Canine Insights

Pradhakshana Duraiswamy

2024-03-27

R Markdown

Introduction

Welcome to the documentation for the Dog Breed Analysis project. This project aims to provide comprehensive insights into various attributes of dog breeds, leveraging data collected from multiple sources including the American Kennel Club (AKC) website and other publicly available datasets.

Dataset Overview

The dataset utilized in this analysis contains information about 277 different dog breeds, sourced from the AKC and other sources. It includes details such as breed characteristics, grooming requirements, energy levels, trainability, and more. Additionally, to enrich the dataset, information on the popularity rankings of dog breeds for the years 2018 to 2022 was extracted from the American Kennel Club website using web scraping techniques with Beautiful Soup.

Disclaimer

It's important to note that the popularity rankings provided by the AKC are based solely on registration statistics. Popularity is determined by the number of registrations for each breed and does not necessarily reflect factors such as entertainment value, intelligence, or appearance. This disclaimer serves to clarify the basis of the popularity rankings utilized in this analysis.

Throughout this documentation, you will find detailed explanations of the various stages of the project, including data cleaning, exploration, feature engineering, and clustering. Visualizations and insights generated from the analysis will also be presented, providing valuable information for breeders, owners, and enthusiasts interested in understanding the characteristics and trends of different dog breeds.

Let's delve into the details of each stage of the analysis and explore the fascinating world of dog breeds!

Load necessary libraries

```r
library(readr)
library(ggplot2)
```

```
## Warning: package ’ggplot2’ was built under R version 4.3.3
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by ’GGally’:
##   method from
##   +.gg   ggplot2
```

```r
library("corrplot")
```

```
## Warning: package ’corrplot’ was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```r
library(rpart)
```

```
## Warning: package ’rpart’ was built under R version 4.3.2
```

```r
library(rpart.plot)
library(cluster)
```

```
## Warning: package ’cluster’ was built under R version 4.3.3
```

```r
library(ggrepel)
```

```
## Warning: package ’ggrepel’ was built under R version 4.3.3
```

Set the working directory and read the file

```r
getwd()
```

```
## [1] "C:/Users/satba/OneDrive/Desktop/MS/Coursera/Capstone_8/Breeds"
```

```r
setwd("C:/Users/satba/OneDrive/Desktop/MS/Coursera/Capstone_8/Breeds")
data<-read_csv("breeds.csv")
```

```
## Rows: 277 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (12): Breed, description, temperament, popularity_2018, avg_weight(kg), ...
## dbl (12): min_height, max_height, avg_height(cm), min_weight, max_weight, mi...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
head(data)
```

```
## # A tibble: 6 x 24
##   Breed             description temperament popularity_2018 min_height max_height
##   <chr>             <chr>       <chr>       <chr>                <dbl>      <dbl>
## 1 Affenpinscher     The Affen’~ Confident,~ 148                   22.9       29.2
## 2 Afghan Hound      The Afghan~ Dignified,~ 113                   63.5       68.6
## 3 Airedale Terrier The Aireda~ Friendly, ~ 60                    58.4       58.4
## 4 Akita             Akitas are~ Courageous~ 47                    61.0       71.1
## 5 Alaskan Malamute  The Alaska~ Affectiona~ 58                    58.4       63.5
## 6 American Bulldog  The Americ~ Loyal, Sel~ <NA>                  50.8       63.5
## # i 18 more variables: ‘avg_height(cm)‘ <dbl>, min_weight <dbl>,
## #   max_weight <dbl>, ‘avg_weight(kg)‘ <chr>, min_expectancy <dbl>,
## #   max_expectancy <dbl>, ‘avg_lifespan(years)‘ <chr>, group <chr>,
## #   grooming_frequency_value <dbl>, grooming_frequency_category <chr>,
## #   shedding_value <dbl>, shedding_category <chr>, energy_level_value <dbl>,
## #   energy_level_category <chr>, trainability_value <dbl>,
## #   trainability_category <chr>, demeanor_value <dbl>, ...
```

```r
colnames(data)
```

```
##  [1] "Breed"                       "description"
##  [3] "temperament"                 "popularity_2018"
##  [5] "min_height"                  "max_height"
##  [7] "avg_height(cm)"              "min_weight"
##  [9] "max_weight"                  "avg_weight(kg)"
## [11] "min_expectancy"              "max_expectancy"
## [13] "avg_lifespan(years)"         "group"
## [15] "grooming_frequency_value"    "grooming_frequency_category"
## [17] "shedding_value"              "shedding_category"
## [19] "energy_level_value"          "energy_level_category"
## [21] "trainability_value"          "trainability_category"
## [23] "demeanor_value"              "demeanor_category"
```

```r
dim(data)
```

```
## [1] 277  24
```

1) Data Cleaning

Created avg__weight and avg__height columns using min and max values provided for them in Excel

Subsetting the dataset and keeping only the necessary columns

Converting the chr() variables to factor()

Removing missing values

```r
##Subset the data-set to include only the necessary variables
#?subset()
new_data<-subset(data,select=-c(description,popularity_2018,min_height,
                                max_height,min_weight,max_weight,min_expectancy,
                                max_expectancy))
head(new_data)
```

```
## # A tibble: 6 x 16
##   Breed      temperament ‘avg_height(cm)‘ ‘avg_weight(kg)‘ ‘avg_lifespan(years)‘
##   <chr>      <chr>                  <dbl> <chr>            <chr>
## 1 Affenpins~ Confident,~             26.0 3.855535145      13.5
## 2 Afghan Ho~ Dignified,~             66.0 24.94758035      13.5
## 3 Airedale ~ Friendly, ~             58.4 27.2155422       12.5
## 4 Akita      Courageous~             66.0 45.359237        11.5
## 5 Alaskan M~ Affectiona~             61.0 36.2873896       12
## 6 American ~ Loyal, Sel~             57.2 36.2873896       11
## # i 11 more variables: group <chr>, grooming_frequency_value <dbl>,
## #   grooming_frequency_category <chr>, shedding_value <dbl>,
## #   shedding_category <chr>, energy_level_value <dbl>,
## #   energy_level_category <chr>, trainability_value <dbl>,
## #   trainability_category <chr>, demeanor_value <dbl>, demeanor_category <chr>
```

```r
dim(new_data)
```

```
## [1] 277  16
```

```r
colnames(new_data)
```

```
##  [1] "Breed"                       "temperament"
##  [3] "avg_height(cm)"              "avg_weight(kg)"
##  [5] "avg_lifespan(years)"         "group"
##  [7] "grooming_frequency_value"    "grooming_frequency_category"
##  [9] "shedding_value"              "shedding_category"
## [11] "energy_level_value"          "energy_level_category"
## [13] "trainability_value"          "trainability_category"
## [15] "demeanor_value"              "demeanor_category"
```

```r
str(new_data)
```

```
## tibble [277 x 16] (S3: tbl_df/tbl/data.frame)
##  $ Breed                      : chr [1:277] "Affenpinscher" "Afghan Hound" "Airedale Terrier" "Akita
##  $ temperament                : chr [1:277] "Confident, Famously Funny, Fearless" "Dignified, Profou
##  $ avg_height(cm)             : num [1:277] 26 66 58.4 66 61 ...
##  $ avg_weight(kg)             : chr [1:277] "3.855535145" "24.94758035" "27.2155422" "45.359237" ...
##  $ avg_lifespan(years)        : chr [1:277] "13.5" "13.5" "12.5" "11.5" ...
##  $ group                      : chr [1:277] "Toy Group" "Hound Group" "Terrier Group" "Working Group
##  $ grooming_frequency_value   : num [1:277] 0.6 0.8 0.6 0.8 0.6 0.2 0.2 0.4 0.2 NA ...
##  $ grooming_frequency_category: chr [1:277] "2-3 Times a Week Brushing" "Daily Brushing" "2-3 Times a
##  $ shedding_value             : num [1:277] 0.6 0.2 0.4 0.6 0.6 0.6 0.4 0.6 0.6 NA ...
##  $ shedding_category          : chr [1:277] "Seasonal" "Infrequent" "Occasional" "Seasonal" ...
##  $ energy_level_value         : num [1:277] 0.6 0.8 0.6 0.8 0.8 0.8 0.8 0.8 0.8 NA ...
##  $ energy_level_category      : chr [1:277] "Regular Exercise" "Energetic" "Regular Exercise" "Energ
##  $ trainability_value         : num [1:277] 0.8 0.2 1 1 0.4 0.6 0.6 1 0.4 NA ...
##  $ trainability_category      : chr [1:277] "Easy Training" "May be Stubborn" "Eager to Please" "Eag
##  $ demeanor_value             : num [1:277] 1 0.2 0.8 0.6 0.8 0.6 0.6 1 0.8 NA ...
##  $ demeanor_category          : chr [1:277] "Outgoing" "Aloof/Wary" "Friendly" "Alert/Responsive" ..
```

```
##convert chr() variables to factor()
new_data$Breed <- as.factor(new_data$Breed)
new_data$temperament <- as.factor(new_data$temperament)
new_data$group<-as.factor(new_data$group)
new_data$grooming_frequency_category<-as.factor(new_data$grooming_frequency_category)
new_data$shedding_category<-as.factor(new_data$shedding_category)
new_data$energy_level_category<-as.factor(new_data$energy_level_value)
new_data$trainability_category<-as.factor(new_data$trainability_category)
new_data$demeanor_category<-as.factor(new_data$demeanor_category)
new_data$`avg_height(cm)` <- as.numeric(new_data$`avg_height(cm)`)
new_data$`avg_weight(kg)`<-as.numeric(new_data$`avg_weight(kg)`)
```

```
## Warning: NAs introduced by coercion
```

```
new_data$`avg_lifespan(years)` <- as.numeric(new_data$`avg_lifespan(years)`)
```

```
## Warning: NAs introduced by coercion
```

```
str(new_data)
```

```
## tibble [277 x 16] (S3: tbl_df/tbl/data.frame)
##  $ Breed                      : Factor w/ 277 levels "Affenpinscher",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ temperament                : Factor w/ 267 levels "Active, Outgoing, Sweet-Natured",..: 79 103 13
##  $ avg_height(cm)             : num [1:277] 26 66 58.4 66 61 ...
##  $ avg_weight(kg)             : num [1:277] 3.86 24.95 27.22 45.36 36.29 ...
##  $ avg_lifespan(years)        : num [1:277] 13.5 13.5 12.5 11.5 12 11 11.5 14 12 15 ...
##  $ group                      : Factor w/ 9 levels "Foundation Stock Service",..: 8 3 7 9 9 1 3 5 3
##  $ grooming_frequency_value   : num [1:277] 0.6 0.8 0.6 0.8 0.6 0.2 0.2 0.4 0.2 NA ...
##  $ grooming_frequency_category: Factor w/ 5 levels "2-3 Times a Week Brushing",..: 1 2 1 2 1 3 3 5 3
##  $ shedding_value             : num [1:277] 0.6 0.2 0.4 0.6 0.6 0.6 0.4 0.6 0.6 NA ...
##  $ shedding_category          : Factor w/ 5 levels "Frequent","Infrequent",..: 5 2 3 5 5 5 3 5 5 NA
##  $ energy_level_value         : num [1:277] 0.6 0.8 0.6 0.8 0.8 0.8 0.8 0.8 0.8 NA ...
##  $ energy_level_category      : Factor w/ 5 levels "0.2","0.4","0.6",..: 3 4 3 4 4 4 4 4 4 NA ...
##  $ trainability_value         : num [1:277] 0.8 0.2 1 1 0.4 0.6 0.6 1 0.4 NA ...
##  $ trainability_category      : Factor w/ 5 levels "Agreeable","Eager to Please",..: 3 5 2 2 4 1 1 2
##  $ demeanor_value             : num [1:277] 1 0.2 0.8 0.6 0.8 0.6 0.6 1 0.8 NA ...
##  $ demeanor_category          : Factor w/ 5 levels "Alert/Responsive",..: 4 2 3 1 3 1 1 4 3 NA ...
```

```
summary(new_data)
```

```
##                Breed                                  temperament
##  Affenpinscher    :  1   Friendly, Smart, Willing to Please:  3
##  Afghan Hound     :  1   Active, Proud, Very Smart         :  2
##  Airedale Terrier:  1   Affectionate, Smart, Energetic    :  2
##  Akita           :  1   Friendly, Alert, Intelligent      :  2
##  Alaskan Malamute:  1   Loyal, Alert, Intelligent         :  2
##  American Bulldog:  1   (Other)                           :265
##  (Other)         :271   NA's                              :  1
##  avg_height(cm)  avg_weight(kg)  avg_lifespan(years)
##  Min.   :16.51   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:36.83   1st Qu.:10.55   1st Qu.:11.50
##  Median :49.53   Median :20.41   Median :13.00
##  Mean   :48.47   Mean   :22.59   Mean   :12.57
##  3rd Qu.:60.96   3rd Qu.:28.35   3rd Qu.:13.50
##  Max.   :80.01   Max.   :81.65   Max.   :17.00
##                  NA's   :2       NA's   :3
##                        group    grooming_frequency_value
##  Foundation Stock Service:68   Min.   :0.2000
##  Hound Group             :32   1st Qu.:0.2000
##  Sporting Group          :32   Median :0.4000
##  Terrier Group           :31   Mean   :0.4259
##  Working Group           :31   3rd Qu.:0.6000
##  Herding Group           :30   Max.   :1.0000
##  (Other)                 :53   NA's   :7
##              grooming_frequency_category shedding_value    shedding_category
##  2-3 Times a Week Brushing: 50          Min.   :0.2000   Frequent  :  6
##  Daily Brushing           : 18          1st Qu.:0.4000   Infrequent: 37
##  Occasional Bath/Brush    : 75          Median :0.6000   Occasional: 59
##  Specialty/Professional   :  8          Mean   :0.5292   Regularly : 30
##  Weekly Brushing          :119          3rd Qu.:0.6000   Seasonal  :125
##  NA's                     :  7          Max.   :1.0000   NA's      : 20
##                                         NA's   :20
##  energy_level_value energy_level_category trainability_value
##  Min.   :0.2000     0.2 :  1              Min.   :0.2000
##  1st Qu.:0.6000     0.4 : 19              1st Qu.:0.4000
##  Median :0.6000     0.6 :118              Median :0.6000
##  Mean   :0.7129     0.8 : 92              Mean   :0.6245
##  3rd Qu.:0.8000     1   : 41              3rd Qu.:0.8000
##  Max.   :1.0000     NA's:  6              Max.   :1.0000
##  NA's   :6                                NA's   :24
##      trainability_category demeanor_value                   demeanor_category
##  Agreeable      :77         Min.   :0.2000   Alert/Responsive       :75
##  Eager to Please:50         1st Qu.:0.4000   Aloof/Wary             : 8
##  Easy Training  :39         Median :0.6000   Friendly               :77
##  Independent    :66         Mean   :0.6206   Outgoing               :19
##  May be Stubborn:21         3rd Qu.:0.8000   Reserved with Strangers:73
##  NA's           :24         Max.   :1.0000   NA's                   :25
##                             NA's   :25
```

```
##look for missing values
anyNA(new_data)
```

```
## [1] TRUE
```

```
new_data <- na.omit(new_data)
dim(new_data) ##235 X 16
```

```
## [1] 235  16
```

##To make the data more interesting let's scrap popularity ranking of dog breeds for years (2018-2022) using Beautiful soup and clean and read them here.

```
popularity_2018<-read_csv("popularity_2018.csv")
```

```
## Rows: 192 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (1): Rank_2018
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
popularity_2019<-read_csv("popularity_2019.csv")
```

```
## Rows: 195 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (1): Rank_2019
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
popularity_2020<-read_csv("popularity_2020.csv")
```

```
## Rows: 195 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (1): Rank_2020
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
popularity_2021<-read_csv("popularity_2021.csv")
```

```
## Rows: 197 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (1): Rank_2021
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
popularity_2022<-read_csv("popularity_2022.csv")
```

```
## Rows: 199 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Breed
## dbl (1): Rank_2022
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Merge popularity ranking data for different years with the main data-set after matching the breed names in all CSV files.

```
new_data <- merge(new_data, popularity_2018, by = "Breed", all.x = TRUE)

new_data <- merge(new_data, popularity_2019, by = "Breed", all.x = TRUE)

new_data <- merge(new_data, popularity_2020, by = "Breed", all.x = TRUE)

new_data <- merge(new_data, popularity_2021, by = "Breed", all.x = TRUE)

new_data <- merge(new_data, popularity_2022, by = "Breed", all.x = TRUE)

##Finding the intersection of breed names between multiple data frames
intersection <- Reduce(intersect, list(new_data$Breed, popularity_2018$Breed,
                                       popularity_2019$Breed, popularity_2020$Breed,
                                       popularity_2021$Breed, popularity_2022$Breed))
head(intersection)
```

```
## [1] "Affenpinscher"              "Afghan Hound"
## [3] "Airedale Terrier"           "Akita"
## [5] "Alaskan Malamute"           "American English Coonhound"
```

```
new_data<-na.omit(new_data)
```

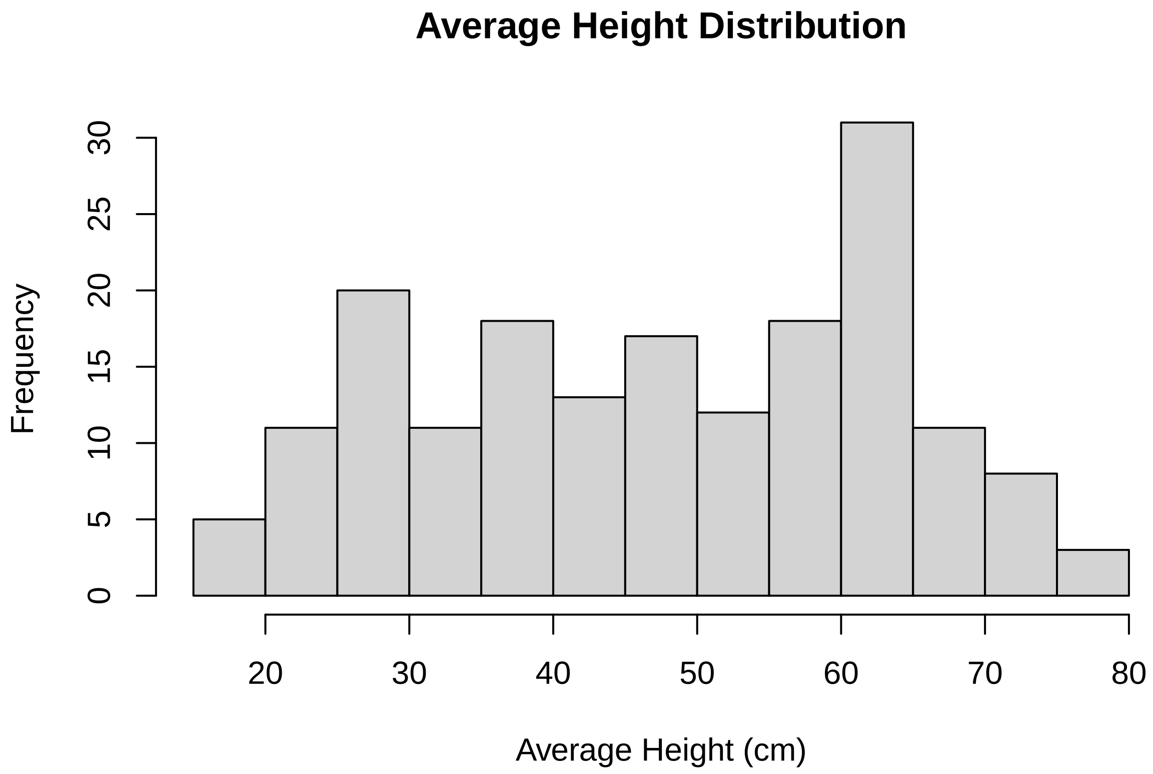
Of the 235 breeds only 178 have the data for popularity ranking columns. Hence, I am eliminating those data that have missing rank values after merging.

2) Data Exploration and Feature Engineering

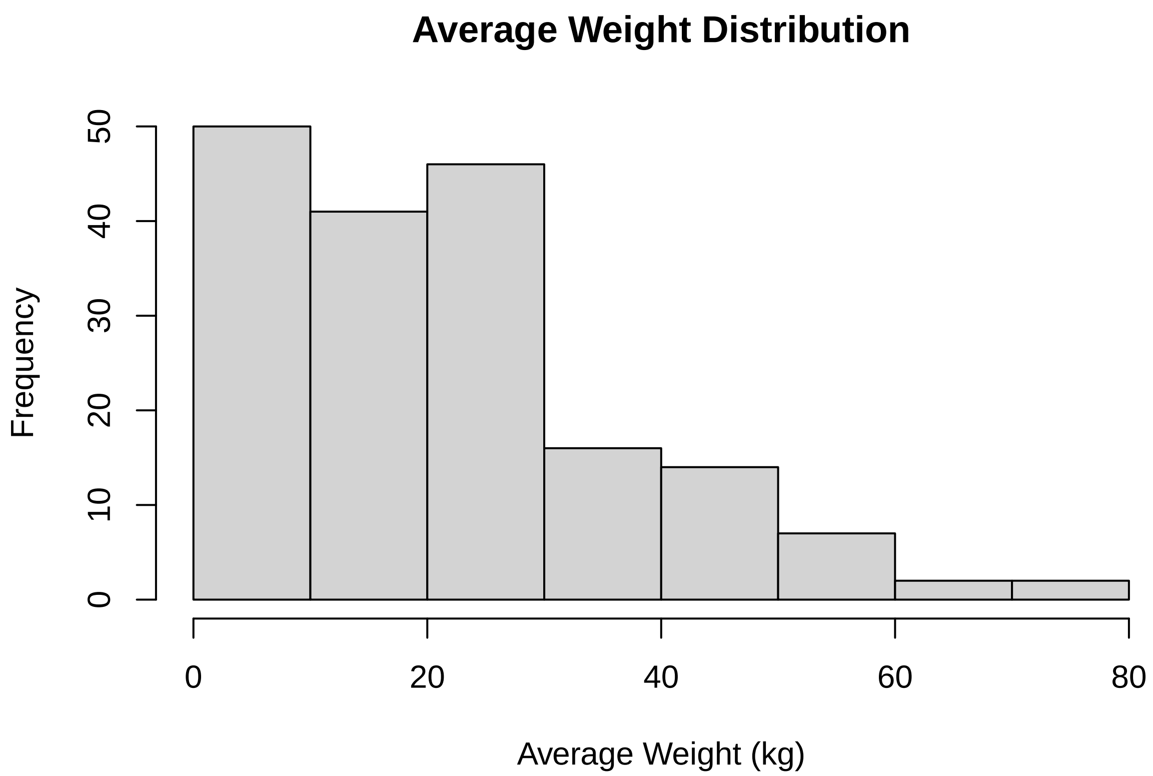
Used various visualizations to get insights about the data

Used histogram to understand the distribution of weight and height of breeds and based on the understanding decided on the cut-off points to create a new column named size (small, medium, large)

```
##Explore the distribution of height and weight variables using histograms
hist(new_data$`avg_height(cm)`, main = "Average Height Distribution", xlab = "Average Height (cm)")
```



```
hist(new_data$`avg_weight(kg)`, main = "Average Weight Distribution", xlab = "Average Weight (kg)")
```



```r
##Feature engineering for column 'Size'
##Create a size column and categorize dog breeds into small, medium, and large based on height and weig
new_data$Size <- NA  ##Create a new column and initialize with NA values

##Define cutoff points for height and weight categories
height_cutoffs <- c(0, 31, 61, 81)
weight_cutoffs <- c(0, 10, 35, 82)

##Categorize size based on average height and weight
new_data$Size[new_data$`avg_height(cm)` <= height_cutoffs[2] &
                new_data$`avg_weight(kg)` <= weight_cutoffs[2]] <- "Small"
new_data$Size[new_data$`avg_height(cm)` > height_cutoffs[3] &
                new_data$`avg_weight(kg)` > weight_cutoffs[3]] <- "Large"

##For cases that don't fit into Small or Large categories, assign Medium
new_data$Size[is.na(new_data$Size)] <- "Medium"

##Print the first few rows to verify the new column
head(new_data$Size)
```

```
## [1] "Small"  "Medium" "Medium" "Large"  "Medium" "Medium"
```

```r
##Calculate the sum of occurrences of the categories
sum(new_data$Size == "Small")
```

```
## [1] 34
```

```r
sum(new_data$Size == "Medium")
```

```
## [1] 113
```

```r
sum(new_data$Size == "Large")
```

```
## [1] 31
```

```r
##Convert 'Size' column to a factor
new_data$Size <- as.factor(new_data$Size)
```

Feature engineering for column 'score'. Create a score column with data score for each dog breed by assigning weights to variables based on personal knowledge.

```r
##Assigning weights to attributes
weight_longevity <- 0.8
weight_temperament <- 0.8
weight_activity <- 0.6
weight_trainability <- 0.6
weight_grooming <- 0.4
weight_shedding <- 0.4

##Calculate score for each breed
new_data$score <- with(new_data, (new_data$'avg_lifespan(years)' * weight_longevity) +
                         (demeanor_value * weight_temperament) +
                         (energy_level_value * weight_activity) +
                         (trainability_value * weight_trainability) -
                         (grooming_frequency_value * weight_grooming) -
                         (shedding_value * weight_shedding))

##Print first few rows to verify
head(new_data$score)
```

```
## [1] 11.96 11.16 11.20 10.20 10.48 10.28
```

```r
summary(new_data$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.76   10.21   11.20   10.85   11.88   14.76
```

```r
##Write the merged data-set to a CSV file
write.csv(new_data, file = "new_data.csv", row.names = FALSE)
dim(new_data)
```

```
## [1] 178  23
```

This csv file was used to build Dashboards in Tableau. The packaged workbook and the png image of the dashboards can be found in this project folder.

Look for outliers

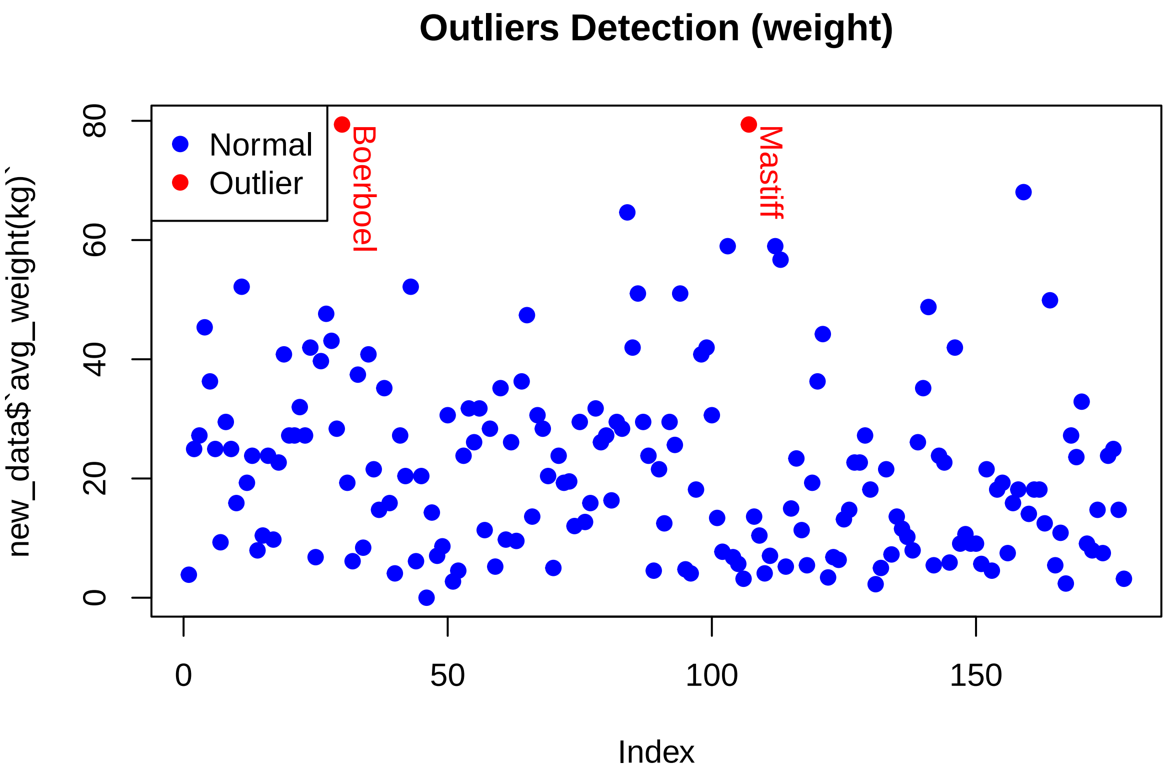
A Z-score, also known as a standard score, is a method to look for outliers and is a measure of how many standard deviations a data point is from the mean of the data-set. It indicates how

far a particular observation is from the mean in terms of standard deviation units.

```
z_scores <- scale(new_data$`avg_weight(kg)`)
outlier_indices <- which(abs(z_scores) > 3)
outlier_indices
```

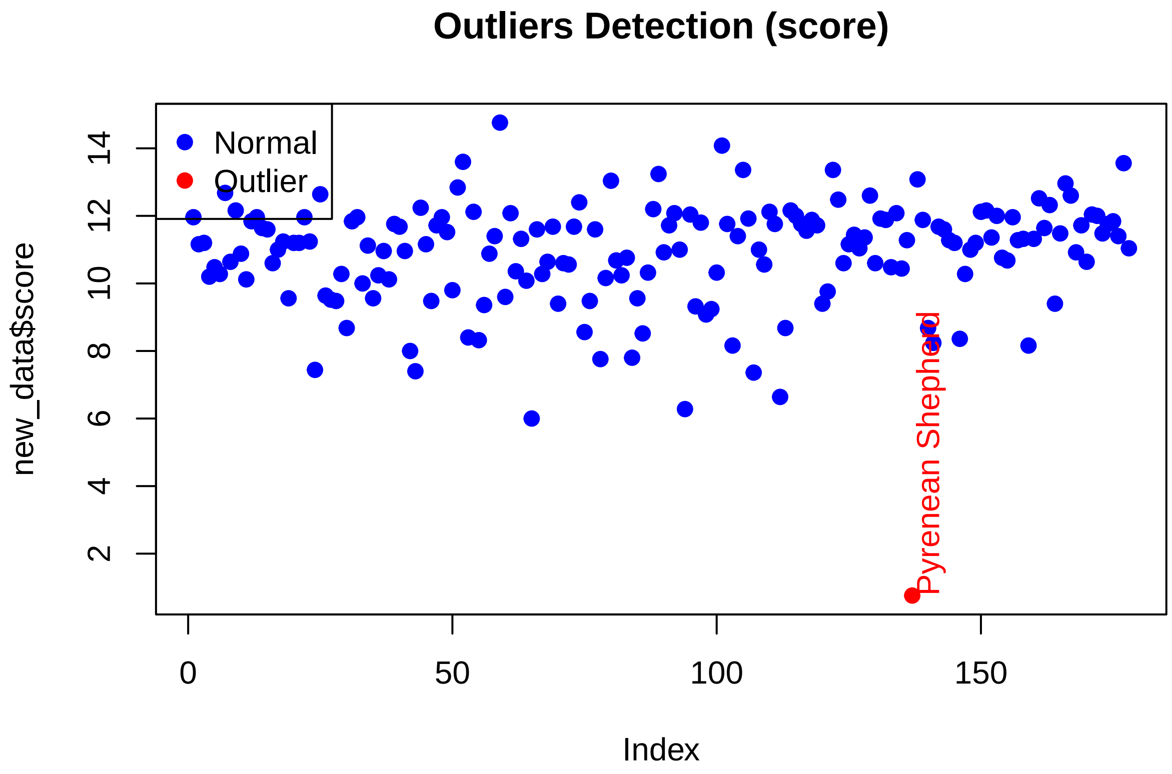
```
## [1]  30 107
```

```
##plot the outliers
plot(new_data$`avg_weight(kg)`, main = "Outliers Detection (weight)", pch = 19,
     col = ifelse(abs(z_scores) > 3, "red", "blue"))
legend("topleft", legend = c("Normal", "Outlier"), col = c("blue", "red"), pch = 19)
text(x = outlier_indices, y = new_data$`avg_weight(kg)`[outlier_indices],
     labels = new_data$Breed[outlier_indices], pos = 4, col = "red", srt = -90)
```



```
z_scores1 <- scale(new_data$score)
outlier_indices <- which(abs(z_scores1) > 3)
```

```
plot(new_data$score, main = "Outliers Detection (score)", pch = 19,
     col = ifelse(abs(z_scores1) > 3, "red", "blue"))
legend("topleft", legend = c("Normal", "Outlier"), col = c("blue", "red"), pch = 19)
text(x = outlier_indices, y = new_data$score[outlier_indices],
     labels = new_data$Breed[outlier_indices], pos = 4, col = "red", srt = 90)
```



3) Further Exploration

Did further exploration with data using visualizations.

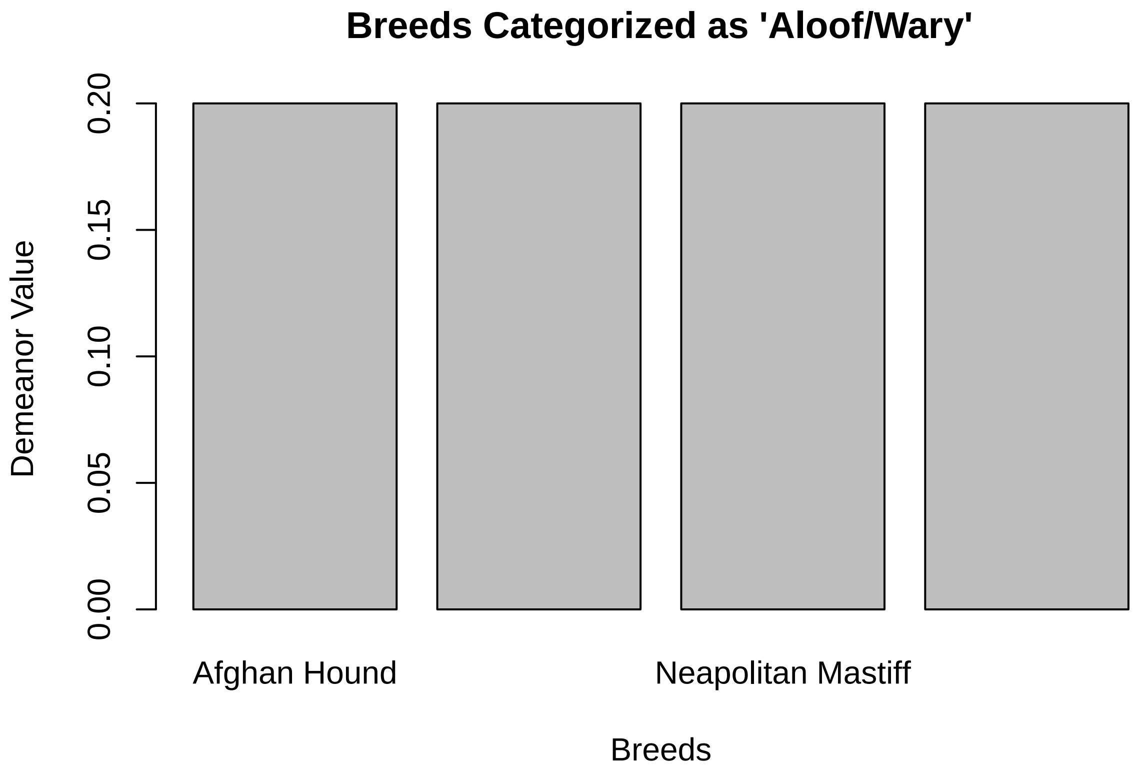
Created a data frame with only 3 variables and did some visualizations to understand the data.

```
##Looking for dog breeds with a particular demeanor category.
##Create a data frame to store breed information along with demeanor values and categories
df <- data.frame(new_data$Breed, new_data$demeanor_value, new_data$demeanor_category)

##Rename columns for better readability
colnames(df) <- c("Breed", "Demeanor Value", "Demeanor Category")

##Filter the data frame for breeds categorized as "Aloof/Wary"
```

```
aloof_breeds <- subset(df, new_data$demeanor_category == "Aloof/Wary")
aloof_breeds
```

```
##                       Breed Demeanor Value Demeanor Category
## 2              Afghan Hound            0.2        Aloof/Wary
## 95        Italian Greyhound            0.2        Aloof/Wary
## 112      Neapolitan Mastiff            0.2        Aloof/Wary
## 130 Polish Lowland Sheepdog            0.2        Aloof/Wary
```

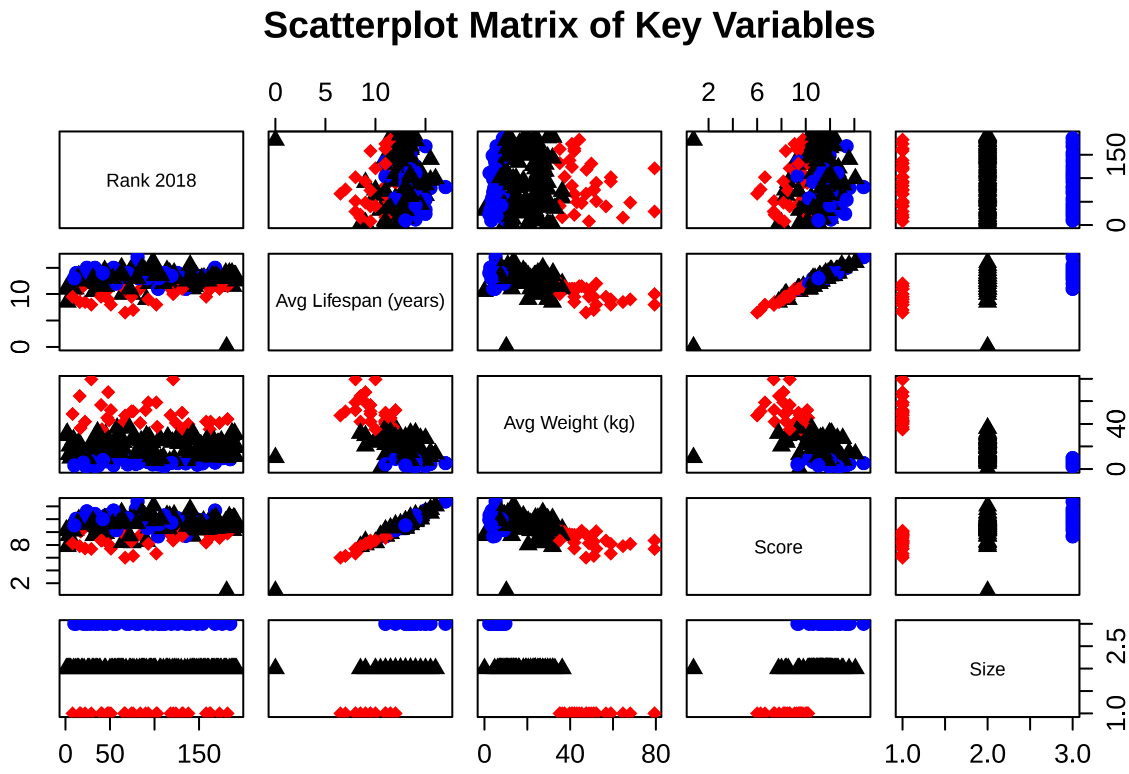
```
##Plot the breeds categorized as "Aloof/Wary"
barplot(aloof_breeds$`Demeanor Value`, names.arg = aloof_breeds$Breed,
        main = "Breeds Categorized as 'Aloof/Wary'", xlab = "Breeds", ylab = "Demeanor Value")
```



Similarly, other categories within variables can be explored.

Experimented with scatterplot to understand the correlations/relationships between some variables. Found a strong correlation between weight and lifespan.

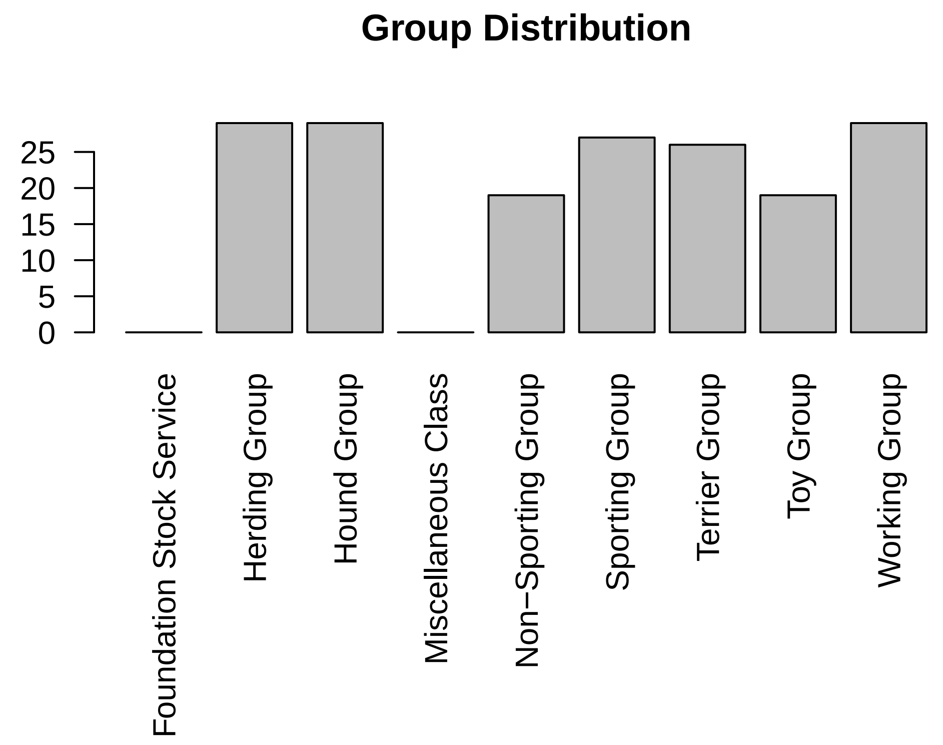
```
##Scatterplot matrix to view the relationship between continuous variables.
pairs(new_data[, c("Rank_2018", "avg_lifespan(years)", "avg_weight(kg)", "score", "Size")],
      col = ifelse(new_data$Size == "Small", "blue", ifelse(new_data$Size == "Medium", "black", "red"))
      pch = ifelse(new_data$Size == "Small", 16, ifelse(new_data$Size == "Medium", 17, 18)),
      main = "Scatterplot Matrix of Key Variables",
      labels = c("Rank 2018", "Avg Lifespan (years)", "Avg Weight (kg)", "Score","Size"),
      cex.axis = 1.2, cex.lab = 1.2,
      cex = 1.5)
```



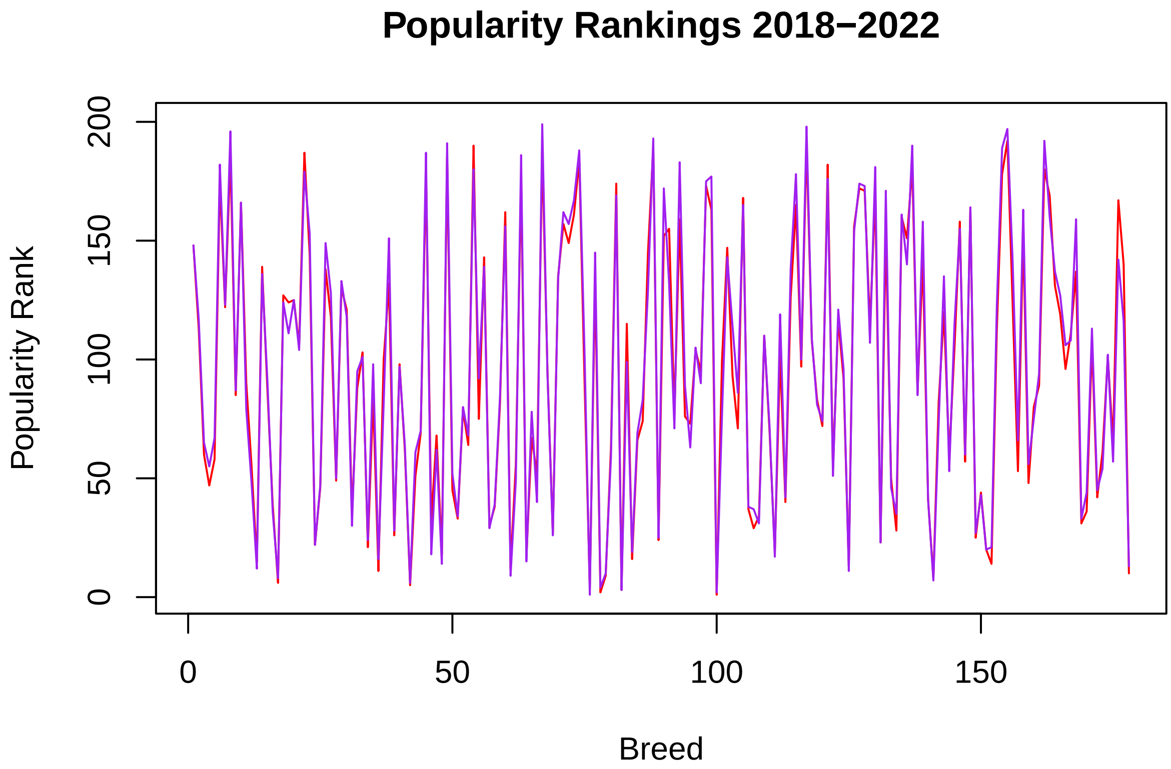
If the points on the plot form a clear linear pattern, it indicates a strong correlation between the variables.For example Avg_weight(kg) and Avg_Lifespan(Years).

Experimented with Barplot and Line Graph.

```
##Bar plots can be used to explore categorical variables
par(mar = c(12, 5, 5, 5))
barplot(table(new_data$group), main = "Group Distribution", las = 2)
```



```r
##Line plot for popularity rankings (years 2018 and 2022)
plot(new_data$Rank_2018, type = "l", col = "red", ylim = c(1,200), xlab = "Breed",
     ylab = "Popularity Rank", main = "Popularity Rankings 2018-2022")
lines(new_data$Rank_2022, col = "purple")
```



We can see from the line graph that the lines deviate from each other at some points but intersect mostly.

4) K-means Clustering

Used k-means algorithm to divide dogs into 4 categories

Chose k-means to cluster as I was aware of the categories I wanted

Clustered dogs into hotdogs, overlooked treasures, rightly ignored, and overrated dogs

Used ggplot to plot this visualization

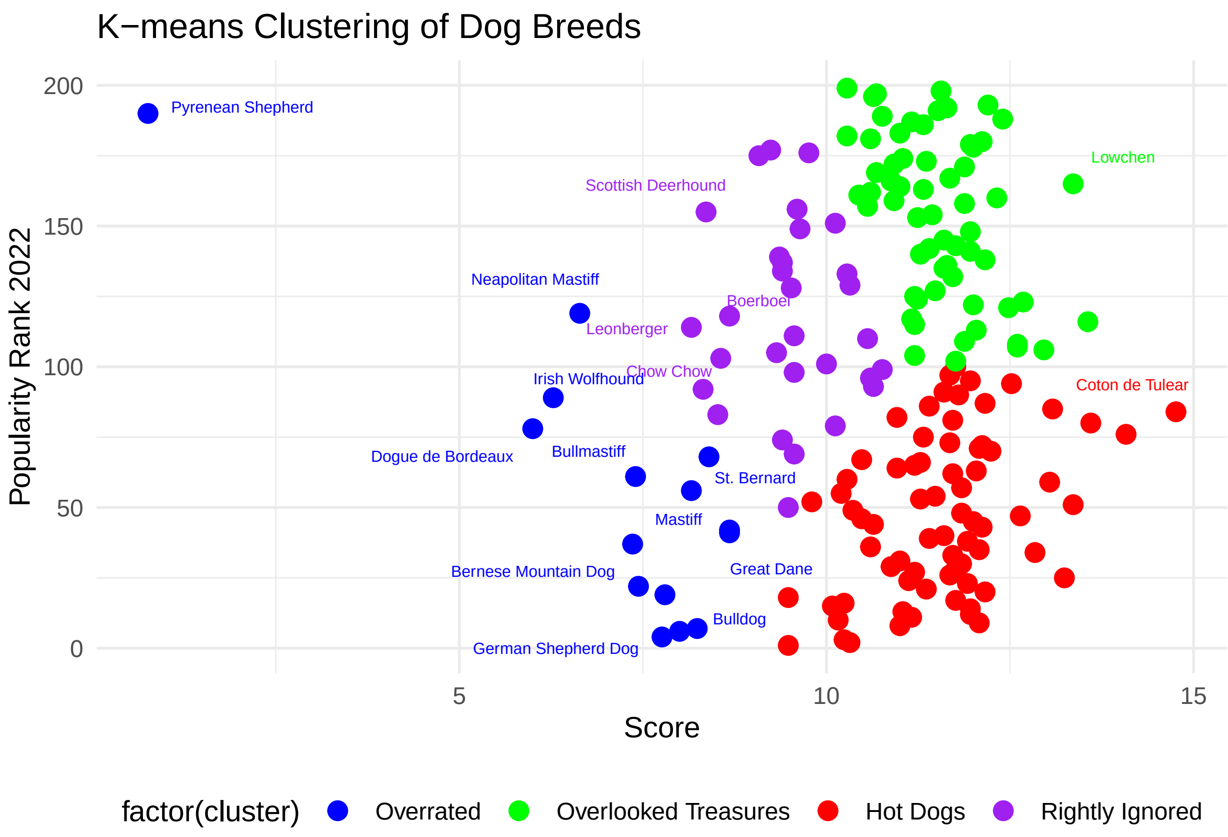
```
##Select relevant features for clustering
features <- c("score", "Rank_2022")

##Prepare data by selecting features
clustering_data <- new_data[, features]

##Scale the features to ensure they have equal importance
```

```r
scaled_data <- scale(clustering_data)

set.seed(123)
##Determine the no. of clusters (k = 4 for hotdogs, overlooked, overrated & rightly ignored)
k <- 4

##Perform k-means clustering
kmeans_result <- kmeans(scaled_data, centers = k)

##Assign cluster labels to each breed
cluster_labels <- kmeans_result$cluster

##Add cluster labels back to the original dataset
new_data$cluster <- cluster_labels

##Display the count of breeds in each cluster
table(new_data$cluster)
```

```
##
##  1  2  3  4
## 15 61 72 30
```

```r
##Assign custom labels to the clusters based on their characteristics
cluster_names <- c("Overrated", "Overlooked Treasures", "Hot Dogs","Rightly Ignored")

##Define colors for the clusters
cluster_colors <- c("blue", "green", "red", "purple")

##Rename cluster labels in the dataset
new_data$cluster_label <- cluster_names[new_data$cluster]

##Create a ggplot object
ggplot(new_data, aes(x = score, y = Rank_2022, color = factor(cluster))) +
  geom_point(size = 3) +  # Adjust point size
  labs(title = "K-means Clustering of Dog Breeds", x = "Score", y = "Popularity Rank 2022") +
  scale_color_manual(values = cluster_colors, labels = cluster_names) +
  geom_text_repel(aes(label = Breed), box.padding = 0.5, point.padding = 0.1,
                  segment.color = "transparent", size = 2, max.overlaps = 10) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## Warning: ggrepel: 161 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


K−means Clustering of Dog Breeds
Popularity Rank 2022
200
150
100
50
0
Score
5
10
15
Pyrenean Shepherd
Lowchen
Scottish Deerhound
Neapolitan Mastiff
Boerboel
Leonberger
Irish Wolfhound
Chow Chow
Coton de Tulear
Dogue de Bordeaux
Bullmastiff
St. Bernard
Mastiff
Bernese Mountain Dog
Great Dane
Bulldog
German Shepherd Dog
factor(cluster)
Overrated
Overlooked Treasures
Hot Dogs
Rightly Ignored