RMarkdown dokument

Vukoje, Leo

Supplement / Prilog

Publication year / Godina izdavanja: **2024**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:052438

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-10-15**

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

UNIRI DIGITALNA KNJIŽNICA

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Preparation and exploratory analysis of a dataset with car prices

Leo Vukoje

25. 6. 2024.

Required Libraries

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(caret)
library(GGally)
library(corrplot)
library(DescTools)
```

Data loading

```
my_cars <- read.csv("carprice.csv", header = TRUE)
head(my_cars)
```

```
##   car_ID symboling                CarName fueltype aspiration doornumber
## 1      1         3       alfa-romero giulia      gas        std        two
## 2      2         3      alfa-romero stelvio      gas        std        two
## 3      3         1 alfa-romero Quadrifoglio      gas        std        two
## 4      4         2              audi 100 ls      gas        std       four
## 5      5         2               audi 100ls      gas        std       four
## 6      6         2                 audi fox      gas        std        two
##       carbody drivewheel enginelocation wheelbase carlength carwidth carheight
## 1 convertible        rwd          front      88.6     168.8     64.1      48.8
## 2 convertible        rwd          front      88.6     168.8     64.1      48.8
## 3   hatchback        rwd          front      94.5     171.2     65.5      52.4
## 4       sedan        fwd          front      99.8     176.6     66.2      54.3
## 5       sedan        4wd          front      99.4     176.6     66.4      54.3
## 6       sedan        fwd          front      99.8     177.3     66.3      53.1
##   curbweight enginetype cylindernumber enginesize fuelsystem boreratio stroke
## 1       2548       dohc           four        130       mpfi      3.47   2.68
## 2       2548       dohc           four        130       mpfi      3.47   2.68
## 3       2823       ohcv            six        152       mpfi      2.68   3.47
## 4       2337        ohc           four        109       mpfi      3.19   3.40
## 5       2824        ohc           five        136       mpfi      3.19   3.40
## 6       2507        ohc           five        136       mpfi      3.19   3.40
##   compressionratio horsepower peakrpm citympg highwaympg price
## 1              9.0        111    5000      21         27 13495
## 2              9.0        111    5000      21         27 16500
## 3              9.0        154    5000      19         26 16500
## 4             10.0        102    5500      24         30 13950
## 5              8.0        115    5500      18         22 17450
## 6              8.5        110    5500      19         25 15250
```

```
class(my_cars)
```

```
## [1] "data.frame"
```

Descriptive statistics

Range, median, mode and quartiles for each variable except the car identifier

```
summary(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)))
```

```
##    symboling         wheelbase        carlength        carwidth
##  Min.   :-2.0000   Min.   : 86.60   Min.   :141.1   Min.   :60.30
##  1st Qu.: 0.0000   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10
##  Median : 1.0000   Median : 97.00   Median :173.2   Median :65.50
##  Mean   : 0.8341   Mean   : 98.76   Mean   :174.0   Mean   :65.91
##  3rd Qu.: 2.0000   3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90
##  Max.   : 3.0000   Max.   :120.90   Max.   :208.1   Max.   :72.30
##    carheight       curbweight     enginesize      boreratio        stroke
##  Min.   :47.80   Min.   :1488   Min.   : 61.0   Min.   :2.54   Min.   :2.070
##  1st Qu.:52.00   1st Qu.:2145   1st Qu.: 97.0   1st Qu.:3.15   1st Qu.:3.110
##  Median :54.10   Median :2414   Median :120.0   Median :3.31   Median :3.290
##  Mean   :53.72   Mean   :2556   Mean   :126.9   Mean   :3.33   Mean   :3.255
##  3rd Qu.:55.50   3rd Qu.:2935   3rd Qu.:141.0   3rd Qu.:3.58   3rd Qu.:3.410
##  Max.   :59.80   Max.   :4066   Max.   :326.0   Max.   :3.94   Max.   :4.170
##  compressionratio   horsepower       peakrpm        citympg
##  Min.   : 7.00    Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.60    1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00    Median : 95.0   Median :5200   Median :24.00
##  Mean   :10.14    Mean   :104.1   Mean   :5125   Mean   :25.22
##  3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00    Max.   :288.0   Max.   :6600   Max.   :49.00
##    highwaympg        price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7788
##  Median :30.00   Median :10295
##  Mean   :30.75   Mean   :13277
##  3rd Qu.:34.00   3rd Qu.:16503
##  Max.   :54.00   Max.   :45400
```

Standard Deviation

```
sapply(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)), sd)
```

```
##        symboling        wheelbase        carlength         carwidth
##        1.2453068        6.0217757       12.3372885        2.1452039
##        carheight       curbweight       enginesize        boreratio
##        2.4435220      520.6802035       41.6426934        0.2708437
##           stroke compressionratio       horsepower          peakrpm
##        0.3135970        3.9720403       39.5441668      476.9856431
##          citympg       highwaympg            price
##        6.5421417        6.8864431     7988.8523317
```

Variance

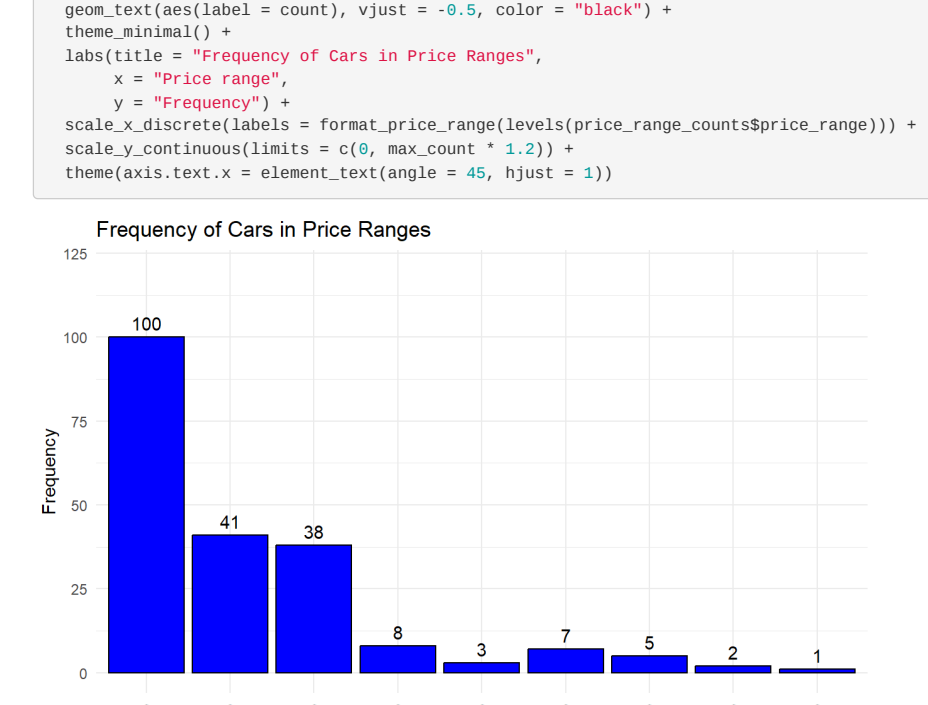
```
sapply(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)), var)
```

```
##        symboling        wheelbase        carlength         carwidth
##     1.550789e+00     3.626178e+01     1.522087e+02     4.601900e+00
##        carheight       curbweight       enginesize        boreratio
##     5.970800e+00     2.711079e+05     1.734114e+03     7.335631e-02
##           stroke compressionratio       horsepower          peakrpm
##     9.834309e-02     1.577710e+01     1.563741e+03     2.275153e+05
##          citympg       highwaympg            price
##     4.279962e+01     4.742310e+01     6.382176e+07
```

Visualization

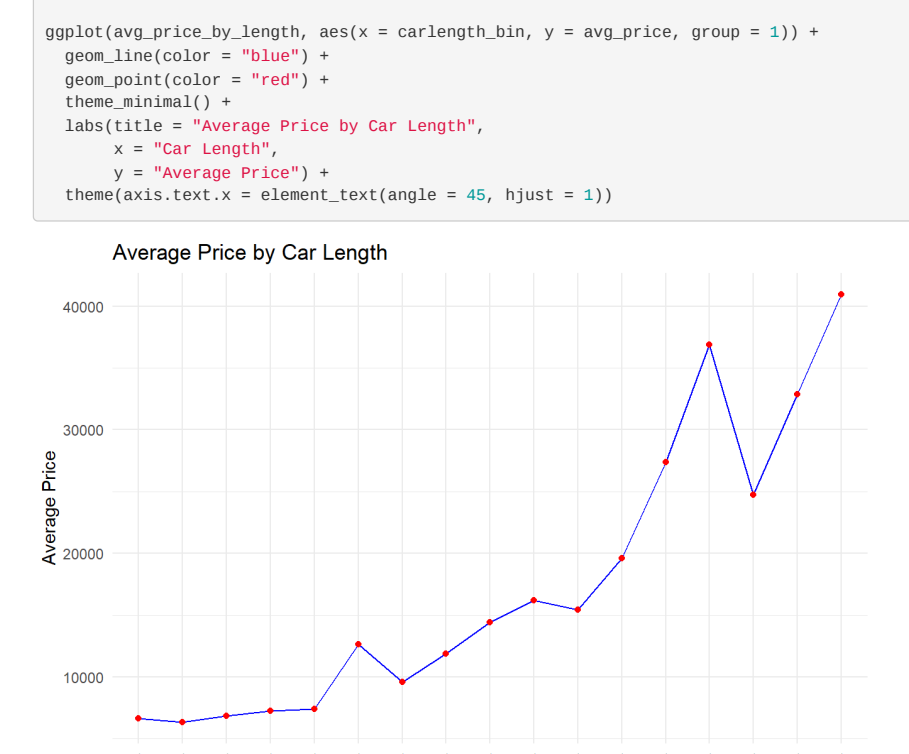
Price Range Frequency Histogram

```
price_breaks <- seq(0, max(my_cars$price + 5000, na.rm = TRUE), by = 5000)

my_cars <- my_cars %>%
  mutate(price_range = cut(price, breaks = price_breaks, include.lowest = TRUE, right = FALSE))

price_range_counts <- my_cars %>%
  group_by(price_range) %>%
  summarize(count = n())

format_price_range <- function(labels) {
  sapply(labels, function(x) {
    range <- gsub("\\[|\\)|\\]", "", x)
    range <- strsplit(range, ",")[[1]]
    paste0(format(as.numeric(range[1]), big.mark = ","), " - ", format(as.numeric(range[2]), big.mark = ","))
  })
}

max_count <- max(price_range_counts$count)

ggplot(price_range_counts, aes(x = price_range, y = count)) +
  geom_bar(stat = "identity", fill = "blue", color = "black") +
  geom_text(aes(label = count), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Frequency of Cars in Price Ranges",
       x = "Price range",
       y = "Frequency") +
  scale_x_discrete(labels = format_price_range(levels(price_range_counts$price_range))) +
  scale_y_continuous(limits = c(0, max_count * 1.2)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



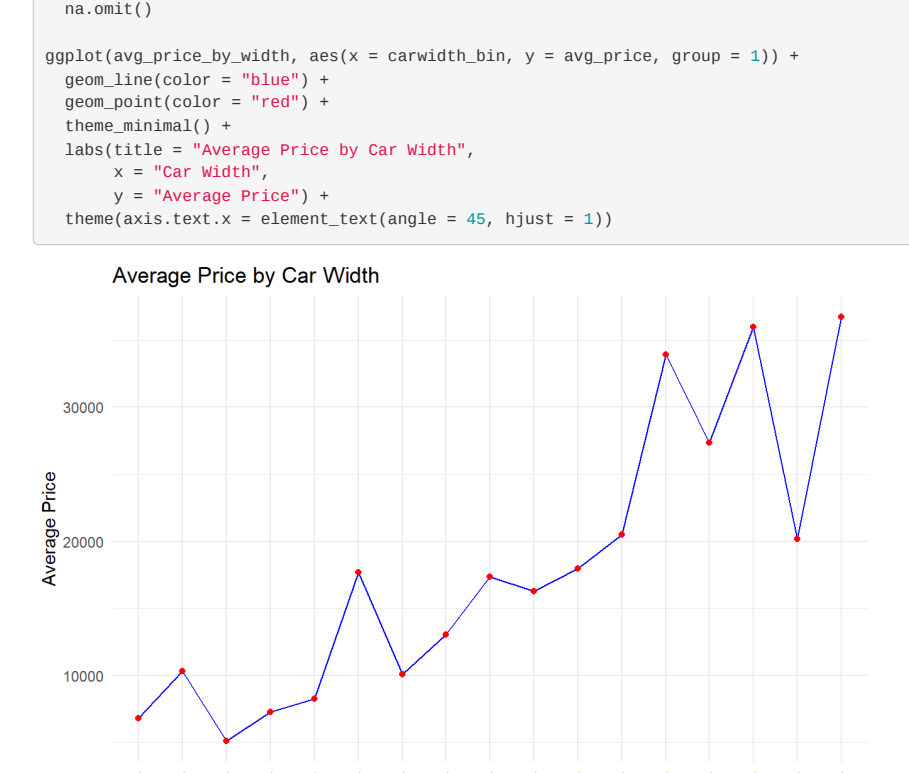
Average Price by Car Length Line Chart

```
avg_price_by_length <- my_cars %>%
  mutate(carlength_bin = cut(carlength, breaks = seq(min(carlength), max(carlength), length.out = 20))) %>%
  group_by(carlength_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_length, aes(x = carlength_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Car Length",
       x = "Car Length",
       y = "Average Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



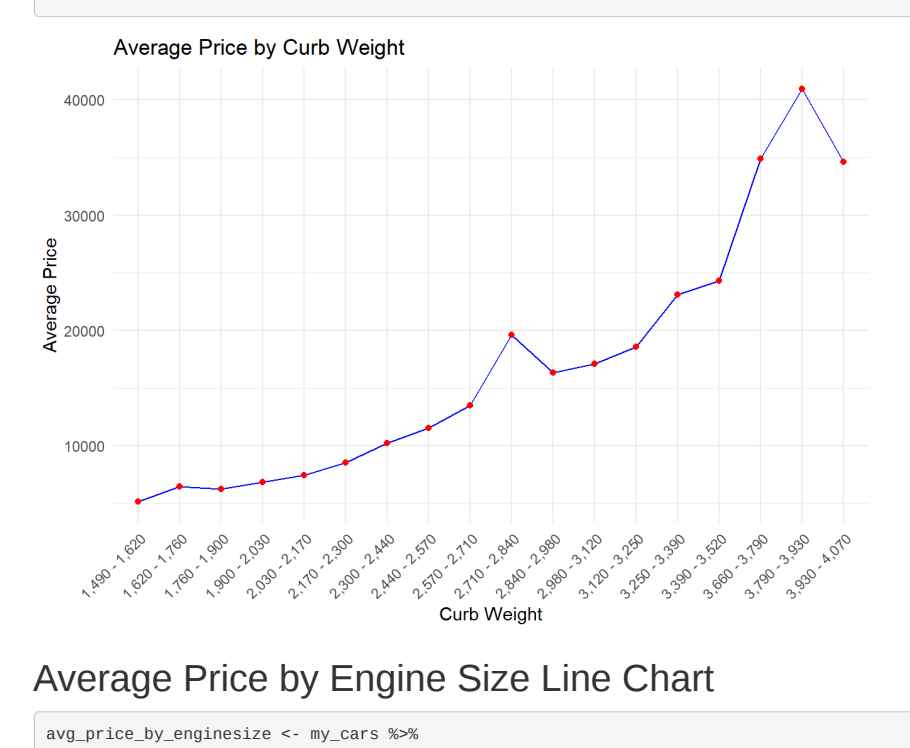
Average Price by Car Width Line Chart

```
avg_price_by_width <- my_cars %>%
  mutate(carwidth_bin = cut(carwidth, breaks = seq(min(carwidth), max(carwidth), length.out = 20))) %>%
  group_by(carwidth_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_width, aes(x = carwidth_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Car Width",
       x = "Car Width",
       y = "Average Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



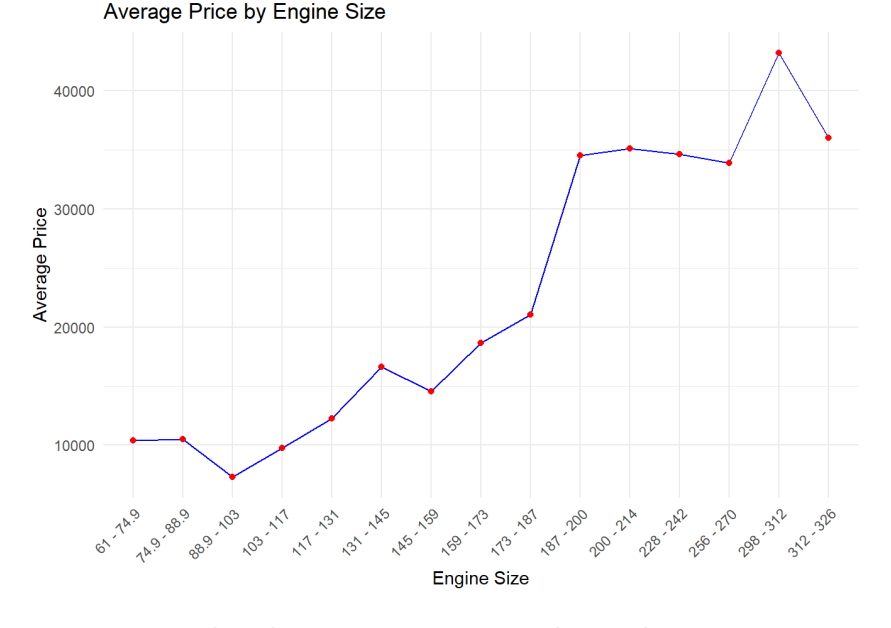
Average Price by Curb Weight Line Chart

```
avg_price_by_weight <- my_cars %>%
  mutate(curbweight_bin = cut(curbweight, breaks = seq(min(curbweight, na.rm = TRUE), max(curbweight, na.rm = TRU
E), length.out = 20), include.lowest = TRUE, right = FALSE)) %>%
  group_by(curbweight_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

format_weight <- function(labels) {
  sapply(labels, function(x) {
    range <- gsub("\\[|\\)|\\]", "", x)
    range <- strsplit(range, ",")[[1]]
    paste0(format(as.numeric(range[1]), big.mark = ","), " - ", format(as.numeric(range[2]), big.mark = ","))
  })
}

ggplot(avg_price_by_weight, aes(x = curbweight_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Curb Weight",
       x = "Curb Weight",
       y = "Average Price") +
  scale_x_discrete(labels = format_weight(levels(avg_price_by_weight$curbweight_bin))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



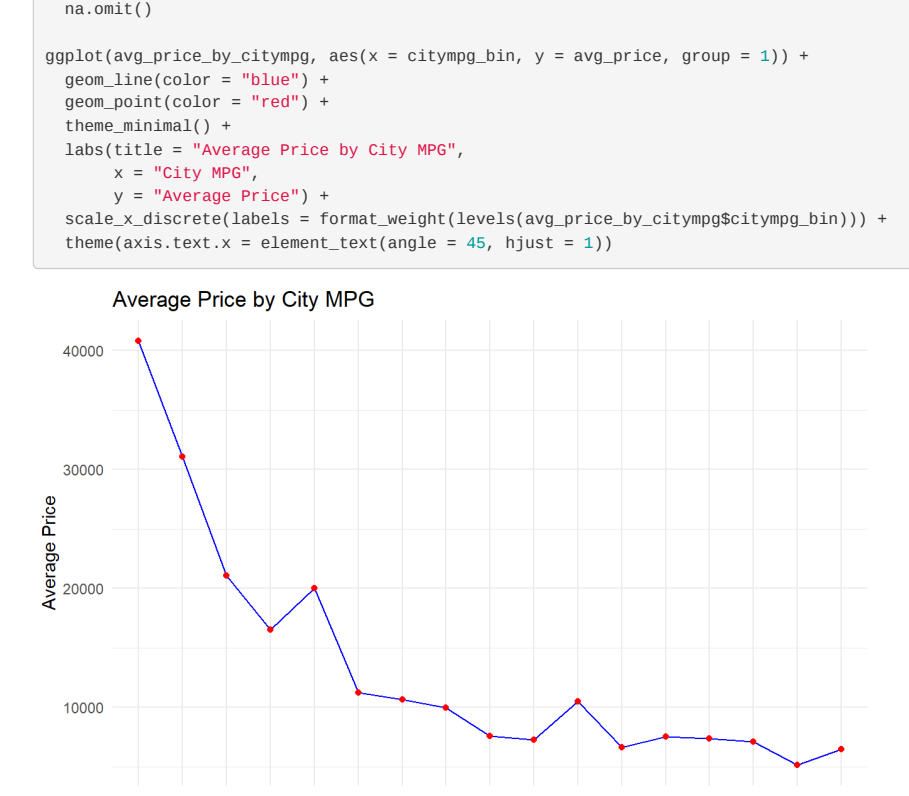
Average Price by Engine Size Line Chart

```
avg_price_by_enginesize <- my_cars %>%
  mutate(enginesize_bin = cut(enginesize, breaks = seq(min(enginesize, na.rm = TRUE), max(enginesize, na.rm = TRU
E), length.out = 20), include.lowest = TRUE, right = FALSE)) %>%
  group_by(enginesize_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_enginesize, aes(x = enginesize_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Engine Size",
       x = "Engine Size",
       y = "Average Price") +
  scale_x_discrete(labels = format_weight(levels(avg_price_by_enginesize$enginesize_bin))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



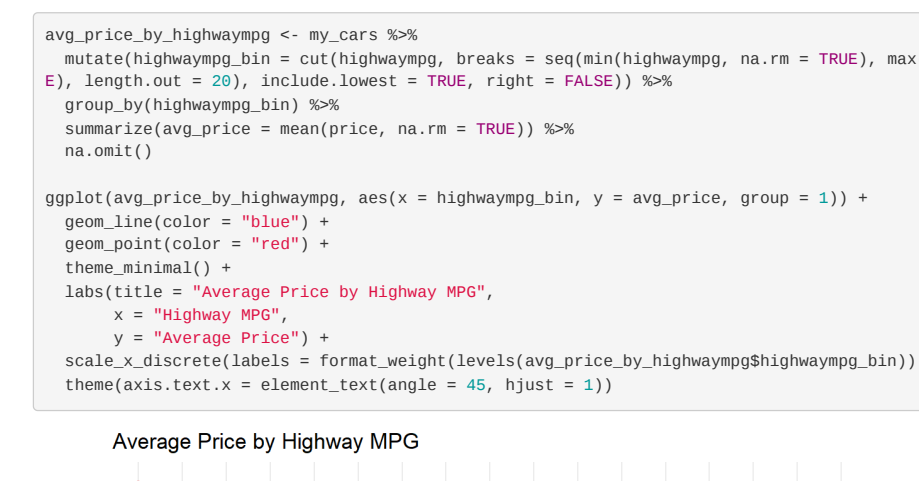
Average Price by Horsepower Line Chart

```
avg_price_by_horsepower <- my_cars %>%
  mutate(horsepower_bin = cut(horsepower, breaks = seq(min(horsepower, na.rm = TRUE), max(horsepower, na.rm = TRU
E), length.out = 20), include.lowest = TRUE, right = FALSE)) %>%
  group_by(horsepower_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_horsepower, aes(x = horsepower_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Horsepower",
       x = "Horsepower",
       y = "Average Price") +
  scale_x_discrete(labels = format_weight(levels(avg_price_by_horsepower$horsepower_bin))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



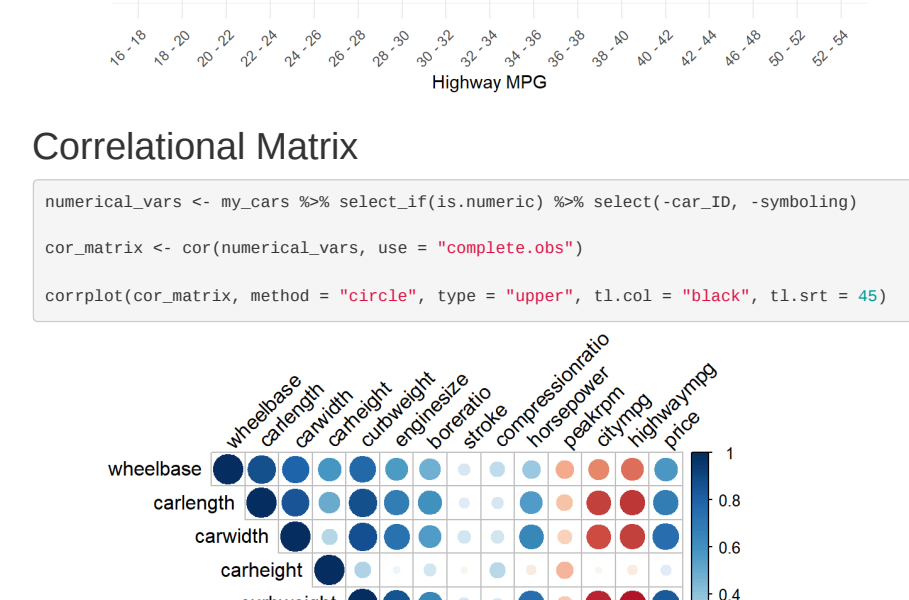
Average Price by City MPG Line Chart

```
avg_price_by_citympg <- my_cars %>%
  mutate(citympg_bin = cut(citympg, breaks = seq(min(citympg, na.rm = TRUE), max(citympg, na.rm = TRUE), length.o
ut = 20), include.lowest = TRUE, right = FALSE)) %>%
  group_by(citympg_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_citympg, aes(x = citympg_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by City MPG",
       x = "City MPG",
       y = "Average Price") +
  scale_x_discrete(labels = format_weight(levels(avg_price_by_citympg$citympg_bin))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



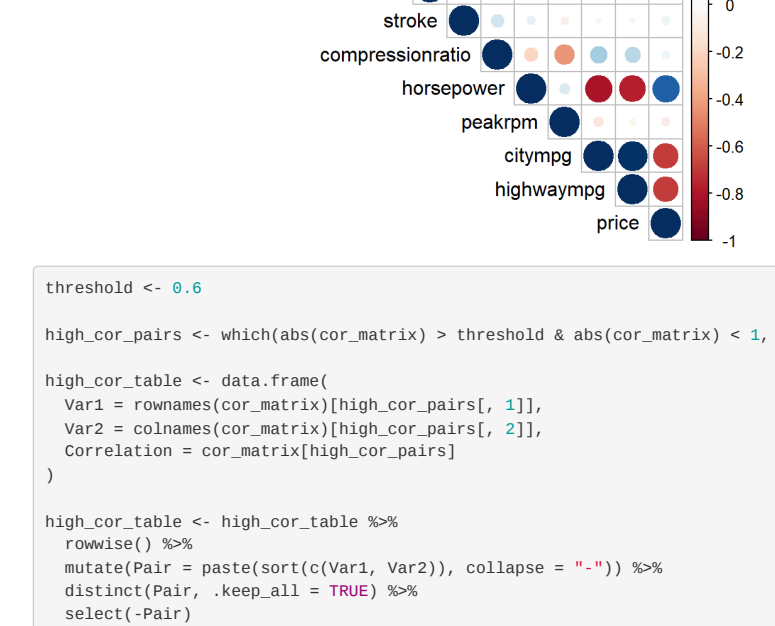
Average Price by Highway MPG Line Chart

```
avg_price_by_highwaympg <- my_cars %>%
  mutate(highwaympg_bin = cut(highwaympg, breaks = seq(min(highwaympg, na.rm = TRUE), max(highwaympg, na.rm = TRU
E), length.out = 20), include.lowest = TRUE, right = FALSE)) %>%
  group_by(highwaympg_bin) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  na.omit()

ggplot(avg_price_by_highwaympg, aes(x = highwaympg_bin, y = avg_price, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Average Price by Highway MPG",
       x = "Highway MPG",
       y = "Average Price") +
  scale_x_discrete(labels = format_weight(levels(avg_price_by_highwaympg$highwaympg_bin))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Correlational Matrix

```
numerical_vars <- my_cars %>% select_if(is.numeric) %>% select(-car_ID, -symboling)

cor_matrix <- cor(numerical_vars, use = "complete.obs")

corrplot(cor_matrix, method = "circle", type = "upper", tl.col = "black", tl.srt = 45)
```



```
threshold <- 0.6

high_cor_pairs <- which(abs(cor_matrix) > threshold & abs(cor_matrix) < 1, arr.ind = TRUE)

high_cor_table <- data.frame(
  Var1 = rownames(cor_matrix)[high_cor_pairs[, 1]],
  Var2 = colnames(cor_matrix)[high_cor_pairs[, 2]],
  Correlation = cor_matrix[high_cor_pairs]
)

high_cor_table <- high_cor_table %>%
  rowwise() %>%
  mutate(Pair = paste(sort(c(Var1, Var2)), collapse = "-")) %>%
  distinct(Pair, .keep_all = TRUE) %>%
  select(-Pair)
```

Inferential Statistics

Hypothesis Testing

T-test

```
t_test_result <- t.test(price ~ fueltype, data = my_cars)
t_test_result
```

```
##
##  Welch Two Sample t-test
##
## data:  price by fueltype
## t = 1.5495, df = 23.566, p-value = 0.1346
## alternative hypothesis: true difference in means between group diesel and group gas is not equal to 0
## 95 percent confidence interval:
##  -945.9072 6622.6108
## sample estimates:
## mean in group diesel    mean in group gas
##             15838.15             12999.80
```

ANOVA test

```
anova_result <- aov(price ~ carbody, data = my_cars)
summary(anova_result)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## carbody       4 1.802e+09 450499206   8.032 5.03e-06 ***
## Residuals   200 1.122e+10  56088213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chi-square test

```
chisq.test(table(my_cars$carbody, my_cars$fueltype))
```

```
## Warning in chisq.test(table(my_cars$carbody, my_cars$fueltype)): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(my_cars$carbody, my_cars$fueltype)
## X-squared = 10.129, df = 4, p-value = 0.0383
```

Contigency table

```
table(my_cars$carbody, my_cars$fueltype)
```

```
##
##               diesel gas
##   convertible      0   6
##   hardtop          1   7
##   hatchback        1  69
##   sedan           15  81
##   wagon            3  22
```

Confidence Intervals

```
t_test_result <- t.test(my_cars$price)
t_test_result$conf.int
```

```
## [1] 12176.59 14376.83
## attr(,"conf.level")
## [1] 0.95
```

```
MeanCI(my_cars$price, conf.level = 0.95)
```

```
##     mean   lwr.ci   upr.ci
## 13276.71 12176.59 14376.83
```

```
MeanCI(my_cars$price, conf.level = 0.70)
```

```
##     mean   lwr.ci   upr.ci
## 13276.71 12696.94 13856.48
```

```
MeanCI(my_cars$price, conf.level = 0.98)
```

```
##     mean   lwr.ci   upr.ci
## 13276.71 11968.41 14585.01
```

Manual Insertion of NA Values

```
my_cars[my_cars == ""] <- NA
sum(is.na(my_cars))
```

```
## [1] 0
```

Determination of Numeric Variables

```
numeric_columns <- c("wheelbase", "carlength", "carwidth", "carheight", "curbweight",
                     "enginesize", "boreratio", "stroke", "compressionratio", "horsepower",
                     "peakrpm", "citympg", "highwaympg")

head(my_cars[numeric_columns])
```

```
##   wheelbase carlength carwidth carheight curbweight enginesize boreratio stroke
## 1      88.6     168.8     64.1      48.8       2548        130      3.47   2.68
## 2      88.6     168.8     64.1      48.8       2548        130      3.47   2.68
## 3      94.5     171.2     65.5      52.4       2823        152      2.68   3.47
## 4      99.8     176.6     66.2      54.3       2337        109      3.19   3.40
## 5      99.4     176.6     66.4      54.3       2824        136      3.19   3.40
## 6      99.8     177.3     66.3      53.1       2507        136      3.19   3.40
##   compressionratio horsepower peakrpm citympg highwaympg
## 1              9.0        111    5000      21         27
## 2              9.0        111    5000      21         27
## 3              9.0        154    5000      19         26
## 4             10.0        102    5500      24         30
## 5              8.0        115    5500      18         22
## 6              8.5        110    5500      19         25
```

Normalization and Standardization of Numeric Variables

```
min_max_normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

z_score_standardize <- function(x) {
  return ((x - mean(x)) / sd(x))
}

numeric_columns <- c("wheelbase", "carlength", "carwidth", "carheight", "curbweight",
                     "enginesize", "boreratio", "stroke", "compressionratio", "horsepower",
                     "peakrpm", "citympg", "highwaympg")

head(my_cars[numeric_columns], 10)
```

```
##    wheelbase carlength carwidth carheight curbweight enginesize boreratio
## 1       88.6     168.8     64.1      48.8       2548        130      3.47
## 2       88.6     168.8     64.1      48.8       2548        130      3.47
## 3       94.5     171.2     65.5      52.4       2823        152      2.68
## 4       99.8     176.6     66.2      54.3       2337        109      3.19
## 5       99.4     176.6     66.4      54.3       2824        136      3.19
## 6       99.8     177.3     66.3      53.1       2507        136      3.19
## 7      105.8     192.7     71.4      55.7       2844        136      3.19
## 8      105.8     192.7     71.4      55.7       2954        136      3.19
## 9      105.8     192.7     71.4      55.9       3086        131      3.13
## 10      99.5     178.2     67.9      52.0       3053        131      3.13
##    stroke compressionratio horsepower peakrpm citympg highwaympg
## 1    2.68              9.0        111    5000      21         27
## 2    2.68              9.0        111    5000      21         27
## 3    3.47              9.0        154    5000      19         26
## 4    3.40             10.0        102    5500      24         30
## 5    3.40              8.0        115    5500      18         22
## 6    3.40              8.5        110    5500      19         25
## 7    3.40              8.5        110    5500      19         25
## 8    3.40              8.5        110    5500      19         25
## 9    3.40              8.3        140    5500      17         20
## 10   3.40              7.0        160    5500      16         22
```

```
my_cars[numeric_columns] <- lapply(my_cars[numeric_columns], min_max_normalize)

head(round(my_cars[numeric_columns], digits = 4), 10)
```

```
##    wheelbase carlength carwidth carheight curbweight enginesize boreratio
## 1     0.0583    0.4134   0.3167    0.0833     0.4112     0.2604    0.6643
## 2     0.0583    0.4134   0.3167    0.0833     0.4112     0.2604    0.6643
## 3     0.2303    0.4493   0.4333    0.3833     0.5178     0.3434    0.1000
## 4     0.3848    0.5299   0.4917    0.5417     0.3293     0.1811    0.4643
## 5     0.3732    0.5299   0.5083    0.5417     0.5182     0.2830    0.4643
## 6     0.3848    0.5403   0.5000    0.4417     0.3953     0.2830    0.4643
## 7     0.5598    0.7701   0.9250    0.6583     0.5260     0.2830    0.4643
## 8     0.5598    0.7701   0.9250    0.6583     0.5687     0.2830    0.4643
## 9     0.5598    0.7701   0.9250    0.6750     0.6199     0.2642    0.4214
## 10    0.3761    0.5537   0.6333    0.3500     0.6071     0.2642    0.4214
##    stroke compressionratio horsepower peakrpm citympg highwaympg
## 1  0.2905           0.1250     0.2625  0.3469  0.2222     0.2895
## 2  0.2905           0.1250     0.2625  0.3469  0.2222     0.2895
## 3  0.6667           0.1250     0.4417  0.3469  0.1667     0.2632
## 4  0.6333           0.1875     0.2250  0.5510  0.3056     0.3684
## 5  0.6333           0.0625     0.2792  0.5510  0.1389     0.1579
## 6  0.6333           0.0938     0.2583  0.5510  0.1667     0.2368
## 7  0.6333           0.0938     0.2583  0.5510  0.1667     0.2368
## 8  0.6333           0.0938     0.2583  0.5510  0.1667     0.2368
## 9  0.6333           0.0813     0.3833  0.5510  0.1111     0.1053
## 10 0.6333           0.0000     0.4667  0.5510  0.0833     0.1579
```

```
my_cars[numeric_columns] <- lapply(my_cars[numeric_columns], z_score_standardize)
```

Mean and Standard Deviation of Numeric Variables

```
my_cars[numeric_columns] %>% colMeans
```

```
##        wheelbase        carlength         carwidth        carheight
##    -5.243096e-17    -1.045234e-16     9.721221e-17     3.198661e-17
##       curbweight       enginesize        boreratio           stroke
##    -1.210075e-17    -4.027943e-17    -2.847147e-16    -5.046776e-17
## compressionratio       horsepower          peakrpm          citympg
##    -3.155504e-17    -3.936553e-17     9.873538e-17     6.796731e-17
##       highwaympg
##    -1.394548e-16
```

```
sapply(my_cars[numeric_columns], sd)
```

```
##        wheelbase        carlength         carwidth        carheight
##                1                1                1                1
##       curbweight       enginesize        boreratio           stroke
##                1                1                1                1
## compressionratio       horsepower          peakrpm          citympg
##                1                1                1                1
##       highwaympg
##                1
```

Standardization and Normalization with dplyr

```
#my_cars <- my_cars %>%
#      mutate(across(all_of(numeric_columns), min_max_normalize)) %>%
#      mutate(across(all_of(numeric_columns), z_score_standardize))

#head(my_cars[numeric_columns], 5)
```

Brand Values Consistency Fix

```
my_cars <- my_cars %>% separate(CarName, into = c("brand", "model"), sep = " ", extra = "merge")
```

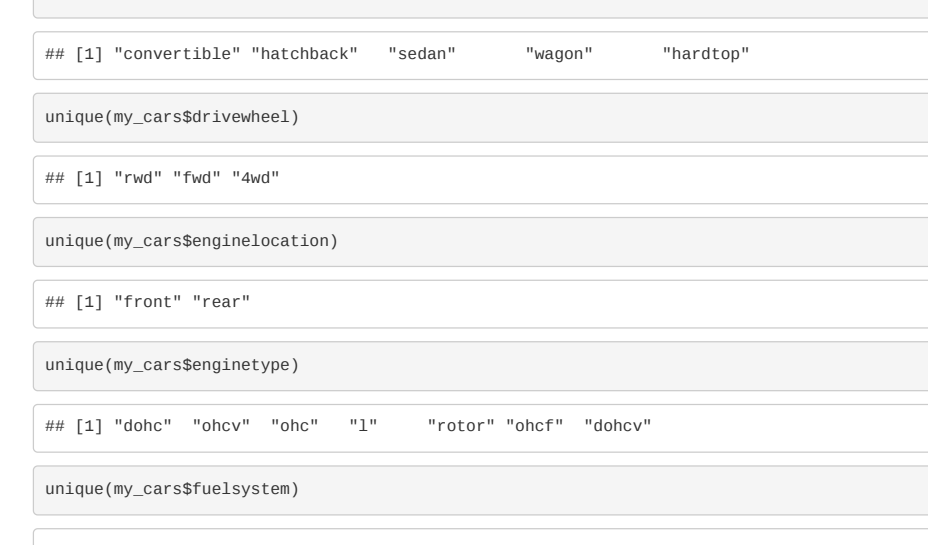
```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [139,
## 142].
```

```
my_cars[my_cars == "volkswagen" | my_cars == "vokswagen"] <- "vw"
my_cars[my_cars == "toyouta"] <- "toyota"
my_cars[my_cars == "porcshce"] <- "porsche"
my_cars[my_cars == "Nissan"] <- "nissan"
my_cars[my_cars == "maxda"] <- "mazda"

head(my_cars, n = 10)
```

```
##    car_ID symboling       brand          model fueltype aspiration doornumber
## 1       1         3 alfa-romero         giulia      gas        std        two
## 2       2         3 alfa-romero        stelvio      gas        std        two
## 3       3         1 alfa-romero   Quadrifoglio      gas        std        two
## 4       4         2        audi         100 ls      gas        std       four
## 5       5         2        audi          100ls      gas        std       four
## 6       6         2        audi            fox      gas        std        two
## 7       7         1        audi          100ls      gas        std       four
## 8       8         1        audi           5000      gas        std       four
## 9       9         1        audi           4000      gas      turbo       four
## 10     10         0        audi 5000s (diesel)      gas      turbo        two
##        carbody drivewheel enginelocation  wheelbase  carlength   carwidth
## 1  convertible        rwd          front -1.6866429 -0.4254799 -0.8427194
## 2  convertible        rwd          front -1.6866429 -0.4254799 -0.8427194
## 3    hatchback        rwd          front -0.7068655 -0.2309477 -0.1901008
## 4        sedan        fwd          front  0.1732736  0.2067498  0.1362086
## 5        sedan        4wd          front  0.1068480  0.2067498  0.2294398
## 6        sedan        fwd          front  0.1732736  0.2634883  0.1828242
## 7        sedan        fwd          front  1.1696574  1.5117367  2.5602206
## 8        wagon        fwd          front  1.1696574  1.5117367  2.5602206
## 9        sedan        fwd          front  1.1696574  1.5117367  2.5602206
## 10   hatchback        4wd          front  0.1234544  0.3364379  0.9286740
##     carheight  curbweight enginetype cylindernumber  enginesize fuelsystem
## 1  -2.0154834 -0.01453071       dohc           four  0.07426712       mpfi
## 2  -2.0154834 -0.01453071       dohc           four  0.07426712       mpfi
## 3  -0.5422002  0.51362457       ohcv            six  0.60257108       mpfi
## 4   0.2353660 -0.41976986        ohc           four -0.43002303       mpfi
## 5   0.2353660  0.51554514        ohc           five  0.21835002       mpfi
## 6  -0.2557284 -0.09327386        ohc           five  0.21835002       mpfi
## 7   0.8083095  0.55395643        ohc           five  0.21835002       mpfi
## 8   0.8083095  0.76521854        ohc           five  0.21835002       mpfi
## 9   0.8901585  1.01873308        ohc           five  0.09828094       mpfi
## 10 -0.7058983  0.95535444        ohc           five  0.09828094       mpfi
##     boreratio     stroke compressionratio  horsepower    peakrpm    citympg
## 1   0.5178038 -1.8348856      -0.28764476  0.17405669 -0.2623181 -0.6449741
## 2   0.5178038 -1.8348856      -0.28764476  0.17405669 -0.2623181 -0.6449741
## 3  -2.3990076  0.6842711      -0.28764476  1.26144842 -0.2623181 -0.9506844
## 4  -0.5160028  0.4610547      -0.03588498 -0.05353693  0.7859315 -0.1864087
## 5  -0.5160028  0.4610547      -0.53940454  0.27520941  0.7859315 -1.1035396
## 6  -0.5160028  0.4610547      -0.41352465  0.14876851  0.7859315 -0.9506844
## 7  -0.5160028  0.4610547      -0.41352465  0.14876851  0.7859315 -0.9506844
## 8  -0.5160028  0.4610547      -0.41352465  0.14876851  0.7859315 -0.9506844
## 9  -0.7375327  0.4610547      -0.46387661  0.90741391  0.7859315 -1.2563947
## 10 -0.7375327  0.4610547      -0.79116432  1.41317750  0.7859315 -1.4092499
##    highwaympg    price     price_range
## 1  -0.5447253 13495.00 [1e+04,1.5e+04)
## 2  -0.5447253 16500.00 [1.5e+04,2e+04)
## 3  -0.6899381 16500.00 [1.5e+04,2e+04)
## 4  -0.1090867 13950.00 [1e+04,1.5e+04)
## 5  -1.2707895 17450.00 [1.5e+04,2e+04)
## 6  -0.8351509 15250.00 [1.5e+04,2e+04)
## 7  -0.8351509 17710.00 [1.5e+04,2e+04)
## 8  -0.8351509 18920.00 [1.5e+04,2e+04)
## 9  -1.5612152 23875.00 [2e+04,2.5e+04)
## 10 -1.2707895 17859.17 [1.5e+04,2e+04)
```

Frequency of Cars per Brand

```
brand_counts <- my_cars %>%
  count(brand) %>%
  arrange(desc(n))

my_cars$brand <- factor(my_cars$brand, levels = brand_counts$brand)

ggplot(my_cars, aes(x = brand)) +
  geom_bar(fill = "blue", color = "black") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Count Plot of Car Brands", x = "Brand", y = "Count") +
  scale_y_continuous(limits = c(0, max(brand_counts$n) * 1.1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Categories of Categorical Variables

```
unique(my_cars$fueltype)
```

```
## [1] "gas"    "diesel"
```

```
unique(my_cars$aspiration)
```

```
## [1] "std"   "turbo"
```

```
unique(my_cars$carbody)
```

```
## [1] "convertible" "hatchback"   "sedan"       "wagon"       "hardtop"
```

```
unique(my_cars$drivewheel)
```

```
## [1] "rwd" "fwd" "4wd"
```

```
unique(my_cars$enginelocation)
```

```
## [1] "front" "rear"
```

```
unique(my_cars$enginetype)
```

```
## [1] "dohc"  "ohcv"  "ohc"   "l"     "rotor" "ohcf"  "dohcv"
```

```
unique(my_cars$fuelsystem)
```

```
## [1] "mpfi" "2bbl" "mfi"  "1bbl" "spfi" "4bbl" "idi"  "spdi"
```

```
unique(my_cars$cylindernumber)
```

```
## [1] "four"   "six"    "five"   "three"  "twelve" "two"    "eight"
```

```
unique(my_cars$doornumber)
```

```
## [1] "two"  "four"
```

Ordinal Encoding

```
number_mapping <- c("two" = 2, "three" = 3, "four" = 4, "five" = 5, "six" = 6, "seven" = 7, "eight" = 8, "nine" =
9, "ten" = 10, "eleven" = 11, "twelve" = 12)

head(my_cars$cylindernumber, 10)
```

```
##  [1] "four" "four" "six"  "four" "five" "five" "five" "five" "five" "five"
```

```
head(my_cars$doornumber, 10)
```

```
##  [1] "two"  "two"  "two"  "four" "four" "two"  "four" "four" "four" "two"
```

```
my_cars$doornumber <- sapply(my_cars$doornumber, function(x) number_mapping[x])
my_cars$cylindernumber <- sapply(my_cars$cylindernumber, function(x) number_mapping[x])

head(my_cars$cylindernumber, 10)
```

```
##  [1] 4 4 6 4 5 5 5 5 5 5
```

```
head(my_cars$doornumber, 10)
```

```
##  [1] 2 2 2 4 4 2 4 4 4 2
```

```
encoded_columns <- c("cylindernumber", "doornumber")

my_cars <- my_cars %>%
      mutate(across(all_of(encoded_columns), min_max_normalize)) %>%
      mutate(across(all_of(encoded_columns), z_score_standardize))

head(my_cars$cylindernumber, 10)
```

```
##  [1] -0.3520252 -0.3520252  1.4983638 -0.3520252  0.5731693  0.5731693
##  [7]  0.5731693  0.5731693  0.5731693  0.5731693
```

```
head(my_cars$doornumber, 10)
```

```
##  [1] -1.1276279 -1.1276279 -1.1276279  0.8824914  0.8824914 -1.1276279
##  [7]  0.8824914  0.8824914  0.8824914 -1.1276279
```

One-hot Encoding

```
dummy_formula <- dummyVars(~ brand + fueltype + aspiration + carbody + drivewheel + enginelocation + enginetype +
fuelsystem, data = my_cars)

carprice_encoded <- predict(dummy_formula, newdata = my_cars)

carprice_encoded <- as.data.frame(carprice_encoded)

my_cars <- cbind(carprice_encoded, my_cars[ , !names(my_cars) %in% c("brand", "fueltype", "aspiration", "carbod
y", "drivewheel", "enginelocation", "enginetype", "fuelsystem")])
```

Persisting Preprocessed Dataset

```
write.csv(my_cars,"preprocessed_carprice.csv", row.names = FALSE)
```