[Date]

Professor **Mostafa K. Ardakani, Ph.D., P.E., PMP**

Fundamentals of Machine Learning

Project Clustering

|  |  |
| --- | --- |
| Team Members | Participation in project |
| Abhinav Chandra, Katha | Complete R coding,PPT, analysis of data and descriptive analysis, and small part in the writing the report |
| Keerthana Vonteddu | Participated in the preparation of in-class PPT slides, generation of report, code interpretation results and formatting the document. |

**Project Goal**

### Title:

Clustering Motorcycle Models Based on Multiple Features Using Machine Learning Techniques

### Introduction:

The project aims to analyze and cluster motorcycle models based on multiple features using machine learning algorithms. By applying clustering techniques such as K-means and DBSCAN, we seek to uncover hidden patterns and relationships within the dataset, ultimately gaining insights into the characteristics and classifications of different motorcycle models.

### Objective:

The primary objective of this project is to group motorcycle models into clusters based on their features, such as engine displacement, power, torque, and other relevant attributes. By clustering these models, we aim to identify similarities and differences between different types of motorcycles, providing valuable insights for manufacturers, retailers, and enthusiasts.

### Methodology:

Data Collection: Gather data on various motorcycle models, including features such as engine displacement, power, torque, engine cylinder count, fuel capacity, weight, and other relevant attributes. This data can be obtained from manufacturers, websites, or specialized databases.

### Data Preprocessing:

Clean the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary. Ensure that the dataset is ready for clustering analysis.

### Feature Selection:

Identify the features that will be used for clustering. These features should be relevant to the characteristics of motorcycle models and can include both quantitative and qualitative attributes.

### Clustering Algorithms:

K-means: Implement the K-means clustering algorithm to group motorcycle models into K clusters based on their features. Choose an appropriate value of K based on evaluation metrics such as the elbow method or silhouette score.

DBSCAN: Utilize the DBSCAN clustering algorithm to identify clusters of motorcycle models based on their density in feature space. Adjust parameters such as epsilon and minimum points to achieve meaningful clusters.

Cluster Analysis: Analyze the clusters obtained from K-means and DBSCAN to understand the characteristics of each cluster. Evaluate the clusters based on their coherence and interpretability, considering the features that differentiate each cluster.

### Visualization:

Visualize the clusters using plots such as scatter plots, heatmaps, or dendrograms to illustrate the relationships between motorcycle models based on their features. This visualization can provide insights into the grouping of similar models and the outliers.

Interpretation and Insights:

Interpret the clustering results to gain insights into the characteristics and classifications of motorcycle models. Identify any trends, patterns, or relationships that emerge from the clustering analysis, providing valuable information for decision-making in the motorcycle industry.

# Overview of the data:

The dataset used in this project comprises of 10340 rows with 18 columns which can ber termed as features and are explained clearly below. To gain insights into the data and understand its characteristics, we conducted thorough data exploration and descriptive analysis. Here's an overview of our findings:

# Data Structure:

The dataset consists of 10340 rows and 18 columns observations and features. These features include

Categorical Variables:  
1. Brand  
2. Model  
3. Category  
4. Rear brakes  
5. Front tires  
6. Rear tire  
7. Colour options

Numerical Variables:  
1. Year  
2. Displacement (ccm)  
3. Power (hp)  
4. Torque (Nm)  
5. Engine cylinder  
6. Engine stroke  
7. Gearbox  
8. Fuel capacity (lts)  
9. Dry weight (kg)  
10. Wheelbase (mm)  
11. Seat height (mm)

So, there are 7 categorical variables and 11 numerical variables in the provided dataset and the dataset has missing values.

Summary Statistics:  
In the provided dataset, we have information on 10,000 observations across 9 variables. These variables include categorical features such as Brand, Engine cylinder, and Category, as well as numerical attributes like Displacement (ccm), Power (hp), Torque (Nm), Wheelbase (mm), Seat height (mm), and Gearbox. Upon initial examination using the `str()` function, we observed that some variables contain missing values, notably Wheelbase (mm), Seat height (mm), and Gearbox. Further exploration revealed that Displacement (ccm) has a mean value of 259.9 and a standard deviation of 259.9, indicating variability in engine sizes across the dataset. Similarly, the mean Power (hp) is 20.91 with a standard deviation of 20.15, suggesting diversity in horsepower ratings among the observed vehicles. Additionally, the summary statistics provide insights into the central tendency, dispersion, and range of numerical variables, aiding in the initial understanding of the dataset's characteristics.

However, it's worth noting that missing values are prevalent in several variables, particularly Wheelbase (mm), Seat height (mm), and Gearbox. These missing values could potentially impact the analysis and interpretation of results and may require imputation or removal based on the extent of missingness and the specific objectives of the project.

Overall, the summary statistics obtained through the `summary()` function offer valuable insights into the distribution and characteristics of the numerical variables in the dataset. These statistics serve as a foundation for further exploratory data analysis and modeling tasks, guiding decisions on data preprocessing, feature engineering, and model selection.

## Missing Values:

We conducted an assessment of missing values in the dataset and applied appropriate strategies to address them effectively. Using the `na.omit()` function in R, we removed observations with any missing values, ensuring that subsequent analyses are conducted on complete cases. After this preprocessing step, we inspected the remaining missing values across variables using the `colSums(is.na(df))` function, which calculates the total number of missing values in each column.The summary of missing values across variables is as follows:

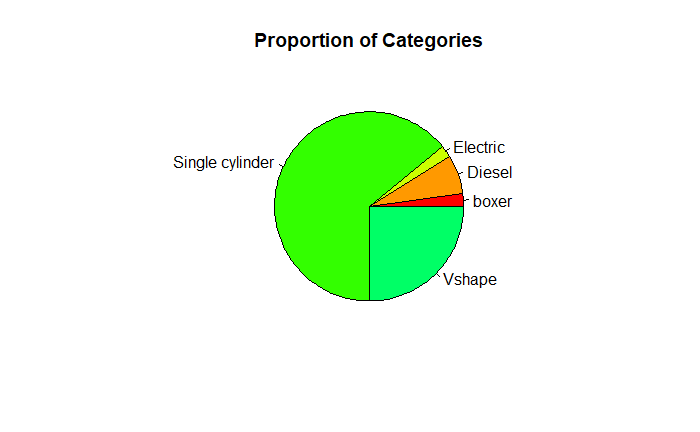
* Brand: 0 missing values
* Engine cylinder: 0 missing values
* Category: 0 missing values
* Displacement (ccm): 0 missing values
* Power (hp): 0 missing values
* Torque (Nm): 0 missing values
* Wheelbase (mm): 0 missing values
* Seat height (mm): 0 missing values
* Gearbox: 0 missing values

This summary indicates that after handling missing values, there are no remaining missing values across any of the variables in the dataset. Consequently, the dataset is now complete and ready for further analysis without concerns about missing data impacting the validity of the results. The absence of missing values ensures the integrity and reliability of subsequent analyses and modeling tasks.4. Distribution of Variables: We visualized the distribution of numerical variables using histograms, boxplots, and density plots. This helped us identify any outliers, skewness, or patterns in the data.

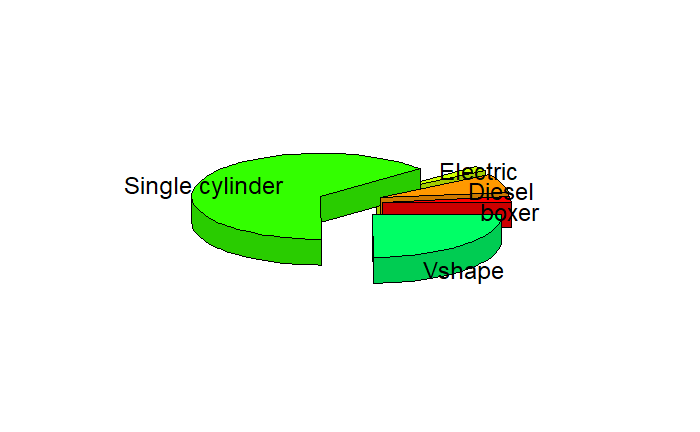
# Data exploration/Descriptive analysis:

The provided code generates two pie charts to visualize the distribution of categories within the 'Engine.cylinder' column:

1. Pie Chart with Base R: It displays the proportion of each category, with slices representing the frequency of each category. The colors, generated using `rainbow(10)`, differentiate categories, and the title "Proportion of Categories" clarifies the chart's purpose. The `cex = 1` parameter adjusts text size for readability.



2. 3D Pie Chart with 'pie3D': Similar to the first chart, but in 3D format for visual appeal. The `explode = 0.25` parameter separates slices slightly, enhancing visual distinction. Colors are assigned based on the same sequence from `rainbow()`. Each segment is labeled with the category name.



3. Frequency as Percentages: Calculated percentages offer insight into each category's relative distribution. The expression `prop.table(table(freq\_table)) \* 100` computes percentages relative to the total observations.

* Boxer="2.10280373831776 %"
* Diesel="6.77570093457944 %"
* Electric ="2.10280373831776 %"
* Single cylinder="64.1355140186916 %"
* V-shape="24.8831775700935 %"

These visualizations aid in understanding the distribution of engine cylinder categories, facilitating comparisons between categories based on their relative frequencies.

Outliers:  
Boxplots

The boxplots for each variable were generated, with the y-axis limited to 2000 to maintain uniformity. The boxplots provided the following insights:

* Displacement (ccm) – Most values were clustered between 125 and 426, but a few outliers exceeded this range.
* Power (hp) – The majority of values were below 93.90, with outliers present.
* Torque (Nm) – Most values were below 72.00, but several outliers were noted.
* Wheelbase (mm) – While most values were within 745 and 2100, outliers were present.
* Seat Height (mm) – The majority of values fell within 508.0 and 998.0, with outliers observed.
* Gearbox – Most values were between 1 and 8, with no significant outliers.

## Statistical Summary

The statistical summary provided detailed insights into each variable's distribution:

### Displacement (ccm)

* Median: 249.0
* Mean: 245.0
* Maximum: 426.0
* Minimum: 125.0

### Power (hp)

* Median: 20.10
* Mean: 22.38
* Maximum: 93.90
* Minimum: 6.50

### Torque (Nm)

* Median: 21.00
* Mean: 20.99
* Maximum: 72.00
* Minimum: 3.80

### Wheelbase (mm)

* Median: 1380
* Mean: 1400
* Maximum: 2100
* Minimum: 745

### Seat Height (mm)

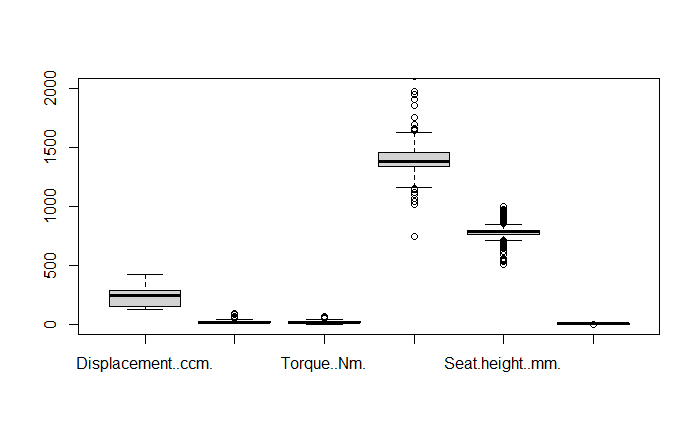
* Median: 785.0
* Mean: 786.1
* Maximum: 998.0
* Minimum: 508.0

### Gearbox

* Median: 6.0
* Mean: 5.974
* Maximum: 8.0
* Minimum: 1.0

The analysis identified outliers across multiple variables, particularly in Power (hp), Torque (Nm), and Wheelbase (mm). The boxplots provided a visual representation of the outliers, while the statistical summary offered numerical insights into the central tendency and spread of each variable.

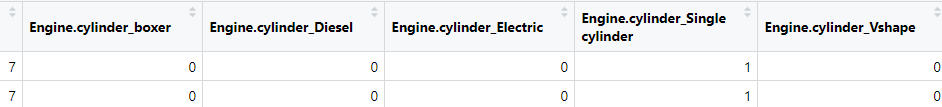
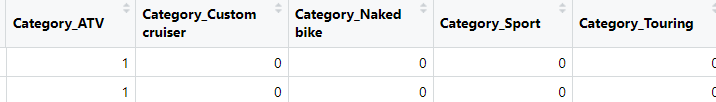
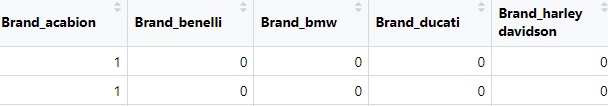
The presence of outliers suggests that some motorcycles in the dataset possess unusually high or low specifications. These outliers could be further investigated or removed based on the study's objectives. The methodology used in this report demonstrates the value of combining boxplots and summary statistics for exploratory data analysis.

Dummy variables

In this report, we focus on transforming categorical variables into dummy variables, which is a crucial step in preparing data for machine learning algorithms and statistical analyses.

#Loading the Required Library

#library(fastDummies)



The purpose of combining multiple data frames containing dummy variables is to create a consolidated data frame that includes all relevant information. This is achieved using the `cbind` function, which combines data frames by column. In this particular case, the dataframe `Engine\_cylinders\_dummies` contains dummy variables for "Engine. Cylinder," while `Brand\_dummies` and `Category\_dummies` are presumed to contain dummy variables for other categorical variables, such as "Brand" and "Category." The resulting dataframe, `transformed\_data`, contains all the dummy variables and the original data, thereby consolidating all necessary information for subsequent analysis.

transformed\_data <- cbind(Engine\_cylinders\_dummies, Brand\_dummies, Category\_dummies)

Creating dummy variables is a critical step in data preparation, especially when working with categorical data. The use of dummy variables allows categorical data to be included in machine learning models and statistical analyses, thereby enhancing the model’s effectiveness and interpretability.

# Feature Selection:

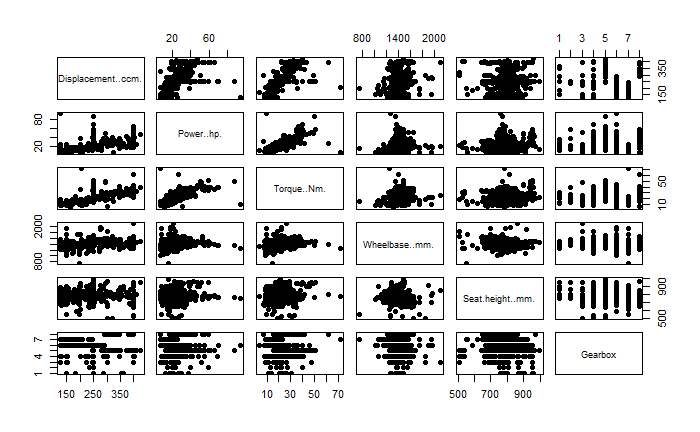
### Selecting Specific Columns for the Pair Plots:

* X1: This variable is a subset of X\_numeric, containing columns 1 to 6, which are assumed to be numerical values.
* X2: This variable contains columns 7 to 11, which relate to the "Engine cylinder" feature, possibly represented as dummy variables.
* X3: This variable includes columns 13 to 17, which pertain to the "Brand" feature, likely represented as dummy variables.
* X4: This variable consists of columns 18 to 22, which are related to the "Category" feature, also represented as dummy variables.

### Creating the Pair Plots:

* pairs(X1, pch = 19): This function call creates a pair plot for the first set of numerical features (columns 1 to 6). The parameter pch = 19 sets the plotting character to a solid circle.
* pairs(X2, pch = 19): This function call creates a pair plot for the features related to "Engine cylinder."
* pairs(X3, pch = 19): This function call creates a pair plot for the features related to "Brand."
* pairs(X4, pch = 19): This function call creates a pair plot for the features related to "Category."

# Analysis of the Pair Plot



The image provided is a pair plot of six numerical variables related to motorcycle specifications:

* Displacement (ccm)
* Power (hp)
* Torque (Nm)
* Wheelbase (mm)
* Seat Height (mm)
* Gearbox

Each cell in the matrix represents a scatterplot of a pair of variables, while the diagonal typically shows the distribution of individual variables. The key elements in the pair plot are:

### 1. Scatterplots

The scatterplots in the lower triangle of the matrix illustrate the relationships between pairs of variables. Key observations include:

#### - Displacement (ccm) vs. Power (hp):

- There is a positive linear relationship, indicating that as the displacement increases, the power tends to increase as well.

#### - Displacement (ccm) vs. Torque (Nm):

- There is also a positive linear relationship, suggesting that larger displacements generally lead to higher torque.

#### - Displacement (ccm) vs. Wheelbase (mm):

- A positive correlation is observed, with larger displacements correlating with longer wheelbases.

#### - Wheelbase (mm) vs. Seat Height (mm):

- There is a positive correlation, suggesting that motorcycles with longer wheelbases tend to have higher seats.

# 2. Histograms

The histograms in the diagonal provide insight into the distribution of individual variables:

### - Displacement (ccm):

- The distribution is right-skewed, with most values concentrated between 100 and 250 ccm.

- Power (hp):

- The distribution is also right-skewed, with most values below 30 hp.

### - Torque (Nm):

- The distribution is right-skewed, with most values below 30 Nm.

### - Wheelbase (mm):

- The distribution is roughly normal, with a peak around 1400 mm.

### - Seat Height (mm):

- The distribution is roughly normal, with a peak around 800 mm.

### - Gearbox:

- The distribution shows discrete values, with 5 and 6 gears being the most common.

## 3. Bar Charts

The bar charts in the rightmost column illustrate the frequency of different gearbox values:

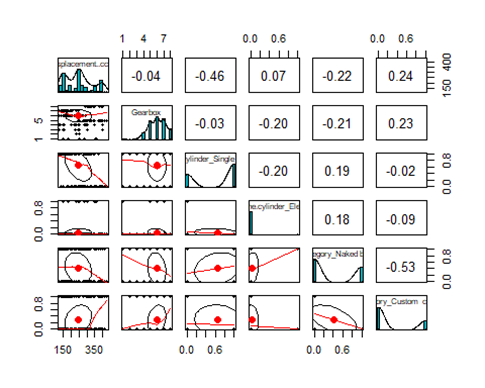
### - Gearbox:

- The bar chart indicates that the majority of motorcycles have either 5 or 6 gears, with fewer having other numbers of gears.

### Conclusion

The pair plot provides a comprehensive overview of the relationships and distributions of the motorcycle specifications. The positive linear relationships between displacement and other variables, along with the skewed distributions of power and torque, offer valuable insights for further analysis or modeling. The discrete nature of the gearbox variable highlights the categorical aspect, which may require special treatment in future analyses.

# Analysis of pair panels:



# Interpretation of the Plot:

## 1. Scatterplots

The scatterplots illustrate the relationships between pairs of variables. Key observations include:

### Displacement (ccm) vs. Gearbox:

The plot shows a weak positive linear relationship, indicating a slight trend where motorcycles with larger displacements tend to have a higher number of gears.

### Displacement (ccm) vs. Cylinder\_Single:

There is a negative linear relationship, suggesting that motorcycles with single-cylinder engines tend to have smaller displacements.

### Displacement (ccm) vs. Category\_Naked:

There is a negative relationship, indicating that motorcycles in the "Naked" category tend to have smaller displacements.

### Gearbox vs. Category\_Custom:

The plot shows a slight positive trend, indicating that custom motorcycles might have higher gear counts.

## 2. Correlation Coefficients

The upper triangle of the matrix displays the correlation coefficients between each pair of variables, indicating the strength and direction of linear relationships:

### Displacement (ccm) and Gearbox:

The correlation coefficient is -0.04, indicating a very weak negative linear relationship.

### Displacement (ccm) and Cylinder\_Single:

The correlation coefficient is -0.46, indicating a moderate negative linear relationship.

### Gearbox and Category\_Custom:

The correlation coefficient is 0.23, indicating a weak positive linear relationship.

## 3. Histograms or Density Plots

The diagonal contains histograms or density plots for individual variables, which show the distribution of each variable:

### Displacement (ccm):

The histogram shows a right-skewed distribution, indicating that most motorcycles have smaller displacements.

### Gearbox:

The histogram indicates that the number of gears is typically between 4 and 7, with most motorcycles having either 5 or 6 gears.

### Cylinder\_Single:

The density plot indicates a left-skewed distribution, meaning that most motorcycles do not have a single-cylinder engine.

### Category\_Naked:

The density plot shows that a smaller proportion of motorcycles fall into the "Naked" category.

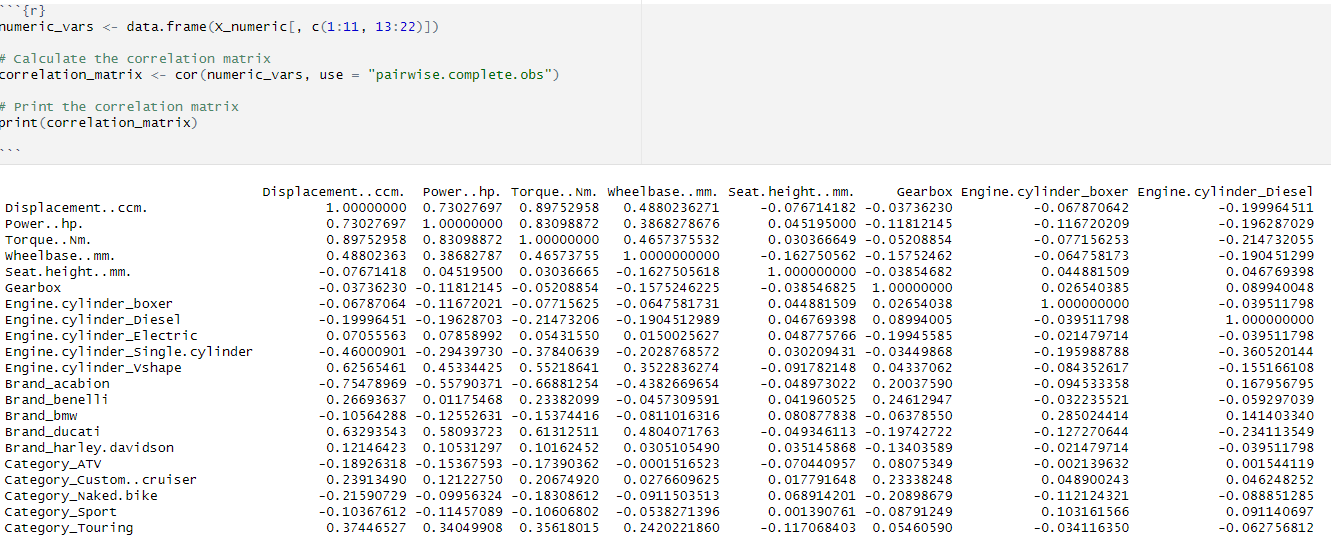
# Conclusion

The pair plot provides insights into the relationships between the motorcycle specifications and categories. The negative relationship between displacement and single-cylinder engines, and the weak relationships between displacement, gear count, and categories, highlight potential trends in the data. The plot serves as a valuable exploratory tool for understanding how these variables interact.

# Feature selection:

## Correlation matrix

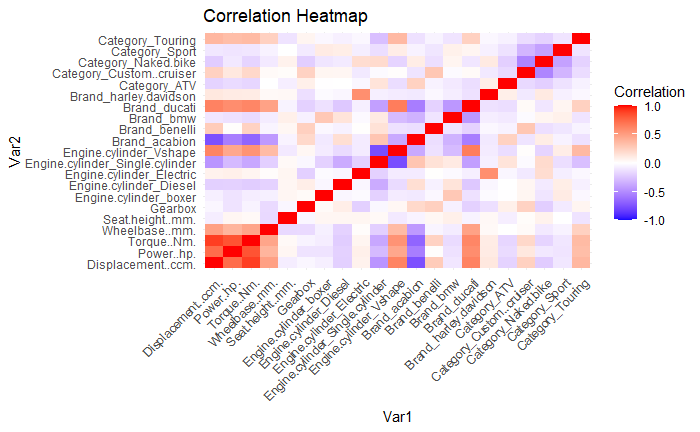
correlation\_matrix <- cor(numeric\_vars, use = "pairwise.complete.obs")



Correlation matrix effectively prepares and calculates a correlation matrix for selected numerical variables, providing valuable insights into linear relationships, aiding feature selection, exploratory data analysis, and detecting multicollinearity. This step is crucial for preparing data for statistical modeling and machine learning.

# Correlation heatmap:

A heatmap visually represents the correlation between pairs of features. The key components of the heatmap include a color scale, which indicates the strength and direction of the correlation. Red represents a strong positive correlation (+1.0), while blue signifies a strong negative correlation (-1.0). White or lighter colors represent weaker correlations. The diagonal elements are all red because they depict the correlation of each variable with itself, which is always +1.0. The off-diagonal elements represent the correlation between different pairs of features. The closer the value is to +1 or -1, the stronger the relationship.



From the heatmap, several key observations can be made. For positive correlations, Displacement (ccm) and Power (hp) exhibit a strong positive correlation, indicating that larger engine displacement tends to be associated with higher power output. Similarly, Wheelbase (mm) and Seat Height (mm) are positively correlated, suggesting that motorcycles with longer wheelbases tend to have higher seats. Additionally, Category\_Touring and Category\_Custom.cruiser are positively correlated, which might indicate overlap in features or similarities in these types of motorcycles.

Conversely, negative correlations were observed between Category\_Naked.bike and Category\_Touring, indicating that these types of motorcycles have distinct and contrasting features. The engine cylinder types Single and VShape also show a negative correlation, highlighting their different motorcycle characteristics.

When considering the top features, Displacement (ccm) shows strong correlations with several other features, including Power (hp), Torque (Nm), Wheelbase (mm), and Seat Height (mm), suggesting that engine displacement is a key determinant of motorcycle specifications. Power (hp), similarly, correlates strongly with these features, emphasizing its importance in the dataset. The feature Engine.cylinder\_Single has strong negative correlations with Engine.cylinder\_VShape, Engine.cylinder\_Eld, and several categories, indicating that motorcycles with single-cylinder engines are distinct from those with other engine configurations. Lastly, Category\_Touring exhibits strong negative correlations with several categories, including Category\_Naked.bike and Category\_Sport, highlighting its unique characteristics among the categories.

# PCA Analysis:

Principal Component Analysis (PCA) is a technique for dimensionality reduction that transforms the features of a dataset into a set of uncorrelated components called principal components. PCA decreases the number of variables in a dataset while preserving most of the variability or information, making it particularly useful for high-dimensional datasets. Additionally, PCA creates new features that are linear combinations of the original features, thereby extracting essential information from the data.

PCA is often used for feature selection and dimensionality reduction as it helps in minimizing redundancy by pinpointing the most important features based on their contribution to the total variance, thus eliminating redundant or less informative features. By reducing the number of features, PCA streamlines models, making them easier to interpret and reducing the risk of overfitting. Moreover, with fewer features, machine learning models become more efficient in terms of computational resources and time. Furthermore, PCA enhances visualization by projecting high-dimensional data into a lower-dimensional space, making it easier to understand patterns or clusters.

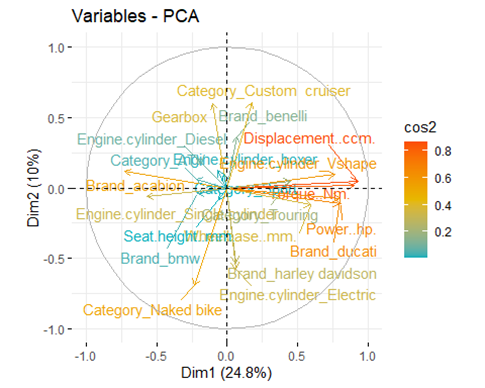
The code uses fviz\_pca\_var from the factoextra package to visualize the variables on the PCA factor map

*# Color by cos2 values: quality on the factor map*

fviz\_pca\_var(res.pca, col.var = "cos2",

gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),

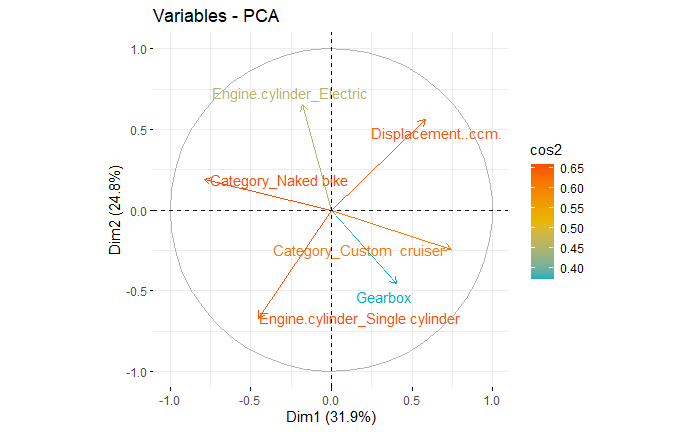
repel = TRUE # Avoid text overlapping



The provided PCA plot visualizes the relationships and importance of various features in a motorcycle dataset using the first two principal components. The plot illustrates how different variables contribute to the overall variability captured by the PCA.

Variable contributions are represented by the cos2 values, which are indicated by the color gradient. Variables with higher cos2 values are better represented in the plot, meaning they contribute more to the principal components shown.

Enhanced and vivid view of shortlisted features based upon the above PCA:



As a keynote, it is highly recommended to scale the data before we perform the actions  
“corrplot” and “heatmap”.



scaled\_data <- scale(data\_selected): The scale() function standardizes the variables in data\_selected to have a mean of zero and a standard deviation of one.

Now, to understand better and to conclude the feature selection and see the correlation strength among the features we draw another “corrplot”(1a) and “heatmap”(1b) with numerical representation using the features that has been chosen from the enhanced PCA(above image) and by excluding the variable “Torque” to avoid multicollinearity.

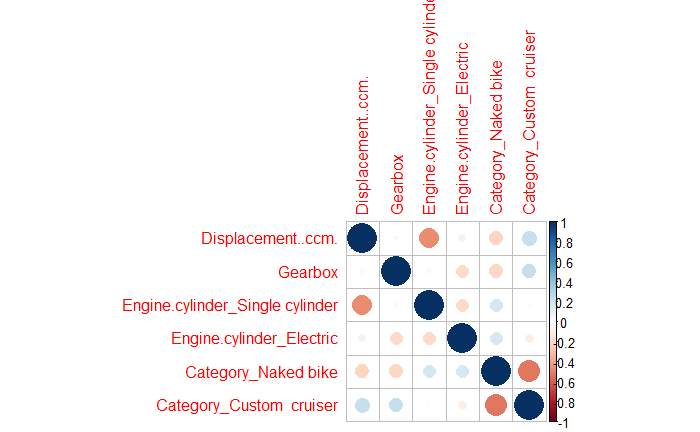


Image 1a

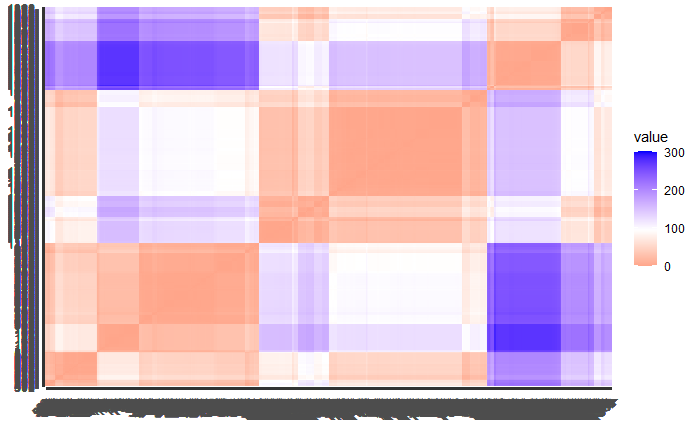


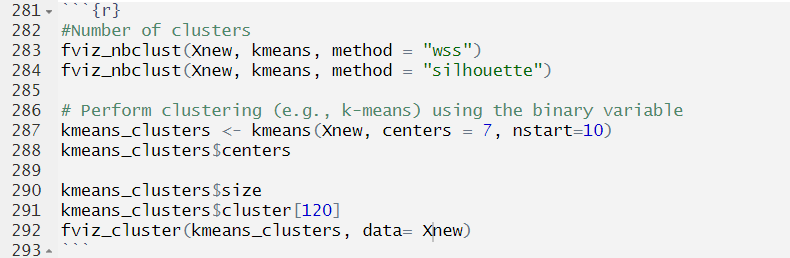
Image 1b

As a next step, we find the optimal number of clusters for the model as it gives us rough idea of how many clusters we need because as an analyst to select the accurate number of clusters is a difficult task and also, the optimal values given the “WSS Elbow Methos” and “Silhouette Method” is not the ultimatum. The choice of optimal clusters depends upon the problem and the challenges faced in the real world with multiple factors impacting the result.  
  
Machine Learning Models

## Optimal number of clusters:

fviz\_nbclust(Xnew, kmeans, method = "wss"): This line uses the "within-cluster sum of squares" (WSS) method to identify the optimal number of clusters. The WSS method looks for an "elbow" in the plot, which suggests the optimal number of clusters.

fviz\_nbclust(Xnew, kmeans, method = "silhouette"): This line uses the silhouette method to identify the optimal number of clusters. The silhouette method measures how similar an object is to its own cluster compared to other clusters, with higher average silhouette widths indicating better cluster configurations.



kmeans\_clusters <- kmeans(Xnew, centers = 7, nstart = 10): The kmeans() function performs k-means clustering with 7 centers and starts the algorithm 10 times to select the best result.

kmeans\_clusters$centers: This extracts the cluster centers, which represent the mean values for each feature in each cluster.

kmeans\_clusters$size: This gives the size of each cluster, indicating how many observations belong to each cluster.

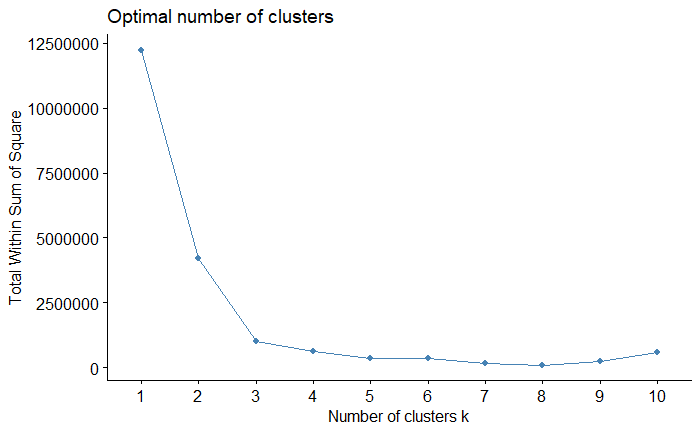
kmeans\_clusters$cluster[120]: This gives the cluster assignment for the 120th observation, showing which cluster it belongs to.

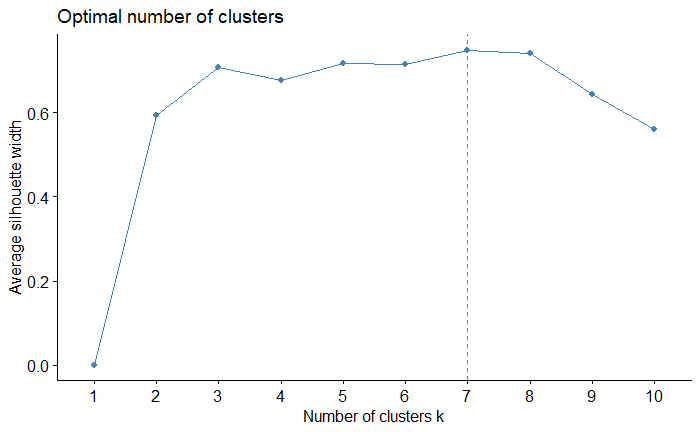
fviz\_cluster(kmeans\_clusters, data = Xnew): The fviz\_cluster() function creates a plot showing the data points colored by their cluster assignments. The cluster centers are also displayed on the plot.

# Elbow method:

## Plot Explanation:

* The y-axis represents the "Total Within Sum of Squares" (WSS), which measures the variance within each cluster.
* The x-axis represents the number of clusters, 𝑘k.
* The plot shows the WSS decreasing as the number of clusters increases.
* The optimal number of clusters is typically found at the "elbow" point, where the rate of decrease slows down significantly.
* In this plot, the elbow appears to be around 𝑘=3. 𝑘=4

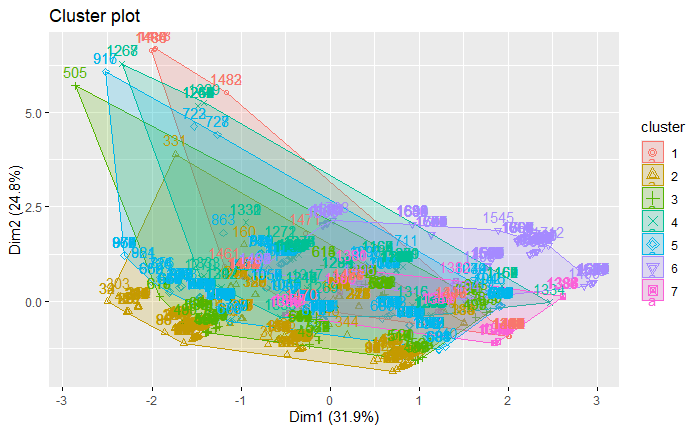
k=4.

Silhouette method:  


## Plot Explanation:

* The y-axis shows the "Average Silhouette Width," which measures how similar each point is to its own cluster compared to other clusters.
* The x-axis shows the number of clusters, 𝑘k.
* The plot shows how the silhouette width changes as the number of clusters increases.
* The optimal number of clusters is typically found at the peak silhouette width, which indicates the best cluster configuration.
* In this plot, the optimal number of clusters appears to be around 𝑘=2 or 𝑘=3.
* Now that, we concluded feature selection and optimal number of clusters for our model. We move onto the next step i.e., implementation of Machine Learning model K-Means and DBScan.

K-Means:  
 K-means clustering is a popular unsupervised machine learning algorithm used to partition data into \(k\) distinct, non-overlapping clusters. The goal is to minimize the sum of squared distances between each data point and its assigned centroid. The algorithm involves initializing \(k\) centroids, assigning each data point to the nearest centroid, updating the centroids as the mean of their assigned points, and repeating until convergence. K-means is useful for data segmentation, pattern recognition, and feature compression. To interpret the results, one should consider the optimal number of clusters, which can be determined using the elbow method or silhouette analysis, the cluster centroids that summarize typical data points, the cluster sizes that reveal the distribution of data points, and the cluster assignments that show group composition and potential outliers. By examining these aspects, we can gain insights into the data structure and make informed decisions based on the clustering results.

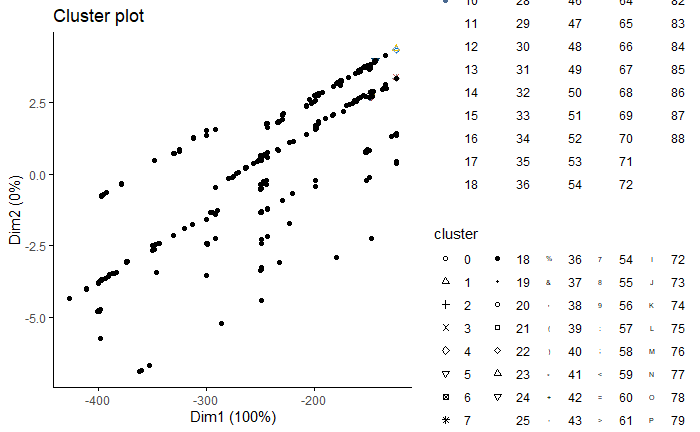


The provided scatterplot illustrates the outcomes of k-means clustering on a dataset, with the x-axis representing PC1 and the y-axis representing PC2, two principal components derived from the data. Each dot represents a data point, color-coded by its assigned cluster, while larger dots or stars indicate the centroids of each cluster. This visualization effectively showcases the grouping of data points into distinct clusters, emphasizing their distribution and separation. The centroids provide insight into the average characteristics of each cluster, aiding in understanding their similarities or differences. By using principal components, the plot captures the most significant variance in the data, offering a clear depiction of its structure and variability.

## DBScan:

DBSCAN, which stands for Density-Based Spatial Clustering of Applications with Noise, is a popular clustering algorithm used to identify clusters in datasets of varying shapes and sizes. Unlike traditional clustering algorithms like K-means, DBSCAN doesn't require specifying the number of clusters beforehand, making it particularly useful for datasets with irregular cluster shapes.

The key idea behind DBSCAN is to group together closely packed data points based on their density. It defines two parameters: epsilon (ε), which specifies the radius within which to search for neighboring points, and minPoints, which sets the minimum number of points required to form a dense region or cluster.



The x-axis (Dim1) represents the first principal component or dimension of the data, which explains 100% of the variance. The y-axis (Dim2) represents the second principal component or dimension of the data, which explains 0% of the variance. This likely indicates that the data is essentially one-dimensional or that Dim2 represents noise. The plot displays several clusters labelled from 0 to 18. The label "0" represents the noise points, which are data points that do not fit into any cluster Each shape on the plot corresponds to a different cluster, and the legend indicates which symbols correspond to which cluster. The clusters seem to be aligned in a linear pattern, which might indicate a linear relationship between the data points. There are several distinct clusters aligned along different lines. The presence of a substantial amount of noise (cluster 0) suggests that many points were not sufficiently close to any cluster centre, or were in sparsely populated areas. The ellipses are not shown, which is often a choice made to simplify the visualization or when the clusters are clearly linear in nature. The DBSCAN algorithm has effectively grouped the data into several distinct clusters, excluding a number of noise points. The clustering suggests that the data might have a linear pattern with several distinct groups.

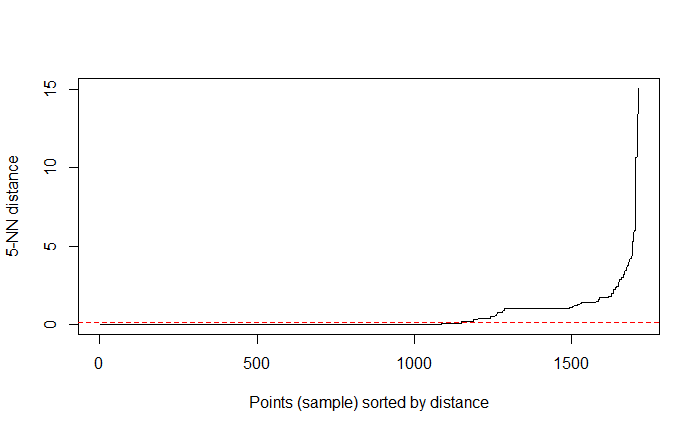
# KNN-Distance plot:

Using the code below, we calculated the distance

# Compute k-nearest neighbors (kNN) distance plot

kNNdistplot(Xnew, k = 5)

abline(h = 0.15, col = "red", lty = 2)



kNN Distance Plot: The kNNdistplot() function generates a plot showing the distances of each point in the dataset (Xnew) to its k-nearest neighbors. This plot helps visualize the distribution of distances and identify any potential clusters or patterns in the data. By examining the distances between points, we can gain insights into the underlying structure of the dataset and make informed decisions about clustering or classification tasks.

## Performance evaluation:

The silhouette width is a measure used to evaluate the quality of clustering algorithms. It quantifies how well-clustered the data points are within their respective clusters, compared to their proximity to points in other clusters. Here's a summary of the comparison between K-means and DBSCAN clustering algorithms based on their average silhouette widths:

### 1. K-means Clustering:

* Average Silhouette Width: 0.7779414
* Interpretation: The high silhouette width indicates that the K-means algorithm has performed well in clustering the data.
* Implication: The clusters generated by K-means are well-separated, and each data point is closer to the center of its own cluster than to the centers of other clusters.

### 2. DBSCAN Clustering:

* Average Silhouette Width: 0.4365029
* Interpretation: The lower silhouette width suggests that the clustering produced by the DBSCAN algorithm is less satisfactory.
* Implication: The clusters from DBSCAN might not be well-separated, or there could be many points classified as noise, leading to a lower overall silhouette width.
* In short, higher average silhouette width of K-means indicates better clustering performance compared to DBSCAN, suggesting more distinct and well-defined clusters in the data.

# Conclusion:

The clustering analysis reveals that K-means provided a superior result, as evidenced by its higher average silhouette width. This suggests that the clusters formed by K-means are more cohesive and better separated compared to those formed by DBSCAN. The poorer performance of DBSCAN might be attributed to inappropriate parameter settings or incompatible cluster shapes for its assumptions. In order to enhance DBSCAN's performance, one could start by tuning its parameters, particularly `eps` and `MinPts`, to better align with the underlying data structure. Additionally, it's crucial to analyze the shapes of the clusters present in the data, as DBSCAN typically excels with non-spherical or varied density clusters. Proper parameter tuning could allow DBSCAN to outperform K-means in such scenarios. However, if the data doesn't contain irregular shapes or if noise handling is not a priority, sticking with K-means might be a more suitable approach.

References:  
1. <https://www.geeksforgeeks.org/k-means-clustering-introduction/>

2. <http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/154-stepwise-regression-essentials-in-r/>

3. <https://www.r-bloggers.com/2021/05/principal-component-analysis-pca-in-r/>

4. <https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/>

5. <https://www.r-bloggers.com/2020/01/how-to-remove-outliers-in-r/>

6.https://www.kaggle.com/datasets/emmanuelfwerr/motorcycle-technical-specifications-19702022