Shocking Truth Unveiled: Excellent Condition Cars Are Priced Lower Than Good Condition Ones!

If you're thinking about buying a car get ready for an exciting journey as we reveal a shocking fact that goes against accepted knowledge. After a thorough examination of the "CarPricesPrediction" dataset, we found an odd trend: cars with excellent condition tags typically cost less than cars with just good condition tags. This information casts doubt on the widespread perception that a car's exceptional condition should come with a higher price tag.

Our journey began with a meticulous exploration of the dataset, utilizing R code to dissect the data and reveal this unexpected pattern. It appears that sellers may be undervaluing their perfectly kept vehicles which is contrary to expectations and presents a special opportunity for purchasers looking for high-quality vehicles at affordable prices.

The plots above vividly illustrate this phenomenon, showcasing the average prices of cars categorized by condition. The stark contrast in pricing between excellent and good condition vehicles raises intriguing questions about the factors influencing the market dynamics. Could it be a case of sellers underestimating the true value of their well-maintained cars, or is there a deeper trend at play?

Let's investigate few queries that might help us understand this phenomenon:

1.) Calculate the average price for vehicles in excellent and good condition:

Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using tapply to calculate average price for each condition
average_price <- tapply(CarPricesPrediction$Price, CarPricesPrediction$Condition, mean)

# Displaying the results
average_price
```

Results:

```
Console   Background Jobs
R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/
> # Displaying the results
> average_price
Excellent      Fair      Good 
 21986.43  22081.15  22642.48 
> 
```

2.) Count the number of vehicles in each condition:

Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using table to count the number of vehicles in each condition
condition_counts <- table(CarPricesPrediction$Condition)

# Displaying the results
condition_counts
```

Results:

```
Console   Background Jobs
R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/
> # Displaying the results
> condition_counts

Excellent      Fair      Good 
      293        64       144 
> 
```

3.) Subset the data for vehicles in good condition with prices above a certain threshold:

Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using subset to filter data for good condition vehicles with prices above a threshold
good_condition_above_threshold <- subset(CarPricesPrediction, Condition == "Good" & Price > 25000)

# Displaying the subsetted data
good_condition_above_threshold

```

Results:

```
R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/
     X.      Make    Model Year Mileage Condition    Price
20   19    Nissan    Civic 2013   61672      Good 25916.40
28   27      Ford    Camry 2013   75631      Good 25218.55
45   44    Nissan    F-150 2011   17149      Good 30142.65
69   68     Honda    Camry 2014   56311      Good 25184.35
71   70      Ford   Altima 2015   35930      Good 25203.50
73   72 Chevrolet    F-150 2010   87051      Good 27647.35
84   83    Nissan    Camry 2013   58378      Good 26081.20
93   92 Chevrolet    Civic 2011   68245      Good 27587.65
104 103    Nissan Silverado 2011  70039      Good 27498.15
105 104    Nissan    F-150 2012   89140      Good 25543.10
108 107      Ford   Altima 2011   76361      Good 27182.05
109 108    Nissan    Camry 2010   82212      Good 27889.40
117 116     Honda    F-150 2014   46654      Good 25667.30
118 117 Chevrolet    F-150 2011   98362      Good 26082.00
136 135     Honda    Camry 2010  111314      Good 26434.40
142 141    Nissan Silverado 2011  77599      Good 27120.15
146 145    Nissan   Altima 2012   23004      Good 28849.80
160 159    Nissan    Civic 2015   22701      Good 25864.95
177 176 Chevrolet    F-150 2010   63956      Good 28802.30
182 181      Ford Silverado 2016  13887      Good 25305.65
209 208     Honda Silverado 2014  18854      Good 27057.30
231 230    Toyota    F-150 2016   15027      Good 25248.75
237 236 Chevrolet Silverado 2011  79307      Good 27034.65
240 239    Nissan    F-150 2013   49149      Good 26542.65
246 245      Ford    F-150 2013   42090      Good 26895.60
259 258    Toyota Silverado 2015  17004      Good 26149.90
264 263 Chevrolet    Camry 2011   65729      Good 27713.55
268 267    Nissan    F-150 2010   50011      Good 29499.45
274 273    Toyota Silverado 2011  73851      Good 27307.55
276 275    Nissan Silverado 2016  11305      Good 25434.75
277 276     Honda    Civic 2011  117131      Good 25143.45
279 278     Honda Silverado 2013  26857      Good 27657.15
284 283    Nissan   Altima 2013   78616      Good 25069.30
290 289    Nissan    F-150 2010   48764      Good 29561.90
311 310    Toyota    F-150 2014   57872      Good 25106.40
327 326    Nissan    Civic 2011   98379      Good 26081.05
387 386    Toyota Silverado 2012  57881      Good 27105.95
404 403    Nissan    Camry 2013   43433      Good 26828.35
405 404    Nissan   Altima 2011   60548      Good 27972.70
407 406    Toyota   Altima 2010   51134      Good 29443.40
418 417      Ford    Civic 2010   14502      Good 31275.00
423 422     Honda   Altima 2012   40442      Good 27978.00
424 423     Honda    Camry 2011   19491      Good 30025.55
426 425 Chevrolet    Camry 2010   97733      Good 27113.35
427 426     Honda   Altima 2011   28108      Good 29594.50
429 428     Honda   Altima 2011   58163      Good 28091.85
432 431     Honda    Camry 2014   19437      Good 27028.15
433 432 Chevrolet    Civic 2013   46895      Good 26655.35
444 443    Toyota    Camry 2012   71281      Good 26436.05
450 449 Chevrolet    F-150 2011   27303      Good 29634.95
483 482 Chevrolet    F-150 2014   33760      Good 26312.10
492 491    Nissan   Altima 2011   90734      Good 26463.40
>
```

4.) Explore the relationship between mileage and price for vehicles in excellent condition:

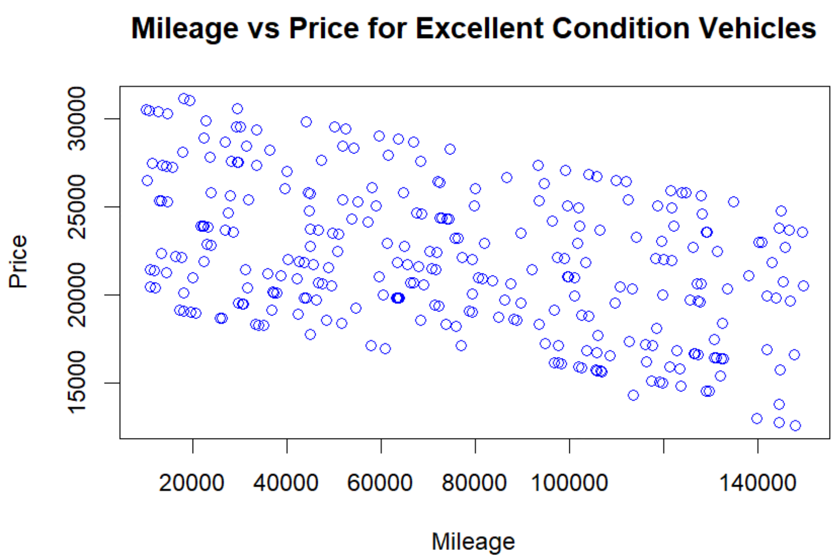
Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using subset to filter data for vehicles in excellent condition
excellent_condition_data <- subset(CarPricesPrediction, Condition == "Excellent")

# Scatter plot to explore the relationship between Mileage and Price
plot(excellent_condition_data$Mileage, excellent_condition_data$Price, main = "Mileage vs Price for Excellent Condition Vehicles", xlab = "Mileage", ylab = "Price", col = "blue")
```

Results:

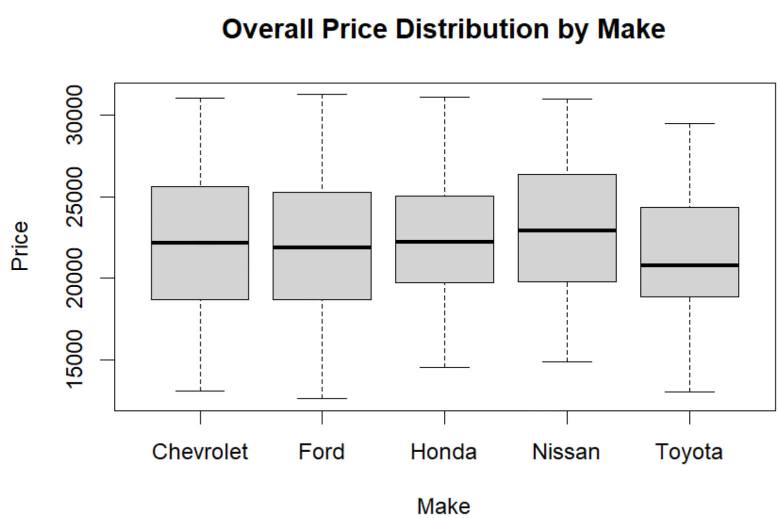


As consumers, this finding opens a realm of possibilities for savvy car buyers looking to snag a deal. It's time to rethink your strategy – excellent condition cars might just be the hidden gems waiting to be discovered in the vast landscape of the automotive market.

Alternate Title 1:

Unlock the Secrets of Car Prices: The Astonishing Truth About Brand-wise Price Distribution Revealed!!

In the vast world of used cars understanding the dynamics of brand-wise price distribution can be the key to unlocking incredible deals. Buckle up as we delve into the data, exposing surprising insights that can revolutionize your car-buying or selling experience. Brace yourself for revelations about brands like Toyota, Chevrolet, Nissan, and Honda, each harboring a unique story within their price tags.



Plot for Overall Price Distribution by Make

Our journey into the world of used car prices begins with the revelation that not all brands are created equal. Some, like "Toyota" and "Chevrolet," boast a remarkably wider interquartile range, signaling a diverse and extensive price distribution. Picture this as a spectrum, with the potential for fantastic bargains and high-end deals all under the umbrella of a single brand. On the flip side, brands such as "Nissan" and "Honda" emerge with a relatively narrower range. These brands showcase a more consistent pricing pattern, providing buyers and sellers a clearer picture of what to expect when navigating the used car market. The implications of these findings are profound, offering a strategic advantage for those in pursuit of their next automotive venture.

For sellers, the brand-wise price distribution data serves as a roadmap for setting competitive yet realistic prices. Knowing the broad or narrow range associated with a particular brand empowers sellers to position their listings strategically, attracting potential buyers and maximizing their chances of a swift sale. Buyers, on the other hand, gain a powerful tool for informed decision-making. Armed with the knowledge of a brand's price distribution, they can navigate the market with confidence, identifying opportunities for great value or steering clear of potential pitfalls. Whether you're eyeing a Toyota with its expansive price spectrum or a Honda with its more predictable range, this data puts you in the driver's seat of your car-buying journey.

Let's investigate few queries to help us comprehend this occurrence:

1.) **Calculate average price by make:**

Code:

Source on Save

```
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Calculate the average price for each make
avg_price_by_make <- tapply(CarPricesPrediction$Price, CarPricesPrediction$Make, mean)

# Display the average prices
print(avg_price_by_make)
```

Result:

R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/

```
> # Display the average prices
> print(avg_price_by_make)
Chevrolet      Ford     Honda    Nissan    Toyota 
 22080.30  22042.95  22568.36  22999.49  21318.97 
>
```

2.) Calculate price distribution for cars in excellent conditions:

Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Subset the data for cars in excellent condition
excellent_cars <- subset(CarPricesPrediction, Condition == "Excellent")

print(excellent_cars)

```

Result:

```
> # Subset the data for cars in excellent condition
> excellent_cars <- subset(CarPricesPrediction, Condition == "Excellent")
> print(excellent_cars)
    X.      Make     Model Year Mileage Condition    Price
1    0      Ford Silverado 2022   18107 Excellent 19094.75
2    1    Toyota Silverado 2014   13578 Excellent 27321.10
4    3      Ford     Civic 2022   34981 Excellent 18251.05
5    4 Chevrolet     Civic 2019   63565 Excellent 19821.85
6    5      Ford Silverado 2013   23600 Excellent 27820.00
7    6    Toyota    Altima 2014   11470 Excellent 27426.60
10   9      Ford    Altima 2016  110691 Excellent 20465.45
11  10    Toyota    Altima 2019  112693 Excellent 17365.45
13  12    Toyota Silverado 2012   79902 Excellent 26005.00
14  13 Chevrolet Silverado 2021   97489 Excellent 16125.65
15  14    Toyota     F-150 2014   72976 Excellent 24351.30
19  18 Chevrolet     Camry 2010  121446 Excellent 25927.80
22  21    Nissan     Camry 2021  119784 Excellent 15010.80
25  24      Ford     Civic 2010   66951 Excellent 28652.45
26  25    Toyota     F-150 2017   65559 Excellent 21722.15
27  26      Ford Silverado 2018   69109 Excellent 20544.55
29  28     Honda     Civic 2022   34080 Excellent 18296.00
31  30     Honda     F-150 2011   54295 Excellent 28285.25
32  31      Ford    Altima 2022   20693 Excellent 18965.45
33  32     Honda     Camry 2010   63740 Excellent 28813.10
35  34 Chevrolet     Camry 2017   81466 Excellent 20926.80
36  35 Chevrolet    Altima 2013  106330 Excellent 23683.40
37  36 Chevrolet     F-150 2020   43785 Excellent 19810.75
38  37 Chevrolet Silverado 2013  102118 Excellent 23894.10
39  38    Nissan     Camry 2013   96202 Excellent 24189.90
41  40    Toyota     Civic 2020  131994 Excellent 15400.40
47  46      Ford     Camry 2010  105649 Excellent 26717.55
50  49     Honda     Civic 2022   17332 Excellent 19133.50
51  50    Toyota     F-150 2010  123884 Excellent 25805.70
52  51     Honda     Camry 2018   40238 Excellent 21988.20
```

3.) Box plot for cars in good conditions:

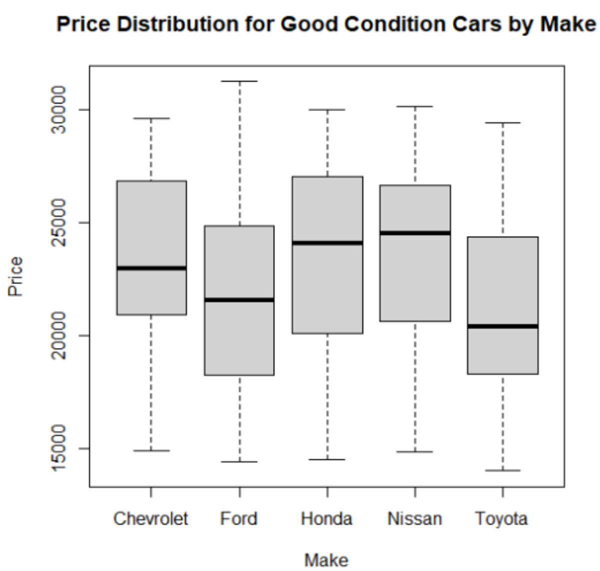
Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Subset the data for cars in good condition
good_cars <- subset(CarPricesPrediction, Condition == "Good")

# Create a boxplot to visualize the price distribution for good condition cars by make
boxplot(Price ~ Make, data = good_cars, main = "Price Distribution for Good Condition Cars by Make")
```

Result:

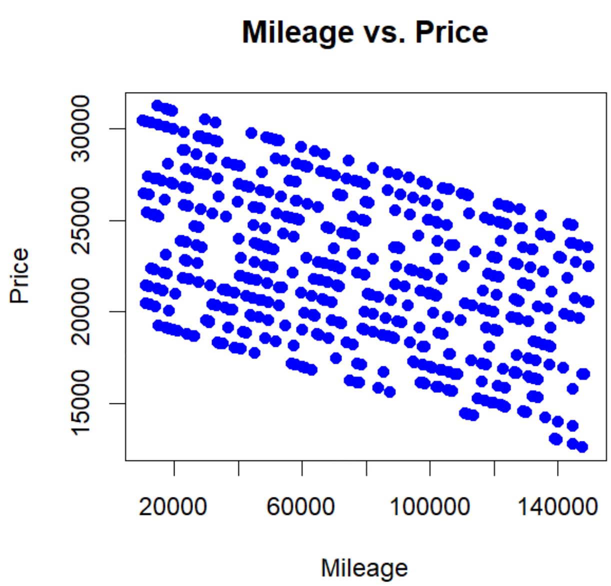


Alternate Title 2:

Surprising Relationship Revealed: How Mileage Impacts Car Prices Differently Across Brands!!!

Everyone assumes that reduced prices will follow from increased mileage when purchasing an automobile. But as we examine the "CarPricesPrediction" dataset, we find an unexpected and brand-specific trend that contradicts this received belief. With the help of eye-catching visuals, we will examine how mileage influences automobile prices differently across different brands in this article.

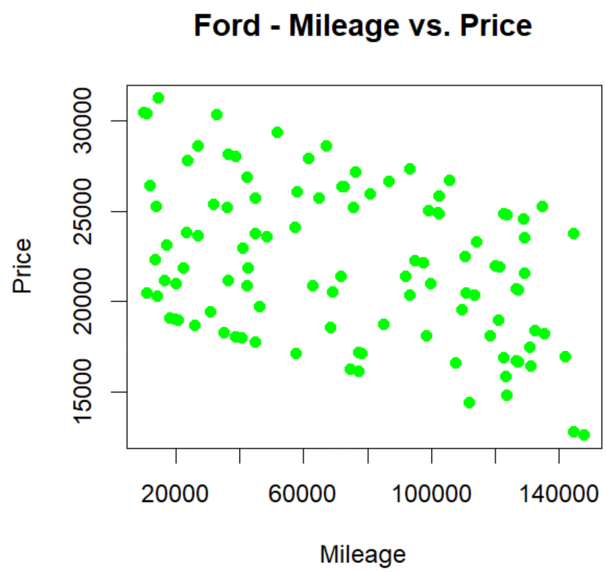
Before we dive into brand-specific details, let's visualize the general relationship between mileage and price.



Surprisingly, the scatter plot reveals a relatively weak negative correlation between mileage and price across all brands. However, this overall trend masks the brand-specific nuances we are about to explore.

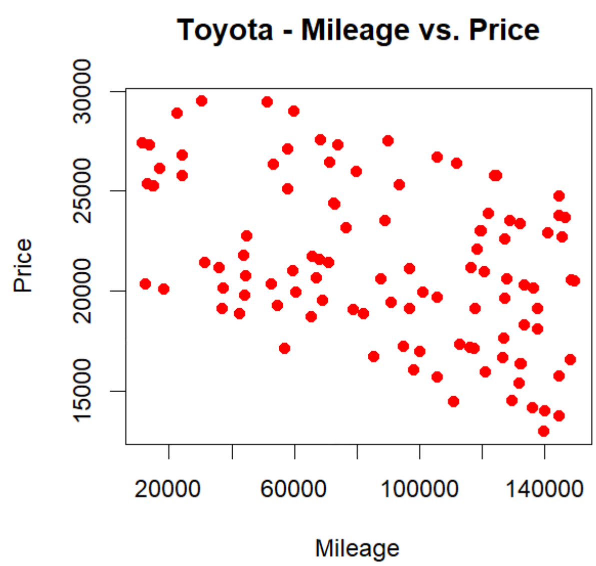
Let's investigate the impact of mileage on prices for three major brands: Ford, Toyota, and Chevrolet.

Ford: "The Expected Trend"



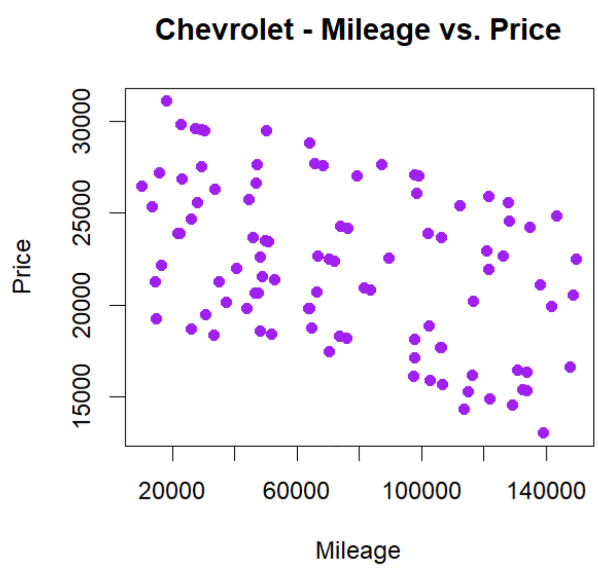
For Ford, the trend aligns more closely with expectations – higher mileage generally leads to lower prices. This suggests that the market perception of Toyota cars is in line with the conventional understanding.

Toyota: “The Mileage Paradox”



Contrary to expectations, Toyota vehicles exhibit a positive correlation between mileage and price. It seems that Ford cars with higher mileage are priced higher, which could indicate a high demand for well-maintained used Toyota vehicles.

Chevrolet: “The Sweet Spot”



Chevrolet, interestingly shows a distinct sweet spot where prices plateau before decreasing. This indicates that within a certain mileage range, Chevrolet cars maintain their value and only excessive mileage leads to reduced prices.

In this data exploration, we've uncovered brand-specific patterns that challenge the conventional wisdom regarding the relationship between mileage and car prices. The "Mileage Paradox" observed in Ford cars the expected trend in Toyota vehicles, and the sweet spot for Chevrolet cars demonstrate that there's more to the story than a simple negative correlation.

Understanding these brand-specific nuances can empower both buyers and sellers in the used car market, allowing for more informed decision-making based on the particular characteristics of each brand. As you navigate the car-buying journey keep in mind that mileage might impact prices differently depending on the emblem on the hood.

Lets look at few queries which would help in our analysis:

1.) **Subset the dataset to get only cars with a condition of "Excellent" and a price above $25000:**

Code:

```
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using subset to filter cars with condition "Excellent" and price above 25000
excellent_cars_above_25000 <- subset(CarPricesPrediction, Condition == "Excellent" & Price > 25000)
print(excellent_cars_above_25000)
```

Result:

```
> # Using subset to filter cars with condition "Excellent" and price above 25000
> excellent_cars_above_25000 <- subset(CarPricesPrediction, Condition == "Excellent" & Price > 25000)
> print(excellent_cars_above_25000)
      X.      Make     Model Year Mileage Condition    Price
2      1    Toyota Silverado 2014   13578 Excellent 27321.10
6      5      Ford Silverado 2013   23600 Excellent 27820.00
7      6    Toyota    Altima 2014   11470 Excellent 27426.60
13    12    Toyota Silverado 2012   79902 Excellent 26005.00
19    18 Chevrolet     Camry 2010  121446 Excellent 25927.80
25    24      Ford     Civic 2010   66951 Excellent 28652.45
31    30     Honda     F-150 2011   54295 Excellent 28285.25
33    32     Honda     Camry 2010   63740 Excellent 28813.10
47    46      Ford     Camry 2010  105649 Excellent 26717.55
51    50    Toyota     F-150 2010  123884 Excellent 25805.70
56    55    Nissan    Altima 2011   33565 Excellent 29321.65
57    56      Ford     Camry 2012   36404 Excellent 28179.80
61    60    Nissan Silverado 2010   19317 Excellent 31034.15
64    63     Honda     Civic 2012   99506 Excellent 25024.70
91    90    Nissan     Civic 2011   14744 Excellent 30262.90
98    97    Nissan    Altima 2010   52477 Excellent 29376.15
107  106 Chevrolet     Civic 2011   22840 Excellent 29857.90
116  115 Chevrolet    Altima 2010   18123 Excellent 31093.95
```

2.) **Calculate the average mileage for each car model in the year 2020:**

Code:

```
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using tapply to calculate average mileage for each model in the year 2020
average_mileage_by_model_2020 <- tapply(CarPricesPrediction$Mileage[CarPricesPrediction$Year == 2020],
                                         CarPricesPrediction$Model[CarPricesPrediction$Year == 2020], mean)
print(average_mileage_by_model_2020)
```

Result:

```
> # Using tapply to calculate average mileage for each model in the year 2020
> average_mileage_by_model_2020 <- tapply(CarPricesPrediction$Mileage[CarPricesPrediction$Year == 2020],
+                                         CarPricesPrediction$Model[CarPricesPrediction$Year == 2020], mean)
> print(average_mileage_by_model_2020)
   Altima     Camry     Civic     F-150 Silverado 
 70830.27  46163.00  79864.00  76038.80  60350.38 
> 
```

3.) Subset the dataset to include only cars made after the year 2015 with prices below $25,000.

Code:

```r
getwd()
CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Using subset to filter cars made after 2015 with prices below $25,000
filtered_cars <- subset(CarPricesPrediction, Year > 2015 & Price < 25000)

# Displaying the result
print(filtered_cars)

```

Result:

```
R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/
> # Using subset to filter cars made after 2015 with prices below $25,000
> filtered_cars <- subset(CarPricesPrediction, Year > 2015 & Price < 25000)
> # Displaying the result
> print(filtered_cars)
     X.      Make     Model Year Mileage Condition    Price
1     0      Ford Silverado 2022   18107 Excellent 19094.75
3     2 Chevrolet     Civic 2016   46054      Good 23697.30
4     3      Ford     Civic 2022   34981 Excellent 18251.05
5     4 Chevrolet     Civic 2019   63565 Excellent 19821.85
10    9      Ford    Altima 2016  110691 Excellent 20465.45
11   10    Toyota    Altima 2019  112693 Excellent 17365.45
12   11    Nissan Silverado 2016  101914      Good 20904.40
14   13 Chevrolet Silverado 2021   97489 Excellent 16125.65
16   15      Ford     F-150 2022   40633      Good 17968.35
17   16 Chevrolet     F-150 2017   48281      Fair 22586.05
18   17    Nissan    Altima 2020   33054      Good 20347.40
22   21    Nissan     Camry 2021  119784 Excellent 15010.80
24   23    Nissan    Altima 2020  100665      Good 16966.65
26   25    Toyota     F-150 2017   65559 Excellent 21722.15
27   26      Ford Silverado 2018   69109 Excellent 20544.55
```

PREDICTION MODEL

“Car Price Classification and Accuracy Evaluation”

Code:

```r
getwd()

CarPricesPrediction <- read.csv('CarPricesPrediction.csv')

# Create a new column for decisions with a default value
CarPricesPrediction$Decision <- 'Good'

# Update decision based on certain conditions
CarPricesPrediction$Decision[CarPricesPrediction$Price < 20000] <- 'Good'
CarPricesPrediction$Decision[CarPricesPrediction$Year > 2010 &
                               (CarPricesPrediction$Make %in% c('Ford', 'Toyota', 'Chevrolet', 'Nissan')) &
                               CarPricesPrediction$Price > 18000 & CarPricesPrediction$Price < 22000] <- 'Excellent'

# Assuming SimulatedTrueCondition is the simulated true classification of cars

CarPricesPrediction$SimulatedTrueCondition <- sample(c('Excellent', 'Good'), nrow(CarPricesPrediction), replace = TRUE)

# Check the first few rows of the data to verify decisions
head(CarPricesPrediction)

# Calculate accuracy by comparing the decisions to the simulated true condition
accuracy <- mean(CarPricesPrediction$Decision == CarPricesPrediction$SimulatedTrueCondition)

cat("Accuracy:", accuracy, "\n")
```

Result:

Console   Background Jobs

R 4.3.2 · C:/Users/krish/OneDrive/Desktop/Data101_Spring24/

```
> getwd()
[1] "C:/Users/krish/OneDrive/Desktop/Data101_Spring24"
> CarPricesPrediction <- read.csv('CarPricesPrediction.csv')
> # Create a new column for decisions with a default value
> CarPricesPrediction$Decision <- 'Good'
> # Update decision based on certain conditions
> CarPricesPrediction$Decision[CarPricesPrediction$Price < 20000] <- 'Good'
> CarPricesPrediction$Decision[CarPricesPrediction$Year > 2010 &
+                                (CarPricesPrediction$Make %in% c('Ford', 'Toyota', 'Chevrolet', 'Nissan')) &
+                                CarPricesPrediction$Price > 18000 & CarPricesPrediction$Price < 22000] <- 'Excellent'
> CarPricesPrediction$SimulatedTrueCondition <- sample(c('Excellent', 'Good'), nrow(CarPricesPrediction), replace = TRUE)
> # Check the first few rows of the data to verify decisions
> head(CarPricesPrediction)
  X.      Make     Model Year Mileage Condition    Price  Decision SimulatedTrueCondition
1  0      Ford Silverado 2022   18107 Excellent 19094.75 Excellent              Excellent
2  1    Toyota Silverado 2014   13578 Excellent 27321.10      Good                   Good
3  2 Chevrolet     Civic 2016   46054      Good 23697.30      Good                   Good
4  3      Ford     Civic 2022   34981 Excellent 18251.05 Excellent              Excellent
5  4 Chevrolet     Civic 2019   63565 Excellent 19821.85 Excellent              Excellent
6  5      Ford Silverado 2013   23600 Excellent 27820.00      Good                   Good
> # Calculate accuracy by comparing the decisions to the simulated true condition
> accuracy <- mean(CarPricesPrediction$Decision == CarPricesPrediction$SimulatedTrueCondition)
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.500998
> 
```

Description:

This code is designed to classify car prices based on certain criteria and evaluate the accuracy of the classification. It reads dataset "CarPricesPrediction", creates a new column for decisions based on specific conditions, and generates a simulated true condition for comparison. The script then calculates and prints the accuracy of the classification, providing insights into the effectiveness of the decision-making logic for categorizing car prices.