



University
of Windsor
Faculty of Science

COMP 8157 Advanced Database Topics

University of Windsor, School of Computer Science

Lab 1

CONFIDENTIALITY AGREEMENT & STATEMENT OF HONESTY

I, **Firas Hussain Mohammed** verify that the submitted work is my own, original work, and that I did not use Generative AI tools (e.g., ChatGPT, Bard) to produce this lab report. I confirm knowing that a mark of 0 may be assigned for sharing or copying this work.

Firas Hussain Mohammed

Student Signature

Firas Hussain Mohammed

Student Name

110173878

Student I.D.

Student Signature

Student Name

Student I.D.

Part 1: Data Exploration

1. Import the Vehicle dataset, summarize it and explain the output?

The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled "Lab 1.R" containing the following code:

```
1 install.packages("stats")
2 install.packages("dplyr", dependencies = TRUE)
3 install.packages("ggplot2")
4 install.packages("ggsignif")
5 install.packages(c("cluster", "rattle", "NbClust"))
6 install.packages("corrplot")
7 install.packages("dbSCAN")
8 library(dbscan)
9 library(corrplot)
10 library(stats)
11 library(ggplot2)
12 library(ggfortify)
13
14 #SOLUTION For Question 1
15 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/vehicle.csv")
16 summary(org_vehicle_dataset)
17
18
```

In the bottom-left pane, the "Console" tab is active, showing the output of the `summary` command:

```
> summary(org_vehicle_dataset)
   Car_Name      Year      Selling_Price Present_Price    Kms_Driven
Length:301   Min.   :2003   Min.   : 0.100   Min.   : 0.320   Min.   : 1000
Class:character 1st Qu.:2012   1st Qu.: 0.100   1st Qu.: 0.320   1st Qu.:15000
Mode :character Median:2014   Median: 3.600   Median: 6.400   Median:32000
               Mean   :2014   Mean   : 4.661   Mean   : 7.628   Mean   :36947
               3rd Qu.:2016   3rd Qu.: 6.000   3rd Qu.: 9.900   3rd Qu.:48767
               Max.  :2018   Max.  :35.000   Max.  :92.600   Max.  :500000
Fuel_Type     Seller_Type Transmission Owner
Length:301   Length:301   Length:301   Min.   :0.00000
Class:character Class:character Class:character 1st Qu.:0.00000
Mode :character Mode :character Mode :character Median:0.04319
                           Mean   :0.04319
                           3rd Qu.:0.00000
                           Max.  :3.00000
```

The right side of the interface shows the "Environment" and "Data" panes. The "Data" pane displays the dataset "org_vehicle_dataset" with 301 observations and 9 variables.

- The output generated showed us complete details of our dataset which includes all column names type of data such as character or numeric, number of records.
- For numeric records it displayed minimum value, 1st Quartile that is 25% of data, Median which is also 2nd Quartile (50% o data), mean, 3rd Quartile that is 75% of data and maximum value of record in our column.
- Our dataset has 302 records and 9 columns.

2. Show the structure and dimension of the dataset and explain it?

The screenshot shows the RStudio interface. In the top-left pane, there is an R script titled "Lab 1.R" containing the following code:

```
1 install.packages("stats")
2 install.packages("dplyr", dependencies = TRUE)
3 install.packages("ggplot2")
4 install.packages("ggsignif")
5 install.packages(c("cluster", "rattle", "NbClust"))
6 install.packages("corrplot")
7 install.packages("dbSCAN")
8 library(dbscan)
9 library(corrplot)
10 library(stats)
11 library(ggplot2)
12 library(ggfortify)
13
14 #SOLUTION For Question 1
15 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/vehicle.csv")
16 summary(org_vehicle_dataset)
17
18 #SOLUTION For Question 2
19 str(org_vehicle_dataset)
20 dim(org_vehicle_dataset)
21
```

In the bottom-left pane, the "Console" tab is active, showing the output of the `str` and `dim` commands:

```
> #SOLUTION For Question 2
> str(org_vehicle_dataset)
'data.frame': 301 obs. of 9 variables:
 $ Car_Name : chr "Fiat207S" "Fiat500" "Fiat500" "Fiat500" ...
 $ Year     : num 2008 2011 2011 2011 2011 2011 2011 2011 2011 ...
 $ Selling_Price: num 3.35 4.75 7.25 2.85 4.6 9.25 6.75 6.5 8.75 7.45 ...
 $ Present_Price: num 5.59 9.54 9.85 4.15 6.87 9.83 8.12 8.61 8.89 8.92 ...
 $ Kms_Driven: num 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 ...
 $ Fuel_Type   : chr "Petrol" "Diesel" "Petrol" "Petrol" ...
 $ Seller_Type : chr "Dealer" "Dealer" "Dealer" "Dealer" ...
 $ Transmission: chr "Manual" "Manual" "Manual" "Manual" ...
 $ Owner      : int 0 0 0 0 0 0 0 0 0 0 ...
> dim(org_vehicle_dataset)
[1] 301 9
```

- Through the structure command we got information like type of data each column has and also displayed a snapshot of the records.
 - Dimension on the other hand returned only the number of records in our dataset and number of columns.
3. Show the column names of the Vehicle dataset and the first 3 rows and the last 6 rows of it?

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project (N)
- Code Editor:** Lab 1.R* showing the R script for Question 3.
- Environment Tab:** Shows the dataset `org_vehicle_dataset` with 301 observations and 9 variables.
- Console Tab:** Displays the R session output for the code run.

```

19 str(org_vehicle_dataset)
20 dim(org_vehicle_dataset)
21
22 #Solution For Question 3
23 names(org_vehicle_dataset)
24 head(org_vehicle_dataset,3)
25 tail(org_vehicle_dataset,6)
26
261 [Top Level] R Script
262
263 Console Terminal Background Jobs
264 > R - R4.2.2 - / - 
265 > #Solution For Question 3
266 > names(org_vehicle_dataset)
267 [1] "Car_Name"      "Year"           "Selling_Price"  "Present_Price" "Kms_Driven"
268 [6] "Fuel_Type"     "Seller_Type"   "Transmission"  "Owner"
269 > head(org_vehicle_dataset,3)
270   Car_Name Year Selling_Price Present_Price Kms_Driven Fuel_Type Seller_Type Transmission
271   ritz    2014       3.35          5.59     27000    Petrol   Dealer    Manual
272   sx4    2013        4.75          9.54     43000    Diesel   Dealer    Manual
273   ciaz   2017       7.25          9.85     6900     Petrol   Dealer    Manual
274   Owner
275   1      0
276   2      0
277   3      0
278 > tail(org_vehicle_dataset,6)
279   Car_Name Year Selling_Price Present_Price Kms_Driven Fuel_Type Seller_Type Transmission
280   city   2015       8.55          13.09    60076    Diesel   Dealer    Manual
281   city   2016       9.50          11.60    33988    Diesel   Dealer    Manual
282   brio  2015        4.00          5.90     60000    Petrol   Dealer    Manual
283   city   2009       3.35          11.00    87934    Petrol   Dealer    Manual
284   city   2017      11.50          12.50    9000     Diesel   Dealer    Manual
285   brio  2016       5.30          5.90     5464     Petrol   Dealer    Manual
286   Owner
287   0
288   0
289   0
290   0
291   0
292 > |

```

4. Show the average Kms_Driven for each type of car (Car_Name) in the dataset.

The screenshot shows the RStudio interface with two panes. The left pane contains R code for various questions, and the right pane shows the environment and the resulting data frames.

Code (Left Pane):

```
13 #Solution For Question 1
14 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/vehicle.csv")
15 summary(org_vehicle_dataset)
16
17 #solution For Question 2
18 str(org_vehicle_dataset)
19 dim(org_vehicle_dataset)
20
21 #solution For Question 3
22 names(org_vehicle_dataset)
23 head(org_vehicle_dataset, 3)
24 tail(org_vehicle_dataset, 6)
25
26 #solution For Question 4
27 avg <- aggregate(Kms_Driven ~ Car_Name, data = org_vehicle_dataset, mean)
28 print(avg)
29
30
```

Environment (Right Pane):

Object	Type	Value
avg	98 obs. of 2 variables	
org_vehicle_dataset	301 obs. of 9 variables	

Console (Bottom Left):

```
R 4.4.2 --n
> #Solution For Question 4
> avg <- aggregate(Kms_Driven ~ Car_Name, data = org_vehicle_dataset, mean)
> print(avg)
```

Car_Name	Kms_Driven
Bajaj Pulsar NS 200	25000.000
Bajaj Avenger 150 street	20000.000
Bajaj Avenger 220	1766.667
Bajaj Avenger 220 dtsi	21800.000
Bajaj Avenger Street 220	24000.000
Bajaj Discover 125	20000.000
Bajaj Dominar 400	12000.000
Bajaj Pulsar NS 200	25000.000

5. What is the average Selling_Price of the cars in each year?

The screenshot shows the RStudio interface with the following components:

- Script Editor (Lab 1.R*):** Displays R code for various solutions to questions 2 through 18. The code includes data manipulation (e.g., `str`, `dim`), aggregate functions (e.g., `aggregate`), and printing results (e.g., `print`).
- Global Environment:** Shows the objects available in the current environment:
 - avg: 98 obs. of 2 variables
 - avg_price_per_year: 16 obs. of 2 variables
 - org_vehicle_dataset: 301 obs. of 9 variables
- Console:** Displays the output of the R code run in the script editor, specifically the results of the aggregate function for question 5.

Year	Selling_Price	
1	2003	1.300000
2	2004	1.500000
3	2005	2.487500
4	2006	1.437500
5	2007	0.160000
6	2008	1.002857
7	2009	2.816667
8	2010	5.262667
9	2011	2.842333
10	2012	3.841304
11	2013	3.540909
12	2014	4.762105
13	2015	5.927049
14	2016	5.213200
15	2017	6.209143
16	2018	9.250000

6. Show the unique combinations of Car_Name, Fuel_Type, Seller_Type, and Transmission in the Vehicle dataset.

```

33 print(avg_price_per_year)
34
35 #Solution For Question 6
36 unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")])
37 print(unique_records)
38

```

Console Terminal Background Jobs

```

> #Solution For Question 6
> unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")])
> print(unique_records)
   Car_Name Fuel_Type Seller_Type Transmission
1      ritz    Petrol     Dealer      Manual
2      sx4     Diesel     Dealer      Manual
3      ciaz    Petrol     Dealer      Manual
4      wagon_r Petrol     Dealer      Manual
5      swift   Diesel     Dealer      Manual
6 vitara_brezza Diesel     Dealer      Manual
7      ciaz   CNG       Dealer      Manual
8      s_cross Diesel     Dealer      Manual
9      ciaz   Diesel     Dealer      Manual
11 alto_800 Petrol     Dealer      Manual
13      ciaz    Petrol     Dealer  Automatic
14      erigia  Petrol     Dealer      Manual
15      dzire  Diesel     Dealer      Manual
16      erigia  Diesel     Dealer      Manual
19      wagon_r CNG       Dealer      Manual
20      sx4    Petrol     Dealer      Manual
21 alto_k10 Petrol     Dealer      Manual
22      ignis  Petrol     Dealer      Manual
23      sx4    Petrol     Dealer  Automatic
26      swift  Petrol     Dealer      Manual
28      swift  Petrol Individual  Manual
31      ritz   Diesel     Dealer      Manual
35      dzire  Diesel     Dealer      Manual
36      sx4   CNG       Dealer      Manual
38      800    Petrol Individual  Manual
41      baleno Petrol     Dealer  Automatic

```

Environment History Connections Tutorial

Data

- avg 98 obs. of 2 variables
- avg_price_per_year 16 obs. of 2 variables
- org_vehicle_dataset 301 obs. of 9 variables
- unique_records 135 obs. of 4 variables

Files Plots Packages Help Viewer Presentation

7. What are the different combinations of Car_Name, Fuel_Type, Seller_Type, and Transmission in the Vehicle dataset, and how many times does it occur? (Display all such in both ascending and descending orders)

```

35 #Solution For Question 6
36 unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")])
37 print(unique_records)
38
39 #Solution For Question 7
40 multiple_combinations <- as.data.frame(table(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")]))
41 ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
42 descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
43 print(ascending)
44 print(descending)
45

```

Console Terminal Background Jobs

```

> #Solution For Question 7
> multiple_combinations <- as.data.frame(table(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")]))
> ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
> descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
> print(ascending)
   Car_Name Fuel_Type Seller_Type Transmission Freq
1      800      CNG     Dealer  Automatic  0
2      Activa_3g  CNG     Dealer  Automatic  0
3      Activa_4g  CNG     Dealer  Automatic  0
4      alto_800  CNG     Dealer  Automatic  0
5      alto_k10  CNG     Dealer  Automatic  0
6      amaze     CNG     Dealer  Automatic  0
7      Bajaj_ct_100 CNG     Dealer  Automatic  0
8      Bajaj_Avenger_150 CNG     Dealer  Automatic  0
9      Bajaj_Avenger_150_street CNG     Dealer  Automatic  0
10     Bajaj_Avenger_220 CNG     Dealer  Automatic  0
11     Bajaj_Avenger_220_dxi CNG     Dealer  Automatic  0
12     Bajaj_Avenger_Street_220 CNG     Dealer  Automatic  0
13     Bajaj_Discover_100 CNG     Dealer  Automatic  0
14     Bajaj_Discover_125 CNG     Dealer  Automatic  0
15     Bajaj_Dominar_400 CNG     Dealer  Automatic  0
16     Bajaj_Pulsar_NS_200 CNG     Dealer  Automatic  0
17     Bajaj_Pulsar_135_LS CNG     Dealer  Automatic  0
18     Bajaj_Pulsar_150 CNG     Dealer  Automatic  0
19     Bajaj_Pulsar_220_F CNG     Dealer  Automatic  0

```

Environment History Connections Tutorial

Data

- ascending 1176 obs. of 5 variables
- avg 98 obs. of 2 variables
- avg_price_per_year 16 obs. of 2 variables
- descending 1176 obs. of 5 variables
- multiple_combinations 1176 obs. of 5 variables
- org_vehicle_dataset 301 obs. of 9 variables
- unique_records 135 obs. of 4 variables

Files Plots Packages Help Viewer Presentation

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows objects in the global environment: ascending (1176 obs. of 5 variables), avg (98 obs. of 2 variables), avg_price_per_year (16 obs. of 2 variables), descending (1176 obs. of 5 variables), multiple_combinations (1176 obs. of 5 variables), org_vehicle_dataset (301 obs. of 9 variables), and unique_records (135 obs. of 4 variables).
- Console:** Displays R script code for Question 7, which prints the descending sorted rows of the dataset.
- Output:** Shows the first 10 rows of the dataset, which contains columns: Car_Name, Fuel_Type, Seller_Type, Transmission, and Freq.

```

35 #Solution For Question 6
36 unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")])
37 print(unique_records)
38
39 #Solution For Question 7
40 multiple_combinations <- as.data.frame(table(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")]))
41 ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
42 descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
43 print(ascending)
44 print(descending)
45
46 #Solution For Question 8
47 anyNA(org_vehicle_dataset)
48
49 #Solution For Question 9
50
51 #Solution For Question 10
52
53 Honda Activa 4G CNG Dealer Automatic 0
54 Honda CB Hornet 160R CNG Dealer Automatic 0
55 Honda CB Shine CNG Dealer Automatic 0
56 Honda CB Trigger CNG Dealer Automatic 0
57 Honda CB twister CNG Dealer Automatic 0
58 Honda CB Unicorn CNG Dealer Automatic 0
59 Honda CBR 150 CNG Dealer Automatic 0
60 Honda Dream Yuga CNG Dealer Automatic 0
61 Honda Karizma CNG Dealer Automatic 0
62 Hyosung GT250R CNG Dealer Automatic 0
63 i10 CNG Dealer Automatic 0
64 i20 CNG Dealer Automatic 0
65 ignis CNG Dealer Automatic 0
[ reached 'max' / getoption("max.print") -- omitted 976 rows ]
> #Solution For Question 8
> anyNA(org_vehicle_dataset)
[1] FALSE
>

```

Part 2: Data Pre-Processing

8. Find if there are any missing values in the Vehicle dataset

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows objects in the global environment: ascending (1176 obs. of 5 variables), avg (98 obs. of 2 variables), avg_price_per_year (16 obs. of 2 variables), descending (1176 obs. of 5 variables), multiple_combinations (1176 obs. of 5 variables), org_vehicle_dataset (301 obs. of 9 variables), and unique_records (135 obs. of 4 variables).
- Console:** Displays R script code for Question 8, which checks for any missing values (NA) in the org_vehicle_dataset.
- Output:** Shows the result of the anyNA function, which returns [1] FALSE, indicating no missing values.

```

31 #Solution For Question 5
32 avg_price_per_year <- aggregate(Selling_Price ~ Year, data = org_vehicle_dataset, mean)
33 print(avg_price_per_year)
34
35 #Solution For Question 6
36 unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")])
37 print(unique_records)
38
39 #Solution For Question 7
40 multiple_combinations <- as.data.frame(table(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission")]))
41 ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
42 descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
43 print(ascending)
44 print(descending)
45
46 #Solution For Question 8
47 anyNA(org_vehicle_dataset)
48
49 #Solution For Question 9
50
51 #Solution For Question 10
52
53 Honda Activa 4G CNG Dealer Automatic 0
54 Honda CB Hornet 160R CNG Dealer Automatic 0
55 Honda CB Shine CNG Dealer Automatic 0
56 Honda CB Trigger CNG Dealer Automatic 0
57 Honda CB twister CNG Dealer Automatic 0
58 Honda CB Unicorn CNG Dealer Automatic 0
59 Honda CBR 150 CNG Dealer Automatic 0
60 Honda Dream Yuga CNG Dealer Automatic 0
61 Honda Karizma CNG Dealer Automatic 0
62 Hyosung GT250R CNG Dealer Automatic 0
63 i10 CNG Dealer Automatic 0
64 i20 CNG Dealer Automatic 0
65 ignis CNG Dealer Automatic 0
[ reached 'max' / getoption("max.print") -- omitted 976 rows ]
> #Solution For Question 8
> anyNA(org_vehicle_dataset)
[1] FALSE
>

```

9. Find which columns contain missing values in the vehicles dataset. What are the total missing values for each column?

```

34 unique_records <- unique(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission", "Year")])
35 print(unique_records)
36
37 #Solution For Question 7
38 multiple_combinations <- ae.data.frame(table(org_vehicle_dataset[,c("Car_Name", "Fuel_Type", "Seller_Type", "Transmission", "Year")]))
39 ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
40 descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
41 print(ascending)
42 print(descending)
43
44 #Solution For Question 8
45 anyNA(org_vehicle_dataset)
46
47 #Solution For Question 9
48 colSums(is.na(org_vehicle_dataset))
49
50
51

```

```

> colSums(is.na(org_vehicle_dataset))
  Car_Name      Year Selling_Price Present_Price Kms_Driven Fuel_Type
0          0        0             0             0         0           0
  Seller_Type Transmission       Owner
0          0        0             0

```

10. Replace the missing values in the dataset with the most repeated value of that field. Check if the missing values were replaced successfully

```

41 ascending <- multiple_combinations[order(multiple_combinations$Freq), ]
42 descending <- multiple_combinations[order(-multiple_combinations$Freq), ]
43 print(ascending)
44 print(descending)
45
46 #Solution For Question 8
47 anyNA(org_vehicle_dataset)
48
49 #Solution For Question 9
50 colSums(is.na(org_vehicle_dataset))
51
52 #Solution For Question 10
53 add_values <- function(col)
54 { col[is.na(col)] <- names(sort(table(col), decreasing = TRUE)[1])
55   return(col)
56 }
57
58 org_vehicle_dataset[] <- lapply(org_vehicle_dataset, add_values)
59 colSums(is.na(org_vehicle_dataset))
60
61
62

```

```

> colSums(is.na(org_vehicle_dataset))
  Car_Name      Year Selling_Price Present_Price Kms_Driven Fuel_Type
0          0        0             0             0         0           0
  Seller_Type Transmission       Owner
0          0        0             0

```

11. Find if the dataset has duplicate rows. Remove them, if exist?

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Contains R code for Question 11, which prints a dataset of car information.
- Console:** Displays the output of the R code, showing a table of car data with columns: Car_Name, Year, Selling_Price, Present_Price, Kms_Driven, Fuel_Type.
- Environment:** Shows the global environment with objects like ascending, avg, avg_price_per_year, descending, multiple_combinations, org_vehicle_dataset, unique_records, and add_values.

Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type
rits	2014	3.35	5.59	27000	Petrol
sx4	2013	4.75	9.54	42000	Diesel
ciaz	2017	7.25	9.85	6900	Petrol
wagon r	2011	2.85	4.15	5200	Petrol
swift	2014	4.6	6.87	42450	Diesel
vitara brezza	2018	9.25	9.83	2071	Diesel
ciaz	2015	6.75	8.12	18796	CNG
s cross	2015	6.5	8.61	33429	Diesel
ciaz	2016	8.75	8.89	20273	Diesel
alto	2017	7.45	8.92	42367	Diesel
800	2017	2.85	3.6	2135	Petrol
ciaz	2015	6.85	10.38	51000	CNG
ciaz	2015	7.5	9.94	15000	Petrol
ertiga	2015	6.1	7.71	26000	Petrol

This screenshot is identical to the one above, showing the same R code, console output, and environment pane.

12. Replace the values of the following attributes:

Fuel_Type: “Petrol”: 0, “Diesel”: 1, “CNG”: 2

Seller_Type: “Dealer”: 0, “Individual”: 1

Transmission: “Manual”: 0, “Automatic”: 1

Show the conversion output of the specific attribute

RStudio

```

File Edit Code View Plots Session Build Debug Profile Tools Help
Source on Save Go to file/function Addins

Lab 1.R* R Script
62 print(org_vehicle_dataset)
63
64 #Solution For Question 12
65 org_vehicle_dataset$Fuel_Type <- as.numeric(factor(org_vehicle_dataset$Fuel_Type, levels = c('
66 org_vehicle_dataset$Seller_Type <- as.numeric(factor(org_vehicle_dataset$Seller_Type, levels =
67 org_vehicle_dataset$Transmission <- as.numeric(factor(org_vehicle_dataset$Transmission, level=
68 print(org_vehicle_dataset)
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269

```

RStudio Environment pane:

- Objects: ascending, avg, avg_price_per_year, descending, multiple_combinations, org_vehicle_dataset, unique_records.
- Functions: add_values.

RStudio Console pane:

```

97 innova 2016    20.5/   25.89   29000   1
98 corolla altis 2017  17.7   18.64   87000   0
99 corolla altis 2013  7.05   18.61   45000   0
100 fortuner 2010   9.65   20.45   50024   1
101 Royal Enfield Thunder 500 2016  1.75   1.9    3000    0
102 UM Renegade Mojave 2017   1.7   1.82   14000   0
Seller_Type Transmission Owner_Age
1          0           0      11
2          0           0      12
3          0           0       8
4          0           0      14
5          0           0      11
6          0           0       7
7          0           0      10
8          0           0      10
9          0           0       9
10         0           0      10
11         0           0       8

```

14. Create a new dataset by selecting only the columns “Car_name”, “Selling_Price”, “Present_Price”, and “Kms_Drive”. Show the output of the new dataset.

RStudio Environment pane:

- Objects: ascending, avg, avg_price_per_year, descending, multiple_combinations, org_vehicle_dataset, unique_records, updated_dataset.
- Functions: add_values.

RStudio Console pane:

```

> #Solution For Question 14
> updated_dataset <- org_vehicle_dataset[c("Car_Name", "Selling_Price", "Present_Price", "Kms_Driven")]
> print(updated_dataset)
  Car_Name Selling_Price Present_Price Kms_Driven
1   ritz        3.35        5.59     27000
2   sx4        4.75        9.54     42000
3   ciaz       7.25        9.85     69000
4   wagon r     2.85        4.15      5200
5   swift       4.6        6.87     42450
6 vitara brezza  9.25        9.83     2071
7   ciaz       6.75        8.12     18796
8   s cross      6.5        8.61     33429
9   ciaz       8.75        8.89     20273
10  ciaz       7.45        8.92     42367
11 alto 800     2.85        3.6      2135
12   ciaz       6.85        10.38    51000
13   ciaz       7.5        9.94     15000

```

15. Shuffle the rows of the Vehicle dataset randomly and show the output

```

74 #Solution For Question 14
75 updated_dataset <- org_vehicle_dataset[c("Car_Name", "Selling_Price", "Present_Price", "Kms_Driven", "Fuel_Type", "Transmission")]
76 print(updated_dataset)
77
78 #Solution For Question 15
79 shuffled_records <- org_vehicle_dataset[sample(nrow(org_vehicle_dataset)), ]
80 print(shuffled_records)
81
82

```

L reached 'max' / getOption("max.print") -- omitted 49 rows]

```

> #Solution For Question 15
> shuffled_records <- org_vehicle_dataset[sample(nrow(org_vehicle_dataset)), ]
> print(shuffled_records)
  Car_Name Year Selling_Price Present_Price Kms_Driven Fuel_Type Transmission
239 verna 2013    4.8      9.4     60241        0
87 land cruiser 2010    35      92.6    78000        1
227 grand i10 2015    5.25      5.7    24678        0
121 Bajaj Pulsar RS200 2016    1.05     1.26     5700        0
118 Royal Enfield Thunder 500 2015    1.1      1.9    14000        0
201 Bajaj Pulsar 150 2006    0.1      0.75    92233        0
133 Bajaj Avenger 160 2017    0.75     0.95     3500        0
17 erica 2015    7.25    10.79    41678        1
270 city 2015    6.7      10     18828        0
43 sx4 2008    1.95      7.15    58000        0
1 ritz 2014    3.35      5.59    27000        0
93 innova 2005    3.51      13.7    75000        0
192 Bajaj Discover 125 2012    0.2      0.57    25000        0
103 KTM RC200 2017    1.65     1.78    4000        0
231 verna 2013    6.15      9.4    45000        1
200 Honda CB Shine 2007    0.12     0.58    53000        0
157 TVS Sport 2017    0.48     0.52    15000        0
107 Hyosung GT250R 2014    1.35     3.45    16500        0
145 Bajaj Pulsar NS 200 2014    0.6      0.99    25000        0
15 dzire 2009    2.25     7.21    77427        0
7 ciaz 2015    6.75     8.12    18796        2
207 xcent 2017    5.75     7.13    12479        0
67 innova 2017    19.75    23.15    11000        0

```

Part 3: Data Visualization

16. Import the Vehicle dataset. Create a scatter plot of the Selling_Price Vs Present_Price. Colour code the points based on the Transmission (5 marks).

- Add labels, title and colour to the plot. The colour should be red for Transmission type ‘0’ and blue for ‘1’.
- Add open triangles to the plot.
- What do you understand from the output?

```

64 #Solution For Question 12
65 org_vehicle_dataset$Fuel_Type <- as.numeric(factor(org_vehicle_dataset$Fuel_Type, levels = c(0, 1)))
66 org_vehicle_dataset$Seller_Type <- as.numeric(factor(org_vehicle_dataset$Seller_Type, levels = c(0, 1)))
67 org_vehicle_dataset$Transmission <- as.numeric(factor(org_vehicle_dataset$Transmission, levels = c(0, 1)))
68 print(org_vehicle_dataset)
69
70 #Solution For Question 13
71 org_vehicle_dataset$Year <- as.numeric(org_vehicle_dataset$Year)
72 org_vehicle_dataset$Age <- 2025 - org_vehicle_dataset$Year
73 print(org_vehicle_dataset)
74
75 #Solution For Question 14
76 updated_dataset <- org_vehicle_dataset[c("Car_Name", "Selling_Price", "Present_Price", "Kms_Driven", "Fuel_Type", "Transmission")]
77 print(updated_dataset)
78
79 #Solution For Question 15
80 shuffled_records <- org_vehicle_dataset[sample(nrow(org_vehicle_dataset)), ]
81 print(shuffled_records)
82
83 #Solution For Question 16
84 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/Vehicle.csv")
85 ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = factor(Transmission)))
86
87 #Solution For Question 16
88 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/Vehicle.csv")
89 ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = factor(Transmission))) + geom_point(shape = 17, size = 3) + scale_color_manual(values = c("red", "blue")) + labs(title = "Selling Price vs Present Price", x = "Present Price", y = "Selling Price")
90 #Solution For Question 16
91 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/Vehicle.csv")
92 ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = factor(Transmission))) + geom_point(shape = 17, size = 3) + scale_color_manual(values = c("red", "blue")) + labs(title = "Selling Price vs Present Price", x = "Present Price", y = "Selling Price")
93 #Solution For Question 16
94 org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/Vehicle.csv")
95 ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = factor(Transmission))) + geom_point(shape = 17, size = 3) + scale_color_manual(values = c("red", "blue")) + labs(title = "Selling Price vs Present Price", x = "Present Price", y = "Selling Price")
96

```

Selling Price vs Present Price

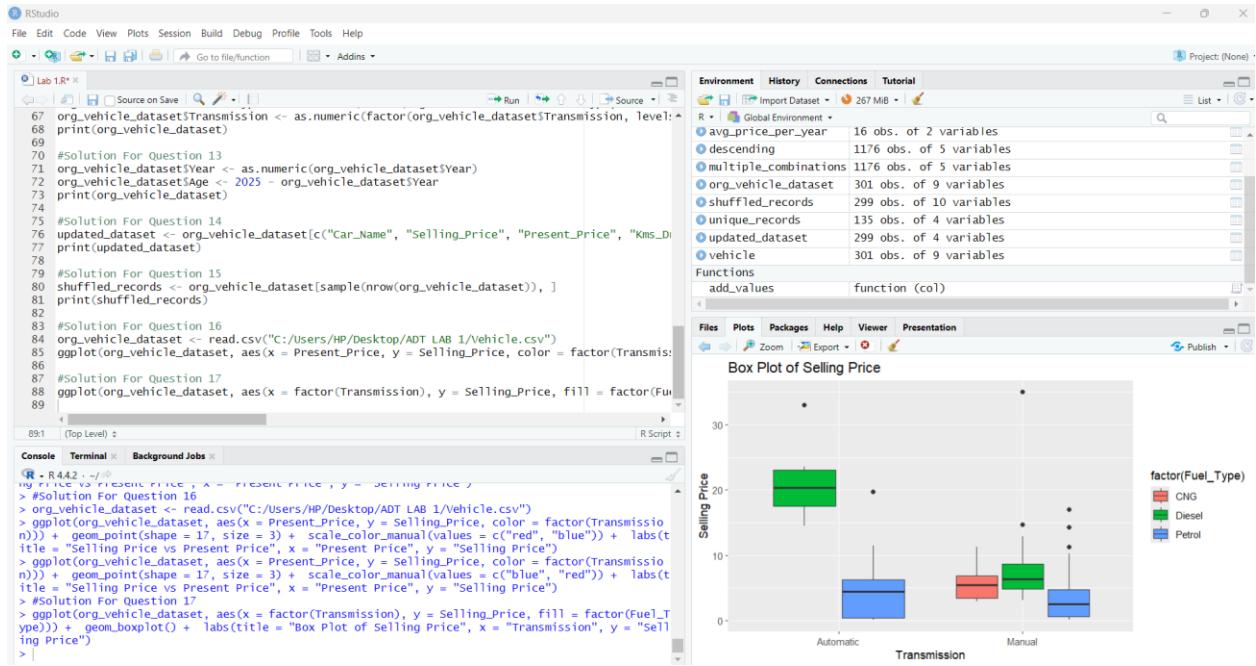
factor(Transmission)

Automatic (Blue Triangles)

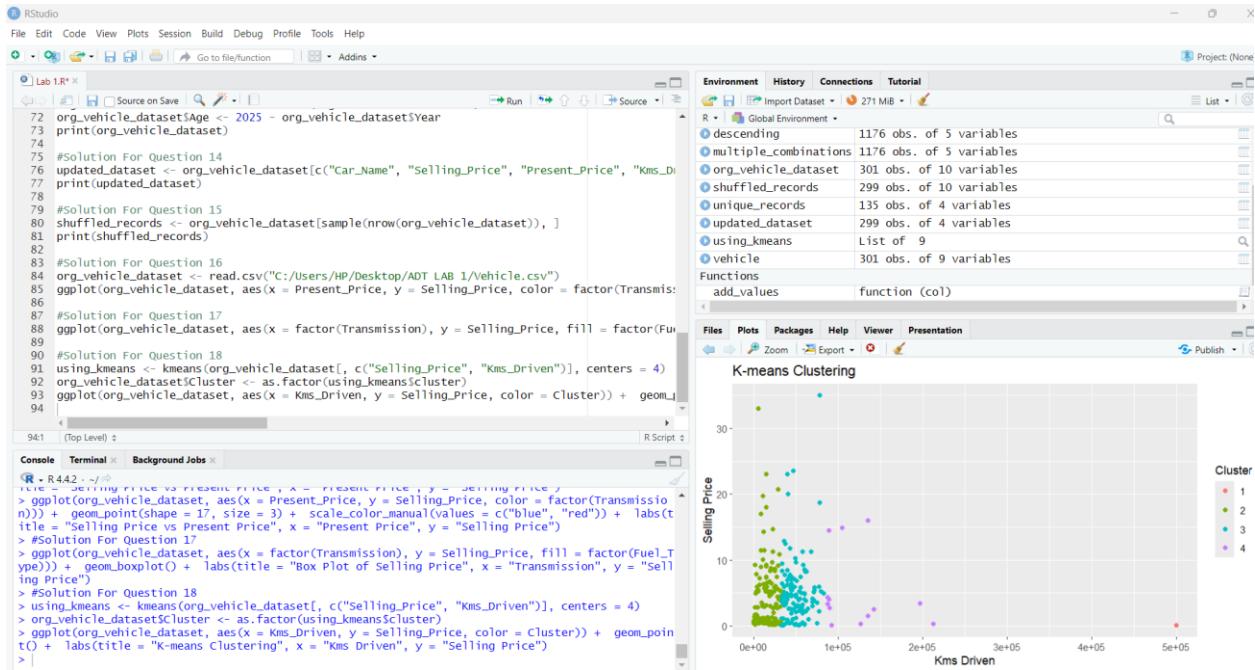
Manual (Red Triangles)

- The scatter plot indicates that manual transmission cars are within a specific range in contrast to automatic transmission.
- The blue clusters formed indicate that mostly everyone prefer manual cars as they are within a range than the automatic transmission.

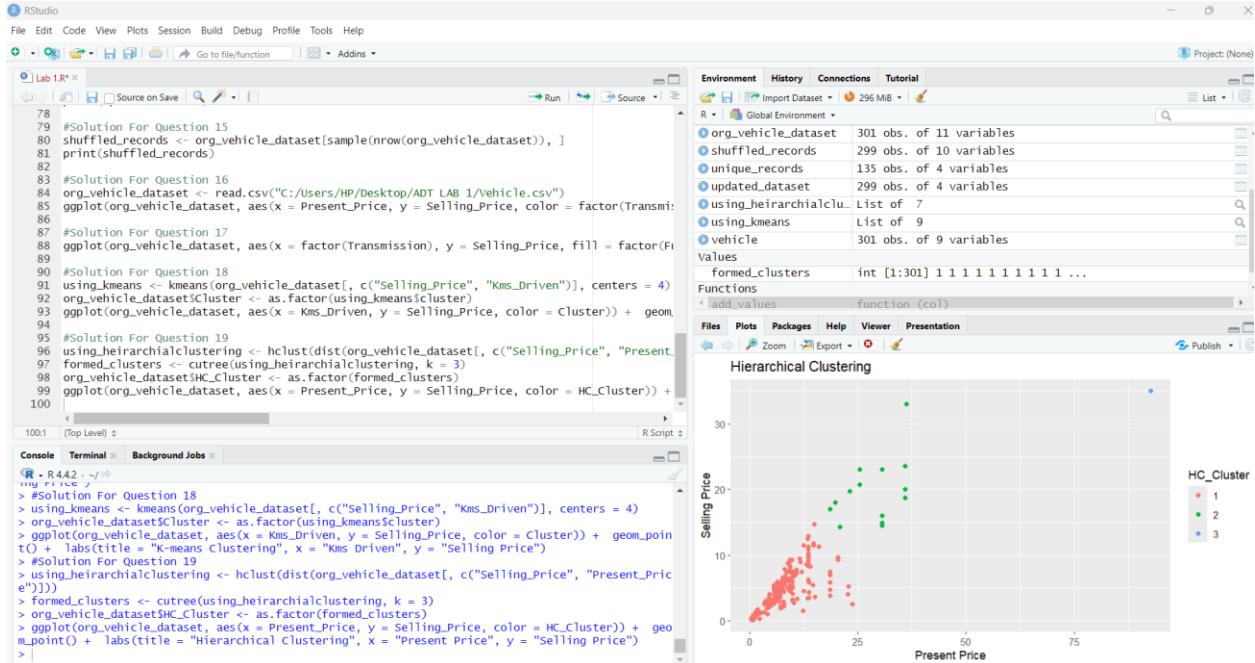
17. Create a box plot of the Selling_Price Vs Transmission and Fuel_Type



18. Create a scatter plot of the Selling_Price Vs Kms_Driven, and use k-means clustering to cluster the points into 4 clusters. Colour-code based on the cluster they belong to.



19. Create a scatter plot of the Selling_Price Vs Present_Price, and use hierarchical clustering to cluster the points into 3 clusters? Colour-code the points based on the cluster they belong to.



20. Add a new field called 'Age', and calculate it using the field 'Year'. Create a barplot for the following fields of the dataset:

- a) ‘Age’, ‘Year’, ‘Transmission’, ‘Seller_Type’, ‘Fuel_Type’ and ‘Owner’
- b) Add labels, titles, and colours to the plot.

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for clustering and generating a bar plot. The code includes sections for Question 16 through 20, involving k-means clustering and ggplot2 for data visualization.
- Console:** Shows the output of the R code, including a frequency table for the 'Owner' variable and the resulting bar plot.
- Environment:** Shows the global environment with objects like `org_vehicle_dataset`, `shuffled_records`, and `using_kmeans`.
- Plots:** A bar plot titled "Bar Plot of Owner" showing the count of entries for each owner category (0, 1, 2, 3).

```

#Solution For Question 16
org_vehicle_dataset <- read.csv("C:/Users/HP/Desktop/ADT LAB 1/vehicle.csv")
ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = factor(Transmission)))
#Solution For Question 17
ggplot(org_vehicle_dataset, aes(x = factor(Transmission), y = Selling_Price, fill = factor(Fuel_Type)))
#Solution For Question 18
using_kmeans <- kmeans(org_vehicle_dataset[, c("Selling_Price", "Kms_Driven")], centers = 4)
org_vehicle_dataset$Cluster <- as.factor(using_kmeans$cluster)
ggplot(org_vehicle_dataset, aes(x = Kms_Driven, y = Selling_Price, color = Cluster)) + geom_point()
#Solution For Question 19
using_heirarchicalclustering <- hclust(dist(org_vehicle_dataset[, c("Selling_Price", "Present_Price")]), method = "average")
formed_clusters <- cutree(using_heirarchicalclustering, k = 3)
org_vehicle_dataset$HC_Cluster <- as.factor(formed_clusters)
ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = HC_Cluster)) + geom_point()
#Solution For Question 20
updated_entry <- c("Age", "Year", "Transmission", "Seller_Type", "Fuel_Type", "Owner")
for (var in updated_entry) { fr<-ggplot(org_vehicle_dataset, aes_string(x = var)) + geom_bar(fills = "blue", color = "black") + labs(title = paste("Bar Plot of", var), x = var, y = "Count") + theme_minimal() + print(fr) }
  
```

21. Create a correlation plot of the whole dataset variables and explain the output. Do not forget to convert some of the variable’s datatype if required and possible.

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for creating a correlation matrix and visualizing it as a heatmap.
- Console:** Shows the output of the R code, including a frequency table for the 'Owner' variable and the resulting correlation matrix plot.
- Environment:** Shows the global environment with objects like `correlation_plot`, `descending`, and `using_kmeans`.
- Plots:** A correlation matrix plot titled "Correlation Plot" showing the correlation coefficients between variables (Year, Selling_Price, Present_Price, Kms_Driven, Owner, Age).

```

#Solution For Question 17
ggplot(org_vehicle_dataset, aes(x = factor(Transmission), y = Selling_Price, fill = factor(Fuel_Type)))
#Solution For Question 18
using_kmeans <- kmeans(org_vehicle_dataset[, c("Selling_Price", "Kms_Driven")], centers = 4)
org_vehicle_dataset$Cluster <- as.factor(using_kmeans$cluster)
ggplot(org_vehicle_dataset, aes(x = Kms_Driven, y = Selling_Price, color = Cluster)) + geom_point()
#Solution For Question 19
using_heirarchicalclustering <- hclust(dist(org_vehicle_dataset[, c("Selling_Price", "Present_Price")]), method = "average")
formed_clusters <- cutree(using_heirarchicalclustering, k = 3)
org_vehicle_dataset$HC_Cluster <- as.factor(formed_clusters)
ggplot(org_vehicle_dataset, aes(x = Present_Price, y = Selling_Price, color = HC_Cluster)) + geom_point()
#Solution For Question 20
updated_entry <- c("Age", "Year", "Transmission", "Seller_Type", "Fuel_Type", "Owner")
for (var in updated_entry) { fr<-ggplot(org_vehicle_dataset, aes_string(x = var)) + geom_bar(fills = "blue", color = "black") + labs(title = paste("Bar Plot of", var), x = var, y = "Count") + theme_minimal() + print(fr) }
#Solution For Question 21
correlation_plot <- cor(org_vehicle_dataset[sapply(org_vehicle_dataset, is.numeric)])
corplot(correlation_plot, method = "circle")
  
```

22. Create a scatter plot of the Selling_Price Vs Kms_Driven, and use DBSCAN clustering to cluster the points into 3 clusters. Color-code based on the cluster they belong to. Add a legend to the plot.

