SINGAPORE MANAGEMENT UNIVERSITY

Group Project for

DSA306 (G1): Big Data Analytics

AY 2023-24 – Term 1

Professor Daniel Preve

Assessing the Resale Car Market Dynamics in Singapore

Prepared by:

Group 5

Gloria Goh Su Yi (01436260)

Kowsalya Ganesan (01424081)

Nguyen Ngoc Quynh Tram (01495579)

Qistina Purnamasari (01443112)

Sneha Murali (01410645)

1. Introduction

1.1 Motivation:

Owning a car in Singapore is a challenging endeavour due to a combination of numerous factors that contribute to the high cost, especially due to shortage of Certificate of Entitlement (COE), which is a licence for owning a vehicle in Singapore. These COEs are allocated through auctions, controlled by the government and only valid for 10 years so the supply is always lower than demand, leading to high bidding prices and cost to car owners. Thus, we forecast a rising demand for used cars as consumers seek cost-effective solutions, which motivated our project to explore the Singapore resale car market.

1.2 Project Objective

The primary objective of our project is to assess the dynamics of Singapore's used car market. We aim to provide valuable insights to both sellers and buyers in this competitive market, specifically:

- Assist sellers and online resale platforms to maximise success rate of sales transactions.
- Empower buyers with a comprehensive understanding of the factors influencing the value of a used car, enabling them to secure better deals.

1.3 Data Source

We collected our dataset by web-scraping from sgCarMart's website, a prominent platform for buying and selling used cars in Singapore. The scraping was achieved using ***rvest*** and ***httr*** packages in R, and further manipulated by ***dplyr*** and ***stringr*** packages. Due to the large number of listings, we also utilised parallel processing through ***foreach*** and ***doParallel*** packages to further speed up the scraping process.

We conducted our own scraping for some reasons. Firstly, we want to obtain the most up-to-date information due to the dynamic nature of the used car market. Additionally, this allowed us to tailor the scraped data to align with our project's objectives. Moreover, by handling the scraping ourselves, we maintain control over data quality and ensure the dataset's reliability. Finally, as part of our commitment to collaborative data science practices, we plan to share this dataset with the GitHub community.

1.3.1 Dataset Features:

The dataset we compiled consists of 22 variables (excluding the index) and 16,393 rows. Here is a brief overview the features:

1. **HEADER**: The title or name of the car listing.
2. **PRICE**: The resale price of the car in Singapore Dollars (SGD).
3. **TYPE**: The type or category of the vehicle (e.g., SUV, Luxury Sudan, MPV, Sports Car).
4. **FEATURES**: Brief description of key features of the car (e.g., engine power, speed).
5. **ACCESSORIES**: Accessories of the car. (e.g., sport rims, leather seats, audio, sensors).
6. **DESCRIPTION**: Description of the deal. (e.g., financing options, interest rate, warranty)
7. **CATEGORIES**: Specific categories or tags associated with the car (e.g., Premium Ad Car, Low Mileage Car, Almost New Car).
8. **MILEAGE**: The number of kilometres driven by the car in its lifespan.
9. **ROAD_TAX**: The annual road tax amount in SGD, calculated based on engine capacity.
10. **DEREG_VAL**: The value in SGD received upon deregistering the vehicle for use in Singapore, calculated from the scraped date.
11. **COE**: Certificate of Entitlement value in SGD when first registered.
12. **ENGINE_CAP**: The engine's displacement in cubic centimetres (cc).
13. **CURB_WEIGHT**: The weight of the car in kilograms (kg) without passengers or items.
14. **MANUFACTURED_YR**: The year the car was manufactured.
15. **TRANSMISSION**: Type of transmission, either "Auto" or "Manual."
16. **OMV**: Open Market Value of the vehicle.
17. **ARF:** Additional Registration Fee in SGD (tax for vehicle registration).
18. **POWER**: The power of the car's engine in kilowatts (kW).
19. **NUM_PAST_OWNERS**: The number of previous owners.
20. **DEPRECIATION_VAL**: Annual straight-line depreciation in SGD.

21. **REG_DATE**: The date when the car was registered.

22. **COE_LEFT**: The number of years and months left on the Certificate of Entitlement.

2. Data Transformation and Wrangling

We recognised that PRICE will be our dependent variable and hence, began to clean possible predictors.

2.1 Removal of irrelevant variables

We first removed the index as well as the 'CATEGORIES' variable present in the scraped dataset as the column did not provide any new information that other columns were not already providing.

2.2 Variable Transformation and Extraction

We then extracted/transformed potentially useful variables from the originally scraped columns. The HEADER column contained values such as "Mercedes-Benz S-Class S320L".

1. BRAND and MODEL_SUBMODEL: Realising that HEADER contained both the brand of the car as well as its model, we extracted the values for it separately by splitting the variable with a " " delimiter, where the first word would represent the BRAND and the remaining substring would refer to the MODEL_SUBMODEL. The functions utilised include **substring_index()** and **split()**.

2. BRAND_CATEGORY: Using dplyr::distinct(), we recognised that there were >50 unique categories of BRAND and some of the categories only had 1-2 observations, which we felt was not representative of a specific brand. Hence, we formulated another variable BRAND_CATEGORY, to match the car brands into broader categories using the **dplyr::case_when()** and **dplyr::mutate()** functions. The groupings, which we formulated based on various online sources (Seow, 2019), are shown in the table below.

Category	**Brands**
Exotic	Aston Martin, Ferrari, Lamborghini, McLaren
Ultra_Luxury	Bentley, Land Rover, Maserati, Porsche, Rolls-Royce, Alpine
Luxury	Audi, BMW, Jaguar, Jeep, Lexus, Lotus, Mercedes-Benz,Volvo, Polestar, Tesla

Mid-Level	Alfa Romeo, Infiniti, MINI, Opel, Volkswagen, BYD, MG
Economy	Chevrolet, Citroen, Fiat, Ford, Honda, Hyundai, Kia, Mazda, Mitsubishi, Nissan, Peugeo, Renault, Skoda, Ssangyong, Subaru, Suzuki, Toyota, Smart, SEAT, Austi, Morris
Budget	Proton, Perodua
Others[1]	All remaining brands in the dataset not in the above mentioned categories

3. DAYS_OF_COE_LEFT: Based on the COE_LEFT column which contained character vectors such as “4yrs 9mths 15day”, we obtained DAYS_OF_COE_LEFT as a numerical variable using the **regexp_extract()** function, such that it could be easily used for the regression analysis.

4. FEATURE_COUNT and ACCESSORIES_COUNT: We have 2 text-based columns, where the features and accessories of the car were listed. We extracted the number of features/accessories by splitting the text based on delimiters such as “.” and “,”, using the **split()** function.

5. AGE_SINCE_MANUFACTURED and DAYS_SINCE_REGISTERED: We transformed the date columns such as MANUFACTURED_YEAR and REG_DATE to obtain these numeric age-based columns using the **datediff**() and **to_date()** functions.

6. NUM_PAST_OWNERS had integers from 1-6 and a string “More than 6”. We modified the string into 7 and converted it into a numerical variable.

2.3 Dealing with invalid outliers

There were cases of invalid entries for certain columns in the dataset. Hence we added in conditions using the **dplyr::ifelse()** and **dplyr:mutate()** to replace such invalid outliers with blanks, so as to deal with them using imputation in the subsequent steps. Such conditions include:

1. MANUFACTURED_YEAR > 2023
2. CURB_WEIGHT_KG < 10
3. DAYS_OF_COE_LEFT < 0 OR DAYS_OF_COE_LEFT > 3650 (since COE can only be up to 10 years in Singapore).

[1] too rare in Singapore, lesser than <5 observations for each of these brands in the dataset

2.4 Imputing missing variables with existing information

Instead of directly eliminating rows with any missing values, we wanted to retain as much data as possible. We realised ENGINE_CAPACITY, CURB_WEIGHT, POWER as well as TYPE were obtained based on MODEL_SUBMODEL of a car. In addition, only cars with the same MODEL_SUBMODEL and MANUFACTURED_YEAR had similar ENGINE_CAPACITY/CURB_WEIGHT/POWER/TYPE values as modifications could have been made to the models in different years.

Hence, we created a new dataframe with all unique combinations of MODEL_SUBMODEL and MANUFACTURED_YEAR and calculated the average/mode values of the 4 mentioned variables for each of the combination, using **dplyr::group_by()** and **dplyr::summarise()**. We then combined our original dataset with this dataframe using a left-join. For whichever rows that had missing values in the 4 mentioned variables, we replaced the blanks with the corresponding average values as per its matching combination using the **is.na()**, **ifelse()** and **mutate()** functions.

Lastly, we then omitted any remaining rows with missing values even after imputations using na.omit().

2.5 Conversion to parquet file

Instead of using a csv file for analysis, we decided to use a parquet file instead, since it is optimised to help us to work with big data. We converted our final dataframe into a parquet file using **sparklyr::spark_write_parquet()**.

3. Data Visualization & Exploratory Data Analysis

3.1 Dropping variables based on correlation matrix

In Appendix Figure 1, we plotted a correlation matrix using **corrplot::corrplot()** between the numerical variables. ROAD_TAX and POWER have a high correlation of 0.96 and 0.9 respectively with ENGINE_CAPACITY_CC, and ARF has a high correlation of 0.96 with OMV. This high correlation is likely due to the fact that ROAD_TAX and POWER are both functions of ENGINE_CAPACITY_CC (Sgcarmart, n.d; Meister, 2023). ARF is also derived from OMV (Sgcarmart, n.d). Thus, we dropped those variables.

Based on Sgcarmart (n.d) & CarSnap (2023), DEPRE_VAL & DEREG_VAL were derived from OMV, which already exists in our dataset, thus we dropped them. We dropped categorical variables DESCRIPTION as we did not extract any data from it to use in our model and BRAND as we already extracted a new BRAND_CATEGORY variable. We also dropped MODEL_SUBMODEL as other variables such as ENGINE_CAPACITY_CC and CURB_WEIGHT_KG already correspond to a car's model and retaining those variables were sufficient.

Therefore, our final dataset will now consist of 14 predictors – 11 numerical (CURB_WEIGHT_KG, OMV, COE_LISTED, ENGINE_CAPACITY_CC, DAYS_OF_COE_LEFT, MILEAGE_KM, FEATURE_COUNT, ACCESSORIES_COUNT, AGE_SINCE_MANUFACTURED, NO_OF_OWNERS and DAYS_SINCE_REGISTERED) and 3 categorical (TYPE, TRANSMISSION and BRAND_CATEGORY).

3.2 Sample Correlation Plot between finalised numerical variables

We then plotted a correlation matrix between the final numerical variables to analyse their relationship with PRICE. Referring to Appendix Figure 2, variables CURB_WEIGHT_KG, OMV, COE_LISTED, ENGINE_CAPACITY_CC, DAYS_OF_COE_LEFT have a positive relationship with PRICE, with OMV having the highest positive correlation of **0.84.** Variables MILEAGE_KM, FEATURE_COUNT, ACCESSORIES_COUNT, AGE_SINCE_MANUFACTURED, NO_OF_OWNERS and DAYS_SINCE_REGISTERED have a negative relationship with PRICE, with MILEAGE_KM having the highest negative correlation of **-0.42.**

The relationship most variables share with price align with our logical understanding - except for FEATURE_COUNT/ACCESORIES_COUNT. One may think that the more features that are mentioned, the higher the price would be but the relationship as per the correlation matrix is negative instead. This could potentially be because these text-based columns are optional to fill and hence may not be very representative of the true nature of the car. Hence, we are unsure as to the usefulness of these 2 predictors and would like to try different models to assess their performance.

3.3 Raster Plots

In Appendix Figure 3, we used **sparklyr::dbplot_raster()** to create raster plots to further visualise and understand the relationship between PRICE and final numerical variables. This also helps to confirm the relationship between predictors and price align with the relationship seen in the correlation matrix.

3.4 Bar Plots

Using ggplot2 package, we used **geom_bar()** to create bar plots of our character variables against price. We intuitively understand that for BRAND_CATEGORY, exotic cars would generally be more expensive and for TYPE, sports cars are the most expensive which align with the relationship seen in our bar plots in Appendix Figure 4. For variable transmission, we saw that manual cars have the highest average price which could be attributed to the high number of sports cars with manual transmission in our dataset.

4. Modelling

We decided to test 2 different pipelines (one with the text-based variables, the other without) to find the best model, based on the RMSE of each of the models. The model including the text-based variables gave a lower RMSE. Hence, we decided to use that as our final model.

4.1 Building a 7-stage pipeline

We first initialise an empty pipeline using **ml_pipeline()** function, and added stages to it. In the *1st stage*: We used the function **ft_vector_assembler()** to combine these numeric variables: MILEAGE_KM + COE_LISTED + ENGINE_CAPACITY_CC + CURB_WEIGHT_KG + TRANSMISSION + OMV + NUM_PAST_OWNERS + DAYS_OF_COE_LEFT + AGE_SINCE_MANUFACTURED + DAYS_SINCE_REGISTERED + FEATURE_COUNT + ACCESSORIES_COUNT into a single-row vector. In the *2nd stage*, we used the function **ft_standard_scaler()** for standardisation by removing the mean and scaling to unit variance in order to optimise the model. In the *3rd-5th stage*, we used the **ft_string_indexer()** function to convert the strings value to index as currently 3 of our variables in our dataset (TYPE, TRANSMISSION, BRAND_CATEGORY) are categorical. In the *6th stage*, we used the

ft_vector_assembler() function again. But this time, to combine data from all the variables (including the converted index values) into a single-row vector. For the *last stage*, we applied linear regression using the **ml_linear_regression()** function.

4.1.2 Obtaining a transformer

We split our final dataset into a training and testing set, such that the probability that a particular observation is assigned to the training set (df_train) is 0.8 and 0.4 for the testing set (df_test). To make the split reproducible, we included the seed argument.

As the pipeline we have earlier is an estimator object, we used the **ml_fit()** function to convert it to an estimator object and obtained a pipeline model with df_train. The resulting model that we obtained is :

PRICE = 13975.081**(MILEAGE_KM)** + 2263.033**(CURB_WEIGHT_KG)** + 185926.974**(OMV)**

+ 728.829**(COE_LISTED)** - 16574.058**(ENGINE_CAPACITY_CC)**

-99.682**(AGE_SINCE_MANUFACTURED)** - 52920.730**(DAYS_SINCE_REGISTERED)**

-9742.942**(NUM_PAST_OWNERS)** + 26257.671**(DAYS_OF_COE_LEFT)**

-2353.743**(FEATURE_COUNT)** - 1024.575**(ACCESSORIES_COUNT)** + 2940.832**(TYPE_index)**

+ 92549.120**(TRANSMISSION_index)** - 5267.821**(BRAND_CATEGORY_index)**

4.1.3 Cross-Validation

Using Sparklyr's **ml_cross_validator()** function, we used our first pipeline to try out the different model specifications by defining a cross-validator. We specified that we want to tune the 7th (last) pipeline stage, by setting elastic_net_param argument to a vector of values 0 and 1 to evaluate the Ridge and Lasso models with lambda values 0.1, 0.005 and 0.001. We also set the number of folds to 10 to carry out a 10-fold cross validation.

Additionally, we utilised the **ml_fit()** function to apply this estimator with df_train. After examining the cross-validation outcomes, we used ml_predict(cv_model$best_model, df_test) to select the best model along with the most suitable alpha and lambda values. We also found the Root Mean Squared-Error (RMSE) of our model to be **78359.33**, with an R^2 of **0.789**.

4.2 Limitations

As we had many variables, it would be good if we could decide the best model based on both RMSE and Adjusted R^2, as Adjusted R^2 takes into account the number of predictors in the model and penalises the inclusion of irrelevant variables. However, because the function ml_linear_regression() does not support adjusted R^2, we could only make our choice based on the RMSE.

5 API with Plumber package

To allow other users to employ our model, we have built a web API using the plumber package. End users can input information into the API to get the predicted price of a car they are interested in buying. For demonstration purposes, we randomly select a resale car and use its variables in the API to predict the price. The predicted price is $121,754.24, while the true resale price is $110,800 in the random sample.

6. Conclusion

We have detailed how our model can present potential car owners with tools and visualisations to make well-informed choices should they want to purchase a resale car. Users will better understand how resale car prices vary with respect to the different predictors. To ensure our model is publicly available for those who need it, we have published the repository at this link.

References

CarSnap. (2023, October 19). *How to Calculate PARF Rebate and COE Rebate For Your Car Easily › Carsnap Blog*. Carsnap Singapore. https://carsnap.sg/blog/calculate-parf-rebate-and-coe-rebate/#:~:text=To%20calculate%20your%20total%20deregistration,COE%20rebate%20and%20PARF%20rebate

Meister, P. (2023, September 14). What is The Relation of CC and Horsepower in Automobile Engine? *CAR FROM JAPAN*. https://carfromjapan.com/article/industry-knowledge/cc-and-horsepower-in-automobile-engine/

Seow, X. J. (2019, November 24). Project 2: Worth it? predicting the price of used cars in Singapore. Medium. https://towardsdatascience.com/project-2-worth-it-predicting-the-price-of-used-cars-in-singapore-e9afe63c75d0

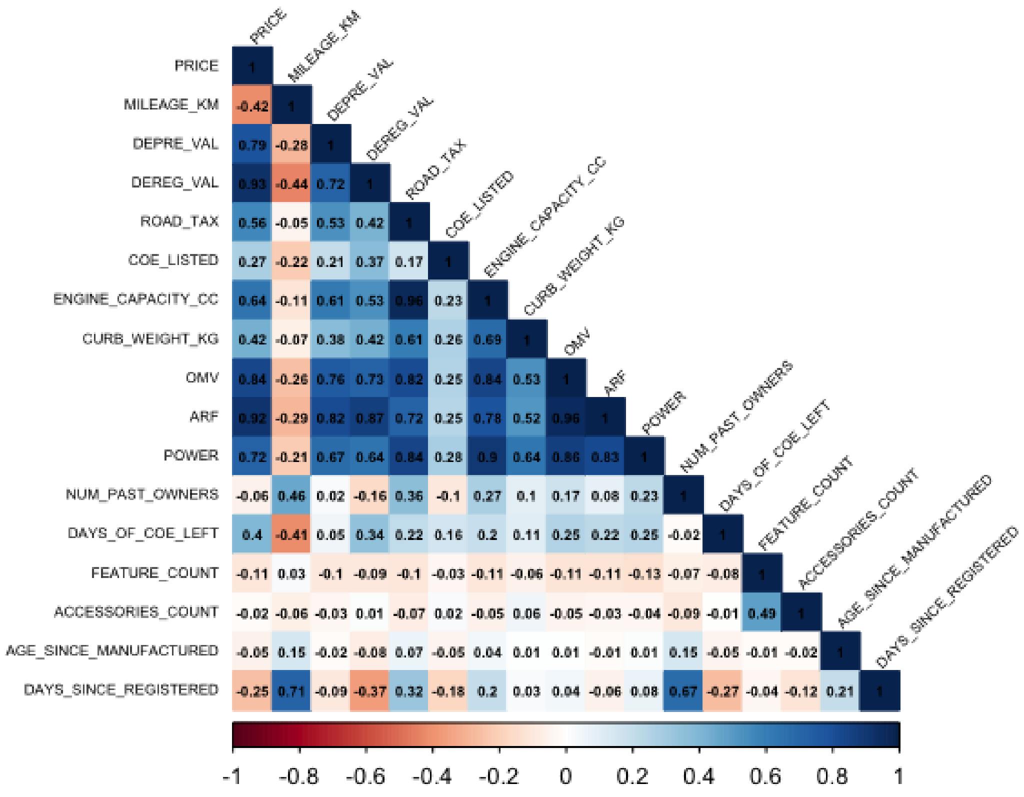
Sgcarmart. (n.d.). *What is ARF? - Sgcarmart*. Sgcarmart.com. https://www.sgcarmart.com/new_cars/popups/whatsARF.php

Sgcarmart. (n.d.). *Road tax Calculator Singapore - SGCarmart*. Sgcarmart.com. https://www.sgcarmart.com/services/roadtax_calculator.php

Sgcarmart. (n.d.). *Used Car Depreciation Calculator & Finder – SGCarmart*. Sgcarmart.com. https://www.sgcarmart.com/tools_tips/depreciation.php

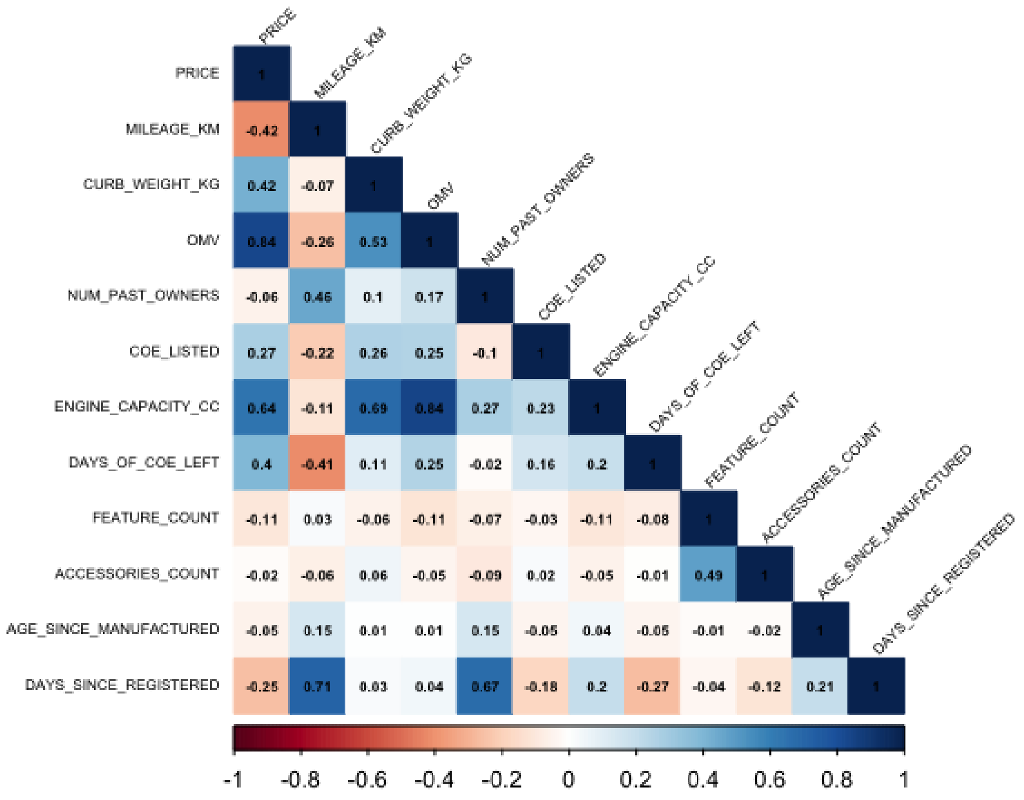
Appendix

Figure 1: Correlation Plot between all numerical variables



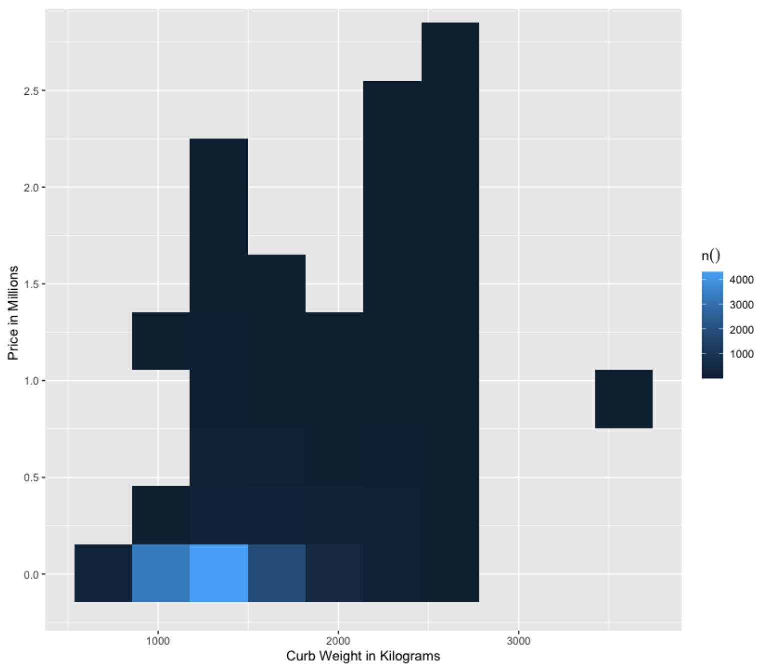
	PRICE	MILEAGE_KM	DEPRE_VAL	DEREG_VAL	ROAD_TAX	COE_LISTED	ENGINE_CAPACITY_CC	CURB_WEIGHT_KG	OMV	ARF	POWER	NUM_PAST_OWNERS	DAYS_OF_COE_LEFT	FEATURE_COUNT	ACCESSORIES_COUNT	AGE_SINCE_MANUFACTURED	DAYS_SINCE_REGISTERED
PRICE	1																
MILEAGE_KM	-0.42	1															
DEPRE_VAL	0.79	-0.28	1														
DEREG_VAL	0.93	-0.44	0.72	1													
ROAD_TAX	0.56	-0.05	0.53	0.42	1												
COE_LISTED	0.27	-0.22	0.21	0.37	0.17	1											
ENGINE_CAPACITY_CC	0.64	-0.11	0.61	0.53	0.96	0.23	1										
CURB_WEIGHT_KG	0.42	-0.07	0.38	0.42	0.61	0.26	0.69	1									
OMV	0.84	-0.26	0.76	0.73	0.82	0.25	0.84	0.53	1								
ARF	0.92	-0.29	0.82	0.87	0.72	0.25	0.78	0.52	0.96	1							
POWER	0.72	-0.21	0.67	0.64	0.84	0.28	0.9	0.64	0.86	0.83	1						
NUM_PAST_OWNERS	-0.06	0.46	0.02	-0.16	0.36	-0.1	0.27	0.1	0.17	0.08	0.23	1					
DAYS_OF_COE_LEFT	0.4	-0.41	0.05	0.34	0.22	0.16	0.2	0.11	0.25	0.22	0.25	-0.02	1				
FEATURE_COUNT	-0.11	0.03	-0.1	-0.09	-0.1	-0.03	-0.11	-0.06	-0.11	-0.11	-0.13	-0.07	-0.08	1			
ACCESSORIES_COUNT	-0.02	-0.06	-0.03	0.01	-0.07	0.02	-0.05	0.06	-0.05	-0.03	-0.04	-0.09	-0.01	0.49	1		
AGE_SINCE_MANUFACTURED	-0.05	0.15	-0.02	-0.08	0.07	-0.05	0.04	0.01	0.01	-0.01	0.01	0.15	-0.05	-0.01	-0.02	1	
DAYS_SINCE_REGISTERED	-0.25	0.71	-0.09	-0.37	0.32	-0.18	0.2	0.03	0.04	-0.06	0.08	0.67	-0.27	-0.04	-0.12	0.21	1

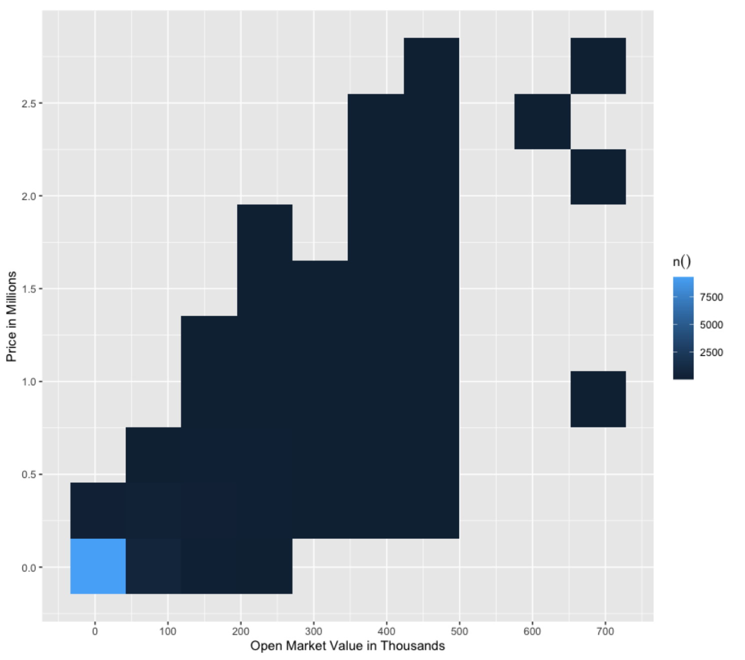
Figure 2: Correlation plot between final numerical variables

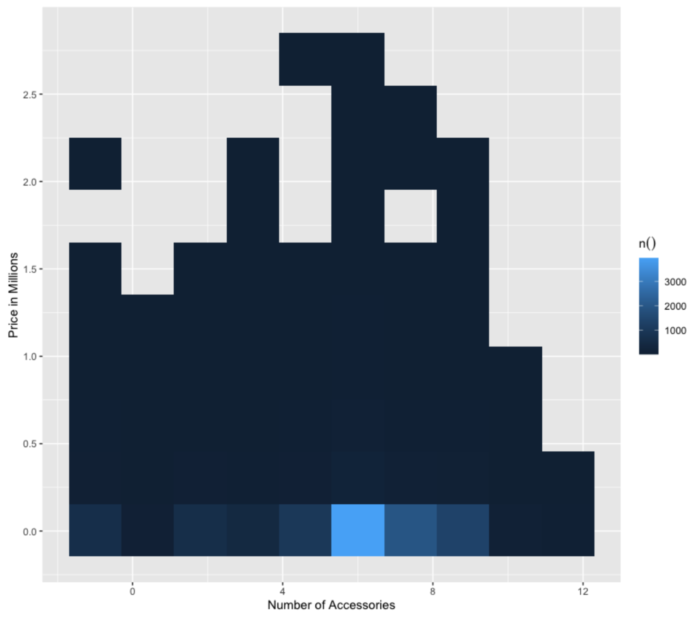


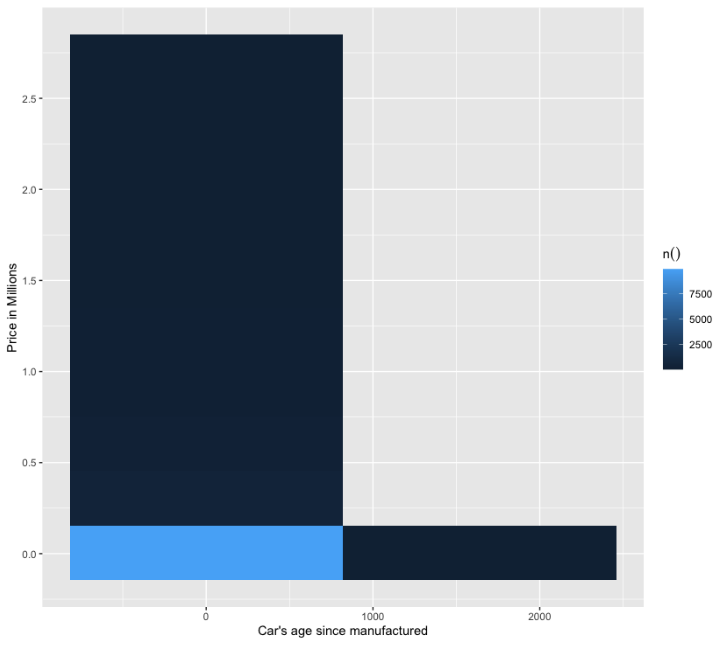
	PRICE	MILEAGE_KM	CURB_WEIGHT_KG	OMV	NUM_PAST_OWNERS	COE_LISTED	ENGINE_CAPACITY_CC	DAYS_OF_COE_LEFT	FEATURE_COUNT	ACCESSORIES_COUNT	AGE_SINCE_MANUFACTURED	DAYS_SINCE_REGISTERED
PRICE	1											
MILEAGE_KM	-0.42	1										
CURB_WEIGHT_KG	0.42	-0.07	1									
OMV	0.84	-0.26	0.53	1								
NUM_PAST_OWNERS	-0.06	0.46	0.1	0.17	1							
COE_LISTED	0.27	-0.22	0.26	0.25	-0.1	1						
ENGINE_CAPACITY_CC	0.64	-0.11	0.69	0.84	0.27	0.23	1					
DAYS_OF_COE_LEFT	0.4	-0.41	0.11	0.25	-0.02	0.16	0.2	1				
FEATURE_COUNT	-0.11	0.03	-0.06	-0.11	-0.07	-0.03	-0.11	-0.08	1			
ACCESSORIES_COUNT	-0.02	-0.06	0.06	-0.05	-0.09	0.02	-0.05	-0.01	0.49	1		
AGE_SINCE_MANUFACTURED	-0.05	0.15	0.01	0.01	0.15	-0.05	0.04	-0.05	-0.01	-0.02	1	
DAYS_SINCE_REGISTERED	-0.25	0.71	0.03	0.04	0.67	-0.18	0.2	-0.27	-0.04	-0.12	0.21	1

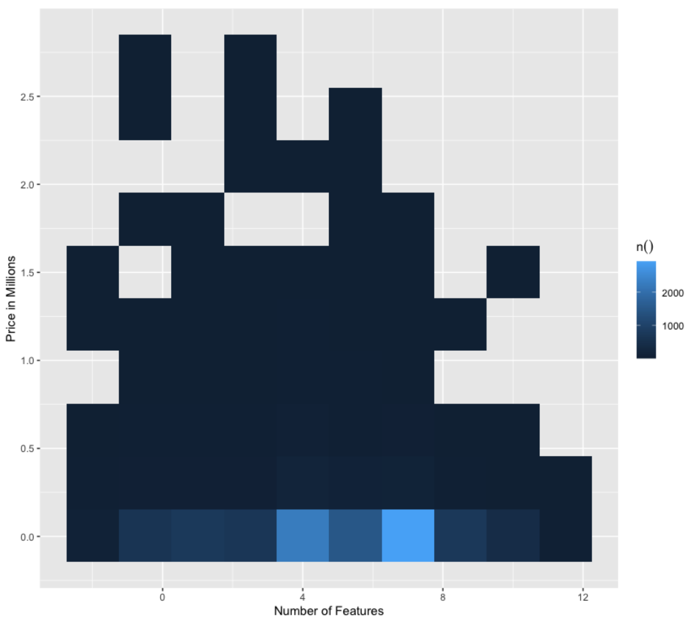
Figure 3: Raster plots

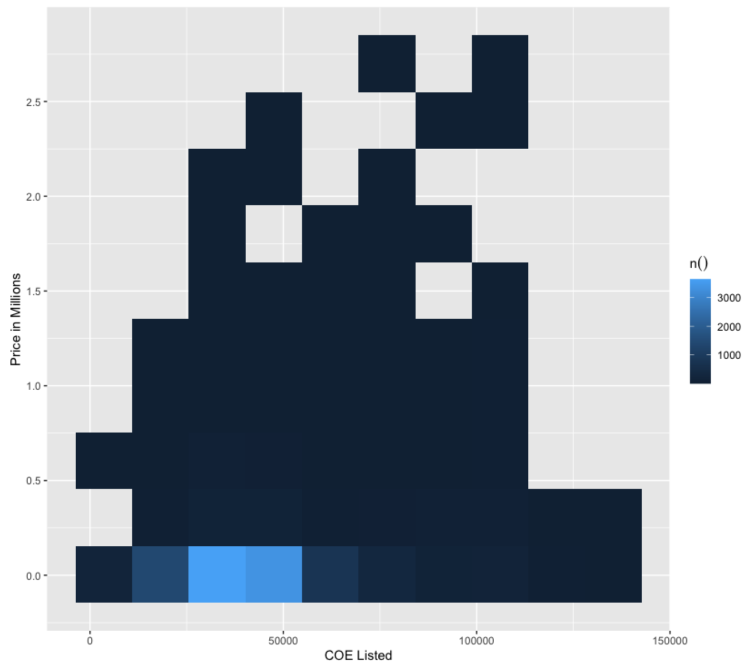


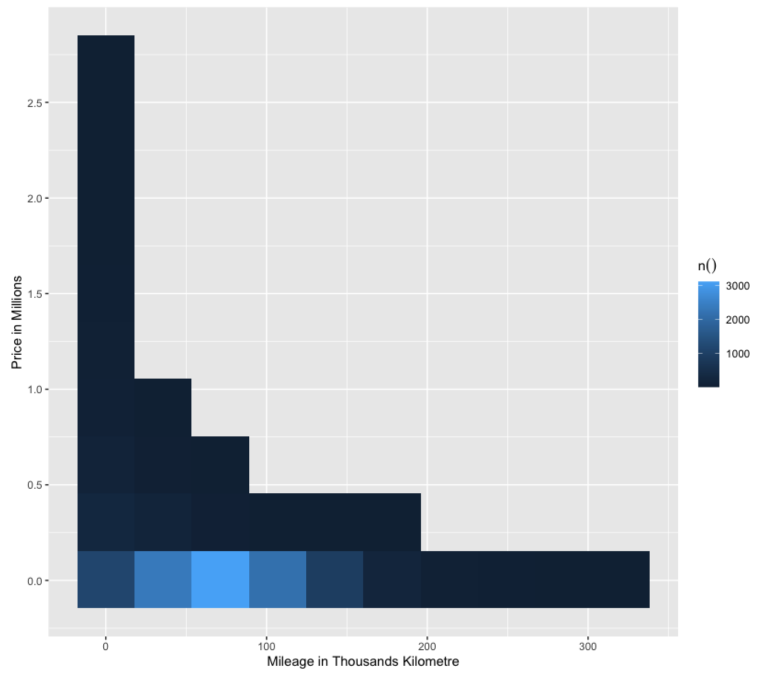


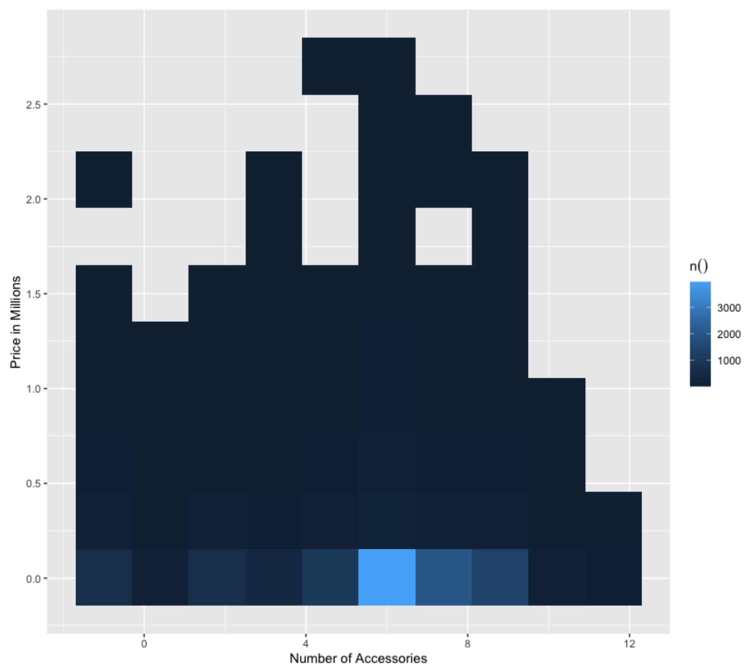


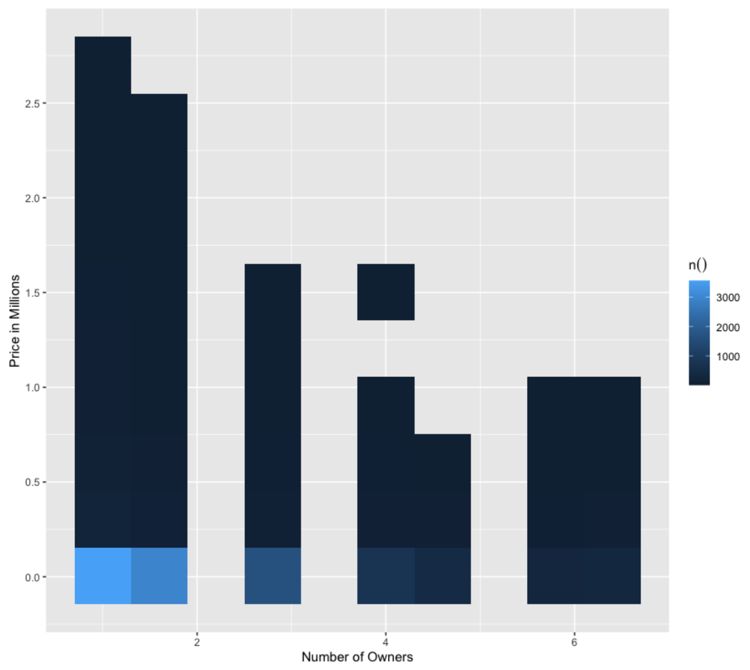





Price in Millions
COE Listed
n()
3000
2000
1000
0.0
0.5
1.0
1.5
2.0
2.5
0
50000
100000
150000


Price in Millions
Mileage in Thousands Kilometre
n()
3000
2000
1000
0.0
0.5
1.0
1.5
2.0
2.5
0
100
200
300


Price in Millions
Number of Accessories
n()
3000
2000
1000
0.0
0.5
1.0
1.5
2.0
2.5
0
4
8
12


Price in Millions
Number of Owners
n()
3000
2000
1000
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6

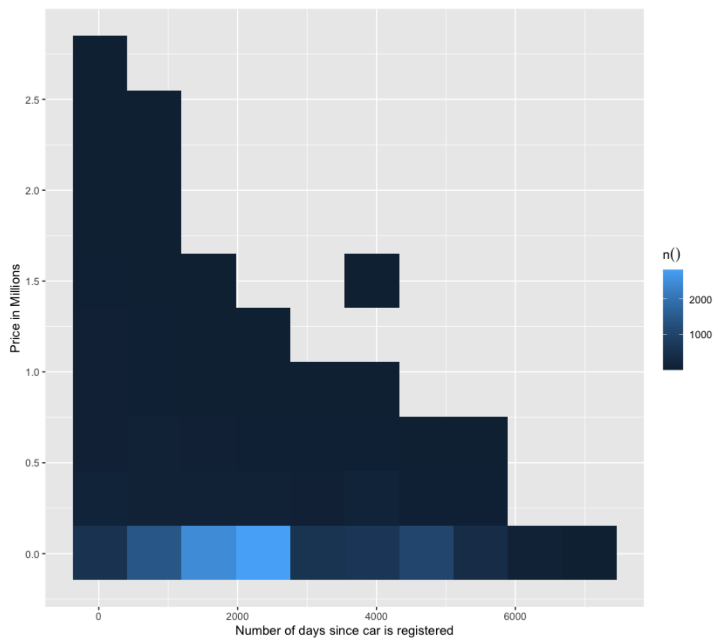
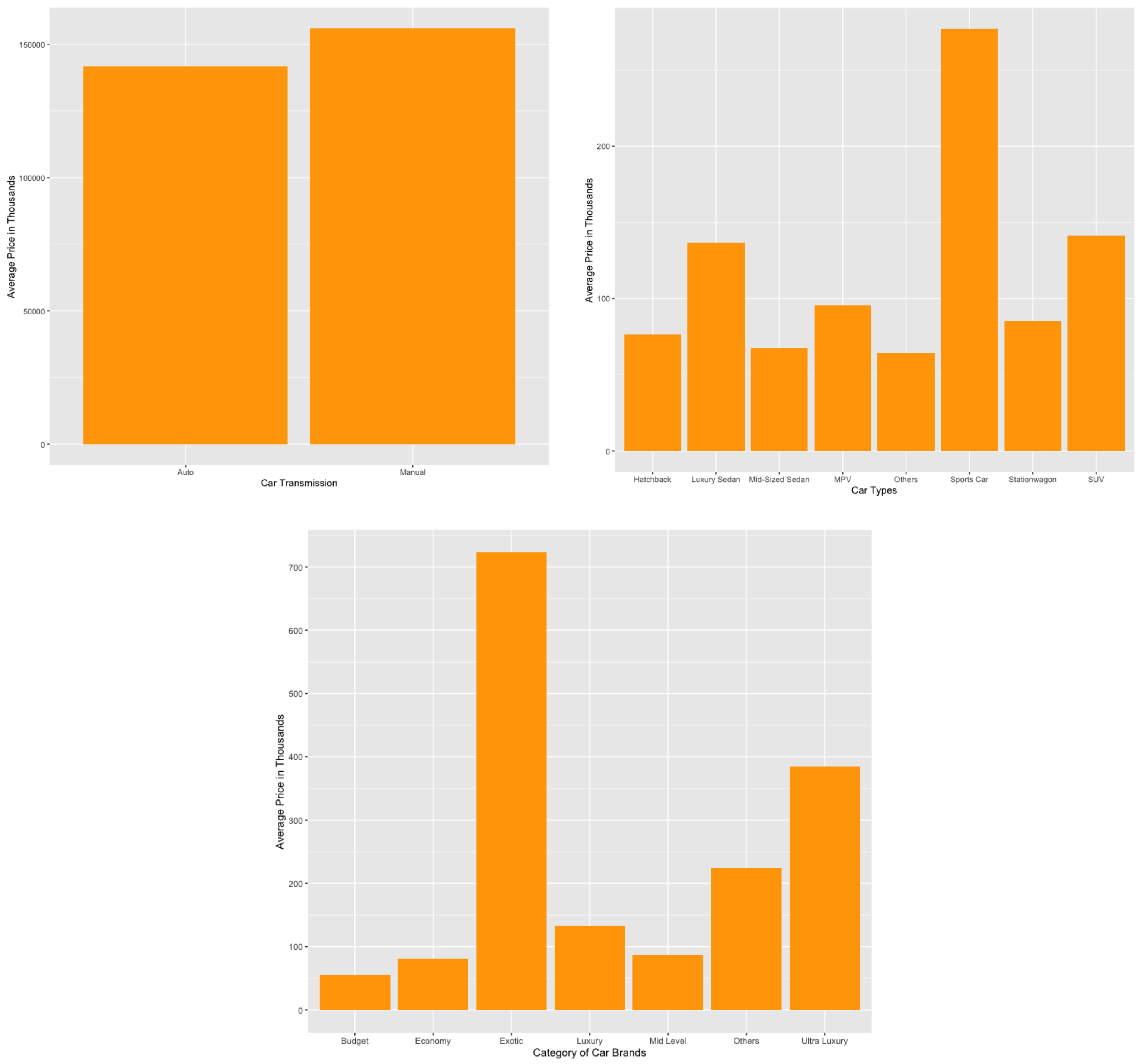

Price in Millions
Number of days since car is registered
n()
2000
1000
0.0
0.5
1.0
1.5
2.0
2.5
0
2000
4000
6000

Figure 4:


Average Price in Thousands
150000
100000
50000
0
Auto
Manual
Car Transmission
Average Price in Thousands
200
100
0
Hatchback
Luxury Sedan
Mid-Sized Sedan
MPV
Others
Sports Car
Stationwagon
SUV
Car Types
Average Price in Thousands
700
600
500
400
300
200
100
0
Budget
Economy
Exotic
Luxury
Mid Level
Others
Ultra Luxury
Category of Car Brands