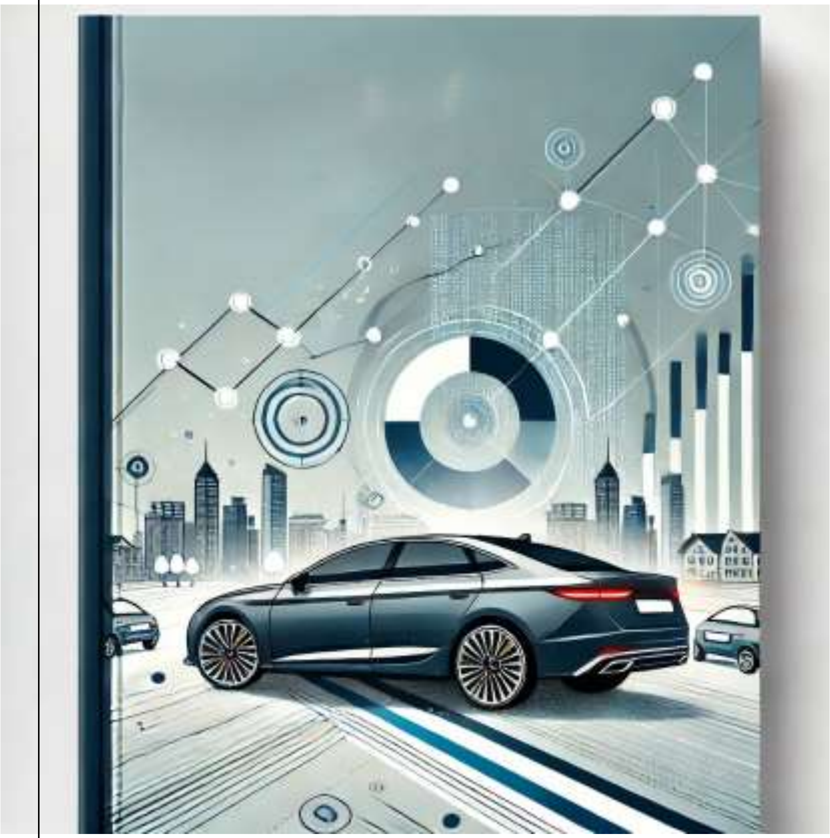


ANALYSIS OF CAR PRICE TRENDS

A Census-Based Study from an Indian City

IS 3001 Sampling Techniques

Group 04

s16290	Shavindi Handagama
s16304	Faith Fernando
s16309	Sethara Gunawardana
s16341	Tharindu Darshana

Table of Contents

1. Introduction

Data Set Name :- Car Price Prediction Dataset.

Source of the data set :- Car Price Prediction Dataset (kaggle.com)

No. of Observations :- 8128

This report is based on a census dataset collected from a mid-sized Indian city, focusing on analyzing vehicle ownership and pricing trends. The dataset offers insight into various factors influencing car prices in the city, such as vehicle age, fuel type, brand, and engine capacity. These factors provide a comprehensive overview of the local automotive market, helping in understanding how social and economic variables shape car purchasing decisions in the region. The dataset is a real-world census collection, and it is a valuable tool for urban planners and policy makers to evaluate market shifts based on current data, identify trends, and inform car regulations.

The objective of this report is to apply sampling techniques to the dataset and estimate various parameters, including mean, totals, and proportions, to gain insights into the car market in the region.

2. Methodology

Sampling Design

To obtain a representative view of car prices across the city, several sampling techniques were applied:

1. **Simple Random Sampling (SRS):** A random subset of car data was selected without any stratification. This method provides an unbiased look at the dataset, ensuring all vehicles have an equal chance of being included.

2. **Stratified Sampling:** The dataset was divided into strata based on the transmission type (Manual , Automatic). From each stratum, a proportionate sample was drawn to ensure that the sample accurately reflects the population composition across different transmission types.

3. **Two-Stage Cluster Sampling:** In the first stage, cars were grouped based on their model. In the second stage, a random selection of cars from each model cluster was analyzed. This method was chosen to reflect model price variations.

Variable Selection

Key variables such as transmission type, maximum power, seller type, owner type and fuel type were selected for analysis. These variables were used to estimate the mean, proportion, and total car price across the population.

Estimation

Under each design, the mean, proportion, and total car prices were calculated along with their standard errors. These estimates were compared against actual values from the complete dataset to assess the accuracy of the sampling methods.

Finally, a regression analysis was performed to determine the relationship between car prices and independent factors like maximum power. The estimates obtained from this regression analysis were compared across different sampling methods for validation.

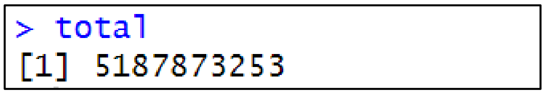
- For all calculations and perform every sample and sample estimation we use R statistical software.

3. Simple Random Sampling

Methodology of the simple random sampling technique will be discussed here. A simple random sample is a subset of a statistical population in which each member of the subset has an equal probability of being chosen, which makes SRS the simplest form of sampling techniques. This assumes homogeneity of the population and then the sampling frame is obtained. We first need to calculate the actual population parameters (means, totals, proportions) from the whole dataset in order to compare with the estimates. R-software makes it easier.

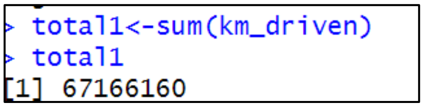
Population

Total : Selling price



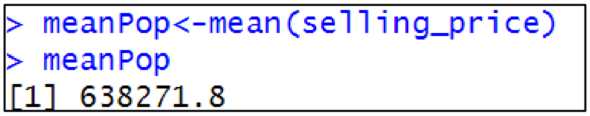
```
> total
[1] 5187873253
```

Km_Driven

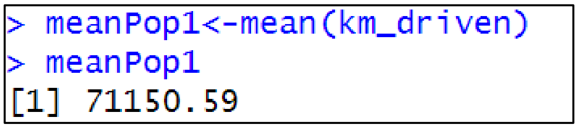


```
> total1<-sum(km_driven)
> total1
[1] 67166160
```

Mean :

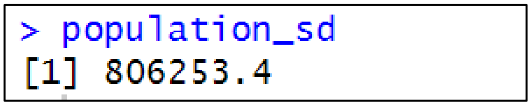


```
> meanPop<-mean(selling_price)
> meanPop
[1] 638271.8
```

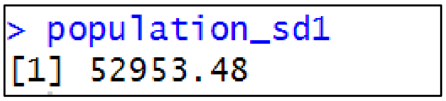


```
> meanPop1<-mean(km_driven)
> meanPop1
[1] 71150.59
```

Standard Deviation :

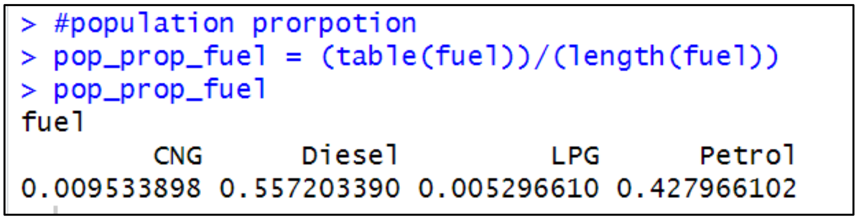


```
> population_sd
[1] 806253.4
```

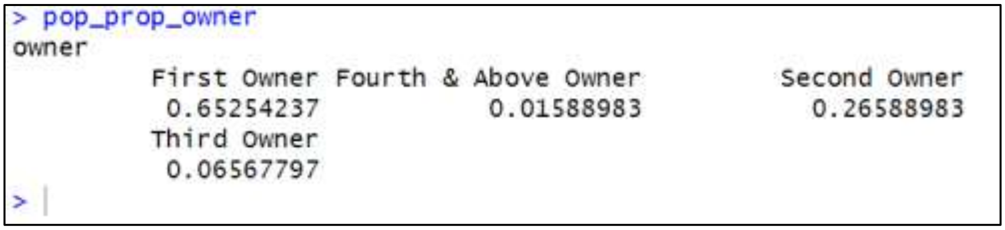


```
> population_sd1
[1] 52953.48
```

Propotion :

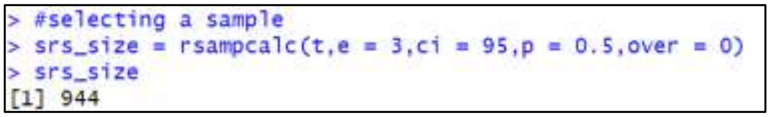


```
> #population prorpotion
> pop_prop_fuel = (table(fuel))/(length(fuel))
> pop_prop_fuel
fuel
        CNG      Diesel         LPG      Petrol 
0.009533898 0.557203390 0.005296610 0.427966102 
```



```
> pop_prop_owner
owner
         First Owner Fourth & Above Owner         Second Owner
          0.65254237           0.01588983           0.26588983
         Third Owner
          0.06567797
> |
```

Here, since R software is used for all the calculations, we straightaway can exercise the function “rsampcalc” in package “sampler” to find the sample size.

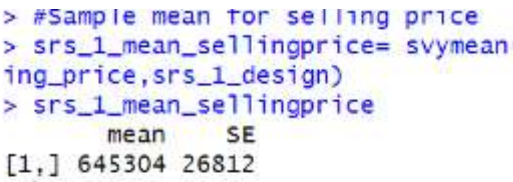


```
> #selecting a sample
> srs_size = rsampcalc(t,e = 3,ci = 95,p = 0.5,over = 0)
> srs_size
[1] 944
```

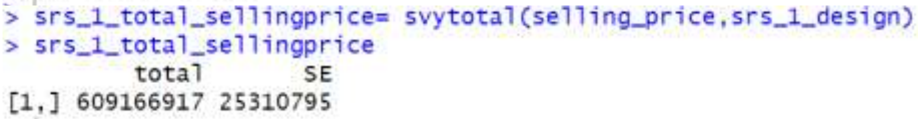
We keep a margin of error of 0.03 and a 5% type 1 error (α). Thus, it allow us to keep a sample size of 944 individuals.

The parameter estimation is done like in below. For Selling price

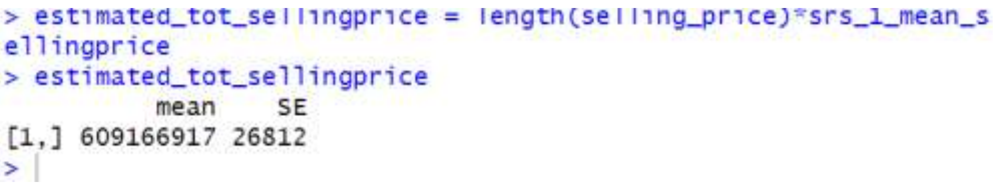
For Sample 1



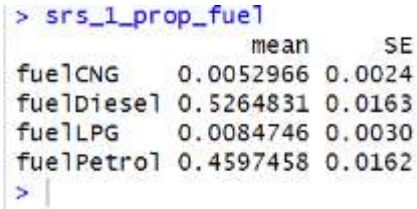
```
> #Sample mean for selling price
> srs_1_mean_sellingprice= svymean
ing_price,srs_1_design)
> srs_1_mean_sellingprice
       mean    SE
[1,] 645304 26812
```



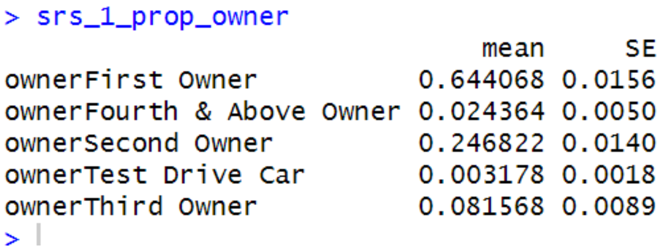
```
> srs_1_total_sellingprice= svytotal(selling_price,srs_1_design)
> srs_1_total_sellingprice
         total       SE
[1,] 609166917 25310795
```



```
> estimated_tot_sellingprice = length(selling_price)*srs_1_mean_s
ellingprice
> estimated_tot_sellingprice
          mean    SE
[1,] 609166917 26812
> |
```

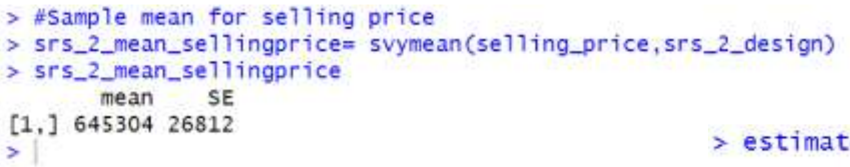


```
> srs_1_prop_fuel
                mean     SE
fuelCNG    0.0052966 0.0024
fuelDiesel 0.5264831 0.0163
fuelLPG    0.0084746 0.0030
fuelPetrol 0.4597458 0.0162
> |
```

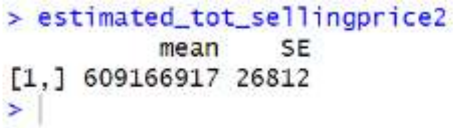


```
> srs_1_prop_owner
                               mean     SE
ownerFirst Owner           0.644068 0.0156
ownerFourth & Above Owner  0.024364 0.0050
ownerSecond Owner          0.246822 0.0140
ownerTest Drive Car        0.003178 0.0018
ownerThird Owner           0.081568 0.0089
> |
```

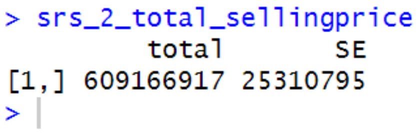
For sample 2



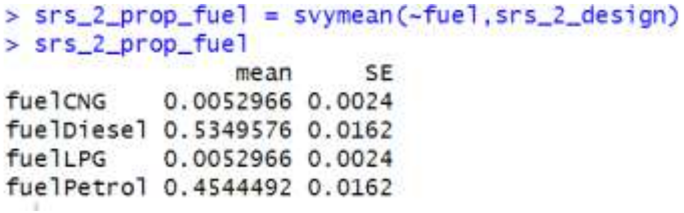
```
> #Sample mean for selling price
> srs_2_mean_sellingprice= svymean(selling_price,srs_2_design)
> srs_2_mean_sellingprice
        mean    SE
[1,] 645304 26812
> 
```



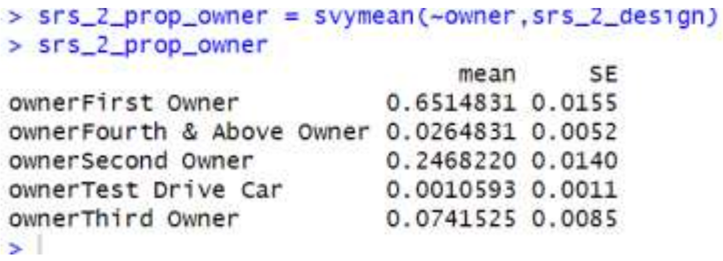
```
> estimated_tot_sellingprice2
          mean    SE
[1,] 609166917 26812
> 
```



```
> srs_2_total_sellingprice
         total       SE
[1,] 609166917 25310795
> 
```



```
> srs_2_prop_fuel = svymean(~fuel,srs_2_design)
> srs_2_prop_fuel
                mean     SE
fuelCNG    0.0052966 0.0024
fuelDiesel 0.5349576 0.0162
fuelLPG    0.0052966 0.0024
fuelPetrol 0.4544492 0.0162
```



```
> srs_2_prop_owner = svymean(~owner,srs_2_design)
> srs_2_prop_owner
                                mean     SE
ownerFirst Owner           0.6514831 0.0155
ownerFourth & Above Owner  0.0264831 0.0052
ownerSecond Owner          0.2468220 0.0140
ownerTest Drive Car        0.0010593 0.0011
ownerThird Owner           0.0741525 0.0085
> 
```

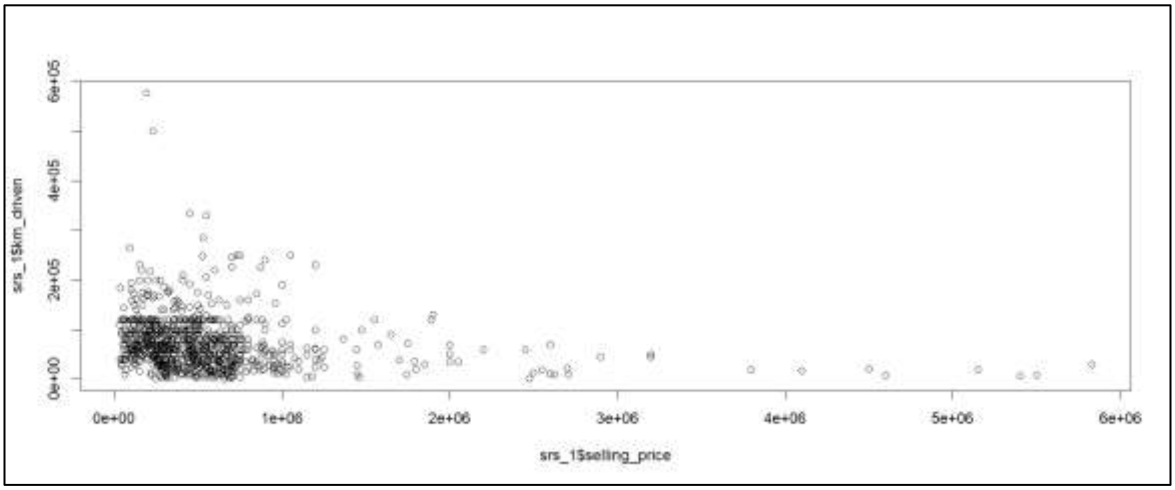
	Population	Sample 1	Sample 2
Total	5187873253	609166917 [SE=25310795]	589828748 [SE= 24528203]
Mean	638171.8	645304 [SE=26812]	624819 [SE =25983]
Standard deviation	806253.4		

- Estimated mean value of Selling price for simple random sample 1 is 645304. When compared with actual mean value, this value is higher . Also, the estimated mean value of Selling price for simple random sample 2 is 624819 which it is little bit lower than the actual value.
- The mean values of Sample 1 for selling price is higher than the mean value of selling price in sample 2
- The estimated the total values of Sample1 is grate1 than actual total in population . but in sample 2 estimated total is higher than population value.
- There are slight deviations from the estimated proportions of both samples for fuel variables with the actual values obtained when considering population. These are, CNG fuel estimations seem to be lesser than actual values while in petrol fuel are a little higher.
- Estimated proportions of both samples for owner types are very close with the actual values obtained when considering population.

Ratio and Regression estimation

Regression Analysis :

For sample 1

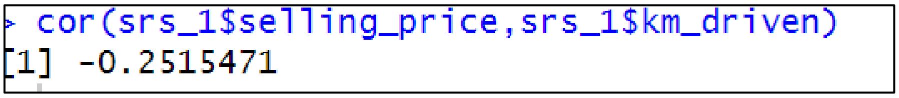


Key Observations:

- **Negative Correlation:** There seems to be a general negative correlation between selling price and kilometers driven. This suggests that as the kilometers driven increase, the selling price tends to decrease.
- **Clustering:** The data points cluster in the lower left corner of the plot, indicating that a majority of the cars have lower selling prices and have been driven fewer kilometers.
- **Outliers:** There are a few outliers, which are the points that are far away from the main cluster. These could represent cars with unique characteristics, conditions, or market factors that influence their selling prices.

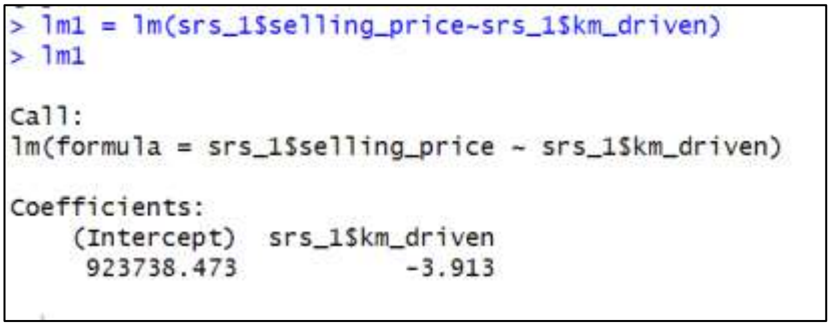
Interpretation:

The plot suggests that the selling price of cars tends to decrease as the number of kilometers driven increases. This is likely due to the general depreciation of vehicles over time. However, there are individual variations, as indicated by the outliers, which might be influenced by factors such as brand, model, condition, and market demand



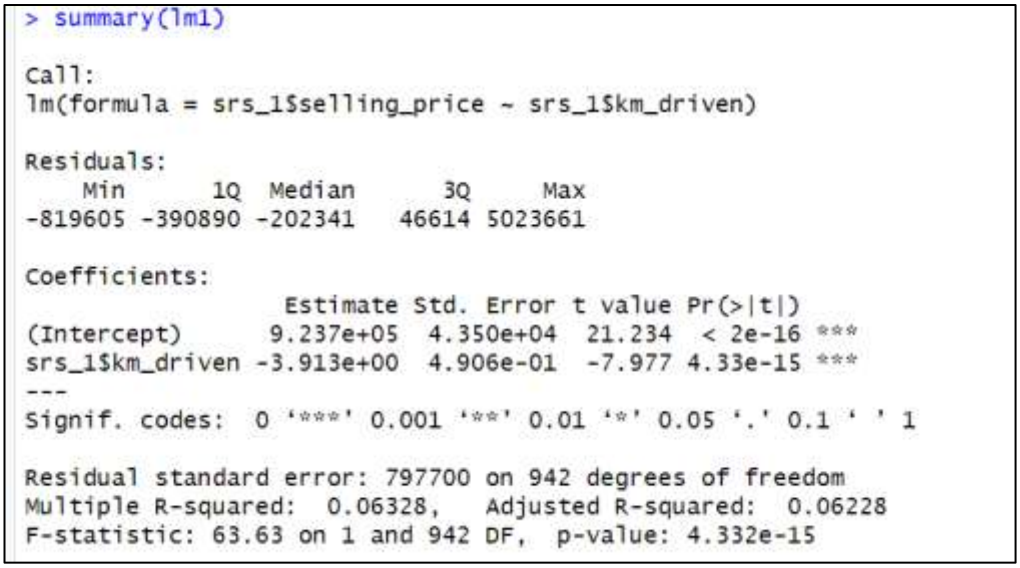
```
> cor(srs_1$selling_price,srs_1$km_driven)
[1] -0.2515471
```

The correlation coefficient is -0.2515471. This indicates a **weak negative correlation** between the selling price and kilometers driven. It means that as the kilometers driven increase, the selling price tends to decrease, but the relationship is not very strong.

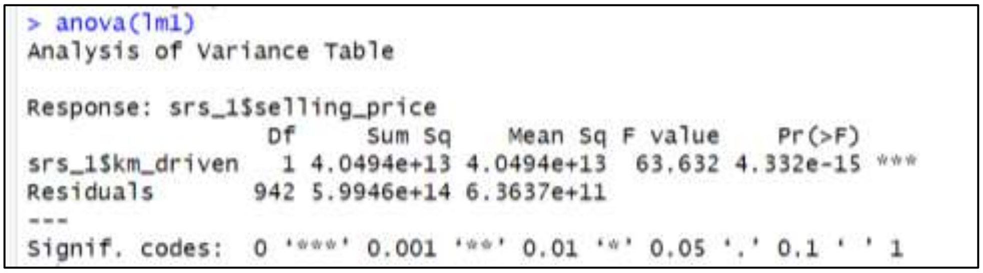


```
> lm1 = lm(srs_1$selling_price~srs_1$km_driven)
> lm1

Call:
lm(formula = srs_1$selling_price ~ srs_1$km_driven)

Coefficients:
    (Intercept)  srs_1$km_driven  
     923738.473           -3.913  
```

For each unit of Km , the price of the car will be decrease by 3.913 units and a brand new car with no km driven the price will be 923738.473 units

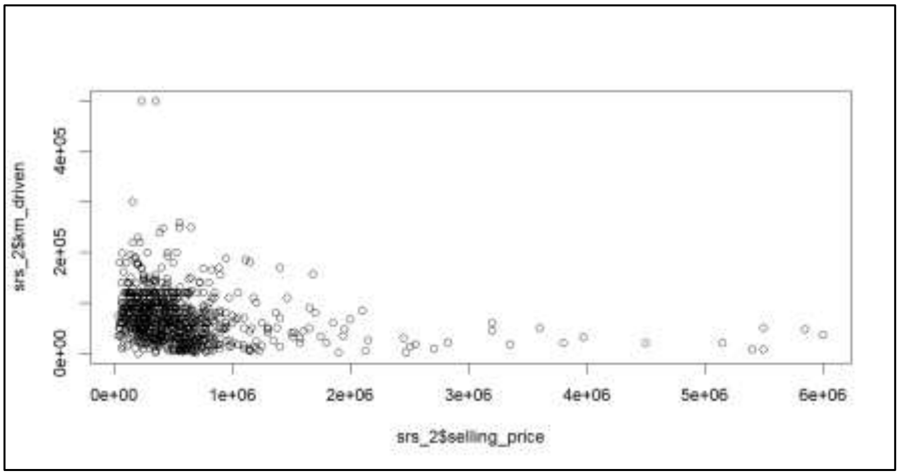


```
> summary(lm1)

Call:
lm(formula = srs_1$selling_price ~ srs_1$km_driven)

Residuals:
    Min      1Q  Median      3Q     Max 
-819605 -390890 -202341   46614 5023661 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.237e+05  4.350e+04  21.234  < 2e-16 ***
srs_1$km_driven -3.913e+00  4.906e-01  -7.977 4.33e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 797700 on 942 degrees of freedom
Multiple R-squared:  0.06328,	Adjusted R-squared:  0.06228 
F-statistic: 63.63 on 1 and 942 DF,  p-value: 4.332e-15
```

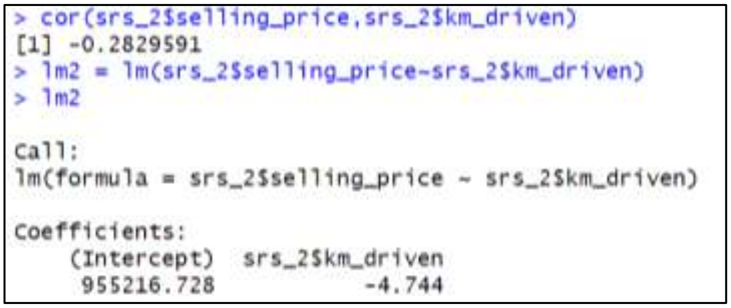


```
> anova(lm1)
Analysis of Variance Table

Response: srs_1$selling_price
                 Df     Sum Sq    Mean Sq F value    Pr(>F)    
srs_1$km_driven   1 4.0494e+13 4.0494e+13  63.632 4.332e-15 ***
Residuals       942 5.9946e+14 6.3637e+11                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

For sample 2 :



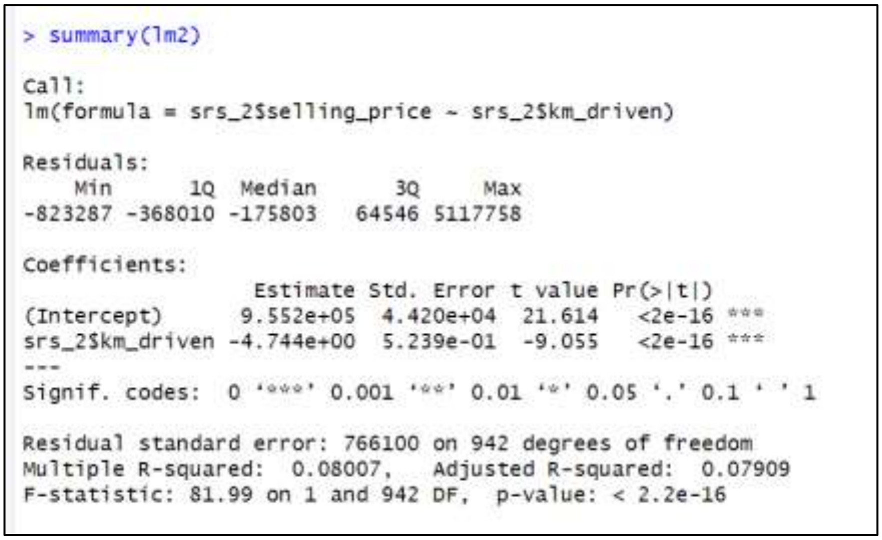
- ➢ It shows same relationships as sample 1 .



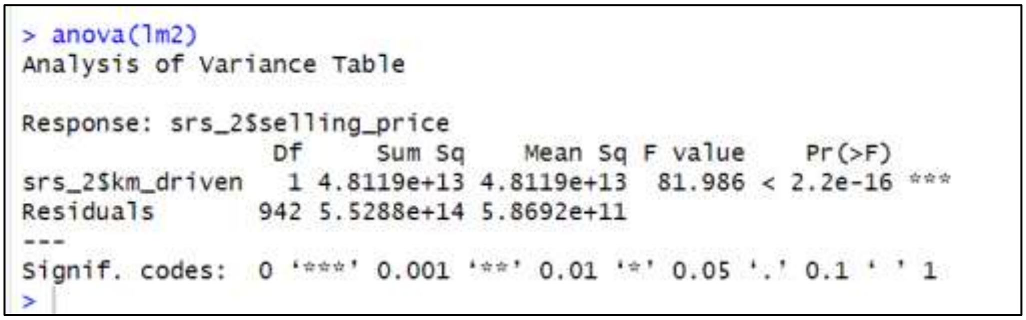
```
> cor(srs_2$selling_price,srs_2$km_driven)
[1] -0.2829591
> lm2 = lm(srs_2$selling_price~srs_2$km_driven)
> lm2

Call:
lm(formula = srs_2$selling_price ~ srs_2$km_driven)

Coefficients:
    (Intercept)  srs_2$km_driven  
     955216.728           -4.744  
```

The correlation coefficient is -0.2829591 This indicates a **weak negative correlation** between the selling price and kilometers driven in sample 2 . It means that as the kilometers driven increase, the selling price tends to decrease, but the relationship is not very strong.

For each unit of Km , the price of the car will be decrease by 34.744 units and a brand new car with no km driven the price will be 955216.728 units.



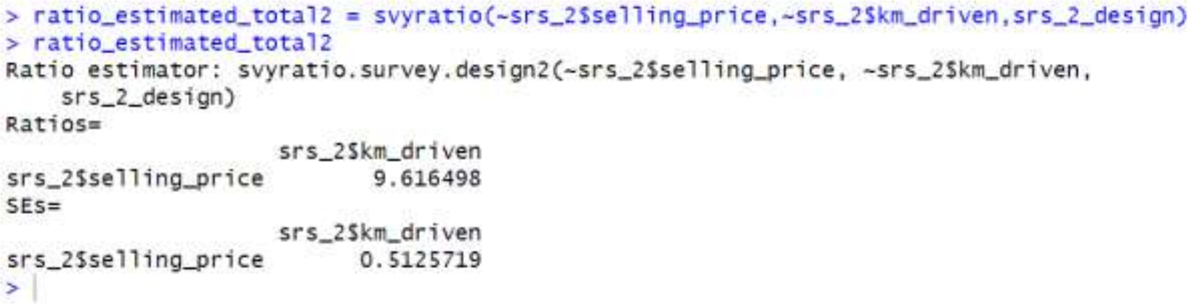
```
> summary(lm2)

Call:
lm(formula = srs_2$selling_price ~ srs_2$km_driven)

Residuals:
    Min      1Q  Median      3Q     Max 
-823287 -368010 -175803   64546 5117758 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.552e+05  4.420e+04  21.614   <2e-16 ***
srs_2$km_driven -4.744e+00  5.239e-01  -9.055   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 766100 on 942 degrees of freedom
Multiple R-squared:  0.08007,	Adjusted R-squared:  0.07909 
F-statistic: 81.99 on 1 and 942 DF,  p-value: < 2.2e-16
```



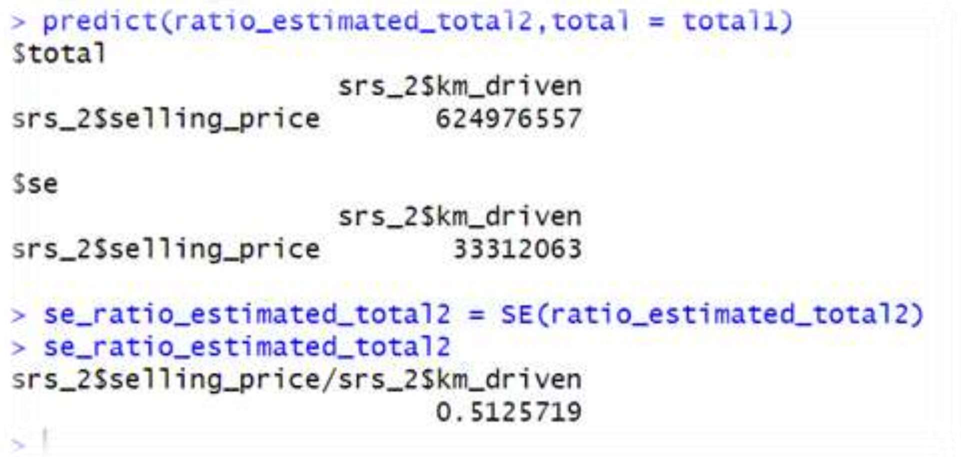
```
> anova(lm2)
Analysis of Variance Table

Response: srs_2$selling_price
                 Df     Sum Sq    Mean Sq F value    Pr(>F)    
srs_2$km_driven   1 4.8119e+13 4.8119e+13  81.986 < 2.2e-16 ***
Residuals       942 5.5288e+14 5.8692e+11                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

Ratio Analysis

Then for estimation of the total for the selling price, a ratio estimation procedure too is carried out.



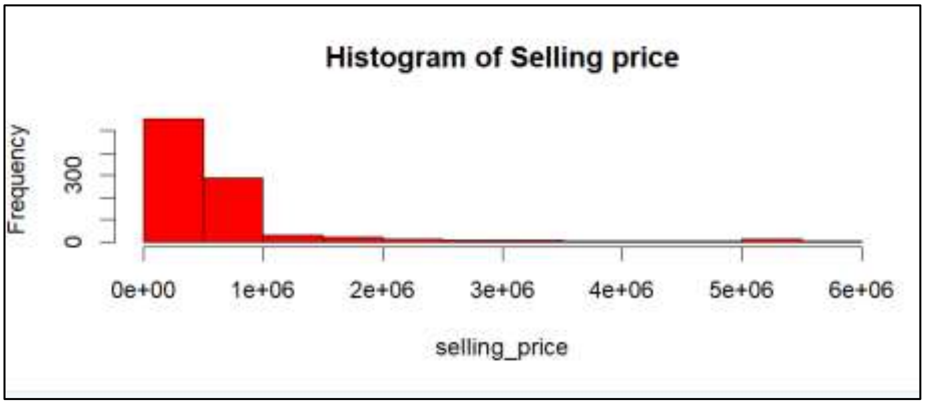
```
> ratio_estimated_total2 = svyratio(~srs_2$selling_price,~srs_2$km_driven,srs_2_design)
> ratio_estimated_total2
Ratio estimator: svyratio.survey.design2(~srs_2$selling_price, ~srs_2$km_driven,
    srs_2_design)
Ratios=
                    srs_2$km_driven
srs_2$selling_price        9.616498
SEs=
                    srs_2$km_driven
srs_2$selling_price       0.5125719
> 
```



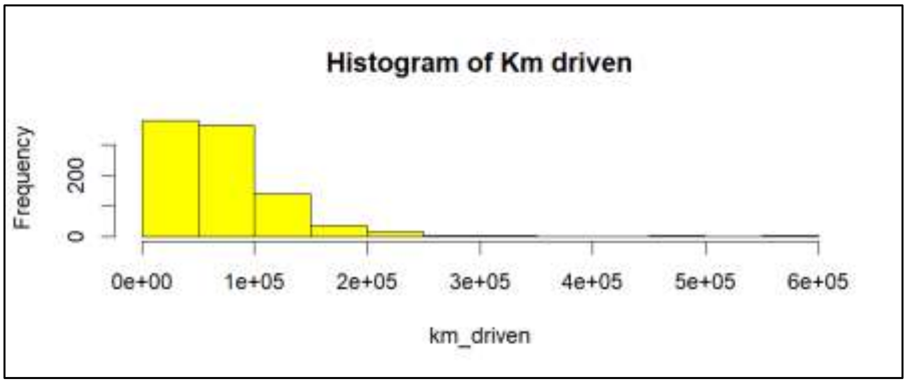
```
> predict(ratio_estimated_total2,total = total1)
$total
                    srs_2$km_driven
srs_2$selling_price       624976557

$se
                    srs_2$km_driven
srs_2$selling_price        33312063

> se_ratio_estimated_total2 = SE(ratio_estimated_total2)
> se_ratio_estimated_total2
srs_2$selling_price/srs_2$km_driven
                          0.5125719
> 
```

Graphical Analysis

For SRS 1

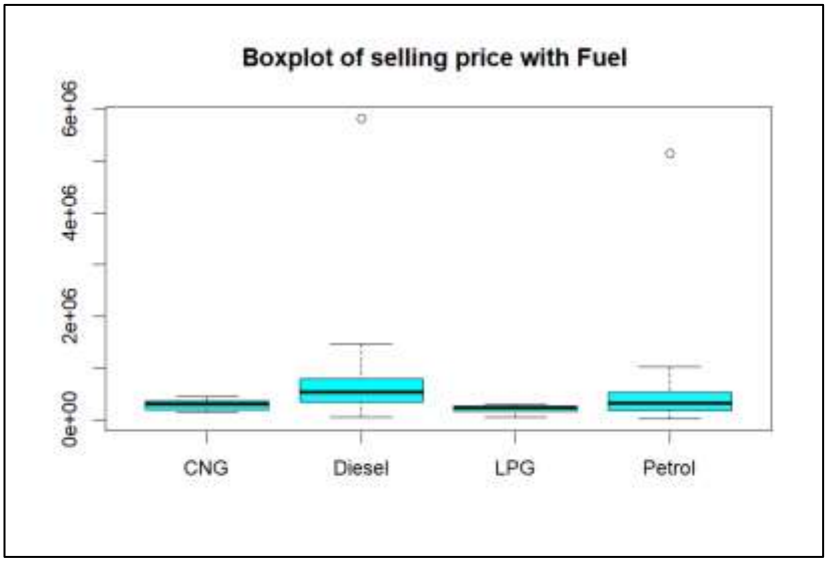


- Most vehicles are sold at lower prices, with a sharp drop in frequency for higher-priced items.
- The long tail towards the right suggests the presence of some outliers or high-value items in the dataset.



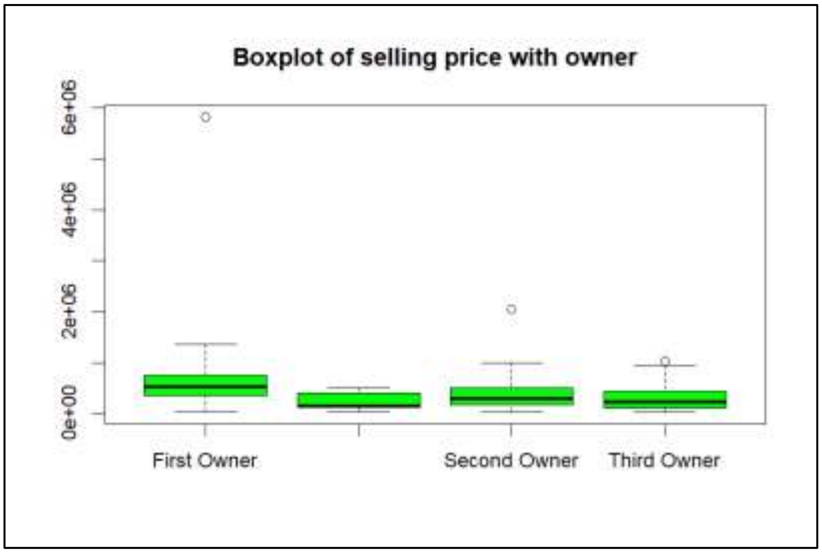
- Most vehicles have driven less than 100,000 km, with the count decreasing significantly as the distance increases.
- A few outliers are visible on the far right, where vehicles have driven up to 600,000 km.

This kind of distribution is typical in datasets related to vehicle usage, where the majority of vehicles fall into a more common mileage range, but some have unusually high values.



- Diesel vehicles tend to have higher selling prices and a larger range of values compared to the other fuel types.
- CNG and LPG vehicles have lower and more compact selling prices, indicating less variability in their prices.
- Petrol vehicles fall between Diesel and CNG/LPG in terms of selling price and variability.

- The presence of outliers for Diesel and Petrol indicates some high-priced vehicles within these categories.

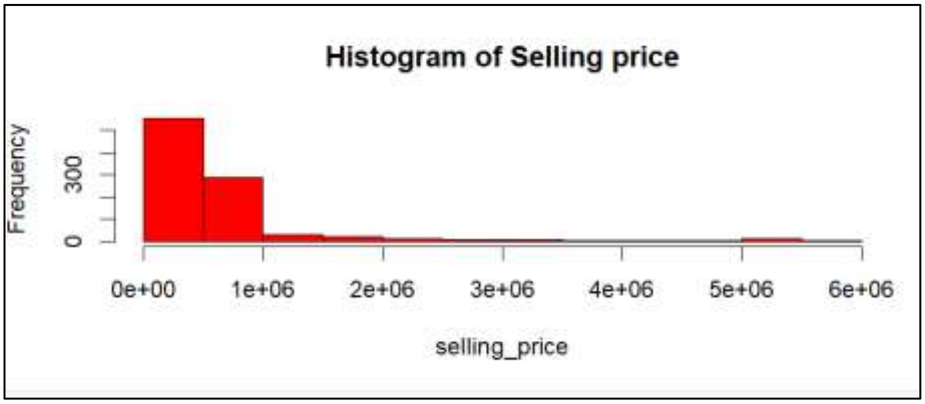


- Median Selling Prices: The median selling price appears to be slightly higher for first owners compared to second and third owners.
- Spread: The spread (IQR) of selling prices seems to be similar across all three owner categories, as indicated by the similar widths of the boxes.

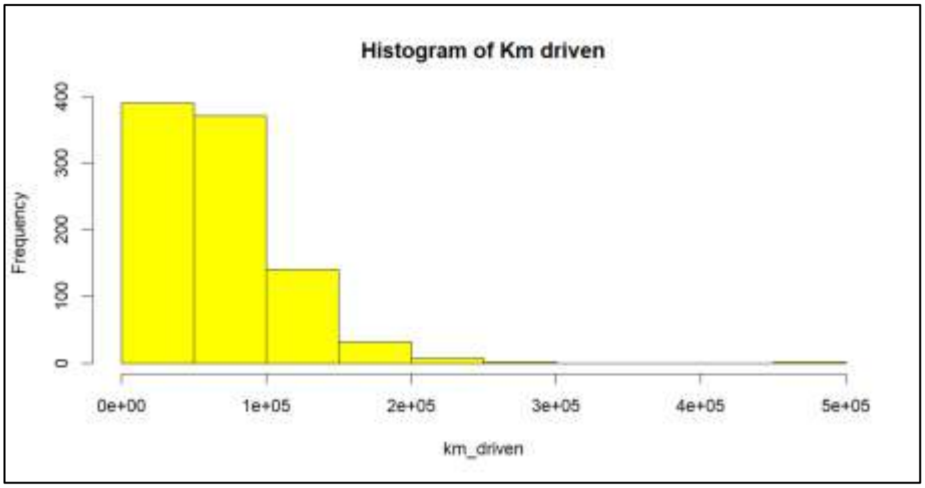
- Outliers: There are a few outliers present in the data, particularly among second and third owners. These outliers might represent cars with unique features, conditions, or market factors that influence their selling prices.

Overall Observations:While the median selling price is slightly higher for first owners, there is some overlap between the three owner categories. The presence of outliers indicates that individual factors beyond ownership can influence the selling price.

For SRS 2

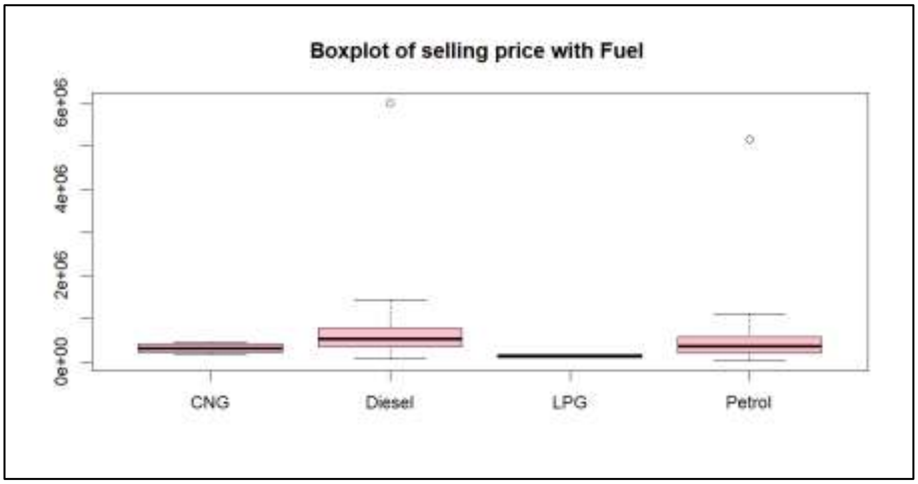


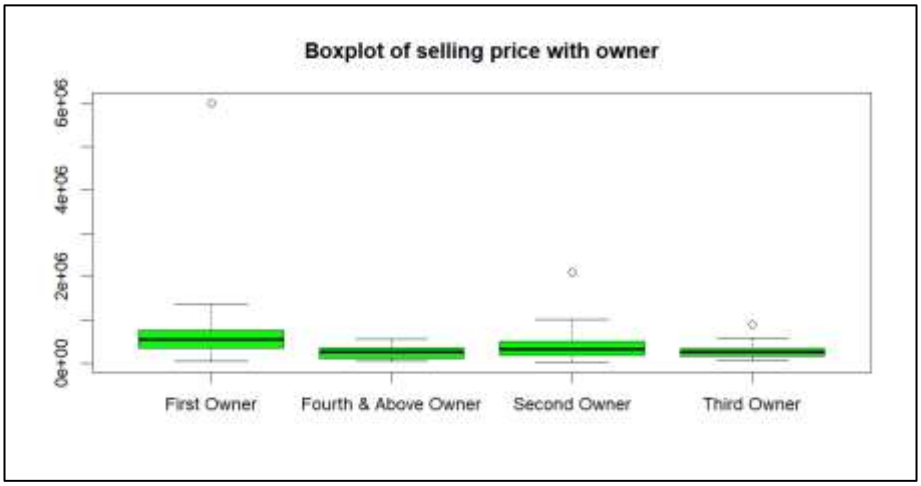
- Most vehicles are sold at lower prices, with a sharp drop in frequency for higher-priced items.
- The long tail towards the right suggests the presence of some outliers or high-value items in the dataset.



- Most vehicles have driven less than 100,000 km, with the count decreasing significantly as the distance increases.
- A few outliers are visible on the far right, where vehicles have driven up to 600,000 km.

This kind of distribution is typical in datasets related to vehicle usage, where the majority of vehicles fall into a more common mileage range, but some have unusually high values.





4. Stratified Random Sampling

- To apply stratified Random Sampling, we should first divide the population into strata. The variable we have selected here for this is the transmission type.
- The Auxiliary variable selected is the maximum power.
- The missing values are ignored in this sampling method.
- First, we decide the size of the simple random sample that needs to be selected from the population. The size obtained is 944.

```
> Sample_size = rsampcalc(nrow(cardekho), e=3,ci=95,0.5)
> Sample_size
[1] 944
```

- Then we decide the sample sizes of each of the stratum.

```
> #sample size for each strata
> strata_size=ssampcalc(cardekho,Sample_size,transmission)
> strata_size
# A tibble: 2 × 4
  transmission    Nh wt[,1] nh[,1]
  <fct>        <int>  <dbl>  <dbl>
1 Automatic     1050  0.129    122
2 Manual        7078  0.871    822
```

- In the dataset we have selected the population can be divided into 2 strata according to the transmission type. For the type “Automatic” we need to take a sample of size 122 and for type “Manual” we need to take a sample of size 822 by using the simple random sampling method.

Comparison of Parameter Estimations and Population Parameters

Variable	Population mean	Sample 01		Sample 02	
		Estimated Mean	S. E	Estimated Mean	S. E
Selling_price	649813.7	648104	21556	649546	22508
Max_power	91.58737	91.668	0.965	92.344	0.9994

Variable	Population total	Sample 01		Sample 02	
		Estimated total	S. E	Estimated total	S. E
Selling_price	5137427277	5488788348	182559269	4889783408	169443289
Max_power	724089.8	776339	8172.6	695163	7523.3

Variable	Population Proportion	Sample 01		Sample 02	
		Estimated Proportion	S. E	Estimated Proportion	S. E
Fuel					
Diesel	0.562167906	0.5409139	0.0162	0.5621679	0.0162
Petrol	0.429330499	0.4473964	0.0162	0.4293305	0.0161
LPG	0.004250797	0.0053135	0.0024	0.0042508	0.0021
CNG	0.004250797	0.0063762	0.0026	0.0042508	0.0021
Seller_type					
Dealer	0.13815090	0.150903	0.0111	0.138151	0.0106
Individual	0.83315622	0.806589	0.0119	0.833156	0.0113
Trustmark Dealer	0.02869288	0.042508	0.0065	0.028693	0.0054
owner					
First owner	0.67162593	0.6641870	0.0153	0.671626	0.0151
Second owner	0.24548353	0.2401700	0.0139	0.245484	0.0139
Third owner	0.07013815	0.0765143	0.0087	0.070138	0.0083
Fourth and above owner	0.01275239	0.0170032	0.0042	0.012752	0.0037

Key Observations:

<u>Selling Price:</u>

- The population mean selling price is 649,813.7.
- The estimated means from Sample 01 and Sample 02 are relatively close to the population mean, suggesting that the sampling was reasonably representative.
- The standard errors (S.E.) for both samples are relatively small, indicating that the estimates are precise.
- The estimated population totals from both samples are close to the actual population total, suggesting that the sampling and estimation methods were effective.

<u>Max Power:</u>

- The population mean max power is 91.58737.
- The estimated means from both samples are slightly higher than the population mean, but the differences are within a reasonable range given the standard errors.
- The standard errors for max power are very small, indicating high precision in the estimates.
- The estimated population totals from both samples are again close to the actual population total, suggesting effective sampling and estimation

Fuel Type:

- The population proportions for diesel, petrol, LPG, and CNG are relatively stable across the samples, suggesting that the sampling was representative.
- The standard errors (S.E.) for all fuel types are small, indicating precise estimates.

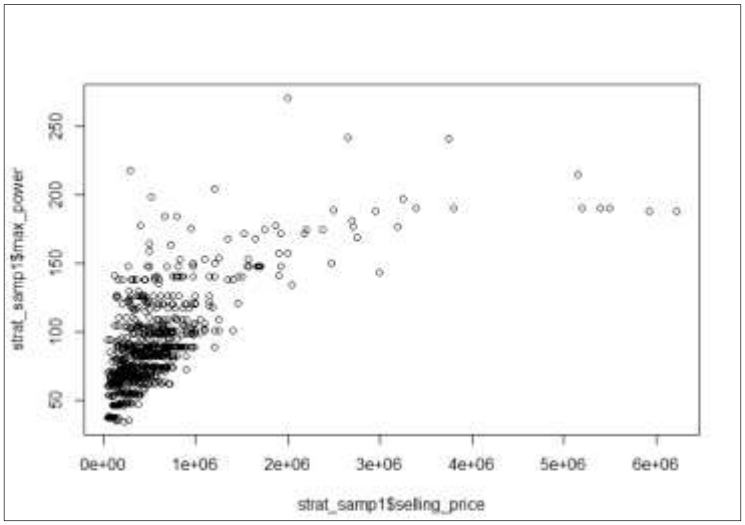
Seller Type:

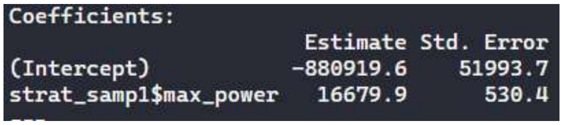
- The population proportion of "Individual" sellers is significantly higher than the others.
- The estimated proportions from the samples are close to the population proportions, suggesting good sampling.
- The standard errors for seller types are generally small, indicating precise estimates.

Dealer Owner:

- The population proportion of "First owner" cars is highest, followed by "Second owner."
- The estimated proportions from the samples are consistent with the population proportions.
- The standard errors for dealer ownership categories are relatively small, suggesting precise estimates

Regression Analysis for Sample 01

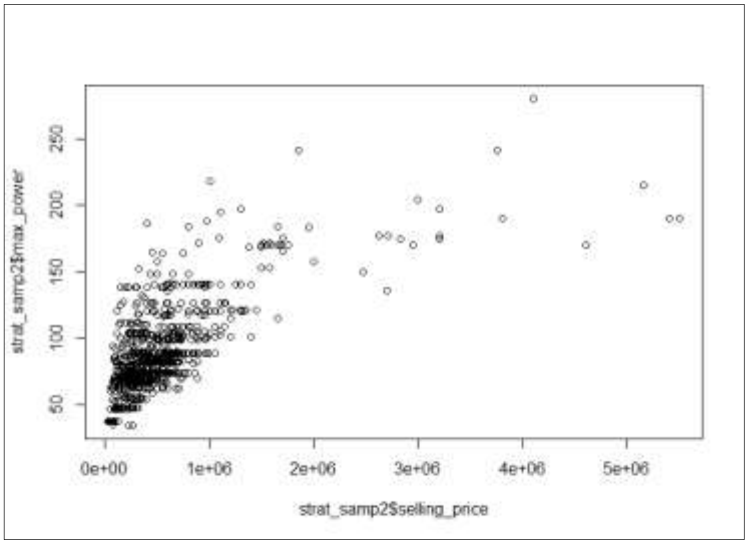


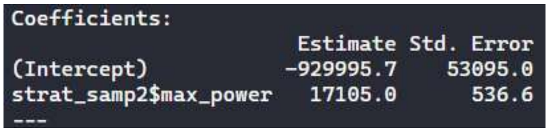


```
Coefficients:
                       Estimate Std. Error
(Intercept)           -880919.6    51993.7
strat_samp1$max_power   16679.9      530.4
```

- Mean of max_power in the population was obtained as 91.58737
- The mean selling_price obtained from the regression model is 646748.6
- The estimated population total for selling_price obtained from the regression model is 5113194217

Regression Analysis for Sample 02



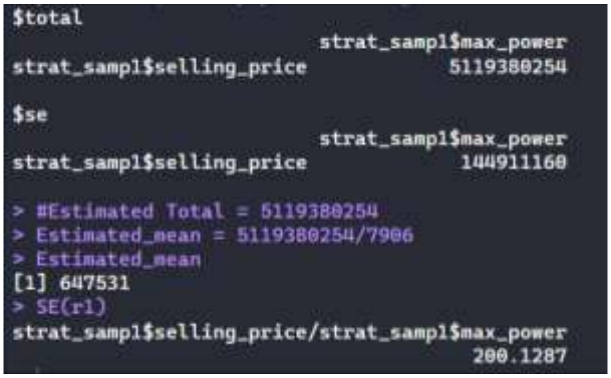

```
Coefficients:
                      Estimate Std. Error
(Intercept)          -929995.7    53095.0
strat_samp2$max_power  17105.0      536.6
---
```

- Mean of max_power in the population was obtained as 91.58737
- The mean selling_price obtained from the regression model is 646748.6
- The estimated population total for selling_price obtained from the regression model is 5033009122

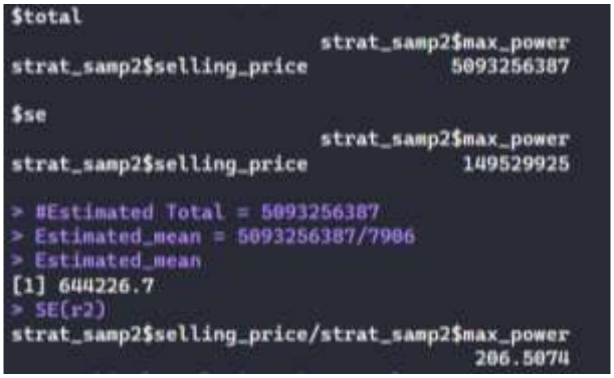
Ratio Analysis

To estimate the mean of selling_price we can carry out a ratio estimation procedure.

Sample 01

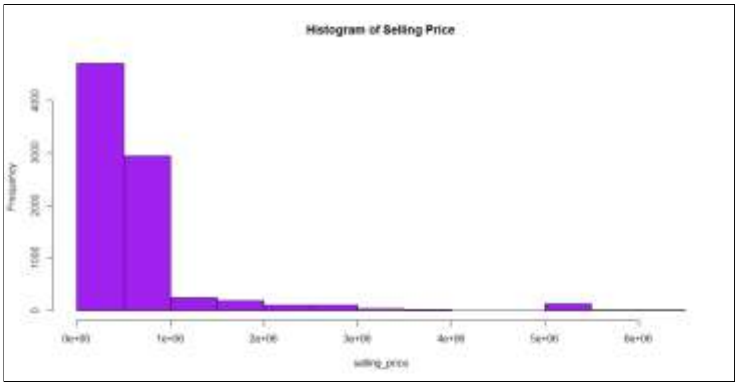

```
$total
                             strat_samp1$max_power
strat_samp1$selling_price               5119380254

$se
                             strat_samp1$max_power
strat_samp1$selling_price                144911160

> #Estimated Total = 5119380254
> Estimated_mean = 5119380254/7906
> Estimated_mean
[1] 647531
> SE(r1)
strat_samp1$selling_price/strat_samp1$max_power
                                       200.1287
```

Sample 02

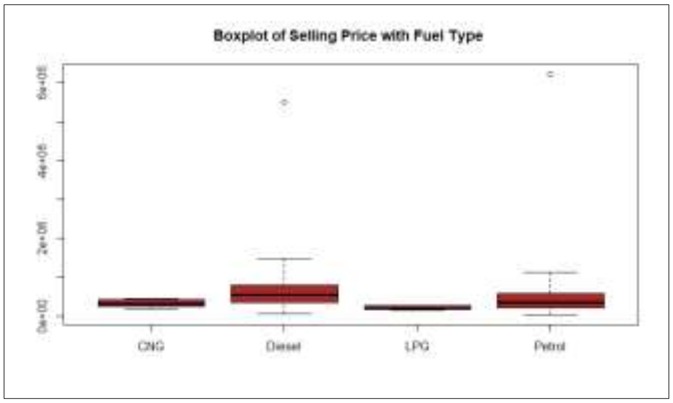

```
$total
                             strat_samp2$max_power
strat_samp2$selling_price               5093256387

$se
                             strat_samp2$max_power
strat_samp2$selling_price                149529925

> #Estimated Total = 5093256387
> Estimated_mean = 5093256387/7906
> Estimated_mean
[1] 644226.7
> SE(r2)
strat_samp2$selling_price/strat_samp2$max_power
                                       206.5074
```

- The estimated mean obtained from the first sample under ratio estimation is 647531, and for the second sample it is 644226.7
- Both values are closer to the population parameter.

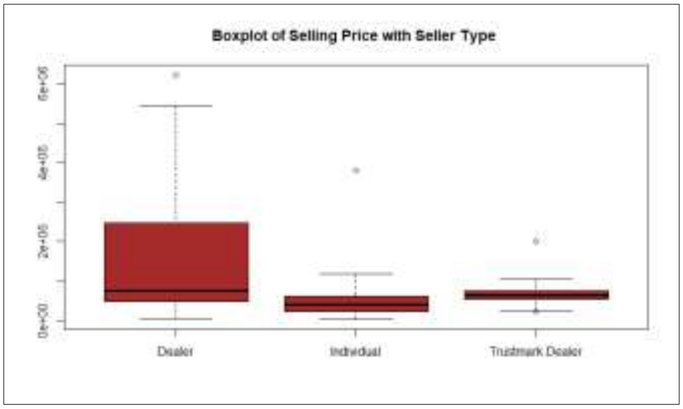
Graphical analysis for Stratified Sampling (For sample 01)



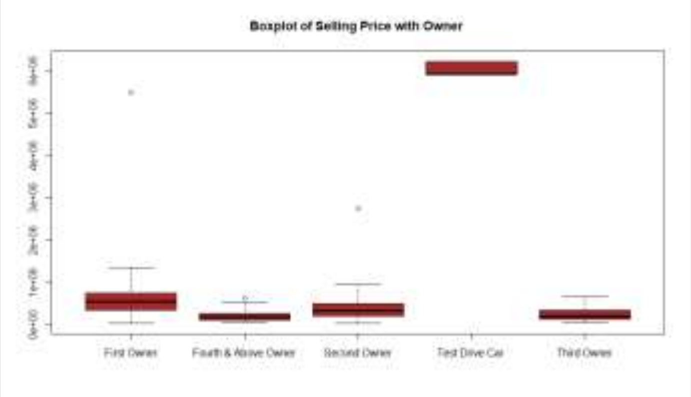
The histogram shows the distribution of selling prices in a dataset. It is skewed to the right, meaning there are lower-priced items than higher-priced ones. The frequency of observations decreases as the price increases.



The boxplot shows the distribution of selling prices for different fuel types (CNG, Diesel, LPG, and Petrol). Diesel vehicles generally have higher selling prices compared to other fuel types, while CNG and LPG vehicles tend to have lower prices. Petrol vehicles have a wider range of selling prices.

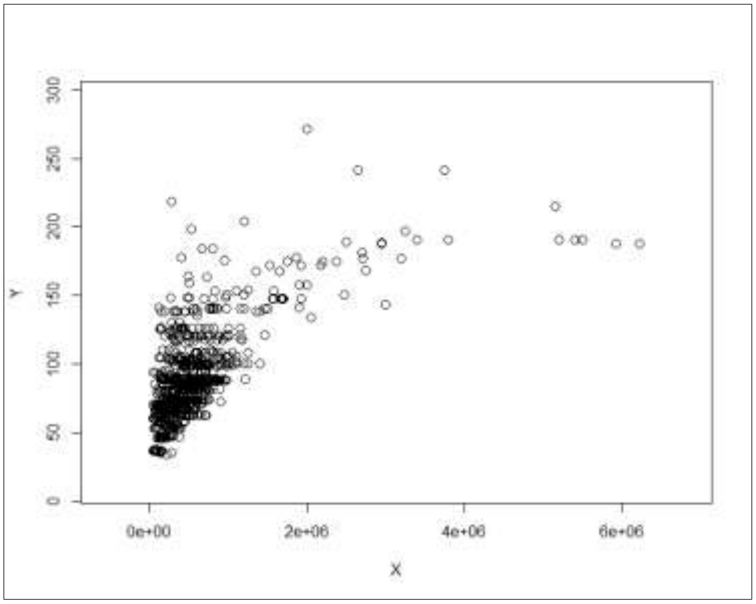


Dealer-sold vehicles tend to have higher selling prices compared to Individual sellers and Trustmark Dealers. The overall distribution of selling prices is relatively similar across all seller types, with some variation in the median and the presence of outliers.



First Owner vehicles tend to have higher selling prices compared to vehicles with subsequent ownerships. The overall distribution of selling prices is relatively similar across all ownership categories, with some variation in the median and the presence of outliers.

The Relationship between the Auxiliary Variable(x) and the Variable of Interest(Y)



Key observations:

• **Positive correlation:** There seems to be a general positive correlation between X and Y, meaning that as X increases, Y tends to increase as well. However, the relationship is not perfectly linear, and there is some scatter around the trend.

• **Clustering:** The data points appear to cluster in certain areas of the plot, suggesting that there may be subgroups or patterns within the data.

- **Outliers**: A few outliers can be observed, which are data points that are significantly different from the majority of the data.

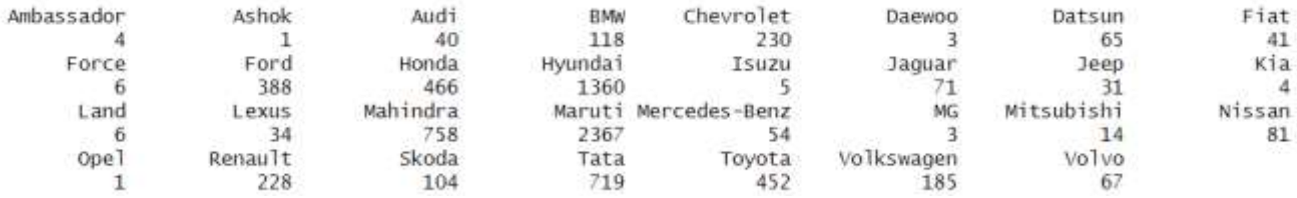
In summary:

The scatter plot suggests a positive correlation between X and Y, with some clustering and outliers present in the data.

5. Two Stage Cluster Sampling

Here we take a two-stage cluster sample to estimate the population mean, total, variance and proportion.

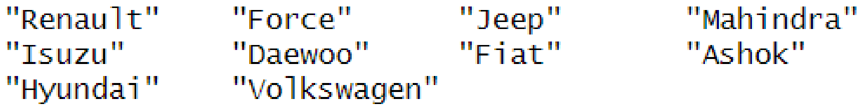
We select the clustering variable as ‘car model’ and we randomly choose several car models and from those selected car models, we again take a simple random sample to obtain the two-stage cluster sample. Here we have 31 total car models as clusters and we randomly choose 10 car models out of them.



```
   Ambassador         Ashok          Audi           BMW     Chevrolet        Daewoo        Datsun          Fiat 
            4             1            40           118           230             3            65            41 
        Force          Ford         Honda       Hyundai         Isuzu        Jaguar          Jeep           Kia 
            6           388           466          1360             5            71            31             4 
         Land         Lexus      Mahindra        Maruti Mercedes-Benz            MG    Mitsubishi        Nissan 
            6            34           758          2367            54             3            14            81 
         Opel       Renault         Skoda          Tata        Toyota    Volkswagen         Volvo 
            1           228           104           719           452           185            67 
```

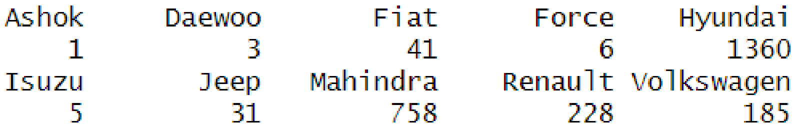
Number of observations in each cluster

Out of these clusters following were selected



```
"Renault"    "Force"      "Jeep"       "Mahindra"
"Isuzu"      "Daewoo"     "Fiat"       "Ashok"
"Hyundai"    "Volkswagen"
```

Following are the number of observations in each sampled clusters



```
     Ashok     Daewoo       Fiat      Force    Hyundai 
         1          3         41          6       1360 
     Isuzu       Jeep   Mahindra    Renault Volkswagen 
         5         31        758        228        185 
```

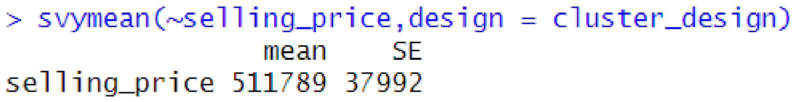
We take simple random samples of size n ≤ 50 from each selected cluster to obtain a sample. Then we have samples from each cluster as following

Car model (Cluster)	Population size	Sample size
Ashok	1	1
Daewoo	3	3
Fiat	41	41
Force	6	6
Hyundai	1360	50
Isuzu	5	5
Jeep	31	31
Mahindra	758	50
Renault	228	50
Volkswagen	185	50

Total sample size = 287

Mean

- Estimate for the mean selling price

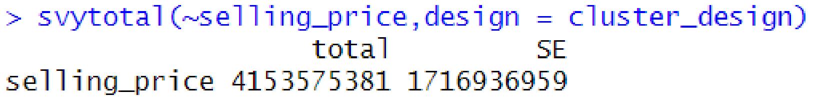


```
> svymean(~selling_price,design = cluster_design)
                mean    SE
selling_price 511789 37992
```

- Population mean selling price = 649813.7

Total

- Estimate for the total selling price

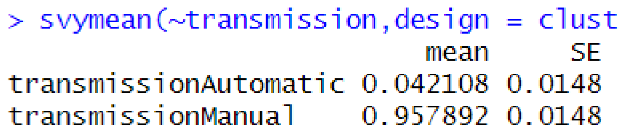


```
> svytotal(~selling_price,design = cluster_design)
                   total         SE
selling_price 4153575381 1716936959
```

- Population total selling price = 5137427277

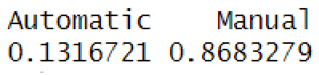
Proportion

- ➢ Estimate for the proportion of transmission



```
> svymean(~transmission,design = cluster_design)
                            mean     SE
transmissionAutomatic 0.042108 0.0148
transmissionManual    0.957892 0.0148
```

- ➢ Population proportion

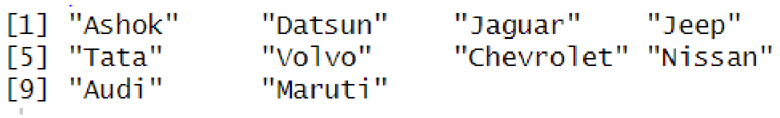


```
Automatic    Manual
0.1316721 0.8683279
```

We will take another two-stage cluster sample.

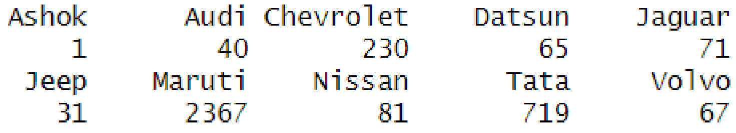
Sample 2

Selected clusters



```
[1] "Ashok"     "Datsun"    "Jaguar"    "Jeep"
[5] "Tata"      "Volvo"     "Chevrolet" "Nissan"
[9] "Audi"      "Maruti"
```

Number of observations in each sampled cluster



```
   Ashok      Audi Chevrolet    Datsun    Jaguar
       1        40       230        65        71
    Jeep    Maruti    Nissan      Tata     Volvo
      31      2367        81       719        67
```

As before, we take simple random samples of size n ≤ 50 from each selected cluster to obtain a sample. Then we have samples from each cluster as following

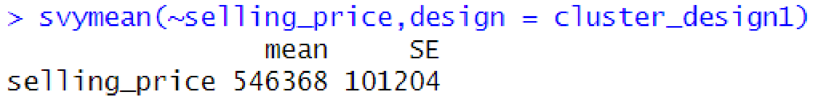
Car model (Cluster)	Population size	Sample size
Ashok	1	1
Audi	40	40
Chevrolet	230	50
Datsun	65	50
Jaguar	71	50

Jeep	31	31
Maruti	2367	50
Nissan	81	50
Tata	719	50
Volvo	67	50

Total sample size = 422

Mean

- Estimate for the mean selling price

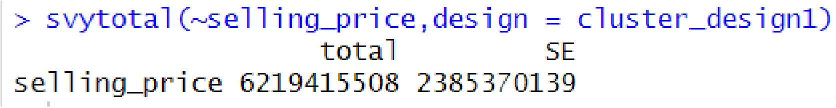


```
> svymean(~selling_price,design = cluster_design1)
                  mean     SE
selling_price 546368 101204
```

- Population mean selling price = 649813.7

Total

- Estimate for the total selling price

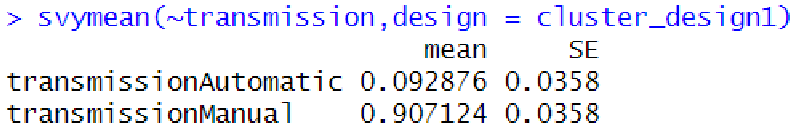


```
> svytotal(~selling_price,design = cluster_design1)
                   total         SE
selling_price 6219415508 2385370139
```

- Population total selling price = 5137427277

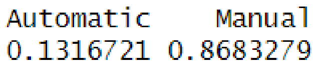
Proportion

- Estimate for the proportion of transmission



```
> svymean(~transmission,design = cluster_design1)
                          mean     SE
transmissionAutomatic 0.092876 0.0358
transmissionManual    0.907124 0.0358
```

- Population Proportion



```
Automatic    Manual
0.1316721 0.8683279
```

Summary

	Population	Sample 1		Sample 2	
	Mean	Estimated population mean	Standard Error	Estimated population mean	Standard Error
Selling price	649813.7	511789	37992	546368	101204

- Here we can see that estimated values from both samples are quite different from the population mean. Sample estimated values are somewhat close to each other. But the standard error of the 2nd sample is much higher than 1st sample

		Population	Sample 1		Sample 2	
		Proportion	Estimated population Proportion	Standard Error	Estimated population Proportion	Standard Error
Transmission	Automatic	0.1316721	0.042108	0.0148	0.092876	0.0358
	Manual	0.8683279	0.957892	0.0148	0.907124	0.0358
Owner	First owner	0.659626	0.669603	0.0354	0.66644336	0.0439
	Second owner	0.254996	0.242750	0.0237	0.21691721	0.0248
	Third owner	0.064508	0.059664	0.0254	0.10015251	0.0335
	Fourth and above owner	0.020238	0.027983	0.0107	0.01566993	0.0083

- Estimated proportions for transmission type are quite different from population proportions. However, estimates obtained from the second sample are closer to the population proportions than the first sample estimates.

	Population	Sample 1		Sample 2	
	Total	Estimated population total	Standard Error	Estimated population total	Standard Error
Selling price	5137427277	4153575381	1716936959	6219415508	2385370139

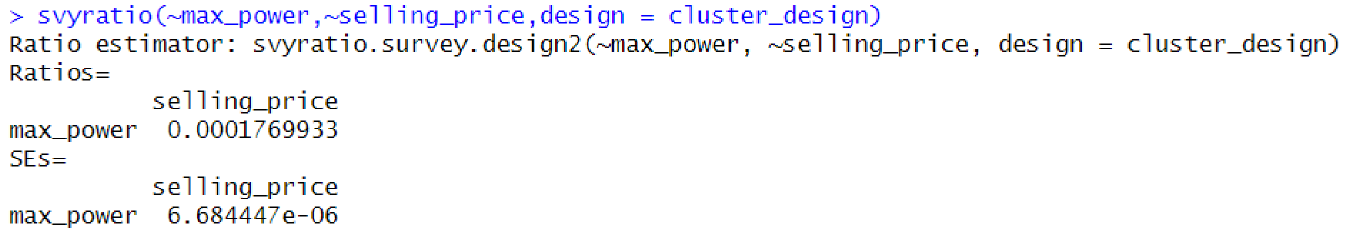
- Here we can see first sample underestimates the population total and the second sample overestimates the population total.

Ratio Estimation

We will use ratio estimation to estimate the selling price of cars. For that, we use 'max_power' as the auxiliary variable as it is highly correlated with the selling price.

Sample 1

B (ratio) = $\frac{mean\ max\ power}{mean\ \text{selling}\ price}$



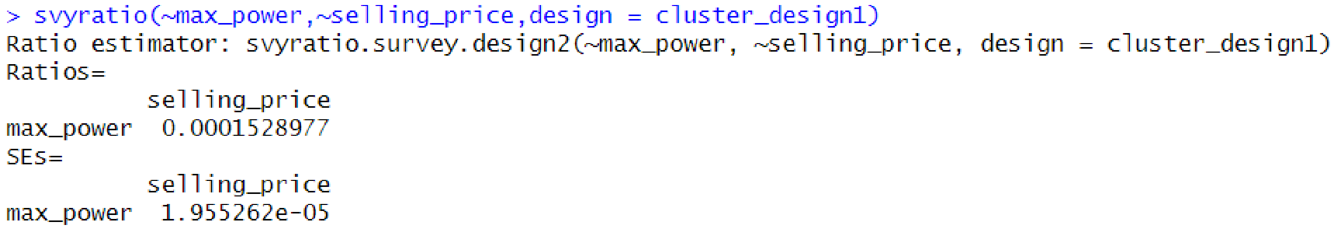
```
> svyratio(~max_power,~selling_price,design = cluster_design)
Ratio estimator: svyratio.survey.design2(~max_power, ~selling_price, design = cluster_design)
Ratios=
          selling_price
max_power  0.0001769933
SEs=
          selling_price
max_power  6.684447e-06
```

The mean max power in population was obtained as 91.58737
Then the estimated selling price = 91.58737 / 0.0001769933

= 517462.4

- Here the ratio estimation of the selling price is significantly different from the population mean value of the selling price (649813.7).

Sample 2



```
> svyratio(~max_power,~selling_price,design = cluster_design1)
Ratio estimator: svyratio.survey.design2(~max_power, ~selling_price, design = cluster_design1)
Ratios=
          selling_price
max_power  0.0001528977
SEs=
          selling_price
max_power  1.955262e-05
```

Since the mean max power in the population is 91.58737, we can obtain the estimated selling price as,

Estimated selling price = 91.58737 / 0.0001528977
= 599010.8

- We can see the estimated selling price is quite close to the population mean selling price (649813.7)

Summary

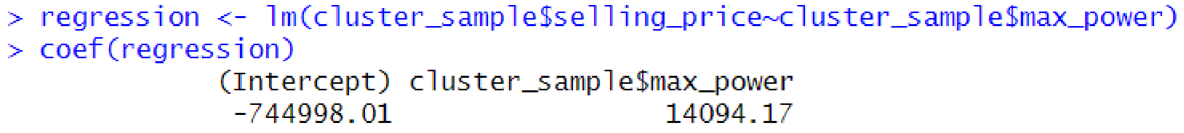
	Population mean	Estimate from sample 1	Estimate from sample 2
Selling price	649813.7	517462.4	599010.8

- The mean selling prices estimated from sample 1 and sample 2 show some difference. Here sample 2 is likely to give a more accurate estimate as it is close to the population mean .

Regression Estimation

We will use ratio estimation to estimate the selling price of cars. Same as in ratio estimation, we use ‘max_power’ as the auxiliary variable as it is highly correlated with the selling price.

Sample 1



```
> regression <- lm(cluster_sample$selling_price~cluster_sample$max_power)
> coef(regression)
              (Intercept) cluster_sample$max_power 
               -744998.01                 14094.17 
```

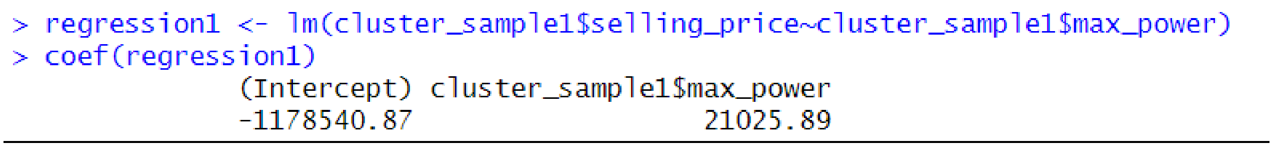
We can write the model as,
selling_price = -744998.01 + 14094.17 (max_power)

The mean max power in population was obtained as 91.58737
Then we can estimate the mean selling price,

Estimated mean selling price = -744998.01 + 14094.17 * 91.58737
= 545850

We can see there’s a difference of more than 100,000 between the estimated mean value and population mean value (649813.7)

Sample 2



```
> regression1 <- lm(cluster_sample1$selling_price~cluster_sample1$max_power)
> coef(regression1)
            (Intercept) cluster_sample1$max_power
            -1178540.87                  21025.89
```

We can write the model as,
selling_price = -1178540.87 + 21025.89 (max_power)

The mean max power in population was obtained as 91.58737
Then we can estimate the mean selling price,

Estimated mean selling price = -1178540.87 + 21025.89 * 91.58737
= 747165.1

From this sample, the mean selling price was overestimated the population mean selling price (649813.7) by a significant amount.

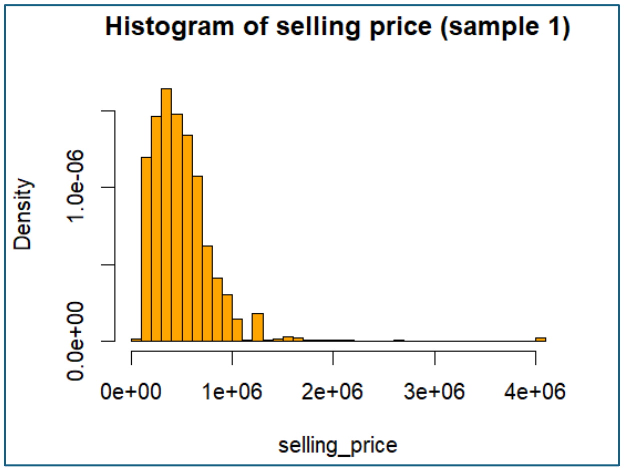
Summary

	Population mean	Estimate from sample 1	Estimate from sample 2
selling price	649813.7	545850	747165.1

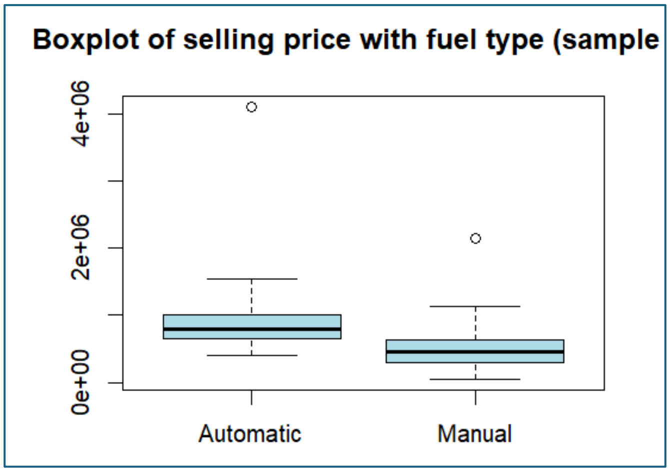
- In regression estimates, the estimated mean from sample 1 and the estimated mean from sample 2 differ significantly and none of them are close to the population mean. Therefore, we can conclude that the regression estimations did not accurately estimate the population mean.

Graphical Analysis

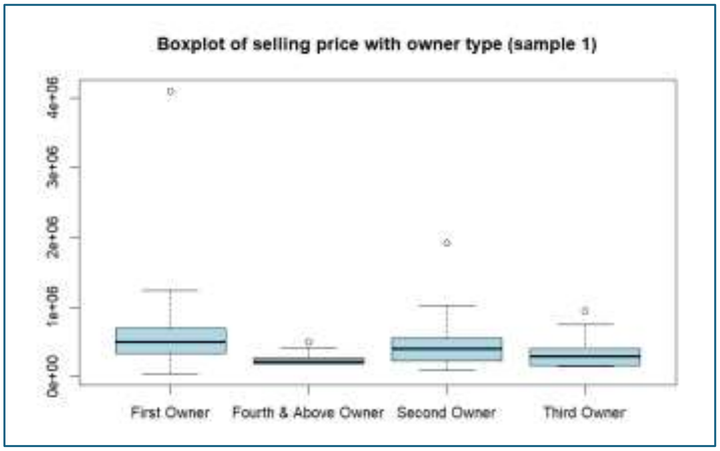
Sample 1



This histogram represents the estimated population distribution of selling prices. The distribution skewed to the right suggesting that there are more lower price cars than higher price cars. Also, there are some outliers present with very high selling prices

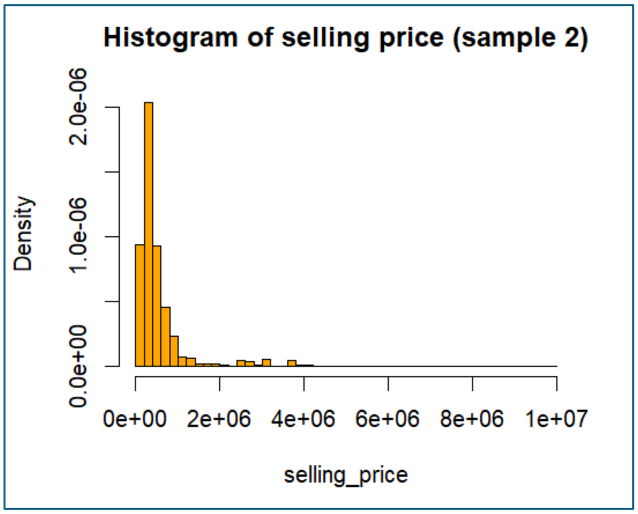


This boxplot represents the estimated distribution of selling price by transmission type. We can see the median selling price of automatic transmission-type cars is higher than that of manual transmission-type cars. Here also we observe some outliers at the higher end of the selling price distribution, indicating that a few cars are priced significantly above the majority.

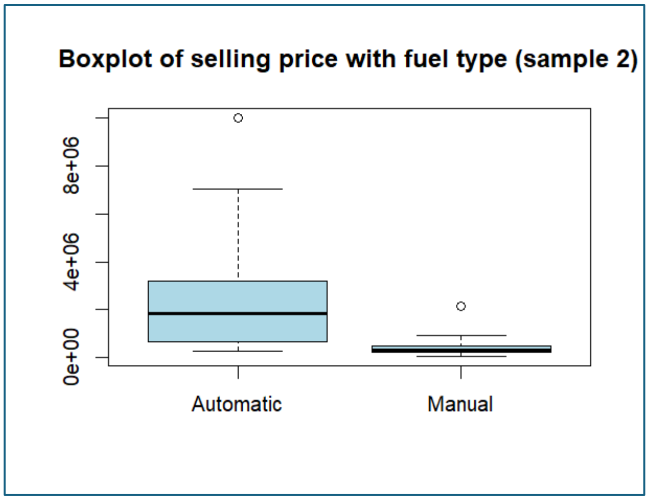


This boxplot depicts the estimated distribution of selling price by owner type. First-owner cars have the highest median selling price. Here also we can observe some outliers present in each category.

Sample 2



This histogram represents the estimated population distribution of selling prices from the second sample. The distribution is right skewed indicating there are more lower price cars than higher price cars. But in this graph, the range is much higher than the range estimated by the first sample. There are outliers with extremely high values.



The median selling price for automatic cars higher than manual cars. But there is a difference in estimate selling price than it estimated by sample 1.

6. Conclusion

The report on car price trends from an Indian city, based on a census dataset, presents a thorough analysis of car prices using various sampling techniques. Simple random sampling, stratified random sampling, and two-stage cluster sampling were applied to estimate car price parameters like mean, total, and proportions. The findings reveal a general negative correlation between car prices and kilometers driven, indicating that as cars accumulate mileage, their selling prices decrease. Outliers and variations were noted, suggesting that factors like brand, model, and condition can influence price beyond mileage.

Each sampling technique yielded slightly different estimates, with stratified sampling proving to be more reliable for estimating means across different car categories. Simple random sampling, while effective, showed minor deviations in total estimates compared to the population values. Two-stage cluster sampling provided mixed results, with one sample underestimating the population total and the other overestimating it.

Overall, the report highlights the importance of choosing the right sampling technique depending on the dataset structure and research goals. The regression and ratio estimation methods further confirmed relationships between variables, although the estimates did not always match the actual population parameters. The insights gained can assist policymakers and urban planners in understanding car ownership patterns and market behavior in the region.

7. R Codes

Simple Random Sampling

```
attach(cardekho)

data <- read.csv("cardekho.csv")

library(survey)
install.packages("sampler")
library(sampler)

card <- svydesign(ids=-1, probs = NULL, strata = NULL, fpc = NULL, weights = NULL, data =
cardekho)
summary(card)


#population total
total <-sum(selling_price)
total

total1<-sum(km_driven)
total1

#population mean
meanPop<-mean(selling_price)
meanPop
meanPop1<-mean(km_driven)
meanPop1

#population standard deviation
population_sd <- sd(selling_price)
population_sd

population_sd1 <- sd(km_driven)
population_sd1
#population prorpotion
pop_prop_fuel = (table(fuel))/(length(fuel))
pop_prop_fuel


pop_prop_owner = (table(owner))/(length(owner))
pop_prop_owner
#Get the number of observations
t <- length(cardekho$selling_price)
```

```
t
# Define the population as a vector of numbers from 1 to 10000
population <- 1:8128
population

#selecting a sample
srs_size = rsampcalc(t,e = 3,ci = 95,p = 0.5,over = 0)
srs_size

# Take a simple random sample (1)without replacement from the population
srs_1 = rsamp(data,n = srs_size,rep = FALSE)
srs_1


# Take a simple random sample (2))without replacement from the population
srs_2 = rsamp(data,n = srs_size,rep = FALSE)
srs_2

# Save the sampled data to a new CSV file
sample1<-write.csv(srs_1, "sampled_data.csv", row.names = TRUE)

#Save the sampled data to a new CSV file
sample2<-write.csv(srs_2, "sampled_data1.csv", row.names = TRUE)

attach(srs_1)
srs_1_design = svydesign(id = ~1,strata = NULL,data = srs_1)
srs_1_design

#Sample mean for selling price
srs_1_mean_sellingprice= svymean(selling_price,srs_1_design)
srs_1_mean_sellingprice

#Sample total for sellingprice
srs_1_total_sellingprice= svytotal(selling_price,srs_1_design)
srs_1_total_sellingprice

estimated_tot_sellingprice = length(selling_price)*srs_1_mean_sellingprice
estimated_tot_sellingprice

#Sample proportion for fuel
srs_1_prop_fuel = svymean(fuel,srs_1_design)
srs_1_prop_fuel

#Sample proportion for owner
srs_1_prop_fuel = svymean(~fuel,srs_1_design)
srs_1_prop_fuel
```

```
srs_1_prop_owner = svymean(~owner,srs_1_design)
srs_1_prop_owner



attach(srs_2)
srs_2_design = svydesign(id = ~1,strata = NULL,data = srs_2)
srs_2_design

#Sample mean for selling price
srs_2_mean_sellingprice= svymean(selling_price,srs_2_design)
srs_2_mean_sellingprice

#Sample total for sellingprice
srs_2_total_sellingprice= svytotal(selling_price,srs_2_design)
srs_2_total_sellingprice

estimated_tot_sellingprice2 = length(selling_price)*srs_2_mean_sellingprice
estimated_tot_sellingprice2

srs_2_prop_fuel = svymean(~fuel,srs_2_design)
srs_2_prop_fuel
srs_2_prop_owner = svymean(~owner,srs_2_design)
srs_2_prop_owner




#GRAPHICAL ANALYSIS USING SAMPLE 01
svyhist(~selling_price,srs_1_design,main = "Histogram of Selling price", col = "Red",probability = FALSE)
svyhist(~km_driven,srs_1_design,main = "Histogram of Km driven", col = "Yellow",probability = FALSE)
svyboxplot(~selling_price~fuel,srs_1_design, main = "Boxplot of selling price with Fuel ",col = "Cyan" )

svyboxplot(~selling_price~owner,srs_1_design, main = "Boxplot of selling price with owner",col = "Green" )

svyboxplot(~posttest~school_type,srs_1_design,
        main = "Boxplot of Posttest with school_type ", col = "Cyan" )

svyplot(pretest~posttest,design = srs_1_design,style = "bubble")
```

```
svyhist(~selling_price,srs_2_design,main = "Histogram of Selling price", col = "blue",probability = FALSE)
svyhist(~km_driven,srs_2_design,main = "Histogram of Km driven", col = "Yellow",probability = FALSE)
svyboxplot(~selling_price~fuel,srs_2_design, main = "Boxplot of selling price with Fuel ",col = "Pink" )

svyboxplot(~selling_price~owner,srs_2_design, main = "Boxplot of selling price with owner",col = "Green" )

svyboxplot(~posttest~school_type,srs_2_design,
        main = "Boxplot of Posttest with school_type ", col = "Cyan" )

svyplot(pretest~posttest,design = srs_2_design,style = "bubble")




#Fitting a linear regression model

plot(srs_1$selling_price,srs_1$km_driven)
cor(srs_1$selling_price,srs_1$km_driven)
lm1 = lm(srs_1$selling_price~srs_1$km_driven)
lm1
summary(lm1)
anova(lm1)
dev.off()

plot(srs_2$selling_price,srs_2$km_driven)
cor(srs_2$selling_price,srs_2$km_driven)
lm2 = lm(srs_2$selling_price~srs_2$km_driven)
lm2
summary(lm2)
anova(lm2)

#  Ratio Analysis

ratio_estimated_total2 = svyratio(~srs_2$selling_price,~srs_2$km_driven,srs_2_design)

ratio_estimated_total2

predict(ratio_estimated_total2,total = total1)

se_ratio_estimated_total2 = SE(ratio_estimated_total2)

se_ratio_estimated_total2
```

Stratified Sampling

```
library("survey")

library("sampler")

attach(cardekho)

cardekho <- na.omit(cardekho)

summary(cardekho)



#Calculating the mean and standard deviations by Stratum

std_error_by_stratum <- tapply(cardekho$selling_price,cardekho$transmission, function(x) sd(x))

print(std_error_by_stratum)


mean_by_stratum <- tapply(cardekho$selling_price,cardekho$transmission, mean)

print(mean_by_stratum)



#Since the costs are not available we will be using the propotional allocation.


#POPULATION PARAMATERS


pop_mean_selling_price=mean(cardekho$selling_price)

pop_mean_selling_price



pop_mean_max_power=mean(cardekho$max_power)

pop_mean_max_power


pop_total_selling_price=sum(cardekho$selling_price)
```

```
pop_total_selling_price

pop_total_max_power=sum(cardekho$max_power)
pop_total_max_power

pop_proportion=table(fuel)/length(fuel)
pop_proportion

pop_proportion=table(seller_type)/length(seller_type)
pop_proportion

pop_proportion=table(owner)/length(owner)
pop_proportion

#stratified
#The Auxiliary Variable used is max_power

library(ggplot2)
ggplot(cardekho, aes(x =max_power, y = selling_price)) +geom_point() + labs(title =
"Scatterplot of Mileage vs. Selling Price",
                                             x = "Mileage",
                                             y = "Selling Price")
cardekho <- na.omit(cardekho)
correlation <- cor(cardekho$max_power, cardekho$selling_price)
print(correlation)

#For Sample 01
set.seed(1005)
Sample_size = rsampcalc(nrow(cardekho),e=3,ci=95,0.5)
```

```
Sample_size

#sample size for each strata
strata_size=ssampcalc(cardekho,Sample_size,transmission)
strata_size

#getting stratified samples
strat_samp1=ssamp(cardekho,Sample_size,transmission)
strat_samp1

transmission_freq <- table(strat_samp1$transmission)
print(transmission_freq)



attach(strat_samp1)

# sample weight for Automatic = 1050/122 = 9
# sample weight for Manual = 7078/822 = 9

strat_samp1$w=9

#define survey design object
strat_design1=svydesign(id=~1,strata =transmission,data = strat_samp1,weights=~w)
summary(strat_design1)

#Sample mean for selling price

sample_1_mean_for_selling_price=svymean(~selling_price,strat_design1)
```

```
sample_1_mean_for_selling_price

#Sample mean for max_power

sample_1_mean_for_max_power=svymean(~max_power,strat_design1)
sample_1_mean_for_max_power

#Sample total for selling price

sample_1_total_for_selling_price=svytotal(~selling_price,strat_design1)
sample_1_total_for_selling_price

#Sample total for max_power

sample_1_total_for_max_power=svytotal(~max_power,strat_design1)
sample_1_total_for_max_power

# sample proportion for Fuel
sample_prop=svymean(~fuel,strat_design1)
sample_prop

# sample proportion for seller_type
sample_prop=svymean(~seller_type,strat_design1)
sample_prop

# sample proportion for owner
sample_prop=svymean(~owner,strat_design1)
sample_prop
```

```
#Regression estimation
#fitting linear regression model
plot(strat_samp1$selling_price,strat_samp1$max_power)
cor(strat_samp1$selling_price,strat_samp1$max_power)
lm1=lm(strat_samp1$selling_price~strat_samp1$max_power,strat_samp1)
lm1
summary(lm1)
anova(lm1)

# mean for max_power in population = 91.58737
#Calculating the  expected mean for selling_price using regression model
mean_selling_price= -880919.6+ 16679.9*91.58737

mean_selling_price
Estimated_Total=7906*mean_selling_price
Estimated_Total

#Ratio Estimation(Combined)
r1=svyratio(~strat_samp1$selling_price,~strat_samp1$max_power, strat_design1)
predict(r1,total=pop_total_max_power)

#Estimated Total = 5119380254
Estimated_mean = 5119380254/7906
Estimated_mean
SE(r1)
```

```
#For Sample 02
set.seed(2106)
Sample_size2 = rsampcalc(nrow(cardekho),e=3,ci=95,0.5)
Sample_size2

#sample size for each strata
strata_size2=ssampcalc(cardekho,Sample_size2,transmission)
strata_size2

#getting stratified samples
strat_samp2=ssamp(cardekho,Sample_size2,transmission)
strat_samp2

transmission_freq <- table(strat_samp2$transmission)
print(transmission_freq)



attach(strat_samp2)

# sample weight for Automatic = 1041/124 = 8
# sample weight for Manual = 6865/817 = 8

strat_samp2$W=8

#define survey design object
strat_design2=svydesign(id=~1,strata =transmission,data = strat_samp2,weights=~W)
```

```
summary(strat_design2)

#Sample mean for selling price

sample_2_mean_for_selling_price=svymean(~selling_price,strat_design2)
sample_2_mean_for_selling_price

#Sample mean for max_power

sample_2_mean_for_max_power=svymean(~max_power,strat_design2)
sample_2_mean_for_max_power

#Sample total for selling price

sample_2_total_for_selling_price=svytotal(~selling_price,strat_design2)
sample_2_total_for_selling_price

#Sample total for max_power

sample_2_total_for_max_power=svytotal(~max_power,strat_design2)
sample_2_total_for_max_power

# sample proportion for Fuel
sample_prop=svymean(~fuel,strat_design2)
sample_prop

# sample proportion for seller_type
sample_prop=svymean(~seller_type,strat_design2)
```

```
sample_prop

# sample proportion for owner
sample_prop=svymean(~owner,strat_design2)
sample_prop

#Regression estimation
#fitting linear regression model
plot(strat_samp2$selling_price,strat_samp2$max_power)
cor(strat_samp2$selling_price,strat_samp2$max_power)
lm2=lm(strat_samp2$selling_price~strat_samp2$max_power,strat_samp2)
lm2
summary(lm2)
anova(lm2)

# mean for max_power in population = 91.58737
#Calculating the  expected mean for selling_price using regression model
mean_selling_price= -929995.7+ 17105.0 *91.58737

mean_selling_price
Estimated_Total=7906*mean_selling_price
Estimated_Total

#Ratio Estimation(Combined)
r2=svyratio(~strat_samp2$selling_price,~strat_samp2$max_power, strat_design2)
predict(r2,total=pop_total_max_power)

#Estimated Total = 5093256387
```

```
Estimated_mean = 5093256387/7906

Estimated_mean

SE(r2)
```

```
#Graphical Analysis Using Sample 01

svyhist(~selling_price,strat_design1, main="Histogram of Selling Price",col="purple",probability = FALSE)


svyboxplot(~selling_price~fuel,strat_design1,main="Boxplot of Selling Price with Fuel Type ", col="brown" )


svyboxplot(~selling_price~seller_type,strat_design1,main="Boxplot of Selling Price with Seller Type ", col="brown" )


svyboxplot(~selling_price~owner,strat_design1,main="Boxplot of Selling Price with Owner ", col="brown" )


svyplot(max_power~selling_price, design=strat_design1, style="bubble")
```

Two Stage Cluster Sampling

```
library(stringr)
library(survey)
library(tidyverse)

cardekho_new <- cardekho_new %>%
  mutate(ID = row_number())

cardekho_new$Car_model <- word(cardekho_new$name,1)
length(unique(cardekho_new$Car_model))

unique(cardekho_new$Car_model)

#select a random sample of clusters
set.seed(159)
sample_models <- sample(unique(cardekho_new$Car_model),size = 10,replace = F)
table(sample_models)

filtered_data <- cardekho_new[cardekho_new$Car_model %in% sample_models,]
table(filtered_data$Car_model)
unique(filtered_data$Car_model)



#create cluster sample
clus <- function(data) {

  cluster_data <- list()
  grouped_data <- split(data, data$Car_model)

  for (group in grouped_data) {

    if (nrow(group) < 50) {
      sampled_group <- group

    }
    else {
      sampled_group <- sample_n(group, size = 50, replace = FALSE)
    }

    cluster_data <- append(cluster_data, list(sampled_group))
  }

  result <- bind_rows(cluster_data)

  return(result)
}

cluster_sample <- clus(filtered_data)

#create fpc columns
cluster_sample$fpc1 <- 31

ssu_counts <- filtered_data %>%
  group_by(Car_model) %>%
  summarize(fpc2 = n())
```

```
cluster_sample <- cluster_sample %>%
  left_join(ssu_counts, by = "Car_model")

#calculate smapling weights
psu_weight <- nrow(filtered_data) / length(unique(filtered_data$Car_model))

cluster_sample <- cluster_sample %>%
  group_by(Car_model) %>%
  mutate(
    N_SSU = nrow(filtered_data[filtered_data$Car_model == Car_model,]),
    n_SSU = n(),
    w_SSU = N_SSU / n_SSU,

    w_Total = psu_weight * w_SSU
  ) %>%
  ungroup() %>%
  select(-N_SSU, -n_SSU, -w_SSU)



add_sample_weight <- function(data, new_col_name) {

  w_psu <- data$fpc1 / 10
  w_ssu <- ifelse(data$fpc2 <= 50, 1, data$fpc2 / 50)
  w_total <- w_psu * w_ssu

  data[[new_col_name]] <- w_total
  return(data)
}


cluster_sample <- add_sample_weight(cluster_sample, "total_weight")


cluster_design <- svydesign(data = cluster_sample,ids = ~Car_model + ID, weights =
~total_weight,fpc = ~fpc1 + fpc2 )
summary(cluster_design)


svymean(~selling_price,design = cluster_design)
mean(cardekho_new$selling_price)


svytotal(~selling_price,design = cluster_design)
sum(cardekho_new$selling_price)


svymean(~transmission,design = cluster_design)
prop.table(table(cardekho_new$transmission))


svymean(~owner,design = cluster_design)
prop.table(table(cardekho_new$owner))


#select the second random sample of clusters
set.seed(161)
sample_models1 <- sample(unique(cardekho_new$Car_model),size = 10,replace = F)
table(sample_models1)


filtered_data1 <- cardekho_new[cardekho_new$Car_model %in% sample_models1,]
table(filtered_data1$Car_model)
unique(filtered_data1$Car_model)
```

```
#create second cluster sample
clus1 <- function(data) {

  cluster_data1 <- list()

  grouped_data <- split(data, data$Car_model)

  for (group in grouped_data) {

    if (nrow(group) < 50) {
      sampled_group <- group

    }
    else {
      sampled_group <- sample_n(group, size = 50, replace = FALSE)
    }

    cluster_data1 <- append(cluster_data, list(sampled_group))
  }

  result <- bind_rows(cluster_data1)

  return(result)
}

cluster_sample1 <- clus(filtered_data1)
```

```
#create fpc columns
cluster_sample1$fpc1 <- 31

ssu_counts <- filtered_data1 %>%
  group_by(Car_model) %>%
  summarize(fpc2 = n())

cluster_sample1 <- cluster_sample1 %>%
  left_join(ssu_counts, by = "Car_model")

psu_weight <- nrow(filtered_data1) / length(unique(filtered_data1$Car_model))

cluster_sample1 <- cluster_sample1 %>%
  group_by(Car_model) %>%
  mutate(
    N_SSU = nrow(filtered_data1[filtered_data1$Car_model == Car_model,]),
    n_SSU = n(),
    w_SSU = N_SSU / n_SSU,

    w_Total = psu_weight * w_SSU
  ) %>%
  ungroup() %>%
  select(-N_SSU, -n_SSU, -w_SSU)

function(data){
  w_psu <- data$fpc1 / 10
  if (data$fpc2 <= 50) {
    w_ssu <- 1
  }
  else{
    w_ssu <- length(data$fpc2) / 50
  }
  w_total <- w_psu*w_ssu
}

add_sample_weight1 <- function(data, new_col_name) {
  # Ensure the new column name is valid
  if (!is.character(new_col_name) || nchar(new_col_name) == 0) {
    stop("Please provide a valid name for the new column.")
  }

  # Calculate weights
  w_psu <- data$fpc1 / 10

  w_ssu <- ifelse(data$fpc2 <= 50, 1, data$fpc2 / 50)

  # Calculate total weight
  w_total <- w_psu * w_ssu

  # Add the new column to the dataframe
  data[[new_col_name]] <- w_total

  return(data)
}

cluster_sample1 <- add_sample_weight1(cluster_sample1, "total_weight")
```

```
cluster_design1 <- svydesign(data = cluster_sample1,ids = ~Car_model + ID, weights = ~total_weight,fpc = ~fpc1 + fpc2 )

summary(cluster_design1)


svymean(~selling_price,design = cluster_design1)

mean(cardekho_new$selling_price)


svytotal(~selling_price,design = cluster_design1)


svymean(~transmission,design = cluster_design1)

prop.table(table(cardekho_new$transmission))


svymean(~owner,design = cluster_design1)


svyratio(~max_power,~selling_price,design = cluster_design)

svyratio(~max_power,~selling_price,design = cluster_design1)


regression <- lm(cluster_sample$selling_price~cluster_sample$max_power)

coef(regression)


regression1 <- lm(cluster_sample1$selling_price~cluster_sample1$max_power)

coef(regression1)


svyhist(~selling_price,design = cluster_design,main = 'Histogram of selling price (sample 1)',col = 'orange',breaks = 30)

svyboxplot(selling_price~transmission,design = cluster_design,col = 'lightblue',main = 'Boxplot of selling price with fuel type (sample 1)')

svyboxplot(selling_price~owner,design = cluster_design,col = 'lightblue', main = 'Boxplot of selling price with owner type (sample 1)')
```

```
svyhist(~selling_price,design = cluster_design1,main = 'Histogram of selling price (sample 2)',col = 'orange',breaks = 40)

svyboxplot(selling_price~transmission,design = cluster_design1,col = 'lightblue',main = 'Boxplot of selling price with fuel type (sample 2)')

svyboxplot(selling_price~owner,design = cluster_design1,col = 'lightblue', main = 'Boxplot of selling price with owner type (sample 2)')

boxplot(cardekho_new$selling_price~cardekho_new$owner)
```