

Proyecto_Final

Proyecto Bimestral

FINISHED

Programacion Avanzada

Nombres:

- Jean Carlos Cuenca
- Emilio Pena
- Santiago Rosales

Took 0 sec. Last updated by anonymous at July 15 2025, 3:37:51 PM.

En este proyecto se opto por trabajar con datos sobre una empresa de renta de vehiculos.

FINISHED

Para contrastar estos datos, se consultaron 3 datasets auxiliares con informacion relacionada como marcas de vehiculos, tipos de vehiculos y sus prestaciones, perfiles de conductores y sus características, etc.

Con estos nuevos datos, el objetivo es hacer un analisis complejo involucrando todas las variables presentes.

Took 0 sec. Last updated by anonymous at July 15 2025, 3:40:33 PM.

Crear la base de datos en terminal

FINISHED

```
CREATE DATABASE cars;
```

Took 0 sec. Last updated by anonymous at July 15 2025, 3:38:21 PM.

Proyecto_Final

```
%spark
val jdbcDF = spark.read
  .format("jdbc")
  .option("url","jdbc:mysql://localhost:3306/cars")
  .option("user","root")
  .option("password","cp4u7")
jdbcDF: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@6d61b976
```

Took 14 sec. Last updated by anonymous at July 15 2025, 10:47:34 AM.

```
%MySQL

use cars;

-- Creacion de las tres tablas auxiliares

CREATE TABLE vehiculo_info (
  vehicle_model VARCHAR(100) PRIMARY KEY,
  fuel_efficiency_km_per_gal DECIMAL(5,2),
  engine_type VARCHAR(50),
  transmission VARCHAR(50),
  seats INT
);

CREATE TABLE vehiculo_tipo (
  vehicle_type VARCHAR(50) PRIMARY KEY,
  standard_daily_rate DECIMAL(10,2),
  insurance_fee_per_day DECIMAL(10,2),
  avg_maintenance_cost_monthly DECIMAL(10,2),
  availability_index DECIMAL(5,2)
);

CREATE TABLE marca_region (
  region_origin VARCHAR(50),
  vehicle_brand VARCHAR(50),
  vehicle_model VARCHAR(100),
  average_driver_age DECIMAL(4,1),
  primary_use VARCHAR(50),
  risk_score DECIMAL(4,2),
  owner_demographic VARCHAR(50),
  PRIMARY KEY (region_origin, vehicle_brand, vehicle_model),
  FOREIGN KEY (vehicle_model) REFERENCES vehiculo_info(vehicle_model)
);
```

Query executed successfully. Affected rows : 0

Query executed successfully. Affected rows : 0

FINISHED

Query executed successfully. Affected rows : 0

Proyecto_Final

Query executed successfully. Affected rows : 0

Took 0 sec. Last updated by anonymous at July 15 2025, 11:07:13 AM.

Para insertar la tabla principal (Rentas), ocurrio un error con algnos tipos de dato. Se opto por crear una tabla temporal con datos genericos y despues insertarlos a la tabla oficial mediante un select.

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 10:56:00 AM.

%MySQL

FINISHED

-- Tabla temporal

```
CREATE TABLE rent_temp (  
  start_date VARCHAR(20),  
  end_date VARCHAR(20),  
  customer_name VARCHAR(100),  
  vehicle_model VARCHAR(100),  
  vehicle_type VARCHAR(50),  
  rental_days VARCHAR(10),  
  total_cost VARCHAR(20),  
  miles_driven VARCHAR(20),  
  return_condition VARCHAR(50),  
  payment_method VARCHAR(50)  
);
```

-- Tabla oficial

```
CREATE TABLE rent (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  start_date DATE,  
  end_date DATE,  
  customer_name VARCHAR(100),  
  vehicle_model VARCHAR(100),  
  vehicle_type VARCHAR(50),  
  rental_days INT,  
  total_cost DECIMAL(10,2),  
  miles_driven DECIMAL(10,2),  
  return_condition VARCHAR(50),  
  payment_method VARCHAR(50),  
  FOREIGN KEY (vehicle_model) REFERENCES vehiculo_info(vehicle_model),  
  FOREIGN KEY (vehicle_type) REFERENCES vehiculo_tipo(vehicle_type)
```

Query executed successfully. Affected rows : 0

Proyecto_Final

Query executed successfully. Affected rows : 0

Took 0 sec. Last updated by anonymous at July 15 2025, 11:24:03 AM.

Al igual que antes, para insertar la tabla secundaria del estudio por provincias, ocurrio un error con algnos tipos de dato. Se opto por crear una tabla temporal con datos genericos y despues insertarlos a la tabla oficial mediante un select.

Took 3 sec. Last updated by anonymous at July 19 2025, 9:30:51 PM.

%MySQL

ERROR

use cars;

-- Tabla temporal

```
CREATE TABLE estudio_rentas_raw (  
  id_registro INT,  
  codigo INT,  
  provincia VARCHAR(50),  
  region VARCHAR(50),  
  modelo_vehiculo VARCHAR(100),  
  tipo_vehiculo VARCHAR(50),  
  cantidad_rentas INT,  
  promedio_dias_renta DECIMAL(5,2),  
  costo_promedio_renta DECIMAL(10,2),  
  rendimiento_km_por_galon DECIMAL(5,2),  
  costo_promedio_por_dia DECIMAL(10,2),  
  terreno VARCHAR(20),  
  poblacion INT,  
  densidad_urbana INT,  
  uso_preferido VARCHAR(50)  
);
```

-- Tabla oficial

```
CREATE TABLE estudio_rentas_provincia (  
  id_registro INT PRIMARY KEY,  
  codigo INT,  
  provincia VARCHAR(50),  
  region VARCHAR(50),  
  id_modelo INT,  
  ... , ...  
);
```

Proyecto_Final

Query executed successfully. Affected rows : 0

Table 'estudio_rentas_raw' already exists

Took 0 sec. Last updated by anonymous at July 19 2025, 9:37:09 PM.

```
%MySQL
Use cars;
```

FINISHED

```
-- Creacion de la tabla catalogo para conectar el dataset principal con el estudio por provincias
```

```
CREATE TABLE catalogo_modelos (
  id_modelo INT AUTO_INCREMENT PRIMARY KEY,
  modelo VARCHAR(100) UNIQUE,
  tipo VARCHAR(50)
);
```

Query executed successfully. Affected rows : 0

Query executed successfully. Affected rows : 0

Took 0 sec. Last updated by anonymous at July 19 2025, 9:35:29 PM.

Carga de datos

FINISHED

Took 1 sec. Last updated by anonymous at July 15 2025, 3:37:09 PM.

Estos comandos se ejecutan en la terminal

FINISHED

Proyecto_Final

```
LOAD DATA LOCAL INFILE '/home/arnaut/Desktop/Vehicle_Info__Full.csv'
INTO TABLE vehiculo_info
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
```

```
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```
LOAD DATA LOCAL INFILE '/home/arnaut/Desktop/Rental_Rates_by_Vehicle_Type.csv'
INTO TABLE vehiculo_tipo
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```
LOAD DATA LOCAL INFILE '//home/arnaut/Desktop/Driver_Profile_by_Brand_and_Region.csv'
INTO TABLE marca_region
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```
LOAD DATA LOCAL INFILE '/home/arnaut/Desktop/Car_Rentals_-_Main_Dataset.csv'
INTO TABLE rent_temp
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```
LOAD DATA LOCAL INFILE '/home/arnaut/Desktop/Estudio_de_Rentas_por_Provincia.csv'
INTO TABLE estudio_rentas_raw
```

FIELDS TERMINATED BY ',' ENCLOSED BY ''

LINES TERMINATED BY '\n'

IGNORE 1 LINES:

Proyecto_Final

Took 0 sec. Last updated by anonymous at July 19 2025, 9:38:51 PM.

%MySQL

FINISHED

-- Insertar los datos desde la tabla temporal con datos genericos a la tabla oficial con los datos correctos

```
INSERT INTO rent (
  start_date, end_date, customer_name, vehicle_model, vehicle_type,
  rental_days, total_cost, miles_driven, return_condition, payment_method
)
```

```
SELECT
  STR_TO_DATE(start_date, '%Y-%m-%d'),
  STR_TO_DATE(end_date, '%Y-%m-%d'),
  customer_name,
  vehicle_model,
  vehicle_type,
  rental_days + 0,
  total_cost + 0.0,
  miles_driven + 0.0,
  return_condition,
  payment_method
FROM rent_temp
WHERE vehicle_model IN (SELECT vehicle_model FROM vehiculo_info)
  AND vehicle_type IN (SELECT vehicle_type FROM vehiculo_tipo)
  AND STR_TO_DATE(start_date, '%Y-%m-%d') IS NOT NULL
  AND STR_TO_DATE(end_date, '%Y-%m-%d') IS NOT NULL;
```

Query executed successfully. Affected rows : 10000

Took 0 sec. Last updated by anonymous at July 15 2025, 11:25:30 AM.

%MySQL

FINISHED

-- Insertar los datos el estudio de provincias desde la tabla temporal

```
INSERT INTO estudio_rentas_provincia (
  id_registro, codigo, provincia, region, id_modelo, cantidad_rentas,
  promedio_dias_renta, costo_promedio_renta, rendimiento_km_por_galon,
  costo_promedio_por_dia, terreno, poblacion, densidad_urbana, uso_preferido
)
SELECT
  r.id_registro, r.codigo, r.provincia, r.region,
```

Proyecto_Final

Query executed successfully. Affected rows : 88

Took 0 sec. Last updated by anonymous at July 19 2025, 9:49:22 PM.

%MySQL

FINISHED

-- Insertar datos filtrados desde la tabla catalogo temporal con datos genericos a la tabla catalogo oficial con los datos correctos

```
INSERT INTO catalogo_modelos (modelo, tipo)
SELECT DISTINCT vehicle_model, vehicle_type
FROM rent
WHERE vehicle_model IS NOT NULL;
```

Query executed successfully. Affected rows : 8

Took 0 sec. Last updated by anonymous at July 19 2025, 9:46:51 PM.

TABLA RENTAS Y AUXILIARES

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 5:10:17 PM.

Tabla de rentas previo a agregar campos auxiliares

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 5:10:15 PM.

%MySQL

FINISHED

```
select * from rent;
```

















settings ▼



Proyecto_Final

| id | start_date | end_date | customer_name | vehicle_model | vehicle_type | rental_days |
|----|------------|------------|-----------------|---------------|--------------|-------------|
| 1 | 2024-12-18 | 2024-12-20 | Meredith Wilson | Vitara | SUV | 2 |
| 2 | 2024-12-07 | 2024-12-20 | Allen Wilson | Vitara | SUV | 13 |
| 3 | 2024-07-16 | 2024-07-25 | Raymond Barker | Hilux | pickup | 9 |
| 4 | 2024-11-07 | 2024-11-16 | Jamie Sullivan | Sentra | sedan | 9 |
| 5 | 2025-04-07 | 2025-04-18 | Heather Morgan | Sentra | sedan | 11 |
| 6 | 2024-10-30 | 2024-11-14 | Ashley Jones | Corolla | sedan | 15 |
| 7 | 2024-08-18 | 2024-08-24 | Ethan Hahn | Hilux | pickup | 6 |
| 8 | 2025-05-28 | 2025-06-02 | Amanda Cruz | Tucson | SUV | 5 |

Took 1 sec. Last updated by anonymous at July 15 2025, 11:27:05 AM. (outdated)

Agregación de atributos de vehiculo_info

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 5:09:59 PM.

%MySQL

FINISHED

```
-- Agregar campos adicionales a la tabla principal rent

ALTER TABLE rent
ADD COLUMN fuel_efficiency_km_per_gal DECIMAL(5,2),
ADD COLUMN engine_type VARCHAR(50),
ADD COLUMN transmission VARCHAR(50),
ADD COLUMN seats INT;

-- Insertar los datos desde la tabla auxiliar Vehiculo_info

UPDATE rent r
JOIN vehiculo_info v ON r.vehicle_model = v.vehicle_model
SET
  r.fuel_efficiency_km_per_gal = v.fuel_efficiency_km_per_gal,
  r.engine_type = v.engine_type,
  r.transmission = v.transmission,
  r.seats = v.seats;
```

Query executed successfully. Affected rows : 10000

Took 0 sec. Last updated by anonymous at July 15 2025, 11:31:34 AM. (outdated)

Proyecto_Final

%MySQL
SELECT * FROM rent;

FINISHED

settings ▼

| id | start_date | end_date | customer_name | vehicle_model | vehicle_type | rental_days |
|----|------------|------------|-----------------|---------------|--------------|-------------|
| 1 | 2024-12-18 | 2024-12-20 | Meredith Wilson | Vitara | SUV | 2 |
| 2 | 2024-12-07 | 2024-12-20 | Allen Wilson | Vitara | SUV | 13 |
| 3 | 2024-07-16 | 2024-07-25 | Raymond Barker | Hilux | pickup | 9 |
| 4 | 2024-11-07 | 2024-11-16 | Jamie Sullivan | Sentra | sedan | 9 |
| 5 | 2025-04-07 | 2025-04-18 | Heather Morgan | Sentra | sedan | 11 |
| 6 | 2024-10-30 | 2024-11-14 | Ashley Jones | Corolla | sedan | 15 |
| 7 | 2024-08-18 | 2024-08-24 | Ethan Hahn | Hilux | pickup | 6 |
| 8 | 2025-05-28 | 2025-06-02 | Amanda Cruz | Tucson | SUV | 5 |
| 9 | 2025-01-14 | 2025-01-22 | Gasper Anderson | Amarok | pickup | 8 |

Output is truncated to 102400 bytes. [Learn more about ZEPPELIN_INTERPRETER_OUTPUT_LIMIT](#)

Took 0 sec. Last updated by anonymous at July 15 2025, 11:32:17 AM. (outdated)

Agregación de atributos de vehiculo_tipo

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 5:09:54 PM.

Proyecto_Final

%MySQL

FINISHED

```
-- Agregar campos adicionales a la tabla principal rent
ALTER TABLE rent
ADD COLUMN standard_daily_rate DECIMAL(10,2),
ADD COLUMN insurance_fee_per_day DECIMAL(10,2),
ADD COLUMN avg_maintenance_cost_monthly DECIMAL(10,2),
ADD COLUMN availability_index DECIMAL(5,2);

-- Insertar los datos desde la tabla auxiliar vehiculo_tipo

UPDATE rent r
JOIN vehiculo_tipo t ON r.vehicle_type = t.vehicle_type
SET
  r.standard_daily_rate = t.standard_daily_rate,
  r.insurance_fee_per_day = t.insurance_fee_per_day,
  r.avg_maintenance_cost_monthly = t.avg_maintenance_cost_monthly,
  r.availability_index = t.availability_index;
```

Query executed successfully. Affected rows : 0

Query executed successfully. Affected rows : 10000

Took 1 sec. Last updated by anonymous at July 15 2025, 11:34:36 AM.

%MySQL

FINISHED

```
select * from rent;
```





settings ▼

| id | start_date | end_date | customer_name | vehicle_model | vehicle_type | rental_days |
|----|------------|------------|-----------------|---------------|--------------|-------------|
| 1 | 2024-12-18 | 2024-12-20 | Meredith Wilson | Vitara | SUV | 2 |
| 2 | 2024-12-07 | 2024-12-20 | Allen Wilson | Vitara | SUV | 13 |
| 3 | 2024-07-16 | 2024-07-25 | Raymond Barker | Hilux | pickup | 9 |
| 4 | 2024-11-07 | 2024-11-16 | Jamie Sullivan | Sentra | sedan | 9 |

Proyecto_Final

| | | | | | | |
|---|------------|------------|----------------|---------|--------|----|
| 5 | 2025-04-07 | 2025-04-18 | Heather Morgan | Sentra | sedan | 11 |
| 6 | 2024-10-30 | 2024-11-14 | Ashley Jones | Corolla | sedan | 15 |
| 7 | 2024-08-18 | 2024-08-24 | Ethan Hahn | Hilux | pickup | 6 |
| 8 | 2025-05-28 | 2025-06-02 | Amanda Cruz | Tucson | SUV | 5 |

Output is truncated to 102400 bytes. Learn more about ZEPPELIN_INTERPRETER_OUTPUT_LIMIT 

Took 0 sec. Last updated by anonymous at July 15 2025, 11:35:24 AM. (outdated)

Agregación de atributos de marca_region FINISHED

Took 0 sec. Last updated by anonymous at July 15 2025, 5:09:50 PM.

%MySQL FINISHED

```
-- Agregar campos adicionales a la tabla principal rent

ALTER TABLE rent
ADD COLUMN average_driver_age DECIMAL(4,1),
ADD COLUMN primary_use VARCHAR(50),
ADD COLUMN risk_score DECIMAL(4,2),
ADD COLUMN owner_demographic VARCHAR(50);

-- Insertar los datos desde la tabla auxiliar marca_region

UPDATE rent r
JOIN (
  SELECT vehicle_model,
         AVG(average_driver_age) AS average_driver_age,
         MAX(primary_use) AS primary_use,
         AVG(risk_score) AS risk_score,
         MAX(owner_demographic) AS owner_demographic
  FROM marca_region
  GROUP BY vehicle_model
) m ON r.vehicle_model = m.vehicle_model
SET
  r.average_driver_age = m.average_driver_age,
  r.primary_use = m.primary_use,
  r.risk_score = m.risk_score,
  r.owner_demographic = m.owner_demographic;
```

Query executed successfully. Affected rows : 0

Proyecto_Final

Query executed successfully. Affected rows : 7468

Took 0 sec. Last updated by anonymous at July 15 2025, 11:39:43 AM.

Configuracion del dataframe

FINISHED

Una vez tenemos la tabla rent lista en la base de datos, configuramos los dataframes para hacer análisis de datos.

Took 3 sec. Last updated by anonymous at July 21 2025, 12:51:39 AM.

```
val df = spark.read
  .format("jdbc")
  .option("url", "jdbc:mysql://localhost:3306/cars")
  .option("dbtable", "rent")
  .option("user", "root")
  .option("password", "4rn4u7")
  .load()
```

FINISHED

df: org.apache.spark.sql.DataFrame = [id: int, start_date: date ... 21 more fields]

Took 0 sec. Last updated by anonymous at July 15 2025, 3:36:33 PM.

```
val catalogo = spark.read
  .format("jdbc")
  .option("url", "jdbc:mysql://localhost:3306/cars")
  .option("dbtable", "catalogo_modelos")
  .option("user", "root")
  .option("password", "4rn4u7")
  .load()
```

FINISHED

catalogo: org.apache.spark.sql.DataFrame = [id_modelo: int, modelo: string ... 1 more field]

Took 1 sec. Last updated by anonymous at July 21 2025, 12:52:14 AM.

Proyecto_Final

rentas_prov: org.apache.spark.sql.DataFrame = [id_registro: int, codigo: int ... 12 more fields]

Took 0 sec. Last updated by anonymous at July 21 2025, 12:52:30 AM.

df.show()

SPARK JOB (http://10.0.2.15:4040/jobs/job?id=3) FINISHED

| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
|--|-------------|--------------|-----------------|---------------------|-----------------------|------------------------------|--------------------|--------------------|------------------|----------------|----------------------------|-------------|--------------|
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
| id | start_date | end_date | customer_name | vehicle_model | vehicle_type | rental_days | total_cost | miles_driven | return_condition | payment_method | fuel_efficiency_km_per_gal | engine_type | transmission |
| per_gal | engine_type | transmission | seats | standard_daily_rate | insurance_fee_per_day | avg_maintenance_cost_monthly | availability_index | average_driver_age | primary_use | risk_score | owner_demographic | | |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----- | | | | | | | | | | | | | |
| 1 | 2024-12-18 | 2024-12-20 | Meredith Wilson | Vitara | SUV | 2 | 81.94 | 276.00 | normal | cash | | | |
| 37.30 | diesel | manual | 5 | 66.58 | | 5.67 | | 298.54 | 0.51 | null | null | | |
| null | | null | | | | | | | | | | | |
| 2 | 2024-12-07 | 2024-12-20 | Allen Wilson | Vitara | SUV | 13 | 978.75 | 2486.00 | normal | cash | | | |
| 37.30 | diesel | manual | 5 | 66.58 | | 5.67 | | 298.54 | 0.51 | null | null | | |
| null | | null | | | | | | | | | | | |
| 3 | 2024-07-16 | 2024-07-25 | Raymond Barker | Hilux | pickup | 9 | 652.40 | 1596.00 | damaged | credit_card | | | |
| 36.90 | electric | manual | 4 | 56.57 | | 12.28 | | 393.55 | 0.44 | 45.0 | business | | |

Took 1 sec. Last updated by anonymous at July 15 2025, 3:36:42 PM.

Análisis de datos

FINISHED

Una vez tenemos listo el DataFrame (df), procedemos con el análisis de sus datos y su interpretacion.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:09:18 PM.

1. Ingresos y Disponibilidad por Tipo de Vehículo

FINISHED

¿Qué tipo de vehículo genera más ingresos?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:07:47 PM.

```
val resumenIngresos = df.groupBy("vehicle_type")
  .agg(
    sum("total_cost").as("total_income"),
    avg("rental_days").as("avg_rental_days"),
    avg("availability_index").as("avg_availability")
  )

z.show(resumenIngresos)
```

SPARK JOB FINISHED

settings ▼

| vehicle_type ▼ | total_income ▼ | avg_rental_days ▼ | avg_availability | |
|----------------|----------------|-------------------|------------------|--|
| SUV | 1833745.08 | 8.01077246452969 | 0.510000 | |
| sedan | 1762519.72 | 7.947397110929408 | 0.650000 | |
| pickup | 1206612.93 | 7.937425742574257 | 0.440000 | |

```
resumenIngresos: org.apache.spark.sql.DataFrame = [vehicle_type: string, total_income: decimal(20,2) ... 2 more fields]
```

Took 2 sec. Last updated by anonymous at July 15 2025, 3:42:00 PM. (outdated)

Proyecto_Final

Análisis

FINISHED

Los resultados nos muestran que los SUV son los que más ingresos generan, lo que los convierte en una opción bastante rentable. Les siguen de cerca los sedán, mientras que los pickup quedan bastante por debajo en cuanto a facturación. A pesar de estas diferencias, el tiempo promedio de alquiler es muy similar para los tres tipos de vehículos (alrededor de una semana), lo que indica que la diferencia en ingresos se debe más a las tarifas que manejan y la demanda que tienen.

En cuanto a disponibilidad, los sedán son los que más tiempo pasan libres (índice de 0.65), lo que podría significar que hay muchas unidades o que simplemente no se alquilan tanto. En cambio, los pickup tienen la menor disponibilidad (0.44), quizás porque hay pocos disponibles o porque se usan para casos muy específicos.

Entonces, podemos concluir que los SUV resultan ser una excelente inversión por su rentabilidad y demanda estable. Los sedán son una opción muy equilibrada y confiable. Los pickup, en cambio parecen ser vehículos de nicho, útiles, pero no como para apostar por ellos a gran escala.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:02:43 PM.

2. Rentabilidad por tipo de motor y tipo de transmisión

FINISHED

¿Qué combinaciones de motor y transmisión generan más ingresos y cuántas veces se alquilan?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:07:17 PM.

```
val rentabilidadPorConfiguracion = df.groupBy("engine_type", "transmission")
  .agg(
    count("*").as("total_rentals"),
    sum("total_cost").as("total_income"),
    avg("rental_days").as("avg_rental_days")
  )
```

 SPARK JOB FINISHED

Proyecto_Final



| engine_type | transmission | total_rentals | total_income | avg_rental_per_day |
|-------------|--------------|---------------|--------------|--------------------|
| diesel | manual | 3774 | 1813444.45 | 7.998410174880 |
| electric | manual | 2557 | 1226913.56 | 7.956589753617 |
| hybrid | automatic | 1232 | 602818.36 | 8.051136363636 |
| diesel | automatic | 1239 | 596350.64 | 7.947538337368 |
| electric | automatic | 1198 | 563350.72 | 7.840567612687 |

rentabilidadPorConfiguracion: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [engine_type: string, transmission: string ... 3 more fields]

Took 1 sec. Last updated by anonymous at July 15 2025, 4:13:45 PM. (outdated)

Análisis

FINISHED

Al observar la combinación entre tipo de motor y transmisión, notamos que los vehículos diésel con transmisión manual son los que generan mayores ingresos. También son los más alquilados, con 3774 rentas registradas. Esto sugiere que son una opción bastante popular y rentable dentro de la flota.

En segundo lugar están los vehículos eléctricos con transmisión manual , que aunque generan menos ingresos, siguen siendo una combinación frecuente con más de 2500 alquileres.

Las transmisiones automáticas , aunque también rentables, aparecen menos veces en el histórico. La configuración híbrido + automática , por ejemplo, tiene una buena cifra de ingresos considerando su volumen, y la diésel + automática también muestra un rendimiento estable

Proyecto_Final

Se puede decir que la configuración diésel manual lidera tanto en rentas como en ingresos, lo que indica que esta combinación debería mantenerse o incluso ampliarse. Las transmisiones automáticas también ofrecen buen desempeño, aunque con menor frecuencia, por lo que podrían tener potencial si se promueven más.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:09:02 PM.

3. Relación entre eficiencia de combustible y costo total o kilometraje

FINISHED

¿Los autos más eficientes generan más ganancias por km?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:06:53 PM.

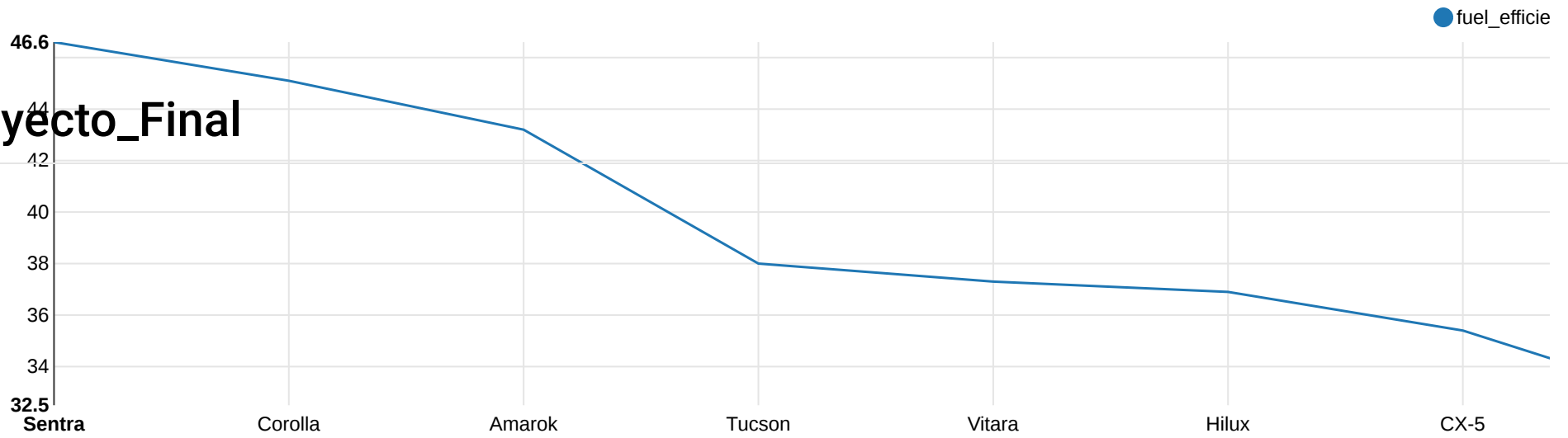
```
val eficienciaVsRentabilidad = df.select(
  col("vehicle_model"),
  col("fuel_efficiency_km_per_gal"),
  col("miles_driven"),
  col("total_cost")
).groupBy("vehicle_model", "fuel_efficiency_km_per_gal")
  .agg(
    avg("miles_driven").as("avg_miles_driven"),
    avg("total_cost").as("avg_total_cost")
  )
  .orderBy(desc("fuel_efficiency_km_per_gal"))

z.show(eficienciaVsRentabilidad)
```

SPARK JOB FINISHED

settings

Proyecto_Final



eficienciaVsRentabilidad: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, fuel_efficiency_km_per_gal: decimal(5,2) ... 2 more fields]

Took 0 sec. Last updated by anonymous at July 15 2025, 4:19:20 PM. (outdated)

settings

SPARK JOB FINISHED

| vehicle_model | fuel_efficiency_km_per_gal | avg_miles_driven | avg_total_cost |
|---------------|----------------------------|------------------|----------------|
| Sentra | 46.60 | 995.108152 | 481.316094 |
| Corolla | 45.10 | 983.123539 | 470.242671 |
| Amarok | 43.20 | 993.397951 | 481.416619 |
| Tucson | 38.00 | 1008.656418 | 485.179339 |
| Vitara | 37.30 | 998.638163 | 469.138836 |
| Hilux | 36.90 | 990.353503 | 474.279650 |

19 of 37

7/21/25, 01:07

| | | | |
|-------|-------|-------------|------------|
| CX-5 | 35.40 | 1022.403382 | 491.146868 |
| Yaris | 32.50 | 1017.939123 | 489.300617 |

Proyecto_Final

```
df = df.withColumn("rentabilidad", org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, fuel_efficiency_kn_per_gal: decimal(5,2) ... 2 more f
ields])
```

Took 0 sec. Last updated by anonymous at July 15 2025, 4:23:02 PM. (outdated)

Análisis

FINISHED

En general, los modelos más eficientes, como el `Sentra` y el `Corolla`, registran un kilometraje y un ingreso promedio bastante similar al resto. Sin embargo, modelos con menor eficiencia como el `Yaris` o el `CX-5` no muestran una caída significativa ni en distancia recorrida ni en ingresos.

Llama la atención que algunos modelos menos eficientes como el `CX-5` tienen incluso el costo promedio más alto (491.15), lo que sugiere que factores como el tipo de vehículo o la tarifa base influyen más que la eficiencia en la rentabilidad.

Aunque en el grafico lineal se muestra una tendencia bajista en la eficiencia entre los modelos, no hay una correlación clara entre eficiencia de combustible y rentabilidad. Esto indica que en el negocio de alquiler, la elección del vehículo depende más de su categoría, tarifas y demanda que del consumo por kilómetro.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:55 PM.

4. Promedio de días de alquiler por tipo de cliente y costo por día

FINISHED

¿Qué perfil demográfico alquila por más tiempo?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:06:42 PM.

```
import org.apache.spark.sql.functions._

val analisisDemografico = df.groupBy("owner_demographic")
```

SPARK JOB FINISHED

Proyecto_Final



 settings ▼

| owner_demographic ▼ | avg_rental_days ▼ | avg_total_cost ▼ | avg_cost_per_day ≡ |
|---------------------|-------------------|------------------|--------------------|
| middle_aged | 8.11352657004831 | 491.146868 | 60.53432669008633 |
| young_adult | 7.969014084507042 | 481.637433 | 60.43877296394486 |
| null | 7.941943127962086 | 475.292275 | 59.84584217514546 |
| senior | 7.880573248407643 | 474.279650 | 60.18339466558901 |

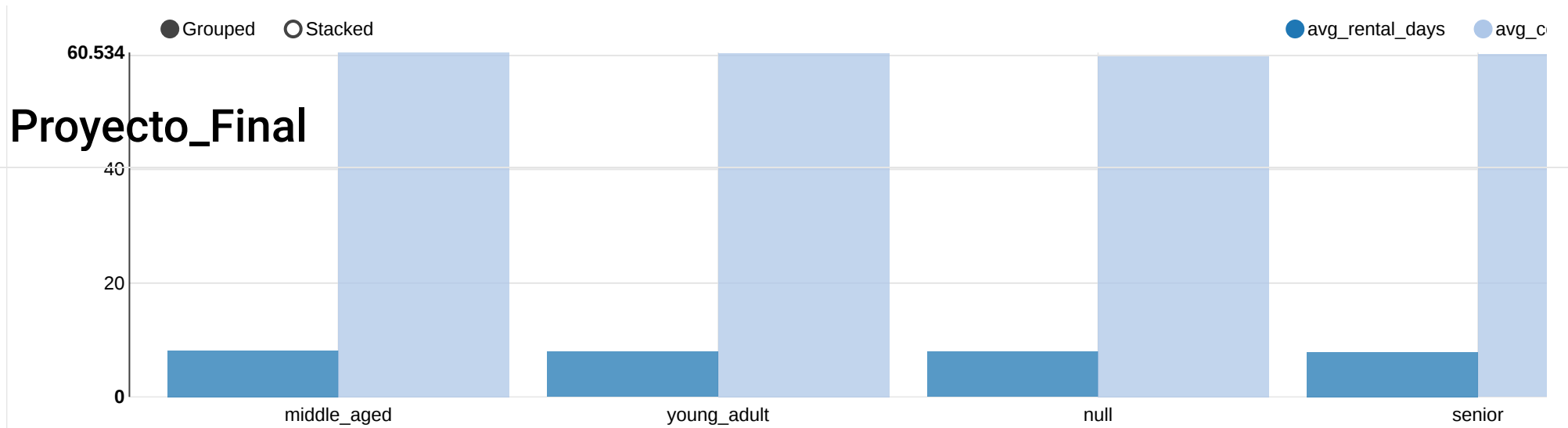
```
import org.apache.spark.sql.functions._  
analysisDemografico: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [owner_demographic: string, avg_rental_days: double ... 2 more fields]
```

Took 0 sec. Last updated by anonymous at July 15 2025, 4:51:25 PM.



 settings ▼

≡ SPARK JOB FINISHED



```
import org.apache.spark.sql.functions._
 analisisDemografico: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [owner_demographic: string, avg_rental_days: double ... 2 more fields]
```

Took 1 sec. Last updated by anonymous at July 15 2025, 4:28:45 PM. (outdated)

Análisis

FINISHED

Los resultados muestran que los clientes `middle_aged` son quienes más días alquilan (8.11 días promedio), seguidos por los `young_adult` y los `senior`, que se mantienen cerca del promedio general. También son quienes pagan un poco más por día (≈ 60.53), aunque la diferencia con los demás grupos no es significativa.

Llama la atención que los `young_adult`, pese a alquilar ligeramente menos tiempo, tienen un costo promedio por día bastante alto, lo que podría reflejar preferencias por modelos más caros o tarifas más elevadas.

unque las diferencias no son drásticas, los adultos de mediana edad tienden a alquilar por más tiempo, mientras que los jóvenes pagan casi lo mismo por día, pero en períodos más cortos. Esta información puede ayudar a personalizar promociones o paquetes

según el perfil del cliente.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:50 PM.

Proyecto_Final

5. Relación entre mantenimiento mensual promedio y rentabilidad

FINISHED

¿Vale la pena tener vehículos con alto costo de mantenimiento?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:06:28 PM.

```
val mantenimientoVsIngreso = df.groupBy("vehicle_model")
  .agg(
    avg("avg_maintenance_cost_monthly").as("avg_maintenance"),
    avg("total_cost").as("avg_total_cost"),
    count("*").as("rental_count")
  )
  .withColumn("cost_ratio", col("avg_total_cost") / col("avg_maintenance"))
  .orderBy(desc("cost_ratio"))

z.show(mantenimientoVsIngreso)

// Creamos una columna extra cost_ratio que compara ingreso promedio por alquiler vs. mantenimiento mensual.
```

SPARK JOB FINISHED

settings ▼

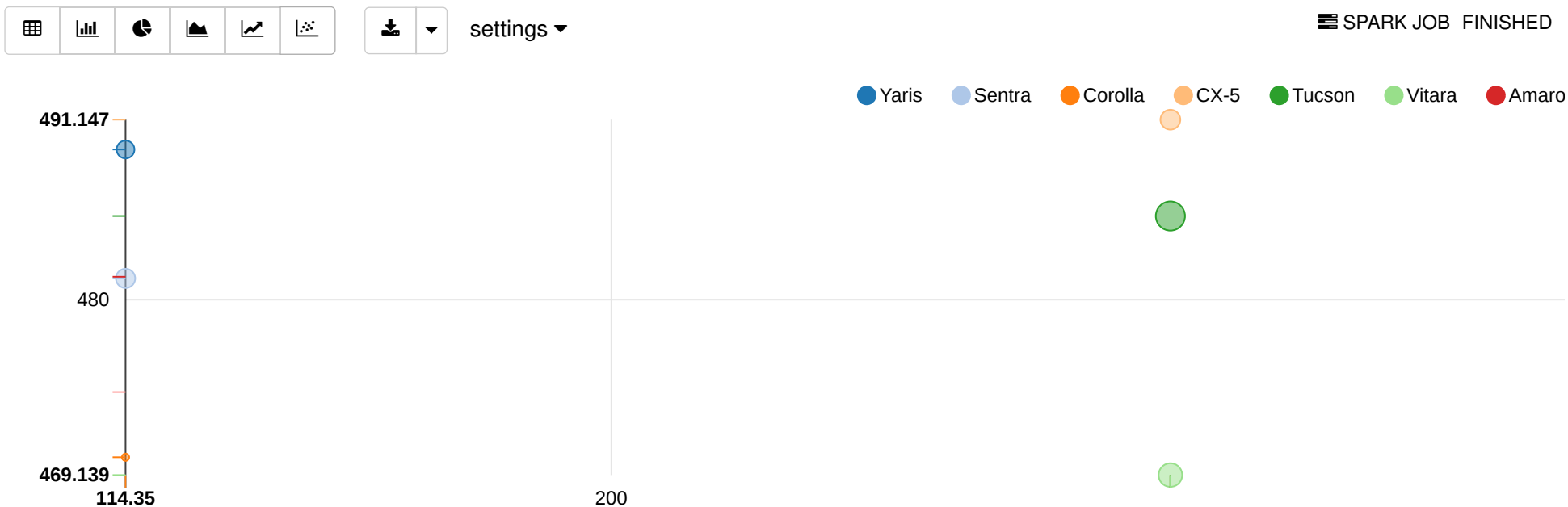
| vehicle_model ▼ | avg_maintenance ▼ | avg_total_cost ▼ | rental_count ▼ | cost_ratio | ⋮ |
|-----------------|-------------------|------------------|----------------|----------------|---|
| Yaris | 114.350000 | 489.300617 | 1232 | 4.278973476165 | |
| Sentra | 114.350000 | 481.316094 | 1239 | 4.209148176650 | |
| Corolla | 114.350000 | 470.242671 | 1198 | 4.112310196764 | |
| CX-5 | 298.540000 | 491.146868 | 1242 | 1.645162685067 | |
| Tucson | 298.540000 | 485.179339 | 1301 | 1.625173641720 | |
| Vitara | 298.540000 | 469.138836 | 1263 | 1.571443813224 | |
| Amarok | 393.550000 | 481.416619 | 1269 | 1.223266723415 | |

| | | | | |
|-------|------------|------------|------|----------------|
| Hilux | 393.550000 | 474.279650 | 1256 | 1.205131876506 |
|-------|------------|------------|------|----------------|

Proyecto_Final

`mantenimientoVsIngreso: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, avg_maintenance: decimal(14,6) ... 3 more fields]`

Took 1 sec. Last updated by anonymous at July 15 2025, 4:42:10 PM. (outdated)



`mantenimientoVsIngreso: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, avg_maintenance: decimal(14,6) ... 3 more fields]`

Took 1 sec. Last updated by anonymous at July 15 2025, 4:45:23 PM. (outdated)

Análisis

FINISHED

El gráfico revela una distinción clara entre dos grupos:

Proyecto_Final

- Yaris , Sentra y Corolla tienen costos de mantenimiento bajos (≈114) y, sin embargo, generan ingresos promedio altos por alquiler (entre 470 y 490). Además, tienen ratios de rentabilidad (cost_ratio) superiores a 4, lo que los convierte en modelos muy eficientes.
- Por otro lado, modelos como el Hilux y el Amarok , con costos de mantenimiento mucho más altos (≈393), no compensan este gasto con ingresos proporcionales, manteniendo ratios por debajo de 1.3.

Podemos concluir que lLos modelos más económicos de mantener, como el Yaris y el Sentra , son también los más rentables en términos de ingresos por mantenimiento. En cambio, los vehículos con altos costos mensuales (como el Hilux) podrían ser revisados, ya que su retorno no parece justificar el gasto operativo.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:43 PM.

6. Comparación de comportamiento por método de pago

FINISHED

¿Hay diferencias en daños, riesgo o duración según cómo se paga?

Took 0 sec. Last updated by anonymous at July 15 2025, 5:06:17 PM.

```
val comportamientoPago = df.groupBy("payment_method")
  .agg(
    avg("rental_days").as("avg_rental_days"),
    avg("total_cost").as("avg_total_cost"),
    count("*").as("total_rentals")
  )
  .withColumn("avg_cost_per_day", col("avg_total_cost") / col("avg_rental_days"))
  .orderBy(desc("total_rentals"))

z.show(comportamientoPago)

// Agregamos una columna (avg_cost_per_day) para ver si el método de pago tiene relación con tarifas más altas.
```

SPARK JOB FINISHED

settings ▼

| payment_method ▼ | avg_rental_days ▼ | avg_total_cost ▼ | total_rentals ▼ | avg_cost_per_day ▼ | avg_cost_per_rental ▼ |
|------------------|-------------------|------------------|-----------------|--------------------|-----------------------|
| | | | | | |

Proyecto_Final

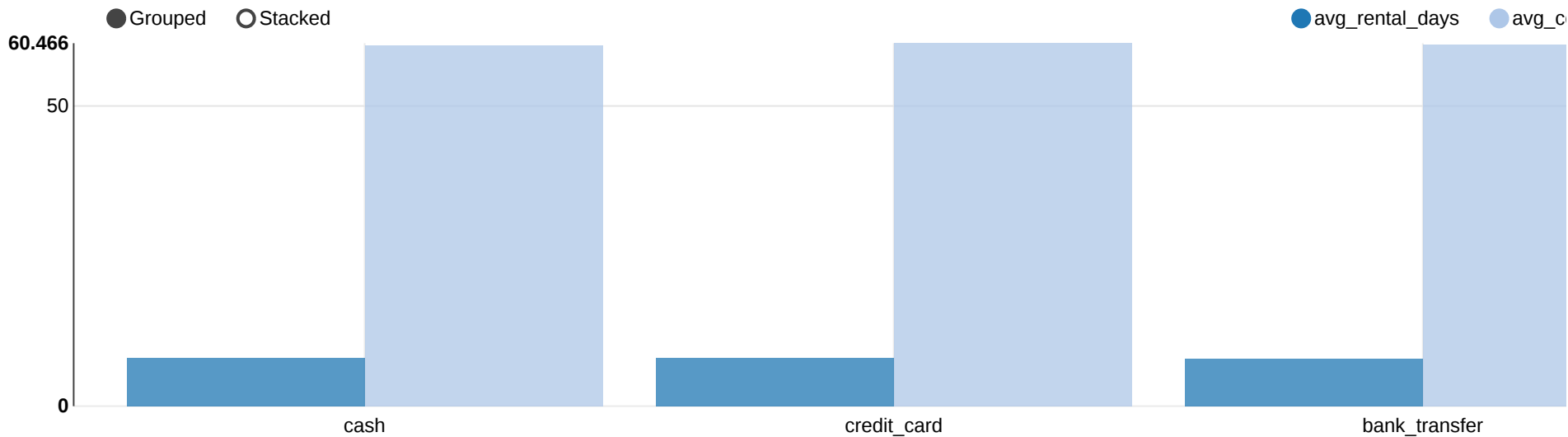
| | | | | |
|---------------|-------------------|------------|------|----------------|
| cash | 8.04016661707825 | 483.680925 | 3361 | 60.15807234300 |
| credit_card | 7.994309673554956 | 483.386301 | 3339 | 60.46629674592 |
| bank_transfer | 7.870909090909091 | 473.696752 | 3300 | 60.18323252483 |

comportamientoPago: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [payment_method: string, avg_rental_days: double ... 3 more fields]

Took 0 sec. Last updated by anonymous at July 15 2025, 5:00:15 PM.

settings ▾

SPARK JOB FINISHED



```
comportamientoPago: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [payment_method: string, avg_rental_days: double ... 3 more fields]
```

Took 0 sec. Last updated by anonymous at July 15 2025, 4:53:15 PM. (outdated)

Proyecto_Final

Análisis

FINISHED

Las diferencias entre métodos de pago son mínimas, pero se pueden observar ciertos patrones. Los clientes que pagaron con `cash` tuvieron la duración de alquiler promedio más alta, con (8.04) días. Le siguen quienes usaron `credit_card`, y en último lugar quienes pagaron por `bank_transfer`.

En cuanto al costo por día, el valor más alto fue para el método `credit_card`, con (60.46), seguido de cerca por `bank_transfer` y `cash`. La diferencia entre ellos es pequeña, pero consistente.

Esto sugiere que los métodos de pago no influyen de forma drástica en el comportamiento de alquiler, aunque quienes usan `cash` tienden a alquilar por más tiempo, y quienes usan `credit_card` pagan más por día en promedio.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:34 PM.

7. Asociación entre índice de disponibilidad y cantidad de días alquilado

FINISHED

¿Qué tan bien se están aprovechando los vehículos?

Took 0 sec. Last updated by anonymous at July 15 2025, 4:56:31 PM.

```
val disponibilidadVsUso = df.groupBy("vehicle_model")
  .agg(
    avg("availability_index").as("avg_availability"),
    avg("rental_days").as("avg_rental_days")
  )
  .withColumn("utilization_score", col("avg_rental_days") / col("avg_availability"))
  .orderBy(desc("utilization_score"))

z.show(disponibilidadVsUso)

// Agregaos una columna (utilization_score) que relaciona los valores (avg_rental_days / availability_index)
```

 SPARK JOB FINISHED

settings ▼

vehicle_model ▼

avg_availability ▼

avg_rental_days ▼

utilization_score

≡

| | | | | |
|---------|----------|--------------------|--------------------|--|
| Amarok | 0.440000 | 7.9936958234830575 | 18.16749050791604 | |
| Hilux | 0.440000 | 7.880573248407643 | 17.910393746381008 | |
| CX-5 | 0.510000 | 8.11352657004831 | 15.908875627545704 | |
| Tucson | 0.510000 | 8.029976940814757 | 15.745052825126974 | |
| Vitara | 0.510000 | 7.889944576405384 | 15.470479561579184 | |
| Yaris | 0.650000 | 8.051136363636363 | 12.386363636363635 | |
| Sentra | 0.650000 | 7.9475383373688455 | 12.226982057490531 | |
| Corolla | 0.650000 | 7.840567612687813 | 12.062411711827405 | |

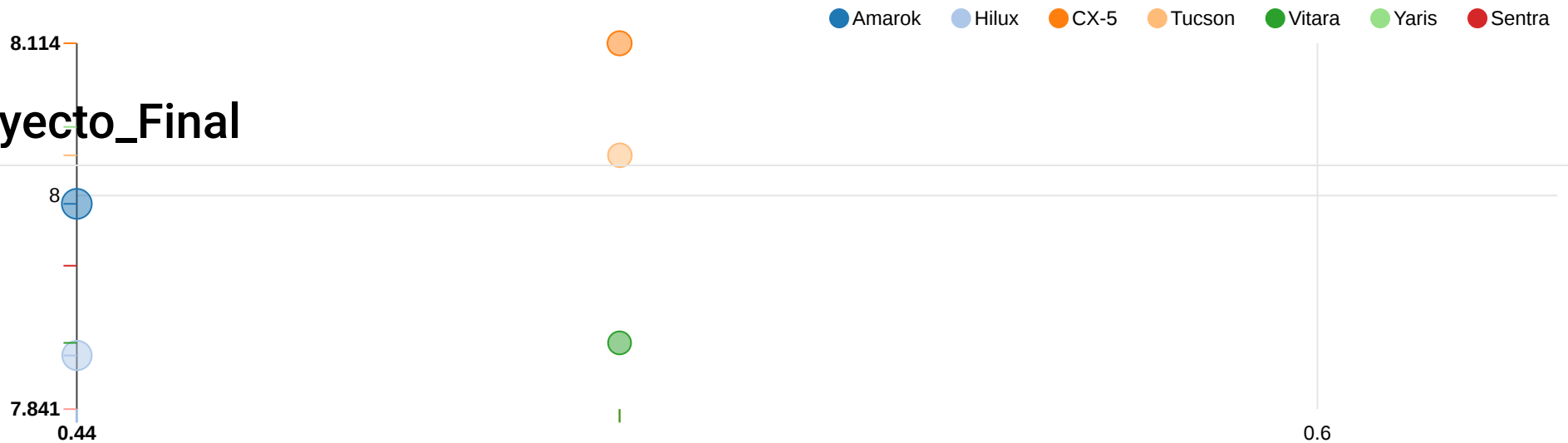
disponibilidadVsUso: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, avg_availability: decimal(9,6) ... 2 more fields]

Took 1 sec. Last updated by anonymous at July 15 2025, 4:57:29 PM. (outdated)

settings ▼

SPARK JOB FINISHED

Proyecto_Final



```
disponibilidadVsUso: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [vehicle_model: string, avg_availability: decimal(9,6) ... 2 more fields]
```

Took 1 sec. Last updated by anonymous at July 15 2025, 4:57:37 PM. (outdated)

Análisis

FINISHED

Los modelos **Amarok** y **Hilux** tienen los valores más altos de `utilization_score`, lo que indica que, a pesar de tener una baja disponibilidad (0.44), se alquilan durante muchos días en promedio. Esto puede interpretarse como una demanda sostenida o incluso saturación de su disponibilidad actual.

En contraste, modelos como el **Corolla**, **Sentra** o **Yaris** muestran una disponibilidad más alta (0.65), pero su tiempo promedio de alquiler es menor, lo que reduce su aprovechamiento relativo.

Esto indica que algunos modelos con baja disponibilidad están siendo subutilizados por limitaciones en cantidad de unidades disponibles, mientras que otros, aunque más accesibles, no logran una utilización proporcional al nivel de oferta.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:28 PM.

8. Capacidad y demanda de vehículos según número de asientos

FINISHED

Proyecto_Final



¿Los vehículos con más asientos se alquilan más o generan mayores ingresos?



Took 0 sec. Last updated by anonymous at July 21 2025, 12:58:29 AM.

```
val capacidadVsDemanda = df.groupBy("seats")
  .agg(
    count("*").as("rental_count"),
    avg("total_cost").as("avg_total_cost"),
    avg("rental_days").as("avg_rental_days")
  )
  .orderBy(desc("rental_count"))

z.show(capacidadVsDemanda)
```

SPARK JOB FINISHED



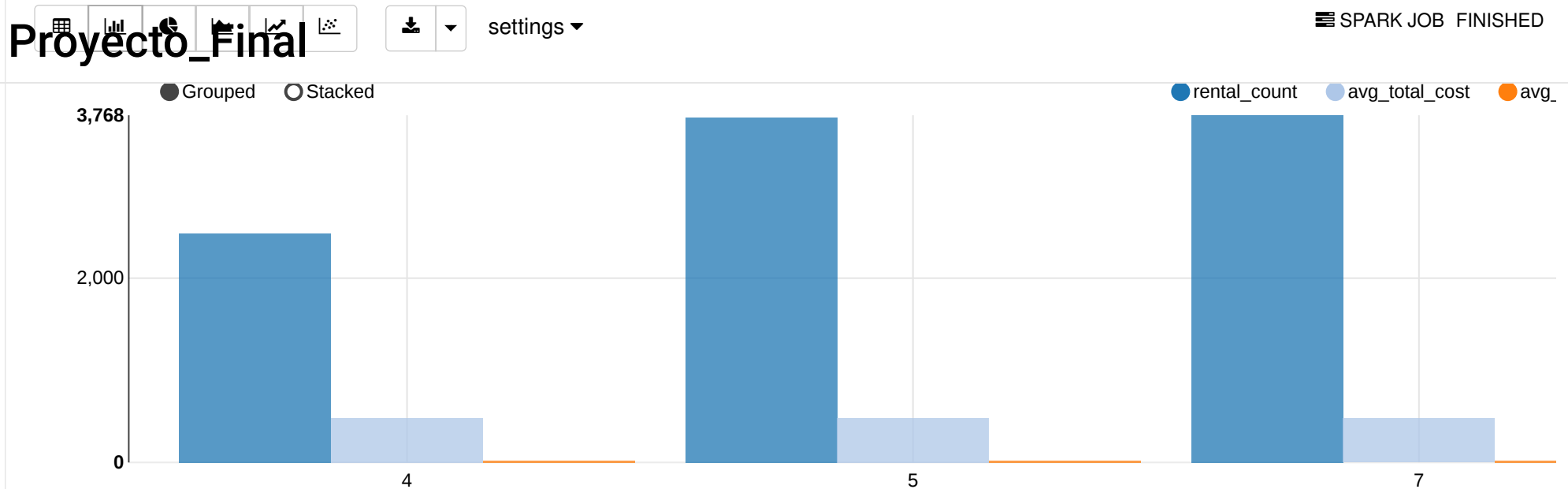


settings ▼

| seats | rental_count | avg_total_cost | avg_rental_days | |
|-------|--------------|----------------|-------------------|--|
| 7 | 3768 | 479.163145 | 7.957537154989384 | |
| 5 | 3744 | 480.469391 | 7.983173076923077 | |
| 4 | 2488 | 481.717685 | 7.965032154340836 | |

capacidadVsDemanda: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [seats: int, rental_count: bigint ... 2 more fields]

Took 1 sec. Last updated by anonymous at July 15 2025, 5:04:15 PM. (outdated)



capacidadVsDemanda: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [seats: int, rental_count: bigint ... 2 more fields]

Took 1 sec. Last updated by anonymous at July 15 2025, 5:04:24 PM. (outdated)

Análisis

FINISHED

Los vehículos con 7 asientos son los que más se alquilan, con 3768 rentas, seguidos muy de cerca por los de 5 asientos. Los de 4 asientos se alquilan considerablemente menos, con apenas 2488 rentas.

Sin embargo, el ingreso promedio por alquiler muestra una diferencia inversa: los vehículos más pequeños tienen un ingreso promedio ligeramente mayor. Los de 4 asientos lideran con 481.71 , mientras que los de 7 asientos están ligeramente por debajo.

En cuanto a duración del alquiler, no hay diferencias relevantes: todos rondan los 7.9 días.

Esto nos muestra que aunque los vehículos más grandes tienen más demanda, los más pequeños podrían estar generando un mejor rendimiento unitario por alquiler.

Took 0 sec. Last updated by anonymous at July 15 2025, 5:08:19 PM.

Proyecto_Final

9. Modelos y usos más demandados en la región Sierra

FINISHED

¿Cuales son los modelos de vehiculo con mayor demanda en la Sierra y para que se utilizan principalmente?

Took 0 sec. Last updated by anonymous at July 21 2025, 12:57:38 AM.

```
val topRentasSierra = rentas_prov
  .filter(col("region") === "Sierra")
  .join(catalogo, rentas_prov("id_modelo") === catalogo("id_modelo"))
  .groupBy(
    catalogo("modelo"),
    catalogo("tipo"),
    rentas_prov("uso_preferido")
  )
  .agg(
    sum("cantidad_rentas").as("total_rentas")
  )
  .orderBy(desc("total_rentas"))
  .limit(5)

z.show(topRentasSierra)
```

SPARK JOB FINISHED

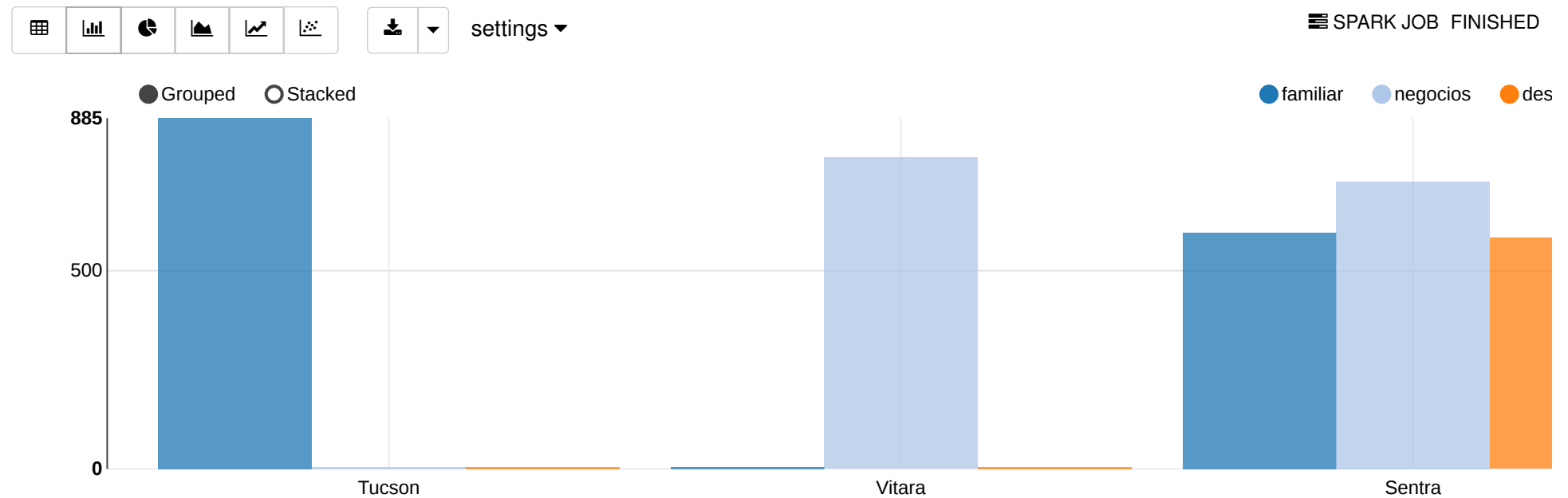
settings ▼

| modelo ▼ | tipo ▼ | uso_preferido ▼ | total_rentas | |
|----------|--------|-----------------|--------------|--|
| Tucson | SUV | familiar | 885 | |
| Vitara | SUV | negocios | 786 | |
| Sentra | sedan | negocios | 725 | |
| Sentra | sedan | familiar | 596 | |
| Sentra | sedan | desplazamiento | 583 | |

Proyecto_Final

topRentasSierra: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [modelo: string, tipo: string ... 2 more fields]

Took 1 sec. Last updated by anonymous at July 21 2025, 1:00:33 AM.



topRentasSierra: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [modelo: string, tipo: string ... 2 more fields]

Took 1 sec. Last updated by anonymous at July 21 2025, 1:00:40 AM. (outdated)

Analisis

FINISHED

En la region de la Sierra , los modelos mas alquilados se agrupan principalmente en torno a dos categorias: SUV y sedan . El Tucson lidera en rentas con uso familiar , seguido por el Vitara enfocado en negocios .

Proyecto_Final, aparece tres veces en el ranking, cubriendo los tres tipos de uso_preferido , lo cual indica que es un modelo versatil y adaptable a distintas necesidades del cliente.

Esto sugiere que mientras los SUV tienen una alta demanda por familias y empresas, los sedan como el Sentra ofrecen una solucion flexible que se ajusta tanto a desplazamientos personales como a fines corporativos.

Took 0 sec. Last updated by anonymous at July 21 2025, 12:56:49 AM.

10. Preferencias de uso de vehiculos por region

FINISHED

¿Que tipo de uso es predominante en cada region del Ecuador?

Took 0 sec. Last updated by anonymous at July 21 2025, 1:00:22 AM.

```
val preferenciasUsoRegion = rentas_prov
  .groupBy("region", "uso_preferido")
  .agg(sum("cantidad_rentas").as("total_rentas"))
  .orderBy(desc("region"), desc("total_rentas"))

z.show(preferenciasUsoRegion)
```

SPARK JOB FINISHED





settings ▼

| region | uso_preferido | total_rentas |
|--------|----------------|--------------|
| Sierra | negocios | 3912 |
| Sierra | desplazamiento | 2306 |
| Sierra | familiar | 2227 |
| Costa | desplazamiento | 2277 |

Proyecto_Final

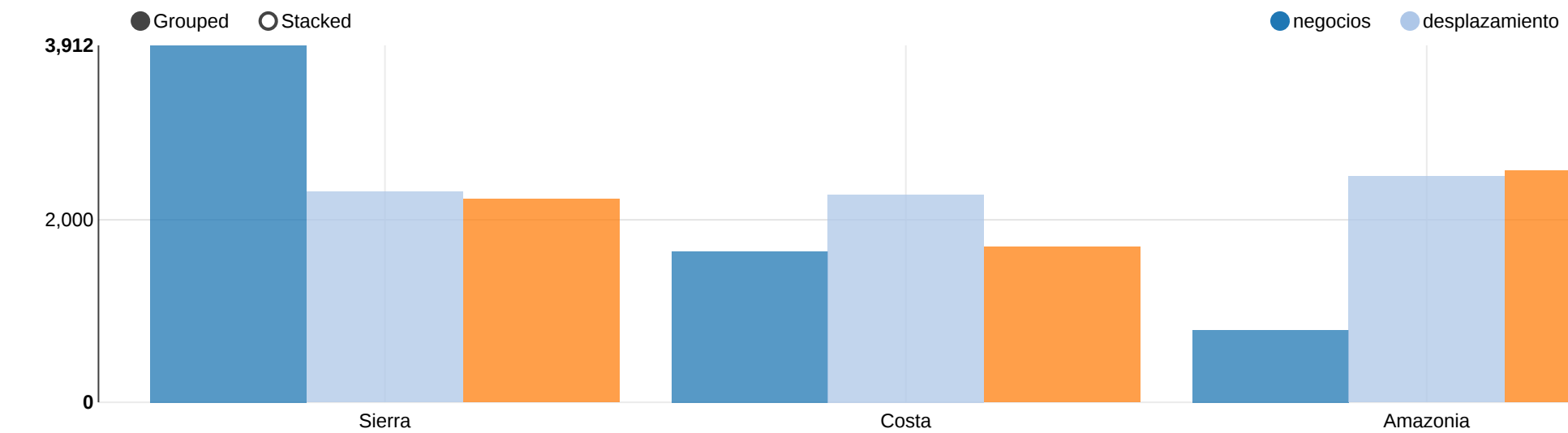
| | | |
|----------|----------------|------|
| Costa | familiar | 1703 |
| Costa | negocios | 1654 |
| Amazonia | familiar | 2536 |
| Amazonia | desplazamiento | 2478 |

preferenciasUsRegion: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [region: string, uso_preferido: string ... 1 more field]

Took 1 sec. Last updated by anonymous at July 21 2025, 1:00:55 AM. (outdated)

settings ▾

SPARK JOB FINISHED



preferenciasUsRegion: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [region: string, uso_preferido: string ... 1 more field]

Took 0 sec. Last updated by anonymous at July 21 2025, 1:00:50 AM. (outdated)

Analisis

FINISHED

Proyecto_Final

En la Sierra , el uso predominante de los vehiculos es para negocios , con mas de 3900 rentas, lo que refleja una fuerte orientacion comercial en esta region. Le siguen los usos desplazamiento y familiar con cifras similares, lo que indica una buena mezcla entre movilidad diaria y viajes personales.

En la Amazonia , las rentas se distribuyen de forma mas equilibrada entre familiar (2536) y desplazamiento (2478), con un uso empresarial mucho menor (793). Esto sugiere una menor presencia corporativa en esa zona y un mayor enfoque en la movilidad personal.

En la Costa , en cambio, no hay un uso claramente dominante. Las rentas se reparten casi equitativamente entre los tres tipos, aunque desplazamiento lidera ligeramente. Esto podria reflejar la diversidad de actividades en las provincias costeras, desde trabajo hasta turismo y vida familiar.

Se puede decir que el comportamiento por region muestra patrones claros: la Sierra con fuerte uso profesional, la Amazonia enfocada en necesidades personales, y la Costa como un punto de equilibrio entre distintos perfiles de cliente.

Took 0 sec. Last updated by anonymous at July 21 2025, 1:05:08 AM.

Conclusiones generales

FINISHED

El analisis realizado sobre el comportamiento de los alquileres de vehiculos revela patrones consistentes y utiles para la toma de decisiones operativas y estrategicas. Se identificaron configuraciones de vehiculos altamente rentables, como los de motor diesel con transmision manual , y se observo que modelos con bajo costo de mantenimiento como el Yaris y el Sentra ofrecen un excelente retorno frente a sus gastos operativos.

Tambien se evidencio que, aunque la mayoría de los clientes alquilan por periodos similares, existen ligeras diferencias segun el owner_demographic y el payment_method , que podrian ser aprovechadas para personalizar promociones. Los modelos con menor availability_index , como el Hilux y el Amarok , mostraron un uso intensivo, lo que sugiere una oportunidad para ampliar la flota en esas categorias.

Al analizar atributos menos explorados como el numero de seats , se descubrio que los vehiculos con mayor capacidad tienen mas

demanda, pero no necesariamente son los mas rentables por alquiler. Esto refleja la importancia de considerar el costo-beneficio y no solo la demanda absoluta.

Proyecto_Final

Por otro lado, al incorporar una fuente auxiliar geografica, se pudo estudiar el comportamiento de los modelos en distintas regiones del Ecuador. Se observo, por ejemplo, que en la Sierra predominan los usos negocios , mientras que en la Amazonia destacan los usos familiar y desplazamiento , y en la Costa hay una distribucion mas equilibrada. Tambien se identificaron los modelos mas demandados por region y su tipo de uso, lo cual puede ser clave para estrategias regionales de marketing o expansion.

En conjunto, este trabajo demuestra como el cruce de multiples dimensiones tecnicas, demograficas y geograficas, permite comprender mejor el rendimiento de la flota y orientar futuras decisiones de inversion o ajustes operativos en el contexto ecuatoriano.

Took 0 sec. Last updated by anonymous at July 21 2025, 1:05:56 AM.