COE Data Pipeline and Car Recommendation Engine

Turning COE premiums and emission bands into practical car-buying recommendations

Table of Contents

1.Introduction

1.1 Overview:

This project builds a data pipeline and analytical system to assist car buyers in Singapore in selecting a suitable vehicle based on COE premiums and desired carbon emission band. While COE premium datasets are publicly available, users must manually analyze them to make informed decisions. This system consolidates COE bidding data with a curated car reference list (including emission band information) to provide recommended car models (non-exhaustive) that align with a chosen emission band and category. Additionally, it highlights optimal quarters and months with historically lower premiums to guide purchase timing.

1.2 Problem statement:

Although COE premium datasets exist, car buyers face difficulty in quickly identifying which vehicles match their desired carbon emission band while also minimizing COE costs. Users must manually combine COE trends with vehicle specifications to select a suitable model. This project addresses the problem by providing recommended car models (based on a curated reference list) aligned with the user's emission band preference and COE category, while also indicating the historically most cost-effective purchase periods.

2. Data Sources

2.1 Dataset chosen

2.1.1 COE Bidding Results / Prices dataset

- **Source link**: COE bidding results/prices
- **Description**: Contains historical COE (Certificate of Entitlement) bidding results in Singapore from January 2010 to August 2025. Data includes bidding exercises, vehicle categories, quota, bid prices, and successful bids.

2.1.2 CEVS Bands - Car Models

- **Source link**: CEVS Revised Bands Annex A PDF
- **Description**: Contains a list of existing car models in Singapore categorized under the revised CEVS (Carbon Emissions-based Vehicle Scheme) bands. Data includes the band tier (A1–A4, B, C1–C4), car model names, and their CO_2 emission categories (Category A or B). This dataset helps classify vehicles based on emissions for policy and incentive purposes.

2.2 Data dictionary

2.2.1 COE Bidding Results / Prices dataset

Data Dictionary outlining a database on COE bidding results in Singapore from January 2010 to August 2025

Field Name	Data Type	Data Format	Description	Example
month	object	YYYY-MM	Year and month of bidding exercise	2010-01
bidding_no	object	Numeric (1 / 2)	Round of bidding exercise within the month	1
vehicle_class	object	Text	Vehicle category (A, B, C, D, E)	Category A
quota	Int64	Numeric	Number of COEs available	1105
bids_success	float64	Numeric	Number of successful bids	1014
bids_received	float64	Numeric	Number of bids submitted	1120
premium	int64	Numeric	Successful bid price(SGD)	18502

2.2.2 CEVS Bands - Car Models

Data Dictionary outlining a database on CEVS Bands - Car Models

Field Name	Data Type	Data Format	Description	Example
band	object	Text	CEVS band level (A1-4, B, C1-4)	A1
car_model	object	Text	Name of car model	Toyota Prius C
category	object	Text	Vehicle category (A, B)	Cat A

3. Methodology/Approach

3.1 Tech Stack / Tools

- **Python** - Data extraction, transformation and loading
  - Key libraries: pandas, requests, os, dotenv, pdfplumber, regex, psycopg2, sqlAlchemy, sqlalchemy_utils

- **PostgreSQ**L - storage & querying

3.2 ETL & Data Preparation

3.2.1 API-based

Environment Setup

- Use dotenv to securely load API keys. Avoid hardcoding credentials, ensuring security and portability.

Data Extraction

- Fetch COE dataset from data.gov.sg via API.
- Inspect the JSON response structure to prevent downstream errors.

Data Normalization

- Flatten nested JSON using pd.json_normalize to obtain a tabular format. Enables filtering, aggregation, and transformation.

Filtering Relevant Categories

- Retain only categories of interest: Category A, B, and E. Reduces dataset size and aligns with business requirements.

Data Quality Checks

- Identify missing values using .isnull().sum().
- Detects duplicate records using .duplicated().
- Convert numeric fields (quota, bids_success, bids_received, premium) to numeric type, coercing invalid entries to NaN.

Fact Table Transformation

- Map vehicle_class → category_id (foreign key to category dimension).
- Convert month → date_id (foreign key to date dimension).
- Aggregate data per month per category: sum of quota, bids_success, bids_received, average of premium.
- Add fact_id as a surrogate primary key.

Dimension Table Preparation (API-Based)

- **Date Dimension**
  - Extract unique months
  - Create date_id (YYYYMM) as primary key.
  - Derive year, month, and quarter attributes.
  - Facilitates time-based filtering and aggregation.
- **Category Dimension**
  - Define category_id, category_code, and description for each vehicle class.
  - Provides context for the fact table's foreign keys.

3.2.2 PDF Scraping

Environment Setup

- Use pdfplumber for PDF parsing and pandas for tabular transformation.
  - Avoid manual copy-pasting to ensure reproducibility and consistency.

Data Extraction

- Download the CEVS revised bands car models PDF from the NCCS website.
- Parse text content from each page using pdfplumber.
- Concatenate pages into a single text string for uniform downstream processing.

Data Normalization

- Apply **regular expressions** to segment text into bands (A1, A2, A3, etc.).
- Split car model listings by comma while preserving content inside parentheses.
- Extract key fields:
  - **band** (A1, A2, … C4)
  - **car_model** (cleaned text without notes/extra symbols)
  - **category** (from (Cat A) or (Cat B) annotations)

Filtering Relevant Categories

- Map only Category A → 1 and Category B → 2.

- Exclude rows without valid categories (e.g., footnotes, incomplete text).

Data Quality Checks

- Identify missing values using .isnull().sum().
- Detects duplicates across car_model + band + category_id using .duplicated().
- Validate that category_id contains only values {1, 2}.
- Drop invalid rows (e.g., footnotes without proper category).

Fact Table Transformation

- Assign a surrogate primary key (car_id) to each car record.
- Standardize band naming to ensure ordered tiers (A1 → C4).
- Convert category into category_id (foreign key).

Dimension Table Preparation (PDF-Based)

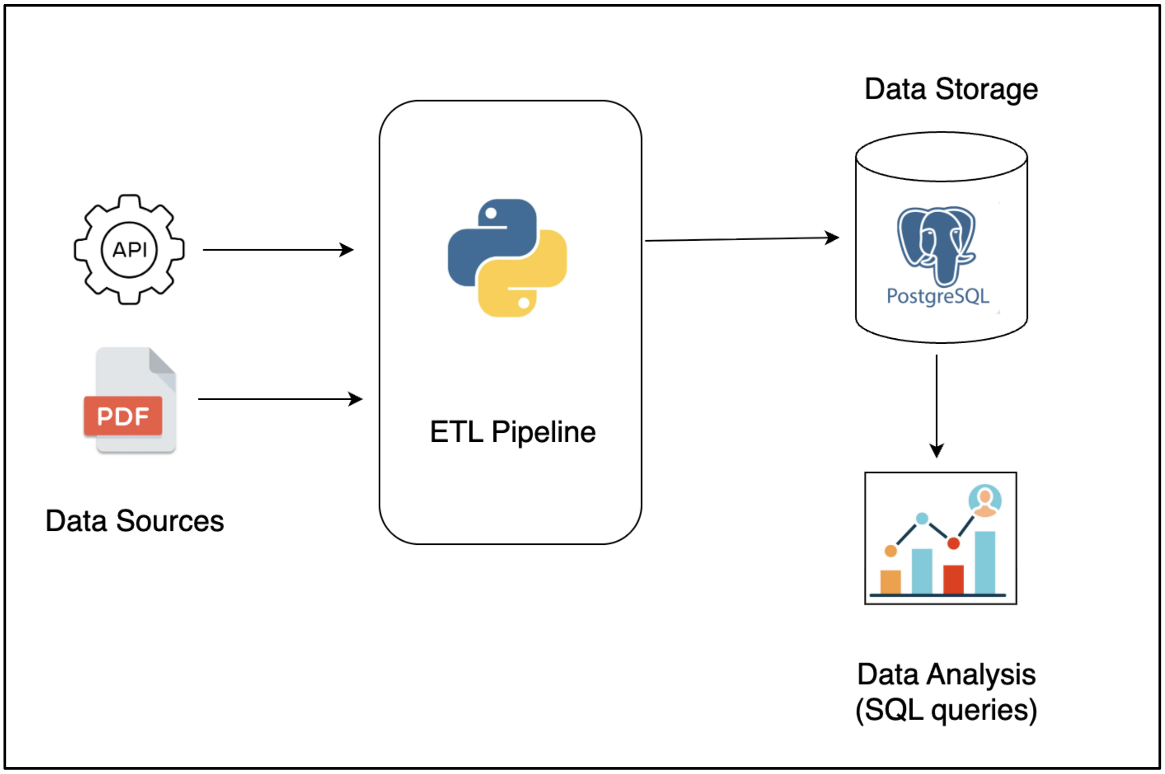
- **Band Dimension**
  - Define band_id, band_code (A1–C4), and description.
  - Enables classification of cars by emission bands.

- **Category Dimension**
  - Define category_id, category_code (A/B), and description.
  - Provides context for the fact table's foreign keys.

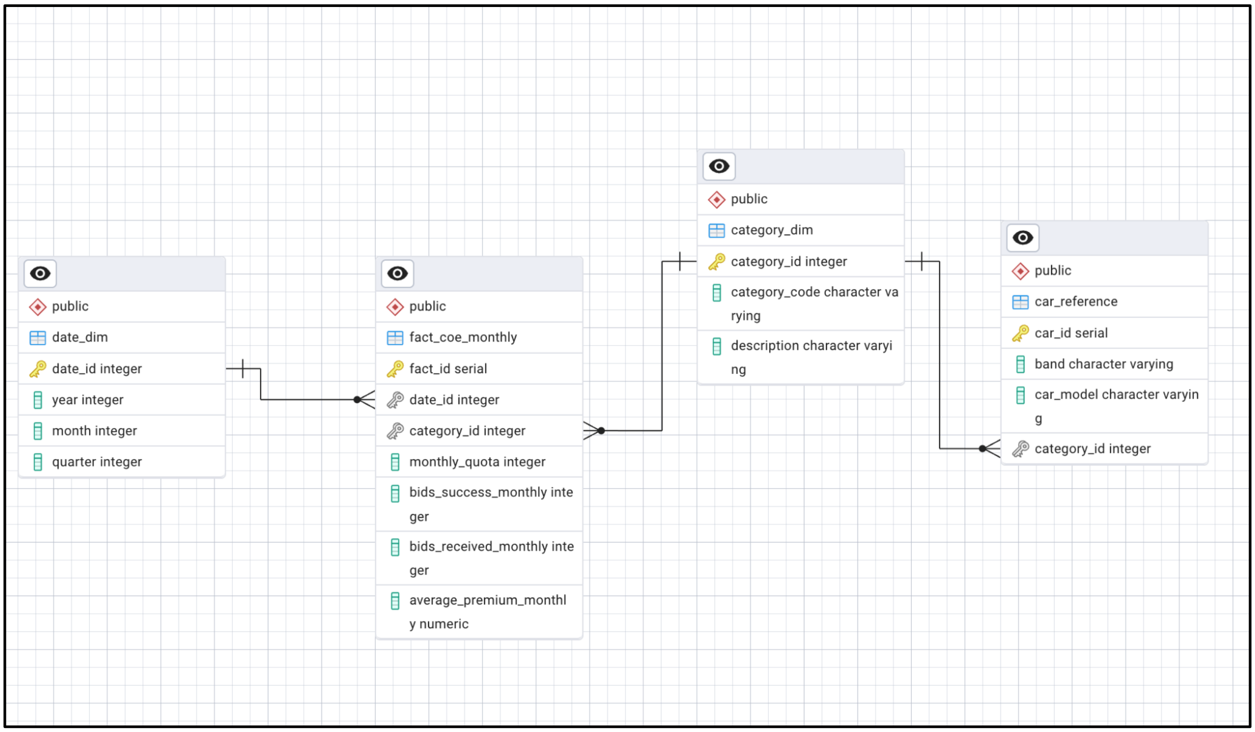
3.2.3 Table Creation & Loading

- Drop tables if they exist to avoid conflicts.
- Create tables with proper primary and foreign keys in dependency order..
- Load DataFrames in dependency order:
  - 1. date_dim 2. category_dim 3. fact_coe_monthly 4. car_reference

3.3 Architecture Diagram



3.4 Database Schema / ERD



The database schema follows a star schema design with a central fact table supported by dimension tables. The fact_coe_monthly table stores the aggregated COE bidding results, including monthly quota, number of successful and received bids, and the average premium. It links to the month_dim table, which provides time attributes such as year, month, and quarter, and the category_dim table, which stores vehicle category details (A, B & E). In addition, a car_reference table serves as a lookup for specific car models and their corresponding categories. This structure enables efficient querying and analysis of COE trends over time and across vehicle categories.

3.5 Table Structures

Table Name	Primary Key	Column Name	Description
fact_coe_monthly	fact_id	fact_id, date_id (FK), category_id (FK), monthly_quota, bids_success_monthly, bids_received_monthly,	Stores aggregated COE bidding results at the monthly level
date_dim	date_id	date_id, year, month, quarter	Stores time-related attributes for COE bidding exercises

category_dim	category_id	category_id, category_code, description	Stores COE category information (A, B, E)
car_reference	car_id	car_id, band, car_model, category_id (FK)	Stores reference information for car models and their associated COE

3.6 Data Validation

- **Missing value check** – Ensure required fields (e.g., car_model, band, category_id) are not empty.
- **Duplication check** – Ensure no repeated records
- **Valid category_id check (PDF scrape)** – Confirm category_id contains only allowed values (1 or 2, corresponding to Category A/B).

4. Challenges/Limitations

4.1 Challenges

1. **Long Runtime -** The initial web scraping approach resulted in very slow execution times (approximately **20–25 minutes per run**). The code was designed to dynamically adapt to future changes on the website, but this flexibility came at the cost of efficiency.

2. **Incomplete source data –** The original PDF contained missing or corrupted text (e.g., truncated model names, overlapping characters, or incomplete listings), which limited the accuracy of extraction.

3. **Inconsistent PDF formatting** – The source file was not structured as a clean table. Instead, car models, bands, and categories were embedded in continuous text with irregular spacing and line breaks, making automated parsing difficult.

4.2 Solutions / Workarounds

1. **Shift to PDF Scraping -** By switching from web scraping to PDF scraping, runtime was drastically reduced from ~25 minutes to just 10–15 seconds, while still ensuring accurate data extraction.

2. **Handling incomplete data source** – Validated extracted data against the PDF visually to detect gaps, flagged incomplete entries for manual review or exclusion, and supplemented missing information from external sources (e.g., LTA open datasets, manufacturer websites)

3. **Handling inconsistent PDF formatting** – Applied regex patterns and text-cleaning techniques (e.g., collapsing duplicate spaces, removing line breaks), defined parsing

rules to segment text into car model/band/category, and implemented validation checks to ensure extracted records matched expected formats.

4.3 Limitations

- Car_reference table is based on a non-exhaustive dataset (77 models only).
- COE premiums fluctuate frequently, recommendations reflect historical trends, not real-time data.
- The recommendation logic does not account for other factors like car specifications, user preferences, or financing.

5. Finding/Analysis

Objective 1: Compare the historical average COE premium across categories to identify which category has the lowest premium.

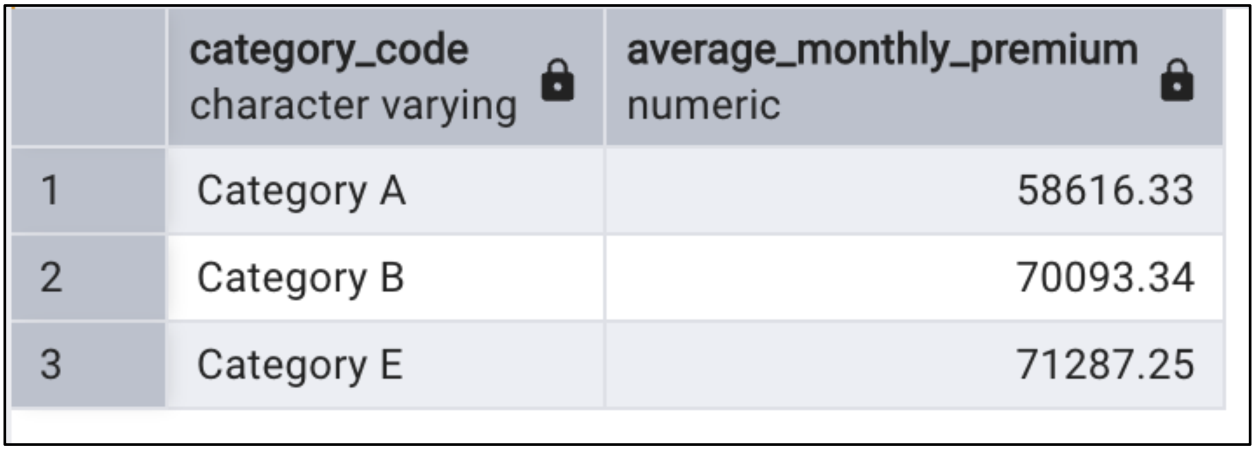
Query:

```
SELECT

c.category_code,
ROUND(AVG(f.average_premium_monthly), 2) AS average_monthly_premium

FROM fact_coe_monthly f
JOIN category_dim c
USING (category_id)
GROUP BY category_code
ORDER BY AVG(f.average_premium_monthly);
```

Output:



	category_code character varying	average_monthly_premium numeric
1	Category A	58616.33
2	Category B	70093.34
3	Category E	71287.25

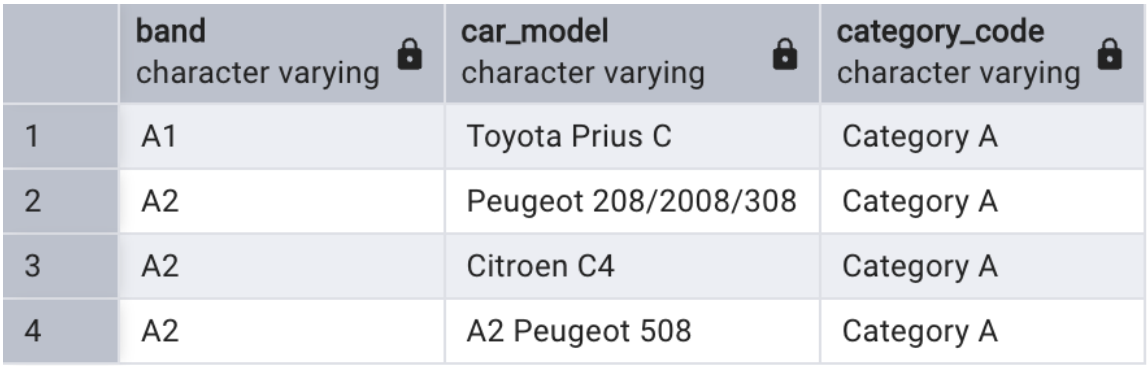
Finding: Over the past 15 years, **Category A** has had the lowest average monthly COE premium. **Categories B and E** show very similar premium levels, higher than Cat A but close to each other.

Objective 2: Identify category A cars with low carbon emission bands (A1 & A2)

Query:

```
SELECT cr.band, cr.car_model, cd.category_code
FROM car_reference cr
JOIN category_dim cd
ON cr.category_id = cd.category_id
WHERE cr.band IN (‘A1’, ‘A2’) AND cd.category_id = '1';
```

Output:



	band character varying	car_model character varying	category_code character varying
1	A1	Toyota Prius C	Category A
2	A2	Peugeot 208/2008/308	Category A
3	A2	Citroen C4	Category A
4	A2	A2 Peugeot 508	Category A

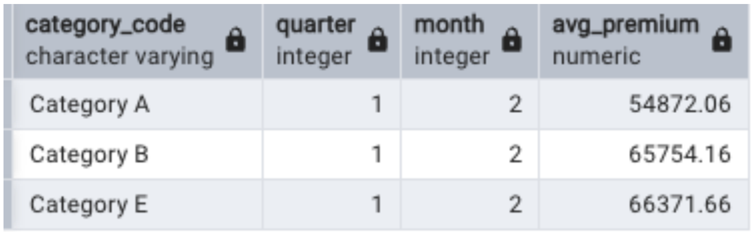
Finding: A total of 4 car models in Category A meet the low-emission criteria (A1 & A2 bands). These models offer both affordability (Cat A COE) and environmental benefits.

Objective 3: Identify the quarter and the specific month with the lowest average COE premium for each category (A, B, and E).

Query:

```
WITH ranked_premiums AS (
    SELECT
        c.category_code,
        d.quarter,
        d.month,
        ROUND(AVG(f.average_premium_monthly), 2) AS avg_premium,
        RANK() OVER (PARTITION BY c.category_code ORDER BY
AVG(f.average_premium_monthly) ASC ) AS rn
    FROM fact_coe_monthly f
    JOIN category_dim c
        ON f.category_id = c.category_id
    JOIN date_dim d
        ON f.date_id = d.date_id
    GROUP BY c.category_code, d.quarter, d.month
)
SELECT
    category_code,
    quarter,
    month,
    avg_premium
FROM ranked_premiums
WHERE rn = 1
ORDER BY category_code, avg_premium;
```

Output:



category_code character varying	quarter integer	month integer	avg_premium numeric
Category A	1	2	54872.06
Category B	1	2	65754.16
Category E	1	2	66371.66

Finding: Across all three categories (A, B, and E), the lowest average COE premiums consistently occur in Q1, with February emerging as the month with the lowest premiums for Category A.

Objective 4: Identifying seasonal fluctuations in COE prices over the years (2010 to 2025)

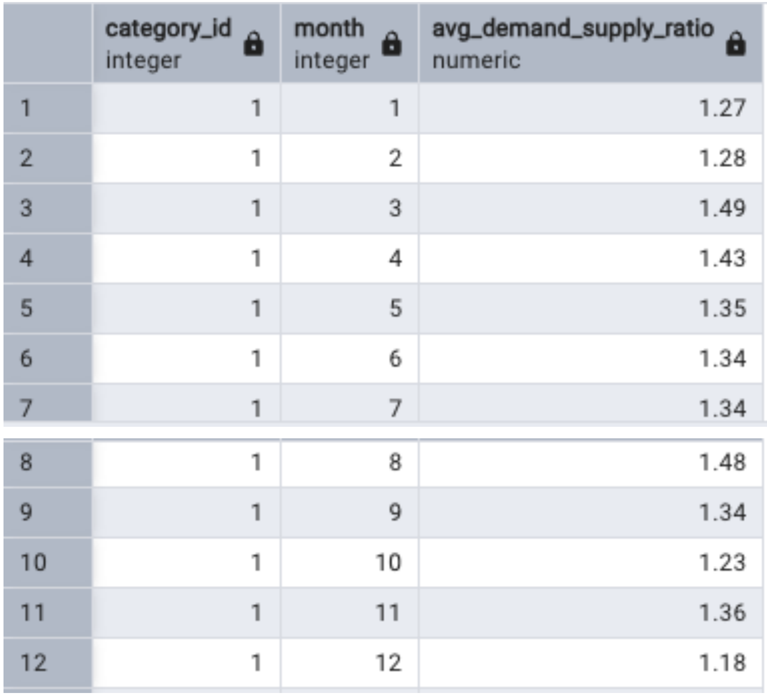
Query:

```
SELECT
        category_id,
        dd.month,
        ROUND(AVG(bids_received_monthly * 1.0 / monthly_quota), 2) AS
avg_demand_supply_ratio
FROM fact_coe_monthly AS fcm

RIGHT JOIN date_dim AS dd
USING(date_id)

WHERE date_id BETWEEN 201001 AND 202508
GROUP BY category_id, dd.month
ORDER BY category_id, dd.month;
```

Output:



	category_id integer	month integer	avg_demand_supply_ratio numeric
1	1	1	1.27
2	1	2	1.28
3	1	3	1.49
4	1	4	1.43
5	1	5	1.35
6	1	6	1.34
7	1	7	1.34
8	1	8	1.48
9	1	9	1.34
10	1	10	1.23
11	1	11	1.36
12	1	12	1.18

Finding:
COE demand follows clear seasonal trends throughout the year. Competition is generally moderate at the start of the year (January–February), increases sharply during peak months (March–April), stabilizes mid-year (May–July), experiences another surge in August, and gradually eases towards the year-end (October–December).

6. Next Steps & Conclusion

6.1 Next Steps

Moving forward, several opportunities exist to enhance and extend the current work:

- Future Improvements
  - Refine the data pipeline for greater reliability and faster processing.
  - Implement stronger data validation and error-handling mechanisms.

- Automation
  - Automate data extraction and transformation workflows using scheduled jobs (e.g., cron, Airflow).
  - Set up CI/CD pipelines to streamline code deployment and testing.

- Scaling
  - Transition to cloud-based infrastructure (AWS, GCP, or Azure) to handle larger datasets and more users.
  - Optimize database queries and indexing for performance at scale.

- Dashboards & Visualization
  - Develop interactive dashboards (e.g., using Tableau, Power BI, or Streamlit) to present insights dynamically.
  - Add filtering, drill-downs, and real-time updates to enhance decision-making.

- Machine Learning Extensions
  - Explore predictive analytics (e.g., forecasting demand, detecting anomalies).
  - Incorporate clustering, classification, or recommendation models to generate deeper insights.

6.2 Conclusion

Although demand (DSR) is moderately high in February, historical data shows that COE premiums are consistently lowest across all three categories during this month.

Based on this trend, this makes February (and generally Q1) as the most favorable period to bid for a COE as buyers can take advantage of lower premiums while competition remains manageable. Conversely, March–April and August represent peak DSR periods, suggesting higher competition and potentially higher premiums.

Further examination of Demand-to-Supply Ratio (DSR) reveals that December typically has the lowest DSR. However, prices in December do not fall correspondingly, which may be explained by several factors:

- **Buyer behavior**: Many buyers delay purchases until January to register a “new year” car, reducing bids without significantly lowering prices.

- **Market expectations**: Dealers and buyers may anticipate future quota or policy changes, preventing sharp declines in premiums.

As a result, December reflects weaker competition but relatively stable prices, whereas February shows the lowest prices potentially due to Chinese New Year slowdowns and softer demand.

Caveat: These findings are based on historical COE price trends and do not account for personal preferences, socioeconomic factors, or unforeseen events (“black swans”). Buyers should use this analysis as a guideline rather than a definitive prediction.