geographic Enrichment The final phase adds geographic coordinates to enable spatial analysis:

python

```python
# Load LGA GeoJSON
lga = gpd.read_file("GJFiles/LGA_2021_AUST_GDA94.geojson")

# Reproject to a projected CRS for accurate centroids
lga = lga.to_crs("EPSG:3577") # GDA94 / Australian Albers
```

```python
# Calculate centroids and convert back to lat/lon
lga['longitude'] = lga.geometry.centroid.to_crs(epsg=4326).x
lga['latitude'] = lga.geometry.centroid.to_crs(epsg=4326).y

# Keep only relevant columns
lga_coords = lga[['LGA_NAME21', 'latitude', 'longitude']]

# Merge using LGA name
df_merged = df_clean.merge(lga_coords, left_on='National LGA Name
2021', right_on='LGA_NAME21')
```

Any remaining missing coordinates are imputed using a sophisticated approach:

python

```python
# Function to fill missing values using LGA name
def fill_coords(row):
    if pd.isna(row['latitude']) or pd.isna(row['longitude']):
        match = lga_lookup.get(row['National LGA Name 2021'])
        if match:
            row['latitude'] = match['latitude']
            row['longitude'] = match['longitude']
    return row

# Apply coordinate fill
df = df.apply(fill_coords, axis=1)

# Fill any remaining with state-level averages
df['latitude'] = df.groupby('State')['latitude'].transform(lambda
x: x.fillna(x.mean()))
df['longitude'] =
df.groupby('State')['longitude'].transform(lambda x:
x.fillna(x.mean()))
```

The final stage involved structuring the data according to our dimensional model:

1. **Dimension Table Creation**:

    a. Created 8 dimension tables (Victim, Crash Event, Date, Time of Day, Location, Remoteness, Road, Vehicle Involvement)
    b. Generated surrogate keys for each dimension table to support proper referential integrity
2. **Fact Table Generation**:
    a. Created the central fact table (fact_crashes) with relationships to all dimension tables
    b. Added measures including fatalities count
3. **PostgreSQL Database Loading**:
    a. Used SQL scripts to create the schema in PostgreSQL
    b. Loaded the transformed data into the respective tables using COPY commands

The entire ETL process was carefully designed to ensure data quality and dimensional model integrity, with validation at each step.

3.2.2 Tools and Techniques Used

The ETL process utilized a comprehensive set of tools and technologies:

- **Python**: Primary programming language for the entire ETL process
- **Pandas**: Core library for data manipulation, transformation, and analysis
- **NumPy**: Used for numerical operations and advanced data handling
- **Scikit-learn**: Employed for advanced imputation techniques, including Random Forest algorithms
- **Matplotlib/Seaborn**: Used for data visualization and pattern recognition during the ETL process
- **Jupyter Notebooks**: Development environment that facilitated iterative development and documentation
- **PostgreSQL**: Database management system for the final data warehouse
- **SQL**: Used for creating the database schema and loading the data

Advanced techniques employed in the ETL process included:

1. **Machine Learning-Based Imputation**: Used Random Forest models to impute missing values in location data with high accuracy
2. **Statistical Data Validation**: Applied statistical methods to identify and handle outliers in the dataset

3. **Domain-Specific Transformation Rules**: Implemented business logic specific to road safety data

The ETL pipeline was built with scalability and reproducibility in mind, allowing for future dataset updates while maintaining consistency in the data warehouse.

3.3 Dimension and Fact Table Population

Approach for Populating Dimension Tables

The dimension tables in this data warehouse were populated following a systematic approach implemented in Python using the pandas library. Each dimension was created to represent a specific analytical aspect of the road safety data:

1. **Data Extraction and Initial Preparation**: The cleaned dataset (Cleaned_Crash_Dwelling_Data.xlsx) containing 51,284 records was loaded and used as the basis for dimension extraction. This dataset had previously undergone comprehensive cleaning to handle missing values, standardize formats, and correct inconsistencies.
2. **Dimension Extraction Strategy**: For each dimension, a focused subset of relevant attributes was extracted from the source dataset using the drop_duplicates() method to ensure uniqueness. This approach effectively normalized the data while preserving all necessary analytical attributes. For example:
3. **Surrogate Key Generation**: Sequential surrogate keys were generated for each dimension table by resetting the index and adding an incremental identifier. For instance:
4. **Derived Attribute Creation**: Several dimensions were enriched with derived attributes to enhance analytical capabilities:
    a. The **Date Dimension** was enhanced with season and quarter determinations
    b. The **Time of Day Dimension** included derived time categories (Morning, Afternoon, Evening, Night)
    c. The **Road Characteristics Dimension** incorporated speed categories based on speed limits
    d. The **Vehicle Involvement Dimension** added an "any_heavy_vehicle" flag and a count of vehicle types involved
5. **Data Type Standardization**: Column names were standardized to follow consistent naming conventions, making the schema more intuitive and database-friendly.

6. **Output Generation**: Each dimension table was exported to CSV format in a dedicated "dimension_tables" directory for subsequent database loading.

4. Data Warehouse Implementation

4.1 Database Setup and Configuration

PostgreSQL was selected as the database management system for this implementation due to its robust support for analytical workloads, advanced querying capabilities, and compatibility with business intelligence tools. The following configuration steps were taken to optimize the database for analysis:

1. **Database Creation**: A dedicated database instance was established specifically for the road safety data warehouse.
2. **Schema Organization**: Tables were organized within a logical schema structure to maintain clear separation between dimension and fact tables.

4.2 Table Creation and Structure

The implementation followed the star schema design with eight dimension tables and one central fact table. Each table was created with appropriate data types, constraints, and indexes to ensure data integrity and query performance.

Key implementation features included:

1. **Primary Key Constraints**: Each dimension table included a surrogate key as its primary key, enforcing uniqueness and providing efficient join paths.
2. **Foreign Key Constraints**: The fact table implemented referential integrity through foreign key constraints to all dimension tables, ensuring data consistency.
3. **Data Type Selection**: Appropriate data types were selected for each column to balance storage efficiency with query performance, such as using INTEGER for numeric keys and VARCHAR with appropriate lengths for text data.
4. **Not-Null Constraints**: Critical columns were defined with NOT NULL constraints to enforce data quality requirements.

4.4 Data Loading Process

The populated dimension and fact tables generated in the previous ETL process were loaded into the PostgreSQL database using an efficient bulk loading approach.

The loading process included:

1. **Sequential Loading**: Dimension tables were loaded first, followed by the fact table, to ensure all foreign key references were satisfied.
2. **Bulk Import**: The PostgreSQL COPY command was used for high-performance bulk data loading instead of row-by-row insertion.
3. **Transaction Management**: Data loading was performed within transactions to ensure atomicity and maintain database consistency.

5. Business Queries and Visualization

5.1 Business Query 1:

Which combination of state, remoteness area, and road type recorded the highest number of fatal crashes in 2024?

5.1.1 StarNet Diagram and Query Footprints

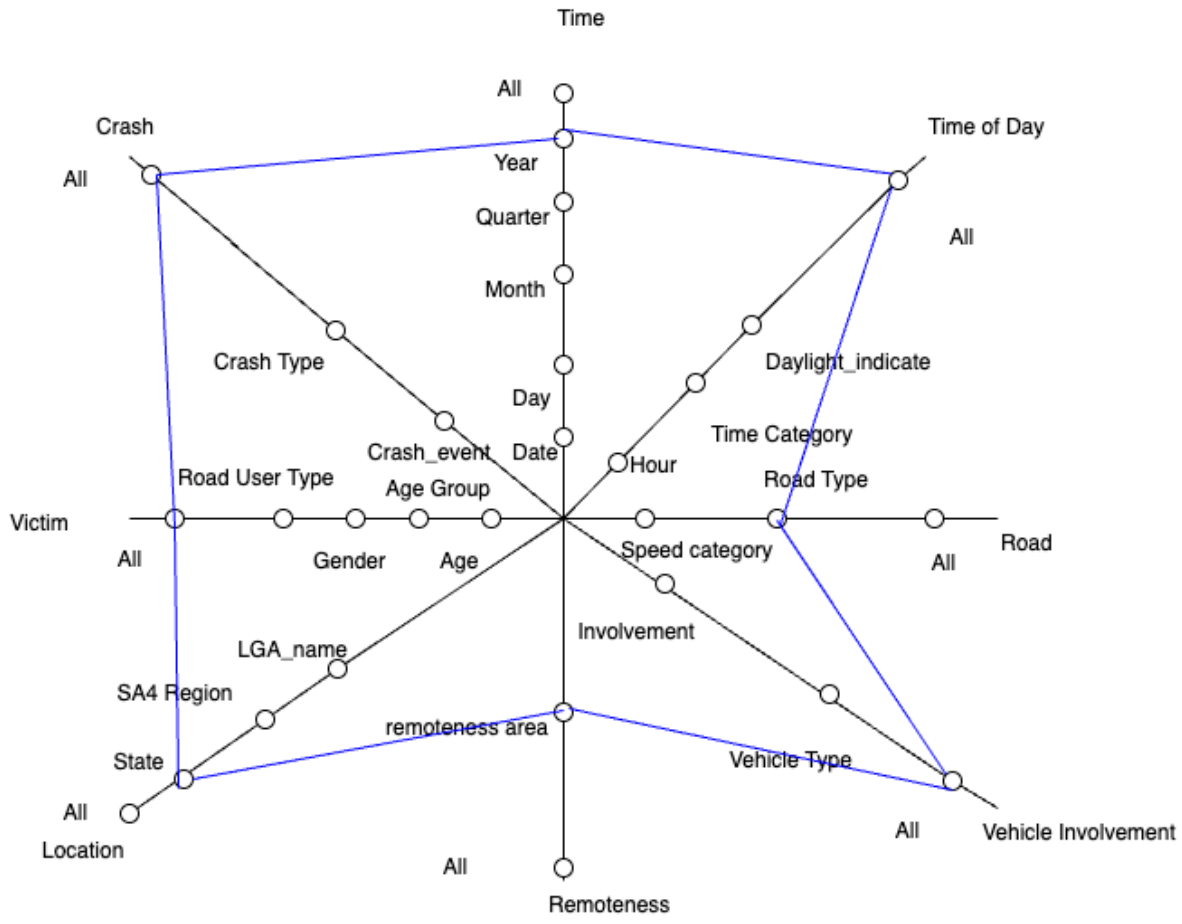

Figure 3: Query Footprints for Query 1

5.1.2 Visualization with analysis

The process involved:

1. Creating a heat map with remoteness areas on the y-axis and states on the x-axis
2. Adding road types as row headers within each remoteness area category
3. Using color intensity to represent fatal crash counts (darker colors indicating higher values)
4. Adding text labels displaying the actual crash counts in each cell
5. Applying a cohesive color palette (blue-green) with varying intensities based on crash numbers

Remotenes..	Road Type	State							
		ACT	NSW	NT	Qld	SA	Tas	Vic	WA
Inner Regional Australia	Access road				1		1	3	
	Arterial Road		34			10	4	25	
	Collector Road	1	1		10	3		2	
	Local Road		18		18	6	3	22	
	National or State Highway		37		24	1	1	28	
	Pedestrian Thoroughfare				1				
	Sub-arterial Road		22		22	9	2	23	
	Undetermined	2			1			41	24
Major Cities of Australia	Access road		1		1			1	
	Arterial Road	3	47			3		40	
	Busway		1						
	Collector Road	3	21		20	9		11	
	Local Road	1	20		29	3		21	
	National or State Highway	1	25		10	3		12	
	Pedestrian Thoroughfare				2			2	
	Sub-arterial Road		1		30	7		14	
	Undetermined				1	1			80
Outer Regional Australia	Access road		2				1	1	
	Arterial Road		17			4	4	7	
	Collector Road				10	1	2	1	
	Local Road		6		19	6		4	
	National or State Highway		32		37	3	8	6	
	Sub-arterial Road		16	2	14	3	2	3	
	Undetermined			43					64
Remote Australia	Access road				1				
	Collector Road				2	1			
	Local Road				1				
	National or State Highway		3		7	3		1	
	Sub-arterial Road		2		1	2			
Very Remote Australia	Arterial Road						1		
	Local Road				5	1			
	National or State Highway		2	1	2	4			
	Sub-arterial Road			1	4				
	Undetermined			4					

Figure 4: Heat Map Diagram for Query 1

5.1.3 Key insights

The heat map effectively visualizes the distribution of fatal crashes across different combinations of states, remoteness areas, and road types in Australia in 2024. Here are the key insights:

Western Australia (WA), Major Cities of Australia, Undetermined road type recorded the highest number of fatal crashes in 2024 with **80 incidents**.

If excluding "Undetermined" road types: The highest combination becomes **NSW, Inner Regional Australia, National or State Highway** with **37 fatal crashes**, closely followed by **WA, Major Cities of Australia, Arterial Road** with **40 fatal crashes**.

5.2 Business Query 2:

What is the distribution of fatalities among different road user types during evening period in 2023-2024?

5.2.1 StarNet Diagram and Query Footprints

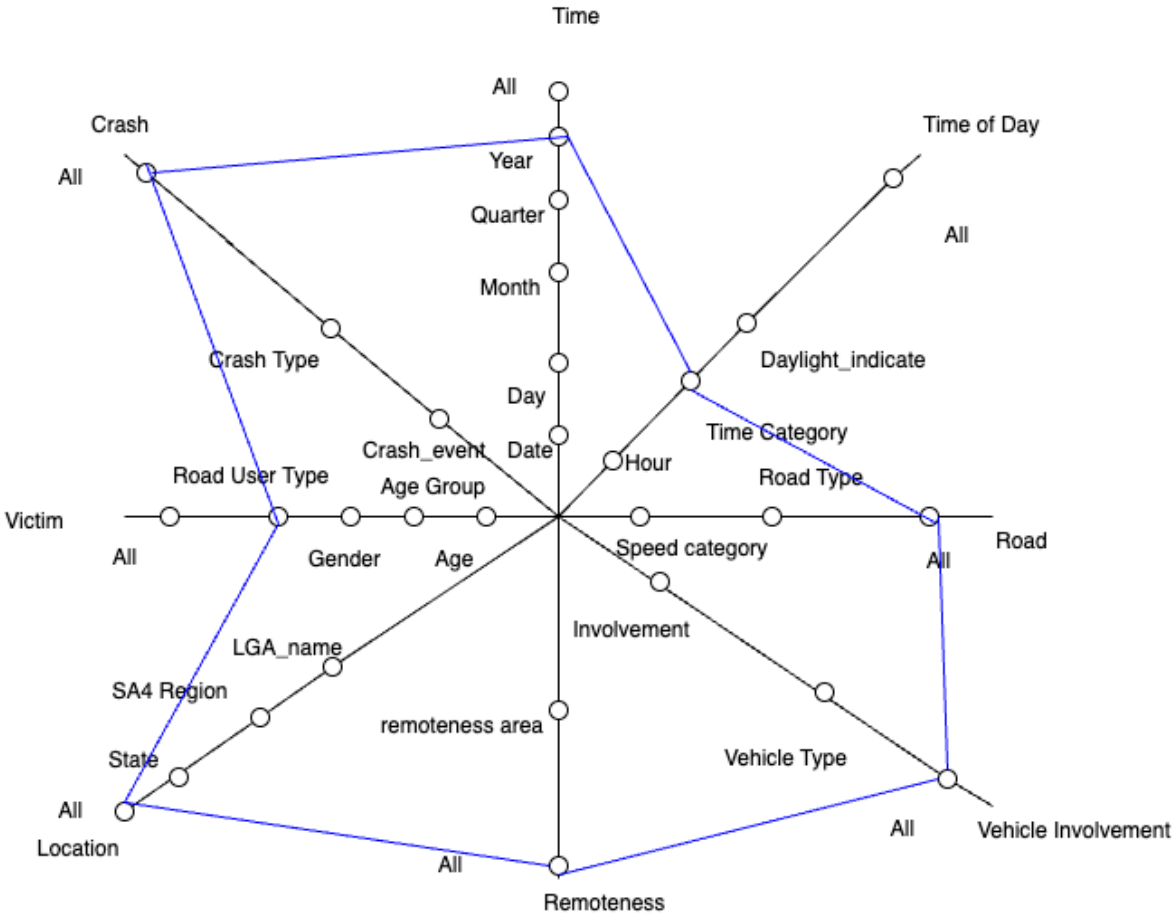

Figure 5: Query Footprints for Query 2

5.2.2 Visualization with analysis

The process involved:

1. Creating a bar chart with road user types on the x-axis and fatality counts on the y-axis
2. Sorting the bars in descending order of fatality counts
3. Applying different colors to distinguish between road user types
4. Adding a title and appropriate axis labels
5. Using horizontal gridlines to improve readability of the fatality values

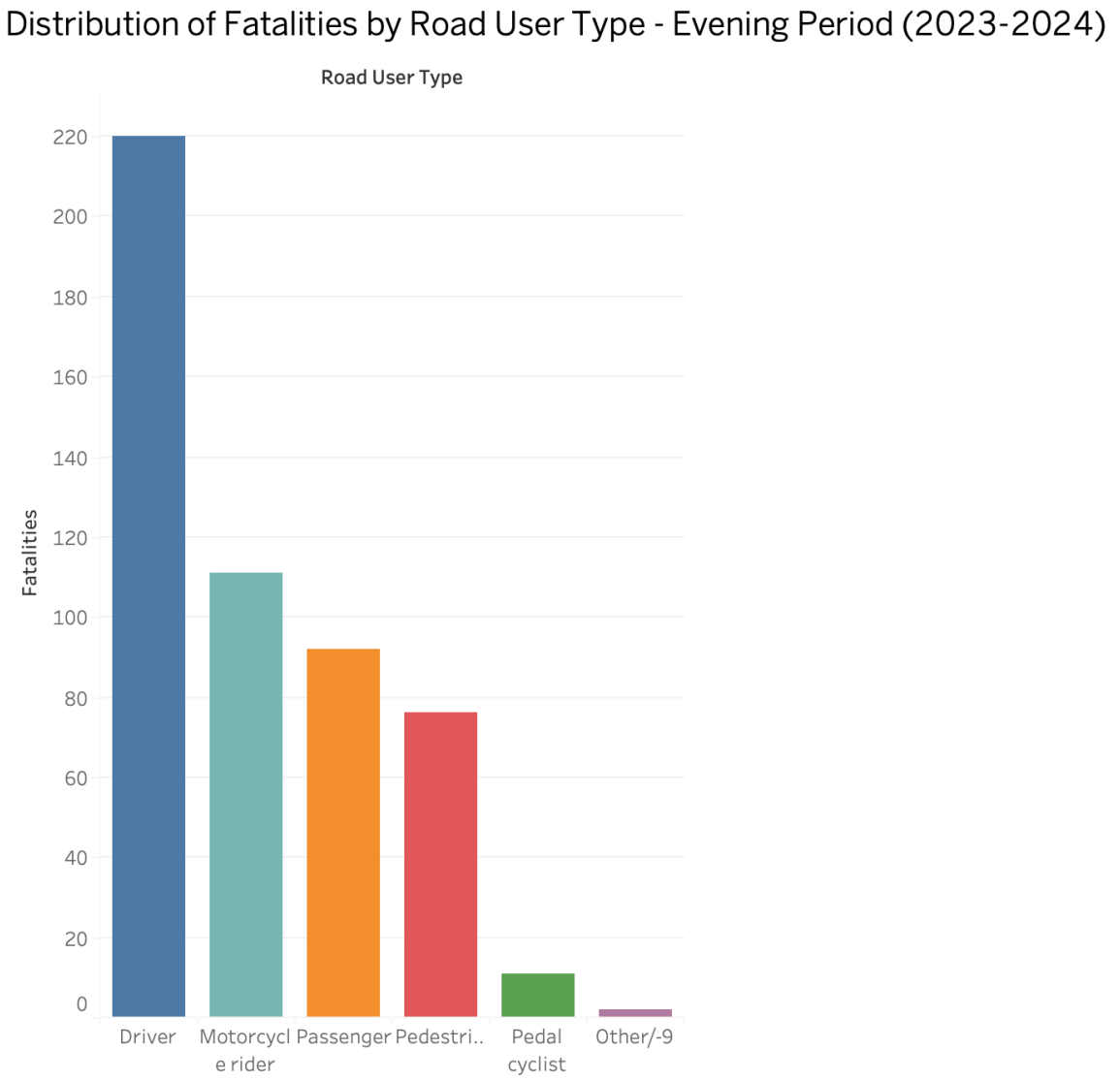

Figure 6: Bar Chart *Diagram* for Query 2

5.2.3 Key insights

The data reveals a clear hierarchy of vulnerability among road users during evening periods, drivers constitute the highest proportion of evening fatalities (approximately 220 deaths), representing nearly 43% of all evening period road deaths in 2023-2024. While drivers represent the largest number of fatalities, motorcycle riders likely face the highest relative risk when accounting for their smaller population proportion among road users.

5.3 Business Query 3:

What is the distribution of fatalities by month and time of day across different states in 2024, and which temporal patterns show the highest risk?

5.3.1 StarNet Diagram and Query Footprints

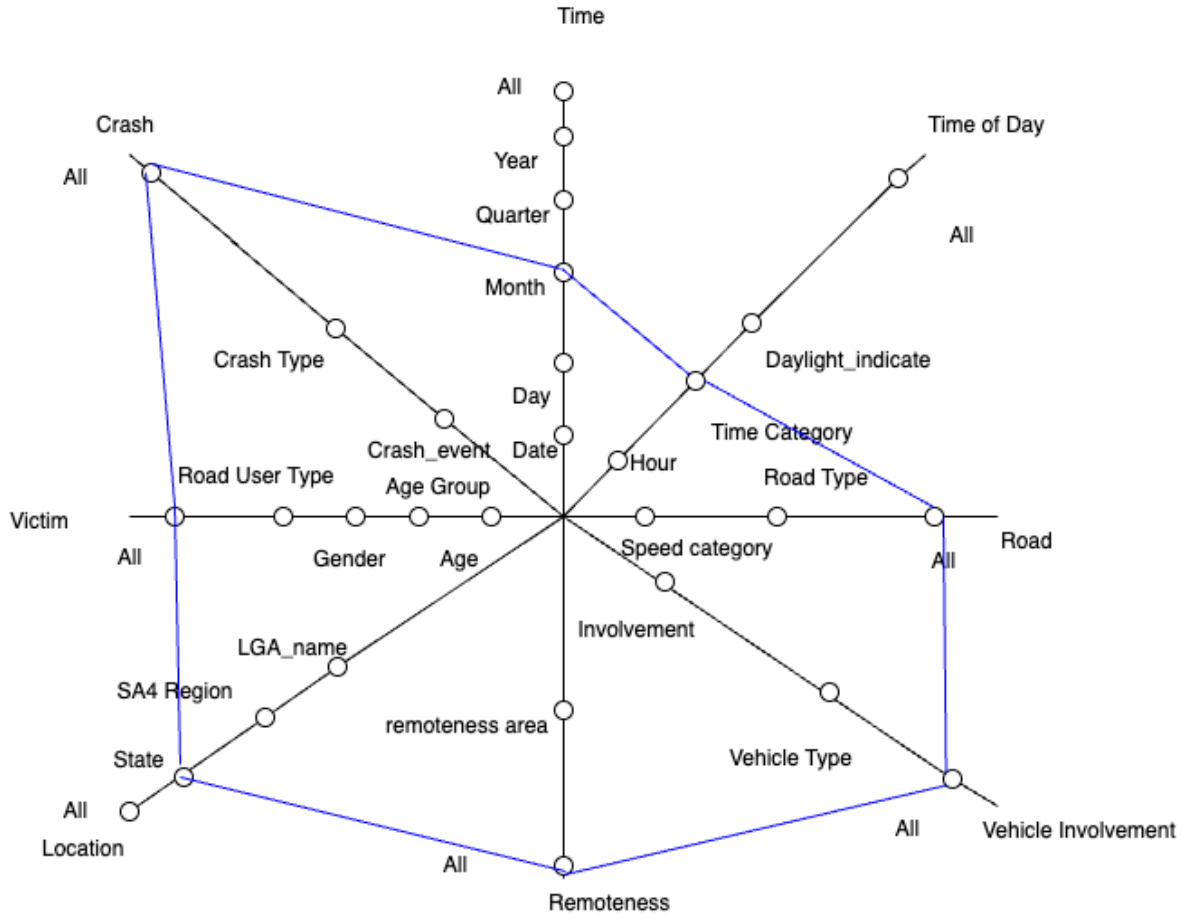

Figure 7: Query Footprints for Query 3

5.3.2 Visualization with analysis

The process involved:

1. Creating line charts with months on the x-axis and fatality counts on the y-axis

2. Using different colors to distinguish between Australian states
3. Adding a title and appropriate axis labels
4. Using gridlines to improve readability of the fatality values
5. Implementing small multiples to separate data by time categories (Morning, Afternoon, Evening, Night)

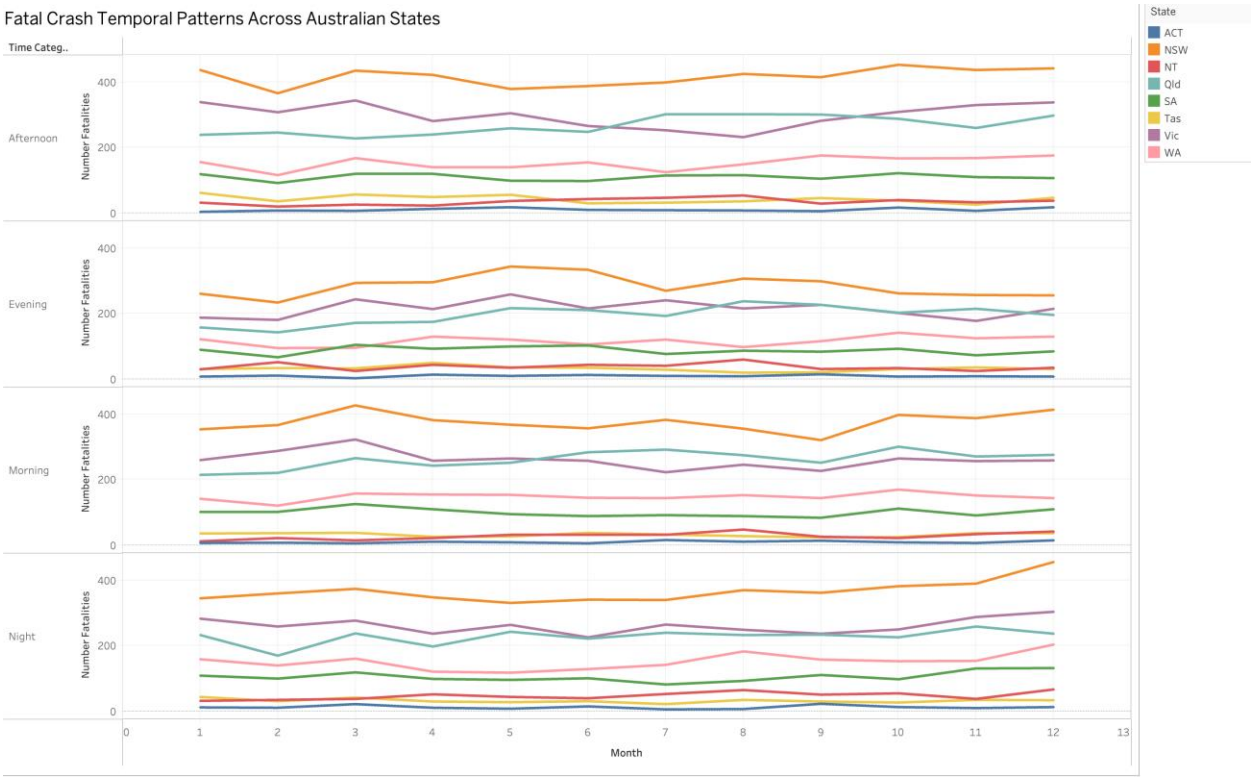

Figure 8: Line Chart *Diagram* for Query 3

5.3.3 Key insights

- NSW consistently shows the highest fatality rates across all months, with peaks typically occurring in March and December
- Victoria and Queensland maintain the second and third highest fatality rates respectively
- Time of day analysis reveals that evening and night hours account for a disproportionate number of fatalities in most states

5.4 Business Query 4:

What is the distribution of road fatalities across Australian states categorized by speed zones?

5.4.1 StarNet Diagram and Query Footprints

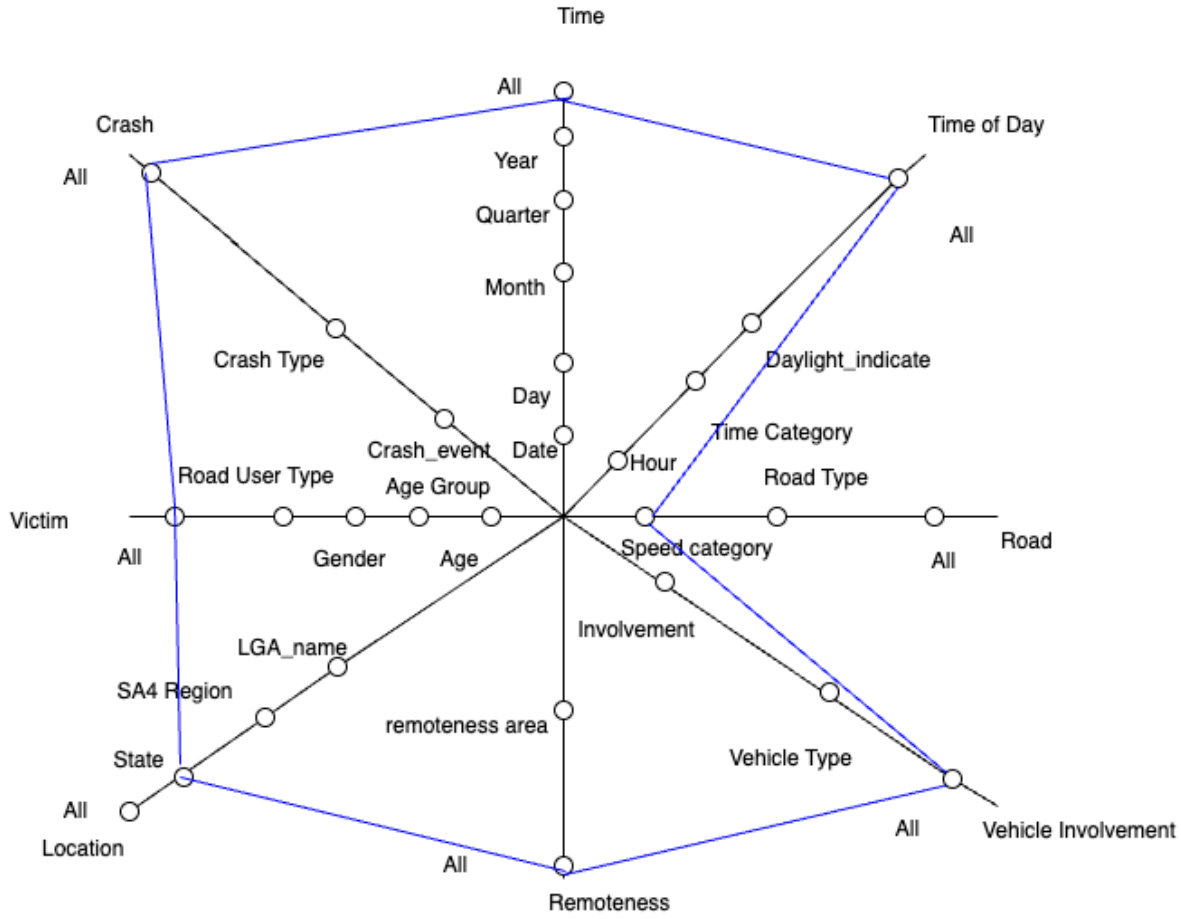

Figure 9: Query Footprint for Query 4

5.4.2 Visualization with analysis

The process involved:

1. Creating horizontal bar charts with states on the y-axis and fatality counts on the x-axis
2. Using different colors to distinguish between speed categories (Low, Medium, High)

3. Adding a title "Fatal Crash Analysis by State and Speed Category" and appropriate axis labels

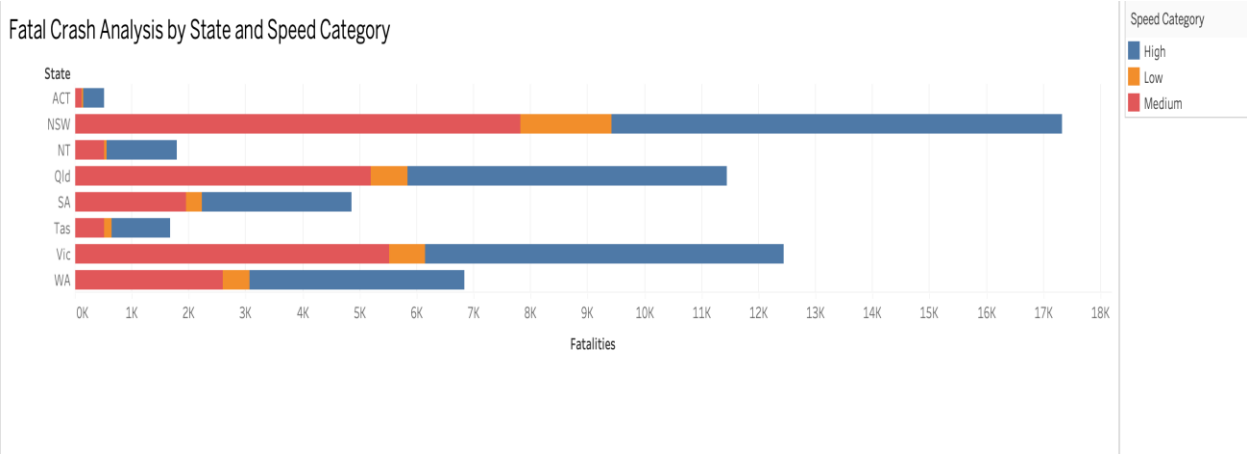

*Figure 10: H*orizontal Bar Chart *Diagram* for Query 4

5.4.3 Key insights

- NSW has the highest total number of fatalities across all speed categories, followed by Queensland and Victoria
- High speed categories contribute significantly to fatalities in most states
- Medium speed zonesaccount for the second-highest fatality count in most states
- Low speed areas generally have the lowest fatality counts, but are still significant in NSW

5.5 Business Query 5:

Which combinations of remoteness areas and road types pose the highest risk factors for fatal crashes, and how does this risk vary across different road environments?

5.5.1 StarNet Diagram and Query Footprints

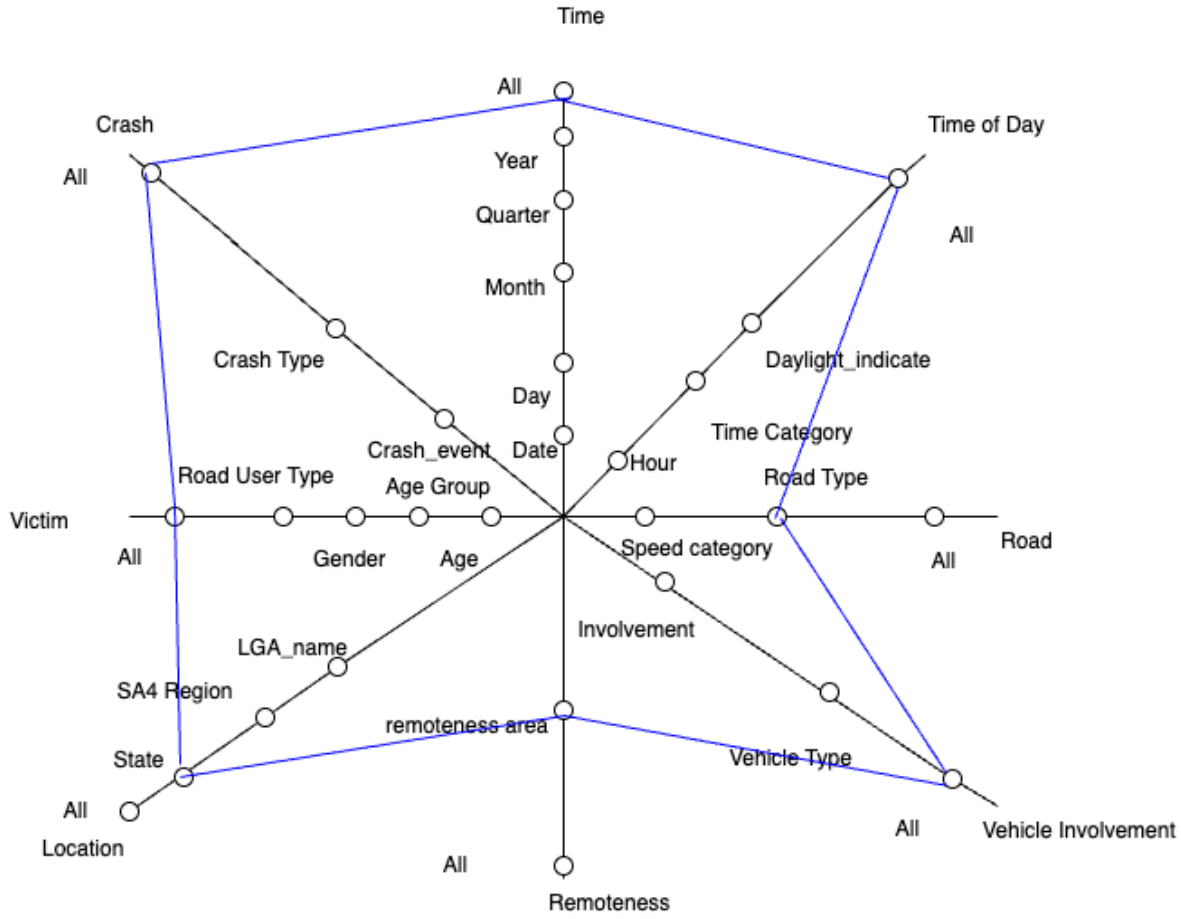

*Figure 11: H*orizontal Bar Chart *Diagram* for Query 5

5.5.2 Visualization with analysis

The process involved:

1. Creating a chat with Remoteness Area and Road Type on the y-axis and Average Risk Factor Score on the x-axis

2. Using a red color gradient to visualize risk intensity (darker red indicating higher risk)
3. Organizing the data hierarchically by remoteness area, then sorting road types by risk factor within each area
4. Adding a title "Risk Factor Analysis by Remoteness Area and Road Type" and appropriate axis labels

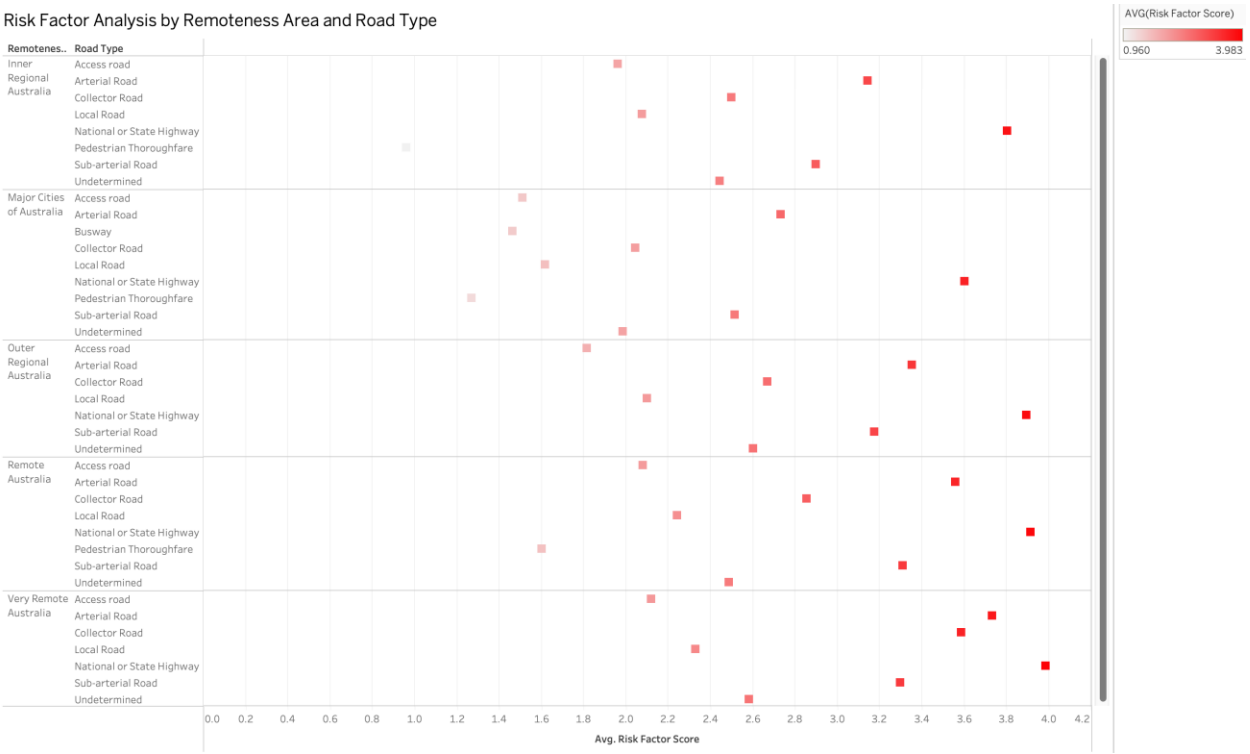

Figure 12: Chart *Diagram* for Query 5

5.5.3 Key insights

1. National or State Highways in Remote and Outer Regional Areas present the highest risk factors (3.5-4.0), suggesting these are the most dangerous road environments when accounting for various risk factors.
1. Arterial Roads in Remote Australia show surprisingly high risk scores (approximately 3.5), indicating that main connecting roads in remote areas warrant special safety attention.
1. Major Cities have lower overall risk scores for most road types compared to regional areas, but their National Highways still present significant risk (approximately 3.0).

6. Association Rules Mining

6.1 Discussing Association Rule Mining Algorithms

6.1.1 Introduction of association rule mining algorithms

First, introduce the major association rule mining algorithms that are applicable to road safety data:

1. **Apriori Algorithm** - this is the classic algorithm for mining association rules using a level-wise approach based on candidate generation. [1]
    - Advantages: Easy to implement, transparent process, effective for market basket analysis
    - Disadvantages: Multiple scans of database required, slow for large datasets, generates many candidates

2. **FP-Growth (Frequent Pattern Growth)** - this algorithm uses a divide-and-conquer approach without candidate generation.[2]
    - Advantages: No candidate generation, faster than Apriori, more efficient for large datasets
    - Disadvantages: Higher memory consumption, complex implementation

3. **ECLAT (Equivalence Class Transformation)** - this uses a depth-first search approach and vertical data format.[3]
    - Advantages: Depth-first search reduces memory requirements, faster than Apriori
    - Disadvantages: Less suitable for sparse datasets, requires transaction ID lists

6.1.2 Application to Road Safety Data Analysis

Discuss how these algorithms apply specifically to road safety data:

For road safety data analysis, Apriori is particularly suitable because:

1. The dataset's dimensionality is manageable because of few key variables
2. The relationships between road safety factors tend to follow the downward closure property that Apriori leverages
3. The interpretability of the algorithm aligns with the need to provide clear recommendations to government stakeholders.

6.2 Top Association Rules with Road User

Based on our association rule mining analysis using the Apriori algorithm in file “associated_data_mining.ipynb” . With a minimum support threshold of 0.1, we identified four significant rules with "Road User" on the right-hand side. These rules are ranked by lift (primary sorting criterion) and confidence (secondary sorting criterion) to prioritize the most meaningful associations.

```
Top 4 Rules with 'Road User' as Consequent (sorted by lift, confidence, support):
  antecedents       consequents  support  confidence    lift
0      Female         Passenger   0.1043      0.3726  1.6788
1        Male  Motorcycle rider   0.1292      0.1794  1.3370
2       40-64            Driver   0.1353      0.5085  1.1125
3       26-39            Driver   0.1176      0.5054  1.1057
Association rule mining analysis complete.
```

6.2.1 Meaning of Four K Rules

Rule 1: Female → Passenger (support=0.1043, confidence=0.3726, lift=1.6788)

This rule reveals that when a person involved in a fatal crash is female, there is a 37.26% probability that they were a passenger rather than a driver or other road user type. The lift value of 1.6788 indicates that females are 67.88% more likely to be passengers in fatal crashes compared to the overall population.

Rule 2: Male → Motorcycle rider (support=0.1292, confidence=0.1794, lift=1.3370)

This rule shows that when a person involved in a fatal crash is male, there is a 17.94% chance that they were a motorcycle rider. The lift value of 1.3370 means that males are 33.70% more likely to be motorcycle riders in fatal crashes than would be expected if gender and road user type were independent.

Rule 3: 40-64 → Driver (support=0.1353, confidence=0.5085, lift=1.1125)

This rule demonstrates that when a person involved in a fatal crash is aged between 40-64 years, there is a 50.85% probability that they were a driver. The lift value of 1.1125 suggests that people in this age group are 11.25% more likely to be drivers in fatal crashes than the average across all age groups.

Rule 4: 26-39 → Driver (support=0.1176, confidence=0.5054, lift=1.1057)

This rule indicates that when a person involved in a fatal crash is aged between 26-39 years, there is a 50.54% probability that they were a driver. The lift value of 1.1057 shows that this age group is 10.57% more likely to be drivers in fatal crashes compared

6.2.2 Plain English Interpretation of the Rules

These rules tell us important patterns about who tends to be involved in fatal road crashes and in what capacity:

1. **Women are much more likely to be passengers** when involved in fatal crashes. This suggests that female road users face different risks than males, potentially being more vulnerable as passengers rather than as drivers.
2. **Men have a stronger tendency to be motorcycle riders** in fatal crashes. This highlights the gender imbalance in motorcycle usage and the heightened risks faced by male motorcycle riders.
3. **Middle-aged and older adults (40-64) are most commonly drivers** in fatal crashes. This age group has both the highest confidence and support values for being drivers, indicating they form a substantial portion of the driving population involved in fatal incidents.
4. **Young to middle-aged adults (26-39) show similar patterns** to the older group, with just slightly lower probability of being drivers in fatal crashes.

6.3 Road Safety Improvement Suggestions

Recommendation 1: Targeted Motorcycle Safety Programs for Male Riders

Given the strong association between males and motorcycle riding fatalities (lift = 1.3370), the government should implement targeted motorcycle safety programs specifically designed for male riders. These programs should include:

- Advanced rider training courses focusing on risk assessment and defensive riding techniques
- Public awareness campaigns highlighting motorcycle visibility and the particular risks faced by male riders

Recommendation 2: Enhanced Passenger Safety Education and Vehicle Standards

With females showing a strong association with being passengers in fatal crashes (lift = 1.6788), improvements in passenger safety should be prioritized:

- Strengthen vehicle safety standards particularly for side-impact protection and rear passenger areas
- Develop passenger-specific safety education campaigns encouraging proper seatbelt use and avoiding distracting drivers

Recommendation 3: Age-Specific Driver Safety Interventions

The similar patterns for both 26-39 and 40-64 age groups as drivers in fatal crashes suggest the need for targeted interventions:

- Implement refresher driver training programs specifically designed for middle-aged drivers (26-64), focusing on complacency and distraction management
- Introduce regular vision and reaction time testing for drivers, particularly in the 40-64 age group
- Develop workplace driver safety programs, as these age groups represent the core working population who may drive for both commuting and work purposes

7. Conclusion

This project has successfully created a comprehensive road safety data warehouse using real-world fatal crash datasets. Through careful dimensional modeling following Kimball's methodology, we designed and implemented an eight-dimension star schema that effectively captures the multifaceted nature of road accidents.

The data warehouse enables powerful analytical capabilities, allowing us to answer critical business questions about geographic risk patterns, demographic vulnerabilities, temporal trends, and infrastructure factors contributing to road fatalities. Our ETL process effectively handled data quality challenges, employing advanced techniques like machine learning-based imputation to create a clean, reliable dataset.

Business queries revealed important insights including the disproportionately high risk on highways in remote areas, the vulnerability of specific demographic groups, and clear temporal patterns in fatal crashes. The association rule mining analysis further uncovered meaningful relationships between demographics and road user types, identifying that females are significantly more likely to be passengers in fatal crashes, males are more commonly motorcycle riders, and middle-aged adults form the largest proportion of drivers in fatal incidents.

These findings provide actionable intelligence for targeted road safety interventions, including enhancing motorcycle safety programs for male riders, strengthening passenger protection standards, and implementing age-specific driver safety measures. By leveraging this data warehouse for ongoing analysis, government authorities can make more informed decisions to reduce road fatalities and improve safety for all road users.

Reference

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 1994, pp. 487-499.

[2] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2000, pp. 1-12.

[3] M. J. Zaki, "Scalable algorithms for association mining," IEEE Trans. Knowl. Data Eng., vol. 12, no. 3, pp. 372-390, May/Jun. 2000.