Automotive Supply Chain Analysis

DSC 202 Group 2
Yi Lien, Jean Hsu, Lucas Lee

Introduction

- **Objective**: Analyze the supply relationships among suppliers, automakers, and their respective models across various automotive components
- **Key questions**
    1. **Supply chain overview**: Who are the primary suppliers for each product, and what is the structure of the key supply network?
    2. **Time series analysis**: Who are the key autosuppliers, and how have their positions evolved over time?
    3. **Maker's similarity**: How can we quantify similarity of makers? How similar are automakers in their supply relationships?

Data Sources

- **MarkLines**
  1. Automotive components supply data, with every csv storing each product data (26 product types total)
     - i.e. abs_esc

Region	Maker	Model	Model year	Supplier
China	BAIC Foton	Foton View i9 (China)	2025	Wuhan Youfin Auto Electronic Control System Co., Ltd.
China	BYD	Fangchengbao Bao 8 (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Fangchengbao Bao 8 (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Han L DM-i (FF) (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Han L DM-i (FF) (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Han L DM-i (FF) (China)	2025	FinDreams Powertrain Co., Ltd.
China	BYD	Han L EV (4WD) (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Han L EV (4WD) (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.

Data Sources

- **MarkLines**

  2. Maker and maker group (partial)

maker_id	maker_name	year	maker_group
1	ARCFOX	2017	BAIC Group
2	ARCFOX	2021	BAIC Group
3	ARCFOX	2022	BAIC Group
4	ARCFOX	2023	BAIC Group
5	ARCFOX	2024	BAIC Group
6	ARCFOX	2025	BAIC Group
7	Abarth	2015	FCA
8	Abarth	2016	FCA
9	Abarth	2020	FCA
10	Abarth	2021	Stellantis
11	Abarth	2022	Stellantis

Data Preprocessing

1. Combine datasets & assigning trading volumes
2. Standardize model region data
3. Extract supply group data from supplier data

Data Preprocessing (Create a base table)

1. Combine datasets & assigning trading volumes (Pandas)

a. Each CSV contains maker-supplier information for a specific product (eg. Chasis)

Region	Maker	Model	Model year	Supplier
China	BAIC Foton	Foton View i9 (China)	2025	Wuhan Youfin Auto Electronic Control System Co., Ltd.
China	BYD	Fangchengbao Bao 8 (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.
China	BYD	Fangchengbao Bao 8 (China)	2025	Bosch Automotive Products (Suzhou) Co., Ltd.

b. Combine all (concat row-wise) CSV together

c. create a new column that is the product type

d. add a new column (trade volume) where ~U(100, 1000)

region	maker	model	model_year	supplier	product	volume
China	BAIC Foton	Foton View i9 (China)	2025.0	Wuhan Youfin Auto Electronic Co	crossmer	696
China	BYD	Fangchengbao Bao 8 (China)	2025.0	Bosch Automotive Products (Suz	abs_esc	997

Data Preprocessing (Modify the model country data)

2. Standardize model region data (partial)

```
with base as
    (select
        replace(region, 'ASEAN?India?Korea', 'ASEAN, India, Korea') as region
      , maker
      , model
      , case
          when right(model, 1) = ')'
          then replace(replace(replace(replace(replace(
                  regexp_replace(regexp_replace(split_part(model, '(', -1), '[^A-Za-z ]', '', 'g'), 'for ', '', 'i')
              , 'Export to China as Tivolan', 'China')
              , 'UK', 'United Kingdom')
              , 'Korea', 'South Korea')
              , 'USA', 'United States')
              , 'Turkiye', 'Turkey')
          else null
          end as model_region
      , substring(model_year, 1, 4)::int as model_year
      , supplier
      , product
      , volume
```

Data Preprocessing (Modify the model country data)

2. Standardize model region data (partial)

region	maker	model	model_region	model_year	supplier	product	yearly_volume
Americas	Alpina	XB7 (USA)	United States	2020	ZF Friedrichshafen AG	automatic_transmission_(at)	446
Americas	Alpina	XB7 (USA)	United States	2022	ZF Friedrichshafen AG	automatic_transmission_(at)	686
Americas	Alpina	XD3 (USA)	United States	2022	ZF Friedrichshafen AG	automatic_transmission_(at)	525
Americas	Alpina	XD4 (USA)	United States	2022	ZF Friedrichshafen AG	automatic_transmission_(at)	492
Americas	Alpina	XB7 (USA)	United States	2023	Continental Automotive Tec	abs_esc	598
Americas	Alpina	XB7 (USA)	United States	2024	Continental Automotive Tec	abs_esc	753
Americas	Audi	A3 (Brazil)	Brazil	2014	KYB Manufacturing do Bra	shock_absorber	443
Americas	Audi	Q5 (Mexico)	Mexico	2016	Hirschvogel Group (Hirschv	transmission_shaft	747
Americas	Audi	Q5 (Mexico)	Mexico	2016	Nemak Poland Sp. Z.o.o.	chassis_frame	805

Data Preprocessing (Create Mapping to Supply Group)

3. Extract supply group data from supplier data

supplier_id	supplier_name	parent_company
1	AAPICO Amata Co., Ltd.	AAPICO
2	AAPICO Hitech Public Co., Ltd.	AAPICO
3	AB SKF	SKF
4	ACPS Automotive GmbH (Formerly BOSAL Automotive Carrier and Protection Systems GmbH)	ACPS
5	ACPS Automotive Kft.	ACPS
6	ACPS Automotive de Mexico SA de CV.	ACPS
7	ADAC Automotive Inc.	ADAC
8	ADVICS Co., Ltd.	ADVICS
9	ADVICS Fuzhou Automobile Parts Co., Ltd.	ADVICS
10	ADVICS Guangzhou Automobile Parts Co., Ltd.	ADVICS
11	ADVICS Manufacturing (Thailand) Co., Ltd.	ADVICS
12	ADVICS Manufacturing Mexico, S. de R.L. de C.V.	ADVICS
13	ADVICS North America, Inc.	ADVICS
14	ADVICS Tianjin Automobile Parts Co., Ltd.	ADVICS
15	AGS Automotive Systems	AGS

Data Preprocessing (Create Mapping to Supply Group)

3. Extract supply group data from supplier data

```
with parent as (
select distinct match,count(*) from (
    select supplier_id,name,unnest(regexp_match(name,'\w+\s')) as match
    from supplier
             ) s
group by match
having count(*)>1),
match as (
select name, "left"(match,length(match)-1) as match_name from supplier
cross join parent
where lower(name) ~* lower(parent.match) and "left"(lower(name),length(match)) = lower(match)
order by lower(match))

update supplier
set parent_company = match_name
from match
where supplier.name = match.name;
```

Data Preprocessing (Create Mapping to Supply Group)

3. Extract supply group data from supplier data (partial)

```
update supplier
set parent_company ='ADAC'
where name like 'ADAC Automotive%'
update supplier
set parent_company = 'AGS'
where name like 'AGS %'
update supplier
set parent_company = 'ATA Casting Technology'
where parent_company = 'ATA'
update supplier
set parent_company = null
where parent_company in ('AT','Changchun','Changsha','Chengdu','China','Chongqing','Foshan','Fuxin','GF','Guangdong','Hangzhou'
,'Harbin','Hebei','Hiroshima','Hubei','Hunan','Jiangsu','Jinzhou','Liuzhou','Lucas','Maxion','Metal','Thai','The','Tianjin'
,'Nanjing','New','Ningbo','Precision','SKH','Shandong','Shanghai','Shenyang','Siam','Sichuan','Taiwan','Wenzhou','Wuhan'
,'Wuhu','Wuxi','Yantai','Yuhuan','Zhejiang')
or name in ('American Axle & Manufacturing Holdings, Inc.','Art Metal Mfg. Co., LTD.','Asahi Tec Aluminium (Thailand) Co., Ltd.',
          'Asahi Tekko Co., Ltd.','Asama Giken Co., Ltd.')
or supplier_id in ('125','126','130','134','142','143','309','310','311')
```

Key Questions

1. **Supply chain overview:** Who are the primary suppliers for each product, and what is the structure of the key supply network?
2. **Time series analysis**: Who are the key autosuppliers, and how have their positions evolved over time?
3. **Maker's similarity**: How can we quantify similarity of makers? How similar are automakers in their supply relationships?

Methodology - Supply Chain Overview

1. **Total trading volume analysis**
    - Data Processing: join supplier group data, categorized years into 5-year intervals using PostgreSQL
    - Supplier Ranking: aggregated trading volumes and get the top 5 suppliers in each product using PostgreSQL

Demonstration - Supply Chain Overview

1. **Total trading volume analysis**
    - Output (partial)

	product	year_range	supplier_group	total_volume
1	abs_esc	2005-2010	ADVICS	920
2	abs_esc	2011-2015	Bosch	48934
3	abs_esc	2011-2015	Continental	23024
4	abs_esc	2011-2015	ZF	12420
5	abs_esc	2011-2015	Beijing West Industries	8838
6	abs_esc	2011-2015	ADVICS	4660
7	abs_esc	2016-2020	Bosch	573248
8	abs_esc	2016-2020	Continental	313680

Methodology - Supply Chain Overview

2. **Key Supply Network Analysis**
    - Data Processing: sum annual trading volume, join supplier group data, categorized years into 5-year intervals, and aggregate trade year count and total trading volume
    - Network visualization: find the top suppliers in each product category and the respective supply network using Neo4j
    - i.e. top 3 suppliers in ‘transmission shaft’

Methodology - Supply Chain Overview

i.e. top 3 suppliers in ‘transmission shaft’

```
// (1) Load PostgreSQL table into Neo4j
CALL apoc.load.jdbc(
'jdbc:postgresql://localhost:5432/dbname?user=username&password=xxx,
'SELECT * FROM groupby_maker'
) YIELD row

// (2) Merge nodes and edges, set the supply product information in the edges
MERGE (m:Maker {name: row.maker})
MERGE (s:Supplier {name: row.supplier_group})
MERGE (p:Product {name: row.product})
MERGE (m)-[r:SUPPLIES {year_range: row.year_range, yearly_trade_cnt:
toInteger(row.yearly_trade_cnt)}]->(s)
ON CREATE SET r.product = row.product;
```

Methodology - Supply Chain Overview

i.e. top 3 suppliers in ‘transmission shaft’

```
// (3) Keep product = 'transmission_shaft'
MATCH (m:Maker)-[r:SUPPLIES]->(s:Supplier)
WHERE r.product = 'transmission_shaft'

// (4) Aggregate the trade count to find the top 3 suppliers
WITH r.year_range AS year_range, s, SUM(r.yearly_trade_cnt) AS total_trades
ORDER BY year_range, total_trades DESC
WITH year_range, COLLECT(s)[0..3] AS top_suppliers
UNWIND top_suppliers AS supplier

// (5) Match the edges back
MATCH (m:Maker)-[r:SUPPLIES]->(supplier)
WHERE r.product = 'transmission_shaft' AND r.year_range = year_range
```

Methodology - Supply Chain Overview

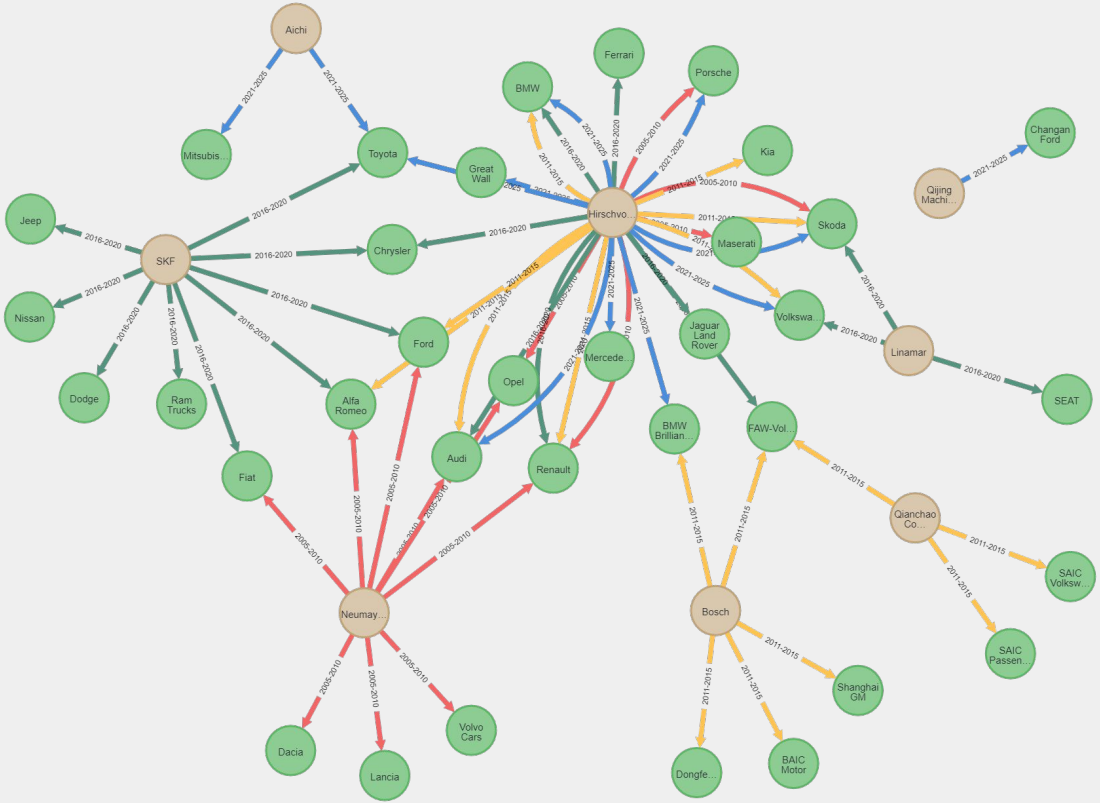
i.e. top 3 suppliers in ‘transmission shaft’

```
// (6) Create virtual edges
WITH m, supplier, r.yearly_trade_cnt AS trade_count, year_range
CALL apoc.create.vRelationship(
    supplier, year_range, {year_range: year_range, trade_count: trade_count}, m)
YIELD rel

// (7) Return makers, top 3 suppliers, and the edges
RETURN m, supplier, rel
```

Demonstration - Supply Chain Overview

based on trade_year_cnt



Node labels

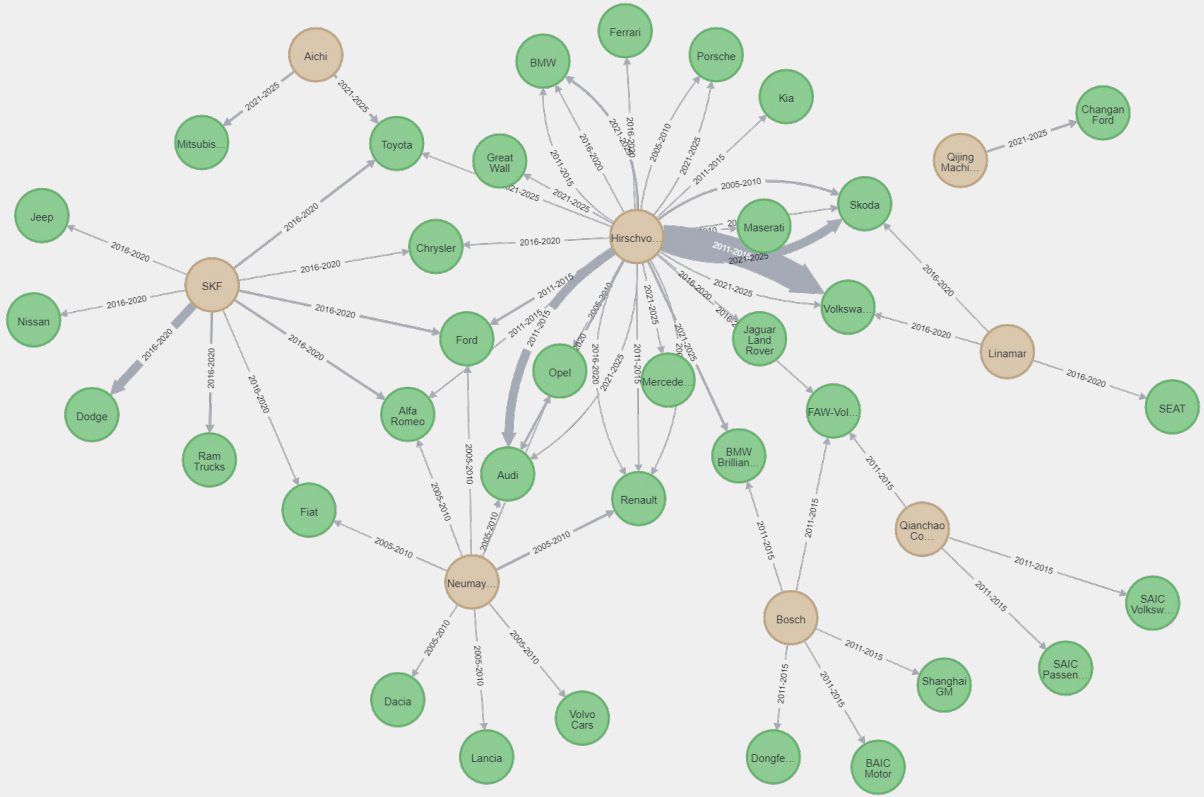
- * (53)
- Maker (46)
- Supplier (7)

Relationship types

- * (62)
- 2005-2010 (4)
- 2011-2015 (28)
- 2016-2020 (23)
- 2021-2025 (7)

Demonstration - Supply Chain Overview

based on trade_year_cnt



Node labels

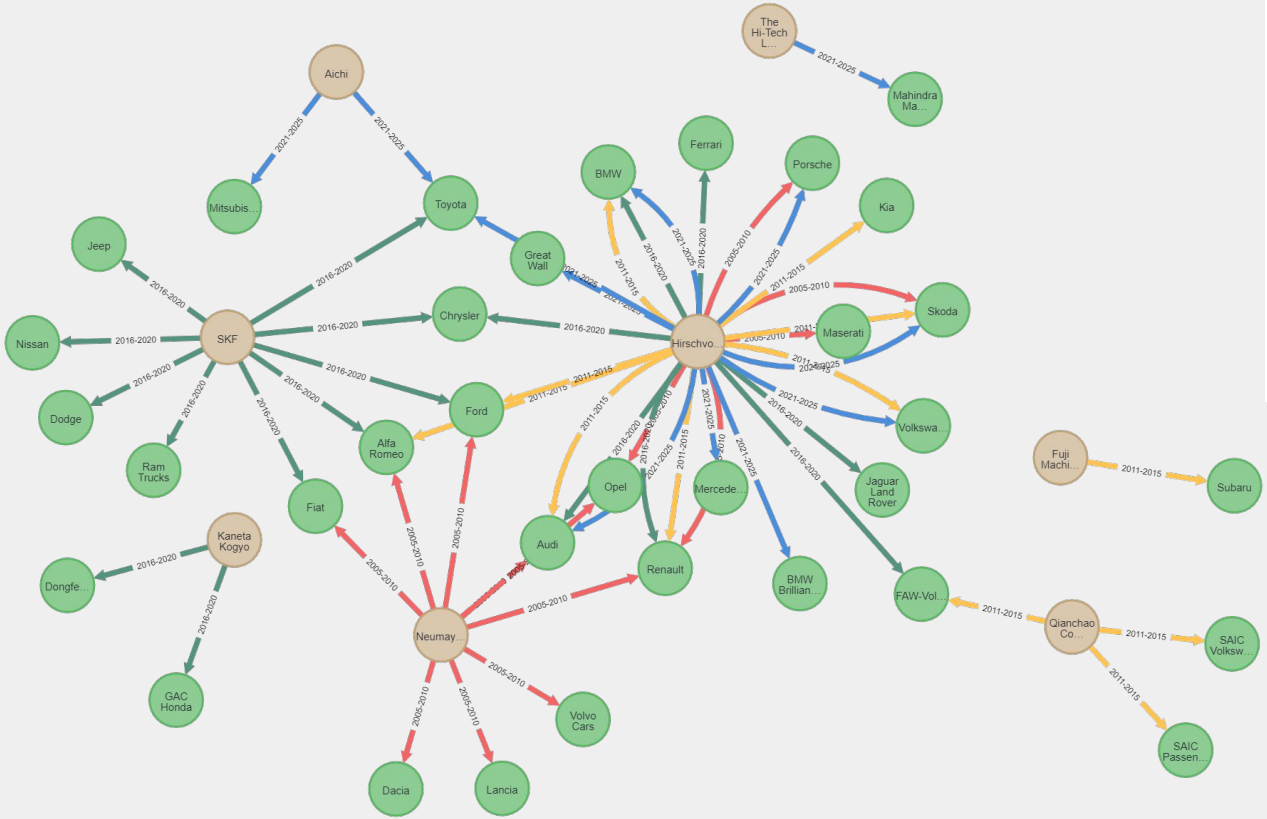
* (43) | Maker (35) | Supplier (8)

Relationship types

1 (44) | 2 (13) | 3 (3) | 4 (1) | * (61)

Demonstration - Supply Chain Overview

based on total_volume



Node labels

* (42) | Maker (34) | Supplier (8)

Relationship types

* (56) | 2005-2010 (14) | 2011-2015 (12) | 2016-2020 (18) | 2021-2025 (12)

Time Series Analysis

2025 Top 10 Automotive Suppliers

- Rank Suppliers
- Calculate YoY Change
- Filter for 2025's Top 10 Suppliers

```sql
with s_rank as (
select
    model_year,
    coalesce(s.parent_company,s.name) as supplier_name,
    rank() over(partition by model_year order by sum(unit) desc) as supplier_rank,
    sum(unit) as unit
from component_fact c
left join supplier s on c.supplier = s.name
group by model_year,supplier_name),
s_share as (
select
    r1.model_year,
    r1.supplier_name,
    r1.supplier_rank,
    r1.unit,
    sum(r1.unit) over(partition by r1.model_year) as unit_year,
    concat(round(cast(r1.unit / sum(r1.unit) over(partition by r1.model_year)*100 as numeric),2),'%') as share,
    case
        when r1.model_year = 2025 then concat(round(cast((r1.unit*12/3 - r2.unit) / r2.unit as numeric)*100,2),'%')
    else concat(round(cast((r1.unit - r2.unit) / r2.unit as numeric)*100,2),'%')
    end as yoy
from s_rank r1
left join s_rank r2 on r1.model_year = r2.model_year+1 and r1.supplier_name = r2.supplier_name)
select model_year, supplier_name, supplier_rank,share ,yoy
from s_share
where
    model_year =2025 and supplier_rank <= 10
order by model_year,supplier_rank;
```

Time Series Analysis

2025 Top 10 Automotive Suppliers

- Top 1 Supplier: Bosch
- High YoY Growth Suppliers: Faurecia and FinDreams

model_year	supplier_name	supplier_rank	share	yoy
2025	Bosch	1	25.64%	-18.94%
2025	Continental	2	7.57%	-43.63%
2025	Hitachi	3	6.74%	21.99%
2025	Faurecia	4	5.28%	307.72%
2025	FinDreams	5	5.22%	307.11%
2025	Wuhu Bethel Electronic Control System Co.,…	6	4.44%	-22.03%
2025	ZF	7	3.08%	-65.52%
2025	Tenneco	8	2.48%	74.49%
2025	Vitesco	9	2.27%	297.88%
2025	Global Technology Co., Ltd.	10	2.25%	241.94%

Time Series Analysis

2025 Top 10 Automotive Suppliers

- High YoY Growth Suppliers: Faurecia and FinDreams

```
select distinct parent_company, region, model_region, product from component_fact c
left join supplier s on c.supplier = s.name
where  parent_company in ('Faurecia','FinDreams') and model_year = 2025
order by parent_company;
```

	parent_company	region	model_region	product
1	Faurecia	ASEAN, India, Korea	India	exhaust_manifold
2	FinDreams	China	China	abs_esc

Time Series Analysis

2025 Top 10 Automotive Suppliers

- extract all yearly data among top 10 suppliers

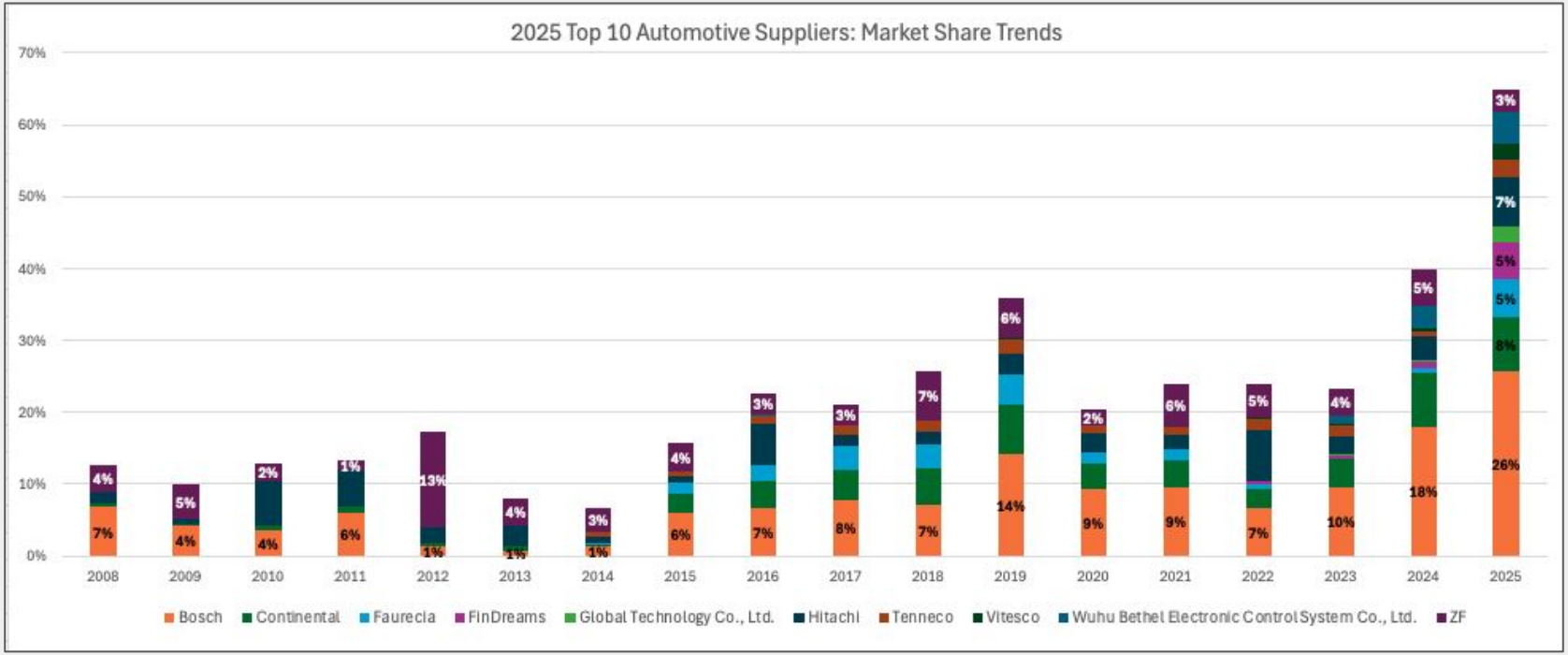
```
where
    supplier_name in
     (select distinct supplier_name from s_rank
        where supplier_rank <= 10 and model_year = 2025)
order by model_year,supplier_rank;
```

	model_year	supplier_name	supplier_rank	share	yoy
1	2008	Bosch	3	6.79%	<null>
2	2008	ZF	6	3.66%	<null>
3	2008	Hitachi	17	1.47%	<null>
4	2008	Continental	36	0.52%	<null>
5	2008	Tenneco	56	0.14%	<null>
6	2009	ZF	5	4.98%	10.75%
7	2009	Bosch	9	4.19%	-49.85%
8	2009	Hitachi	31	0.65%	-63.99%
9	2009	Continental	49	0.19%	-69.34%
10	2010	Hitachi	5	6.34%	2097.52%
11	2010	Bosch	8	3.66%	96.87%
12	2010	ZF	9	2.49%	12.55%
13	2010	Continental	27	0.45%	418.62%
14	2011	Bosch	2	6.05%	898.08%
15	2011	Hitachi	3	5.04%	380.39%

119 rows

Time Series Analysis

Market Share Trends



Time Series Analysis

Average Maker Amount

- Count Distinct Makers Per Year
- Calculate the Average Number of Makers

```
select
    coalesce(parent_company,name) as supplier,
    sum(maker_num) / count(distinct model_year) as avg_maker_amount
    from (
select distinct model_year, c.supplier, count(distinct maker) as maker_num from component_fact c
group by model_year, supplier
order by model_year, supplier) c
left join supplier s on c.supplier = s.name
group by coalesce(parent_company,name)
order by avg_maker_amount desc
limit 10
```

Time Series Analysis

Average Maker Amount

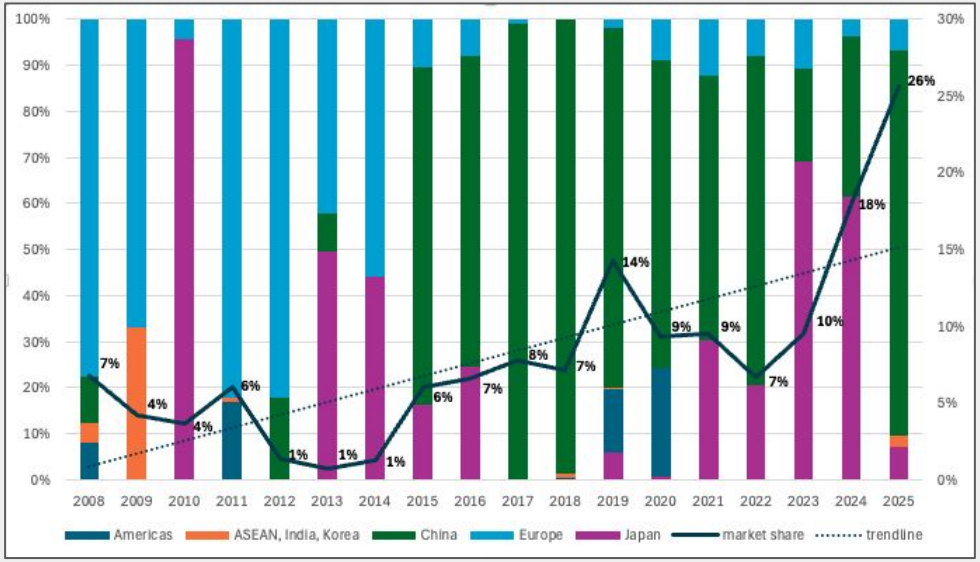
- Count Distinct Makers Per Year
- Calculate the Average Number of Makers

	supplier	avg_maker_amount
1	Bosch	47.6666666666666667
2	ZF	32.7222222222222222
3	Continental	26.1666666666666667
4	Aisin	22.4705882352941176
5	Faurecia	18.3333333333333333
6	Precision Sintered Products (Wuxi) Co., Ltd.	16.5
7	Zhejiang Qibo Machinery Co., Ltd.	15
8	Denso	14.2222222222222222
9	Chengdu Xiling Power Science & Technology Incorpor…	14
10	Hitachi	13.7222222222222222

Time Series Analysis

Bosch - Changes in Maker Regional Distribution

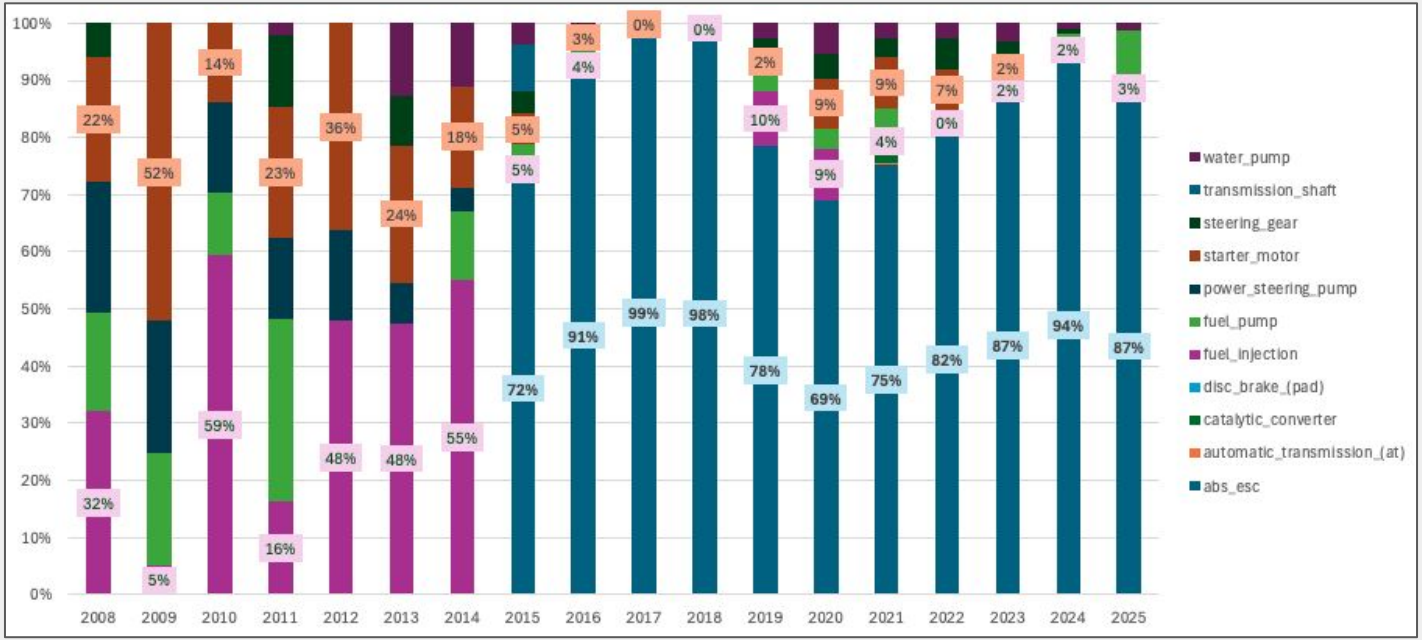
- The loss of cooperation with American automakers in 2012
- Shift in Focus from Europe to China in 2015 (onward)



Time Series Analysis

Bosch - Changes in Product Focus

- Transition from Fuel Injection and Starter Motor to ABS/ESC



Time Series Analysis

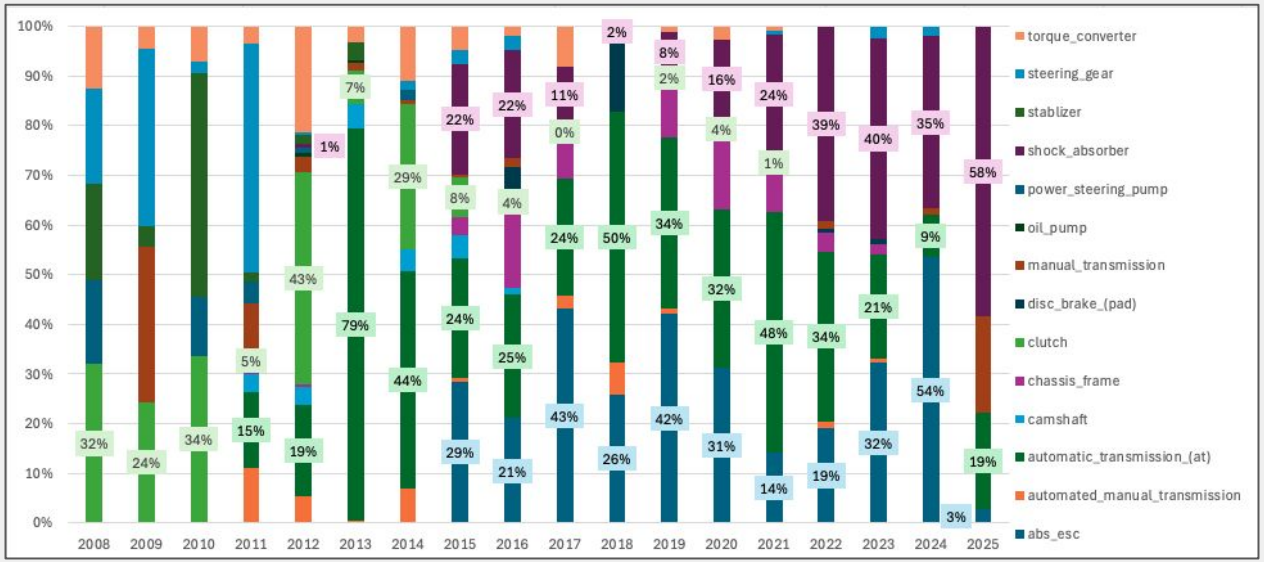
ZF - Changes in Maker Regional Distribution

- maintain high participation in Europe
- growth in China and Japan

Time Series Analysis

ZF - Changes in Product Focus

- Transition from Clutch & Steering Gear to Shock Absorber & Transmission



Time Series Analysis

Top 3 suppliers in each product category

```
with p_rank as (
select
    model_year,
    coalesce(s.parent_company,s.name) as supplier_name,
    product,
    rank() over(partition by model_year,product order by sum(unit) desc) as supplier_rank,
    sum(unit) as unit
from component_fact c
left join supplier s on c.supplier = s.name
group by model_year,supplier_name, product
order by model_year desc, product, supplier_rank)
select supplier_name, count(*) as high_rank_count,
       array_agg(model_year ||':'|| product order by model_year) as high_rank_products
from (select distinct supplier_name, model_year, product from p_rank where supplier_rank<=3) p
group by supplier_name
order by high_rank_count desc;
```

Time Series Analysis

Top 3 suppliers in each product category

	supplier_name	high_rank_count	high_rank_products
1	Aisin	65	{2008:manual_transmission,2008:automatic_transmission_(at),2008:oil_pump,2…
2	ZF	61	{2008:steering_gear,2008:stablizer,2008:power_steering_pump,2008:clutch,20…
3	Bosch	55	{2008:power_steering_pump,2008:fuel_injection,2008:fuel_pump,2008:steering…
4	Denso	38	{2008:starter_motor,2009:fuel_pump,2010:fuel_pump,2010:fuel_injection,2011…
5	Hitachi	38	{2008:piston,2008:shock_absorber,2009:shock_absorber,2010:disc_brake_(pad)…
6	Daihatsu	24	{2011:manual_transmission,2011:drum_brake_(drum),2011:cylinder_block,2011:…
7	MAHLE	24	{2008:piston,2008:camshaft,2009:piston,2010:piston,2011:piston,2012:camsha…
8	JTEKT	23	{2009:steering_gear,2010:power_steering_pump,2011:power_steering_pump,2012…
9	Hyundai	20	{2008:water_pump,2014:manual_transmission,2014:automatic_transmission_(at)…
10	KYB	20	{2008:shock_absorber,2009:shock_absorber,2009:power_steering_pump,2011:sho…

Time Series Analysis

Component Competition

- Count Distinct Suppliers Per Year and Product
- Calculate the Average Number of Suppliers
- Rank Average

```
with competition as (
select distinct model_year, product,
                count(supplier) over(partition by model_year, product) as company_total
from (select distinct  model_year, product, supplier from component_fact
        group by model_year, product, supplier) c
group by model_year, product,supplier
order by model_year)
select distinct product, avg, rank()over(order by avg desc) from
(select distinct product, avg(company_total) over(partition by product) as avg from competition) as rank
order by rank
```

Time Series Analysis

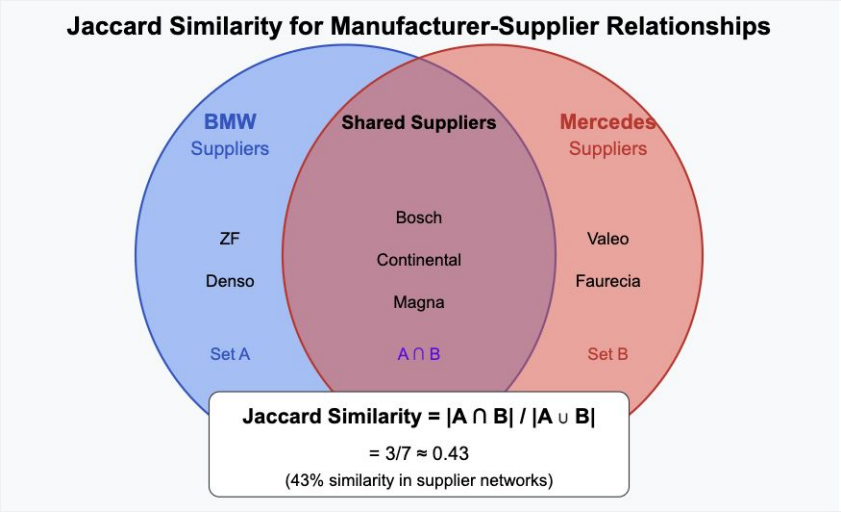
Component Competition (partial)

	product	avg	rank
1	catalytic_converter	65.7272727272727273	1
2	abs_esc	36.2307692307692308	2
3	chassis_frame	20.6111111111111111	3
4	shock_absorber	20.0952380952380952	4
5	automatic_transmission_(at)	18.3333333333333333	5
6	crossmember	15.1875	6
7	water_pump	14.6111111111111111	7
8	clutch	13.9411764705882353	8
9	starter_motor	13.8333333333333333	9
10	camshaft	12.8333333333333333	10
11	manual_transmission	12.8333333333333333	10
12	crankshaft	12.2941176470588235	12
13	fuel_pump	12	13
14	disc_brake_(pad)	11.1764705882352941	14
15	exhaust_manifold	11.1666666666666667	15
16	steering_gear	10.7222222222222222	16

Methodology - Measuring Similarity of Makers

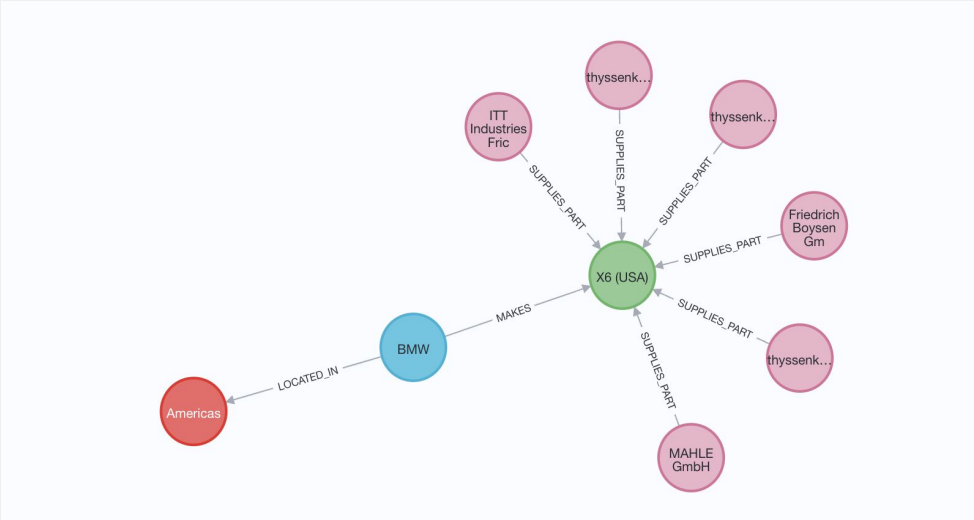
How Similar are different car manufacturers' supply chain overall?

- Need a measure of the similarity of supply chain between makers
  - use jaccard similarity of suppliers between makers
  - calculate this between every pair of makers (Neo4J)



Methodology - Why Jaccard?

- If two car manufacturers have a high jaccard similarity
  - compete for the same limited supplier resources
  - likely produce similar products (cars)
- Structure of output
  - Maker 1, Maker 2, similarity_score, [shared_suppliers]



Methodology - Product Level Analysis

- We can also measure similarity of supply chain within a specific product
  - how a specific component is sourced (eg. transmission)
  - structure of output:
    - maker 1, maker 2, *product, similarity_score, [shared_suppliers]
- Weighted Similarity Aggregation
  - OverallSimilarity(Maker1, Maker2) = Σ [Weight(Product_i) * Similarity(Maker1, Maker2, Product_i)]
    - Use python to calculate this per output of the query

Maker 1	Maker 2	Product	Similarity	Weight	Weighted Similarity
BMW	Audi	Engine	0.8	0.45	0.4
BMW	Audi	Brakes	0.6	0.45	0.18
BMW	Audi	Transmission	0.9	0.1	0.18
			Normalized: 0.77		Total: 0.72

Methodology - Measuring Similarity of Supply Chains

Maker Similarity (Global across all regions)

```cypher
MATCH (m1:Maker)-[:MAKES]->(model1:Model)<-[s1:SUPPLIES_PART]-
(supplier:Supplier)-[s2:SUPPLIES_PART]->(model2:Model)<-[:MAKES]-(m2:Maker)
WHERE m1 <> m2 AND m1 < m2
WITH m1, m2,
     COUNT(DISTINCT supplier) AS intersection,
     COLLECT(DISTINCT supplier) AS sharedSuppliers
WITH m1, m2, intersection, sharedSuppliers,
    [(m1)-[:MAKES]->(model1:Model)<-[:SUPPLIES_PART]-(supplier:Supplier) |
supplier] AS m1Suppliers,
    [(m2)-[:MAKES]->(model2:Model)<-[:SUPPLIES_PART]-(supplier:Supplier) |
supplier] AS m2Suppliers
WITH m1, m2, intersection, sharedSuppliers,
     apoc.coll.union(m1Suppliers, m2Suppliers) AS unionSuppliers
WITH m1, m2, intersection,  unionSuppliers, sharedSuppliers,
     toFloat(intersection) / SIZE(unionSuppliers) AS jaccardSimilarity
WHERE intersection > 5
ORDER BY jaccardSimilarity DESC, m1.name, m2.name
RETURN m1.name AS Maker1, m2.name AS Maker2, jaccardSimilarity, intersection, [s
IN sharedSuppliers | s.name] AS sharedSuppliers
```

Maker Similarity (Global but now considering products)

Cypher

```cypher
MATCH (m1:Maker)-[:MAKES]->(model1:Model)<-[s1:SUPPLIES_PART]-
(supplier:Supplier)-[s2:SUPPLIES_PART]->(model2:Model)<-[:MAKES]-(m2:Maker)
WHERE m1 <> m2 AND s1.product = s2.product AND m1 < m2
WITH m1, m2, s1.product AS Product,
     COUNT(DISTINCT supplier) AS intersection,
     COLLECT(DISTINCT supplier) AS sharedSuppliers
WITH m1, m2, Product, intersection, sharedSuppliers,
    [(m1)-[:MAKES]->(model1:Model)<-[s1:SUPPLIES_PART]-(supplier:Supplier) WHERE
s1.product = Product | supplier] AS m1Suppliers,
    [(m2)-[:MAKES]->(model2:Model)<-[s2:SUPPLIES_PART]-(supplier:Supplier) WHERE
s2.product = Product | supplier] AS m2Suppliers
WITH m1, m2, Product, intersection, sharedSuppliers,
     apoc.coll.union(m1Suppliers, m2Suppliers) AS unionSuppliers
WITH m1, m2, Product, intersection, unionSuppliers, sharedSuppliers,
     toFloat(intersection) / SIZE(unionSuppliers) AS jaccardSimilarity
WHERE intersection > 5 // Filter by intersection size
ORDER BY jaccardSimilarity DESC, m1.name, m2.name, Product
RETURN m1.name AS Maker1, m2.name AS Maker2, Product, jaccardSimilarity,
intersection, [s IN sharedSuppliers | s.name] AS sharedSuppliers
```

Demonstration - Maker Overall Similarity

Joint Ventures

- FAW Toyota & GAC Toyota (0.569, 33)
- Dongfeng Honda & GAC Honda (0.563, 27)
- AIC GM & Shanghai GM (0.531, 34)

Same Parent Company

- Citroen & Peugeot (0.559, 47)
  - PSA Group
- Hyundai & Kia (0.458, 49):
  - Hyundai

Maker1	Maker2	jaccardSimilarity	intersection
FAW Toyota	GAC Toyota	0.5689655172413790	33
Dongfeng Honda	GAC Honda	0.5625	27
Citroen	Peugeot	0.5595238095238100	47
SAIC GM	Shanghai GM	0.53125	34
Hyundai	Kia	0.45794392523364500	49
Beijing Hyundai	Dongfeng Yueda Kia	0.425531914893617	20
Opel	Peugeot	0.41304347826087000	38
Porsche	Skoda	0.410958904109589	30
Mahindra & Mahindra	Tata	0.39285714285714300	22
Bentley	Lamborghini	0.38461538461538500	10
Audi	Porsche	0.3805309734513270	43
Fiat	Peugeot	0.375	30
Audi	Volkswagen	0.37423312883435600	61
Citroen	Opel	0.3655913978494620	34
Brilliance	FAW Haima	0.358974358974359	14
Audi	Daimler	0.3581081081081080	53
Audi	Skoda	0.35514018691588800	38
Citroen	Fiat	0.35443037974683500	28

Demonstration - Maker Overall Similarity

Mahindra & Mahindra and Tata (0.393, 22):

- Two independent Indian major automakers showing significant supplier overlap
- Looking at the supplier list shows sourcing of parts from more local vendors that has low overlap

Conclusion:

- Geo-proximity or domestic markets can drive partial supplier overlap, and corporate linkage tends to push Jaccard scores even higher

Maker1	Maker2	jaccardSimilarity	intersection
FAW Toyota	GAC Toyota	0.5689655172413790	33
Dongfeng Honda	GAC Honda	0.5625	27
Citroen	Peugeot	0.5595238095238100	47
SAIC GM	Shanghai GM	0.53125	34
Hyundai	Kia	0.45794392523364500	49
Beijing Hyundai	Dongfeng Yueda Kia	0.425531914893617	20
Opel	Peugeot	0.41304347826087000	38
Porsche	Skoda	0.410958904109589	30
Mahindra & Mahindra	Tata	0.39285714285714300	22
Bentley	Lamborghini	0.38461538461538500	10
Audi	Porsche	0.3805309734513270	43
Fiat	Peugeot	0.375	30
Audi	Volkswagen	0.37423312883435600	61
Citroen	Opel	0.3655913978494620	34
Brilliance	FAW Haima	0.358974358974359	14
Audi	Daimler	0.3581081081081080	53
Audi	Skoda	0.35514018691588800	38
Citroen	Fiat	0.35443037974683500	28

Maker Product-Level Similarity

- BYD, Brilliance, Dongfeng, FAW, SAIC all share the same supplier for power steering pump

Maker1	Maker2	Product	jaccardSimilarity	intersection
BYD	Brilliance	power_steering_pump	1.0	9
BYD	Dongfeng Passenger Vehicle	power_steering_pump	1.0	9
BYD	FAW Car	power_steering_pump	1.0	9
BYD	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Brilliance	Dongfeng Passenger Vehicle	power_steering_pump	1.0	9
Brilliance	FAW Car	power_steering_pump	1.0	9
Brilliance	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Dongfeng Passenger Vehicle	FAW Car	power_steering_pump	1.0	9
Dongfeng Passenger Vehicle	SAIC Passenger Vehicle	power_steering_pump	1.0	9
FAW Car	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Mazda	Mitsubishi	power_steering_pump	1.0	6
BYD	FAW Haima	power_steering_pump	0.8888888888888890	8
Brilliance	FAW Haima	power_steering_pump	0.8888888888888890	8
Dongfeng Passenger Vehicle	FAW Haima	power_steering_pump	0.8888888888888890	8
FAW Car	FAW Haima	power_steering_pump	0.8888888888888890	8
FAW Haima	SAIC Passenger Vehicle	power_steering_pump	0.8888888888888890	8
Fiat	Renault	starter_motor	0.8571428571428570	6
FAW Toyota	GAC Toyota	abs_esc	0.7777777777777780	7
Mazda	Nissan	torque_converter	0.7777777777777780	7

Maker Product-Level Similarity

- As expected for joint ventures with Toyota, high similarity for ABS/ESC systems
- Not perfect (1.0), suggesting some independent sourcing even within the joint venture

Maker1	Maker2	Product	jaccardSimilarity	intersection
BYD	Brilliance	power_steering_pump	1.0	9
BYD	Dongfeng Passenger Vehicle	power_steering_pump	1.0	9
BYD	FAW Car	power_steering_pump	1.0	9
BYD	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Brilliance	Dongfeng Passenger Vehicle	power_steering_pump	1.0	9
Brilliance	FAW Car	power_steering_pump	1.0	9
Brilliance	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Dongfeng Passenger Vehicle	FAW Car	power_steering_pump	1.0	9
Dongfeng Passenger Vehicle	SAIC Passenger Vehicle	power_steering_pump	1.0	9
FAW Car	SAIC Passenger Vehicle	power_steering_pump	1.0	9
Mazda	Mitsubishi	power_steering_pump	1.0	6
BYD	FAW Haima	power_steering_pump	0.8888888888888890	8
Brilliance	FAW Haima	power_steering_pump	0.8888888888888890	8
Dongfeng Passenger Vehicle	FAW Haima	power_steering_pump	0.8888888888888890	8
FAW Car	FAW Haima	power_steering_pump	0.8888888888888890	8
FAW Haima	SAIC Passenger Vehicle	power_steering_pump	0.8888888888888890	8
Fiat	Renault	starter_motor	0.8571428571428570	6
FAW Toyota	GAC Toyota	abs_esc	0.7777777777777780	7
Mazda	Nissan	torque_converter	0.7777777777777780	7

Maker Weighted Sum of Similarities

Similarity with weights based on safety

- Normalized sum of raw_jaccard: 0.35
- Normalized sum of weighted_jaccard: 0.37

	Maker1	Maker2	Product	jaccardSimilarity	intersection	sharedSuppliers	weight	weighted_jaccard
51	Nissan	Toyota	steering_gear	0.411765	7	[JTEKT Corporation, Rongtai Industrial Develop...	2.00	0.823529
58	Nissan	Toyota	stablizer	0.400000	8	[New Mather Metals, Inc., NHK of America Suspe...	0.70	0.280000
63	Nissan	Toyota	exhaust_manifold	0.388889	7	[Futaba Industrial Co., Ltd., Sango Co., Ltd.,...	0.05	0.019444
74	Nissan	Toyota	abs_esc	0.350000	7	[Bosch (Robert Bosch LLC), Aisin Corporation (...	0.90	0.315000
76	Nissan	Toyota	disc_brake_(pad)	0.344828	10	[Akebono Brake Corporation, ADVICS Co., Ltd., ...	2.00	0.689655
96	Nissan	Toyota	clutch	0.300000	6	[ZF Friedrichshafen AG, Schaeffler AG, P.T. EX...	0.20	0.060000
103	Nissan	Toyota	shock_absorber	0.285714	10	[KYB Americas Corporation - Indiana, thyssenkr...	0.60	0.171429

Conclusion

1. **Supply chain overview:**
   a. Extract top suppliers per product using PostgreSQL
   b. Visualize dominant suppliers and key supply chain shift using Neo4j

2. **Time series analysis**
   a. Analyze market share shifts, YoY changes using PostgreSQL
   b. Conduct statistical analysis in automaker partnerships, component competition, dominant suppliers using PostgreSQL
   c. Dive into Bosch & ZF to inspect maker regional distribution, product focus using PostgreSQL

3. **Maker's similarity**
   a. Find the most similar makers based on supply chain using Neo4j
   b. Weighted maker similarity analysis using Python

Appendix

- GitHub repository: https://github.com/l2lee/Automotive-Supply-Chain-Analysis

Thank you

DSC 202 Group 2
Yi Lien, Jean Hsu, Lucas Lee