Electric Vehicle Project

Clement Cheng

2022-12-08

This is a project related to Electric Vehicle. In this project, I am using the "ElectricCarData_Clean.csv" to solve 5 questions by the author for practicing my skills in R.

Data Source: Kaggle "EVs - One Electric Vehicle Dataset - Smaller" by GEOFF839

Last updated: 12/8/2022

Let's install and load packages!

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("sqldf")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library("tidyverse") # Basic package for data analysis
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("sqldf") # Find one off answers
```

```
## Loading required package: gsubfn
## Loading required package: proto
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
## Loading required package: RSQLite
```

```
library("ggplot2") # For visualization
```

Set the data frame and preview the file

```
df <- read.csv("/cloud/project/ElectricCarData_Clean.csv")
head(df)
```

```
##         Brand                          Model AccelSec TopSpeed_KmH Range_Km
## 1       Tesla  Model 3 Long Range Dual Motor      4.6          233      450
## 2 Volkswagen                       ID.3 Pure     10.0          160      270
## 3   Polestar                               2      4.7          210      400
## 4        BMW                            iX3       6.8          180      360
## 5      Honda                              e       9.5          145      170
## 6      Lucid                            Air       2.8          250      610
##   Efficiency_WhKm FastCharge_KmH RapidCharge PowerTrain   PlugType BodyStyle
## 1             161            940         Yes        AWD Type 2 CCS     Sedan
## 2             167            250         Yes        RWD Type 2 CCS Hatchback
## 3             181            620         Yes        AWD Type 2 CCS  Liftback
## 4             206            560         Yes        RWD Type 2 CCS       SUV
## 5             168            190         Yes        RWD Type 2 CCS Hatchback
## 6             180            620         Yes        AWD Type 2 CCS     Sedan
##   Segment Seats PriceEuro
## 1       D     5     55480
## 2       C     5     30000
## 3       D     5     56440
## 4       D     5     68040
## 5       B     4     32997
## 6       F     5    105000
```

Let's see if the data frame has missing value or duplicates

```
is.na(df)
```

```
##         Brand Model AccelSec TopSpeed_KmH Range_Km Efficiency_WhKm
##    [1,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [2,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [3,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [4,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [5,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [6,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [7,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [8,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##    [9,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [10,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [11,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [12,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [13,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [14,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [15,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [16,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [17,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [18,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [19,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [20,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [21,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [22,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [23,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [24,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##   [25,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
```

```
##  [26,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [27,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [28,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [29,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [30,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [31,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [32,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [33,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [34,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [35,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [36,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [37,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [38,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [39,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [40,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [41,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [42,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [43,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [44,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [45,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [46,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [47,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [48,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [49,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [50,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [51,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [52,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [53,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [54,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [55,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [56,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [57,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [58,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [59,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [60,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [61,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [62,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [63,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [64,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [65,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [66,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [67,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [68,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [69,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [70,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [71,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [72,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [73,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [74,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [75,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [76,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [77,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [78,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
##  [79,] FALSE FALSE    FALSE        FALSE    FALSE          FALSE
```

```
##  [80,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [81,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [82,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [83,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [84,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [85,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [86,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [87,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [88,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [89,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [90,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [91,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [92,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [93,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [94,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [95,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [96,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [97,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [98,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##  [99,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
## [100,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
## [101,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
## [102,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
## [103,] FALSE FALSE    FALSE        FALSE    FALSE           FALSE
##        FastCharge_KmH RapidCharge PowerTrain PlugType BodyStyle Segment Seats
##   [1,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [2,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [3,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [4,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [5,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [6,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [7,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [8,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##   [9,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [10,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [11,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [12,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [13,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [14,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [15,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [16,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [17,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [18,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [19,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [20,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [21,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [22,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [23,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [24,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [25,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [26,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [27,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [28,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [29,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
```

```
##  [30,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [31,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [32,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [33,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [34,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [35,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [36,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [37,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [38,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [39,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [40,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [41,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [42,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [43,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [44,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [45,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [46,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [47,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [48,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [49,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [50,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [51,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [52,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [53,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [54,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [55,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [56,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [57,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [58,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [59,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [60,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [61,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [62,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [63,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [64,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [65,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [66,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [67,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [68,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [69,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [70,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [71,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [72,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [73,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [74,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [75,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [76,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [77,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [78,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [79,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [80,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [81,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [82,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [83,]         FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
```

```
##  [84,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [85,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [86,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [87,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [88,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [89,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [90,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [91,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [92,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [93,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [94,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [95,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [96,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [97,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [98,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##  [99,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
## [100,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
## [101,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
## [102,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
## [103,]          FALSE       FALSE      FALSE    FALSE     FALSE   FALSE FALSE
##        PriceEuro
##   [1,]     FALSE
##   [2,]     FALSE
##   [3,]     FALSE
##   [4,]     FALSE
##   [5,]     FALSE
##   [6,]     FALSE
##   [7,]     FALSE
##   [8,]     FALSE
##   [9,]     FALSE
##  [10,]     FALSE
##  [11,]     FALSE
##  [12,]     FALSE
##  [13,]     FALSE
##  [14,]     FALSE
##  [15,]     FALSE
##  [16,]     FALSE
##  [17,]     FALSE
##  [18,]     FALSE
##  [19,]     FALSE
##  [20,]     FALSE
##  [21,]     FALSE
##  [22,]     FALSE
##  [23,]     FALSE
##  [24,]     FALSE
##  [25,]     FALSE
##  [26,]     FALSE
##  [27,]     FALSE
##  [28,]     FALSE
##  [29,]     FALSE
##  [30,]     FALSE
##  [31,]     FALSE
##  [32,]     FALSE
##  [33,]     FALSE
```

```
##  [34,]    FALSE
##  [35,]    FALSE
##  [36,]    FALSE
##  [37,]    FALSE
##  [38,]    FALSE
##  [39,]    FALSE
##  [40,]    FALSE
##  [41,]    FALSE
##  [42,]    FALSE
##  [43,]    FALSE
##  [44,]    FALSE
##  [45,]    FALSE
##  [46,]    FALSE
##  [47,]    FALSE
##  [48,]    FALSE
##  [49,]    FALSE
##  [50,]    FALSE
##  [51,]    FALSE
##  [52,]    FALSE
##  [53,]    FALSE
##  [54,]    FALSE
##  [55,]    FALSE
##  [56,]    FALSE
##  [57,]    FALSE
##  [58,]    FALSE
##  [59,]    FALSE
##  [60,]    FALSE
##  [61,]    FALSE
##  [62,]    FALSE
##  [63,]    FALSE
##  [64,]    FALSE
##  [65,]    FALSE
##  [66,]    FALSE
##  [67,]    FALSE
##  [68,]    FALSE
##  [69,]    FALSE
##  [70,]    FALSE
##  [71,]    FALSE
##  [72,]    FALSE
##  [73,]    FALSE
##  [74,]    FALSE
##  [75,]    FALSE
##  [76,]    FALSE
##  [77,]    FALSE
##  [78,]    FALSE
##  [79,]    FALSE
##  [80,]    FALSE
##  [81,]    FALSE
##  [82,]    FALSE
##  [83,]    FALSE
##  [84,]    FALSE
##  [85,]    FALSE
##  [86,]    FALSE
##  [87,]    FALSE
```

```
##  [88,]     FALSE
##  [89,]     FALSE
##  [90,]     FALSE
##  [91,]     FALSE
##  [92,]     FALSE
##  [93,]     FALSE
##  [94,]     FALSE
##  [95,]     FALSE
##  [96,]     FALSE
##  [97,]     FALSE
##  [98,]     FALSE
##  [99,]     FALSE
## [100,]     FALSE
## [101,]     FALSE
## [102,]     FALSE
## [103,]     FALSE
```

```
duplicated(df)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

It seems the data is clean for processing

Start the Analysis! Solving Q1: Which car has the fastest 0-100 acceleration? My thoughts: Sort the data by [AccelSec]

```
sort(df$AccelSec) # Using sort the find the lowest Accel Sec
```

```
##   [1]  2.1  2.5  2.8  2.8  2.8  3.0  3.2  3.4  3.5  3.5  3.7  3.8  4.0  4.0  4.0
##  [16]  4.5  4.5  4.6  4.6  4.7  4.8  4.9  5.0  5.0  5.1  5.1  5.1  5.5  5.6  5.7
##  [31]  5.7  5.7  5.9  6.0  6.0  6.2  6.3  6.3  6.5  6.6  6.8  6.8  6.8  6.9  7.0
##  [46]  7.0  7.0  7.3  7.3  7.3  7.3  7.3  7.3  7.5  7.5  7.5  7.5  7.5  7.6  7.8
##  [61]  7.9  7.9  7.9  7.9  7.9  8.1  8.1  8.2  8.3  8.5  8.5  8.7  8.8  9.0  9.0
##  [76]  9.0  9.0  9.0  9.0  9.0  9.5  9.5  9.6  9.7  9.7  9.8  9.9  9.9 10.0 10.0
##  [91] 10.0 10.0 11.4 11.4 11.6 11.9 11.9 12.3 12.3 12.6 12.7 14.0 22.4
```

```
sqldf("SELECT Brand, Model, AccelSec FROM df WHERE AccelSec = 2.1") # Using sqldf to find the details o
```

```
##    Brand     Model AccelSec
## 1 Tesla  Roadster       2.1
```

A1: Tesla's Roadster has the fastest 0-100 acceleration as 2.1.

Solving Q2: Which has the highest efficiency? My thoughts: Sort the data by [Efficiency_WhKm]

```
sort(df$Efficiency_WhKm, decreasing = TRUE)
```

```
##   [1] 273 270 267 261 258 256 244 238 237 232 232 231 228 223 222 219 217 216
##  [19] 216 215 211 209 207 206 206 206 200 200 200 198 197 197 195 194 194 194
##  [37] 193 193 193 193 193 191 188 188 188 184 183 183 181 181 181 180 180 180
##  [55] 180 179 178 178 177 176 176 176 175 175 175 173 173 172 171 171 171 171
```

```
##  [73] 170 168 168 168 168 168 168 167 167 167 167 166 166 166 166 165 165 164
##  [91] 164 164 164 161 161 161 160 156 156 154 153 153 104
```

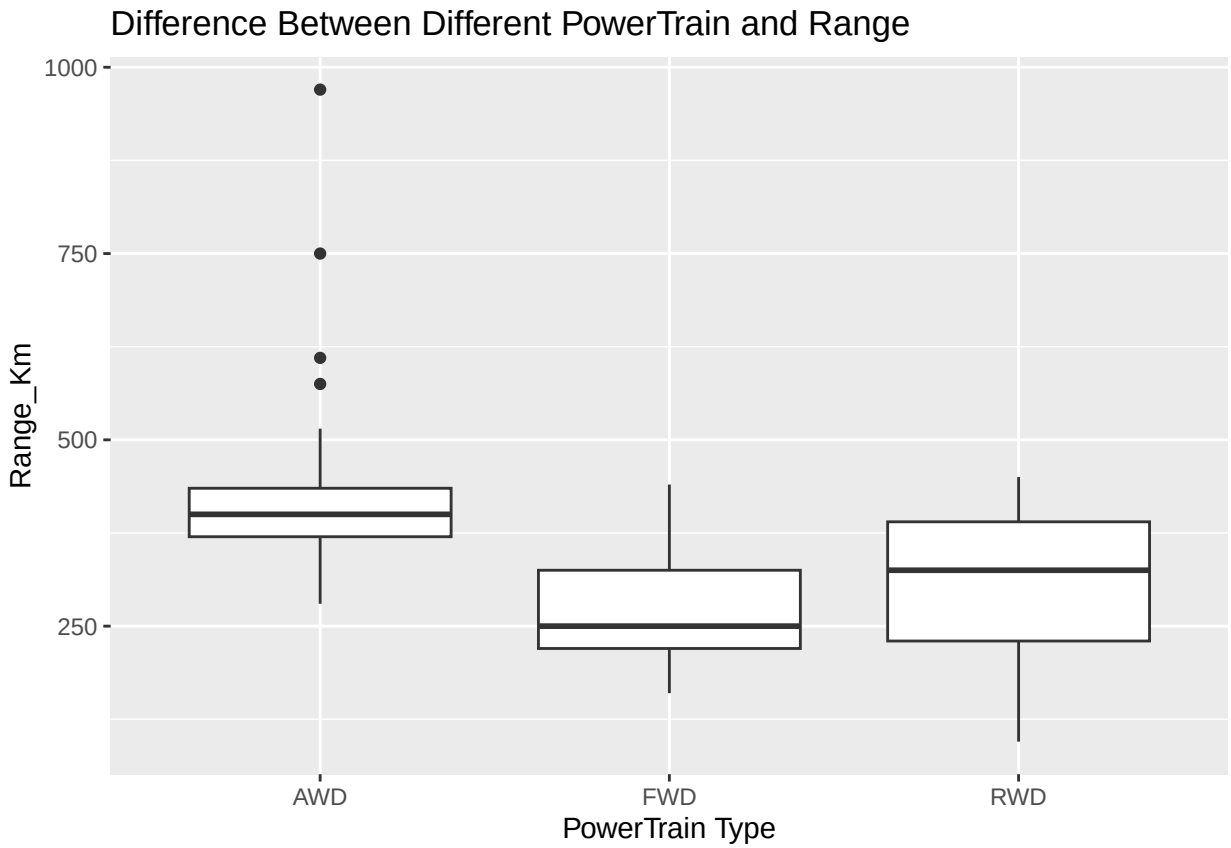
```
sqldf("SELECT Brand, Model, Efficiency_WhKm From df WHERE Efficiency_WhKm = 273")# Using sqldf to find
```

```
##       Brand        Model Efficiency_WhKm
## 1 Mercedes  EQV 300 Long             273
```

A2: Mercedes's EQV 300 Long has the highest efficiency as 273 WhKm.

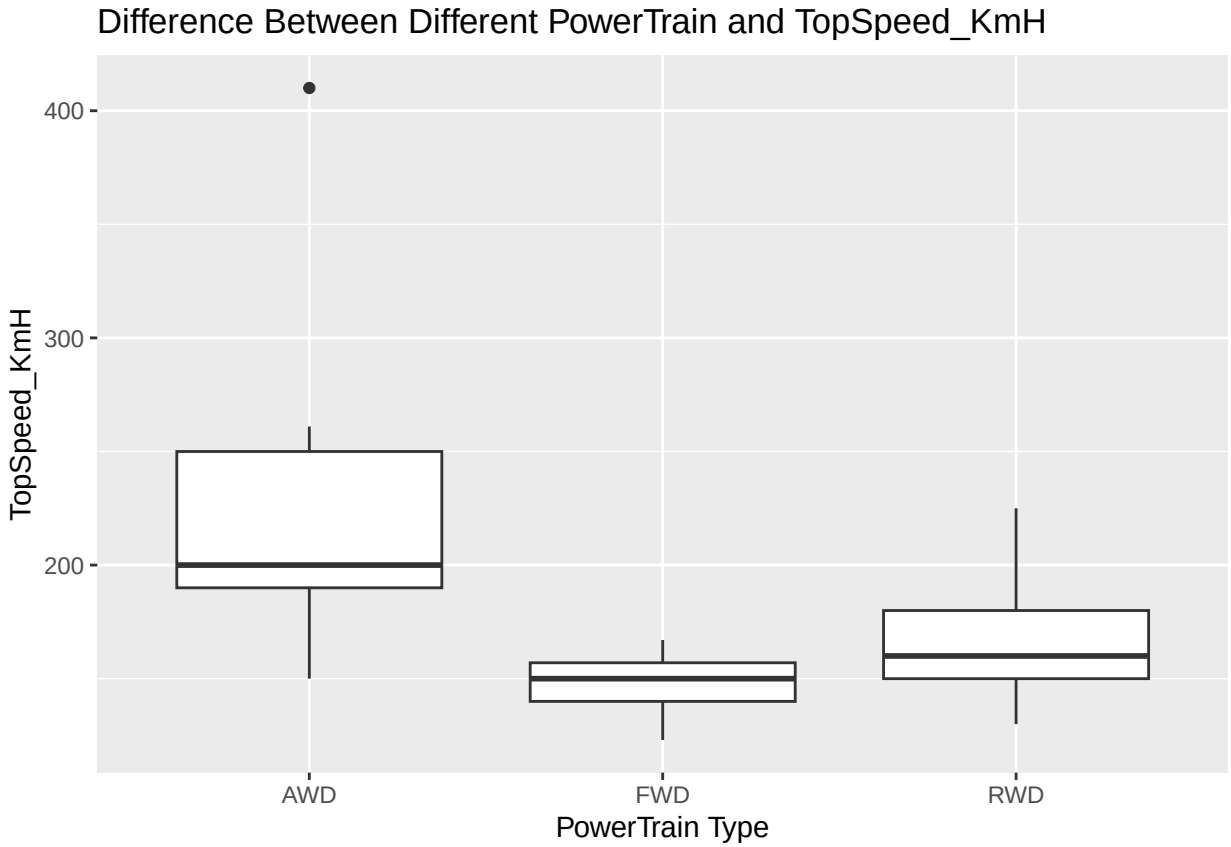
Solving Q3: Does a difference in power train effect the range, top speed, efficiency? My thoughts: Using ggplot to show the distribution between power train and range, power train and speed, and power train and efficiency.

```
ggplot(df, aes(PowerTrain, Range_Km))+geom_boxplot()+labs(title = "Difference Between Different PowerTra
```



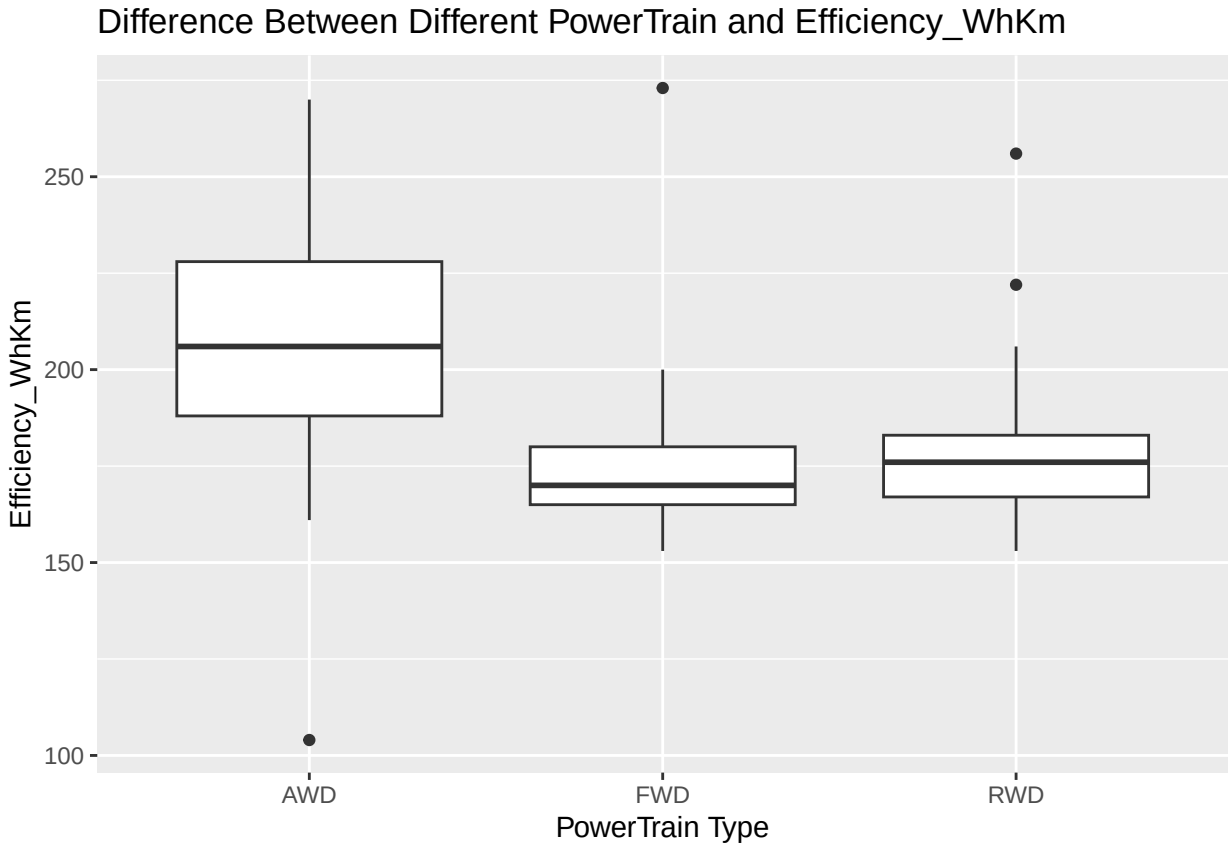
EV's with AWD PowerTrain are the best performer in Range, then FWD, last RWD. It was determined by the minimum range Km the electric vehicle can drove.

```
ggplot(df, aes(PowerTrain, TopSpeed_KmH))+geom_boxplot()+labs(title = "Difference Between Different Pow
```



EV's with AWD PowerTrain are the best performer in Top Speed, then RWD, last FWD. It was determined by the minimum top speed Km per hour.

```
ggplot(df, aes(PowerTrain, Efficiency_WhKm))+geom_boxplot()+labs(title = "Difference Between Different
```



EV's with AWD PowerTrain are the best performer in Efficiency, then RWD, last FWD. It was determined by the average efficiency_WhKm.

A3: Overall, EV's with AWD PowerTrain will have better performance in range, speed, and efficiency.

Solving Q4: Which Manufacturer has the most number of vehicles? My thoughts: Using sqldf for counting vehicles in each brand

```
sqldf("SELECT DISTINCT Brand, COUNT(Brand) as num_vehicels FROM df GROUP BY Brand")
```

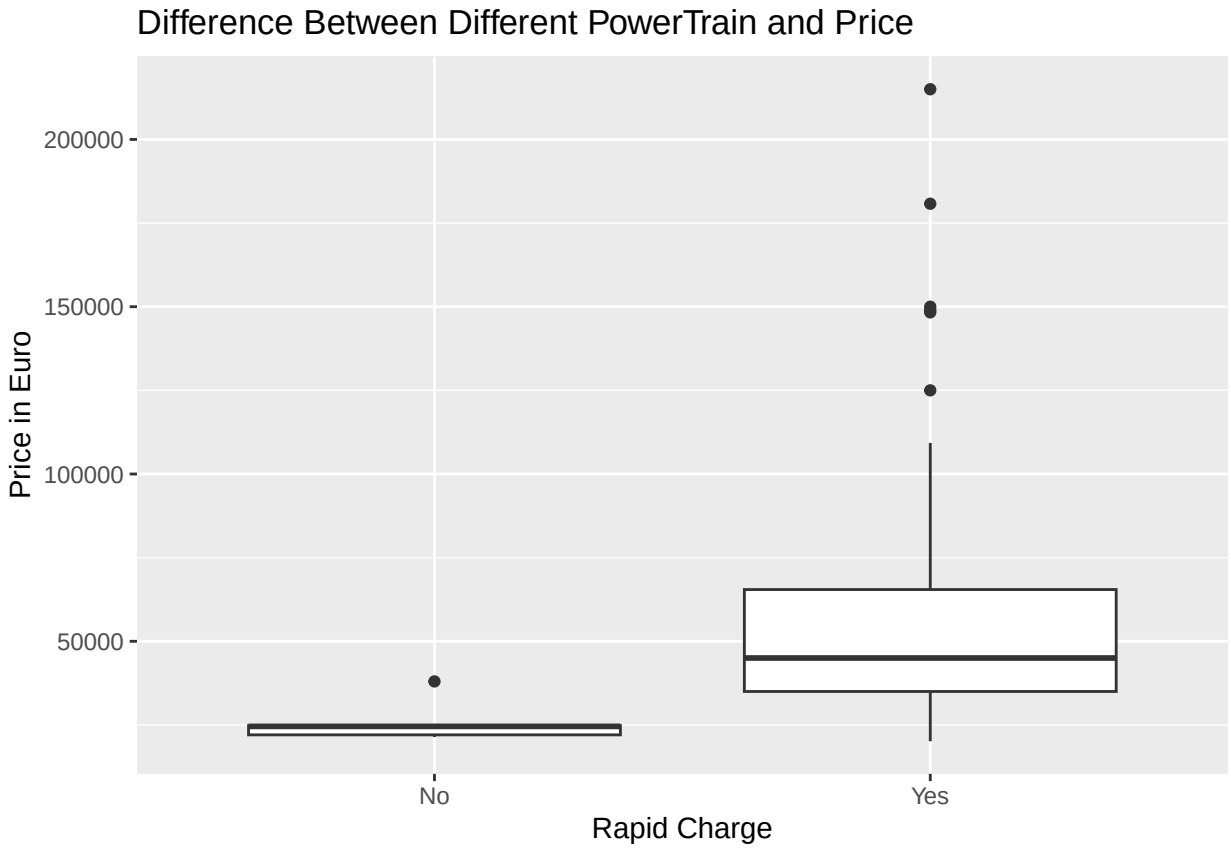
```
##          Brand num_vehicels
## 1      Aiways             1
## 2        Audi             9
## 3         BMW             4
## 4       Byton             3
## 5       CUPRA             1
## 6     Citroen             1
## 7          DS             1
## 8        Fiat             2
## 9        Ford             4
## 10      Honda             2
## 11    Hyundai             3
## 12     Jaguar             1
## 13        Kia             5
## 14      Lexus             1
## 15  Lightyear             1
## 16      Lucid             1
## 17         MG             1
## 18      Mazda             1
```

```
## 19   Mercedes             3
## 20       Mini             1
## 21     Nissan             8
## 22       Opel             3
## 23    Peugeot             2
## 24   Polestar             1
## 25    Porsche             5
## 26    Renault             5
## 27       SEAT             1
## 28      Skoda             6
## 29      Smart             3
## 30       Sono             1
## 31      Tesla            13
## 32 Volkswagen             8
## 33      Volvo             1
```

A4: Tesla has the most number of vehicles.

Solving Q5: How does price related to rapid charging My thoughts: Using ggplot to show the distribution between price and rapid charging

```
ggplot(df, aes(RapidCharge, PriceEuro))+geom_boxplot()+labs(title = "Difference Between Different Power
```



A5: EVs that cost less than 50000 euro has low chance of equip with Rapid charge function, therefore, it means price and rapid charge function has a positive correlation.

That's the end of all questions. Thanks you for reading my project in R.