Managing and Understanding Data

Cristy Medina Armijo

2024-10-03

Contents

Show some registers

```
## data exploration example using used car data
#usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
usedcars <- read.csv("C:/Users/kuris/OneDrive/Escritorio/machine learning/Unidad1 (1)/Unidad1/usedcars(
# Table of 6 first registers

usedcars[1:6, ]
```

```
##   year model price mileage  color transmission
## 1 2011   SEL 21992    7413 Yellow         AUTO
## 2 2011   SEL 20995   10926   Gray         AUTO
## 3 2011   SEL 19995    7351 Silver         AUTO
## 4 2011   SEL 17809   11613   Gray         AUTO
## 5 2012    SE 17500    8367  White         AUTO
## 6 2010   SEL 17495   25125 Silver         AUTO
```

R estructura de los datos

Ejercicios tomados de capítulo 2 de, (Lantz 2023). Las estructuras de datos de R utilizadas con mayor frecuencia en el aprendizaje automático son vectores, factores, listas, matrices y datos. marcos. Ver referencias: (Andrieu et al. 2003) y (Goldberg and Holland 1988)

Vectors

```r
# create vectors of data for three medical patients
subject_name <- c("John Doe", "Jane Doe", "Steve Graves")
temperature <- c(98.1, 98.6, 101.4)
flu_status <- c(FALSE, FALSE, TRUE)

# access the second element in body temperature vector
temperature[2]
```

```
## [1] 98.6
```

```r
## examples of accessing items in vector
# include items in the range 2 to 3
temperature[2:3]
```

```
## [1]  98.6 101.4
```

```r
# exclude item 2 using the minus sign
temperature[-2]
```

```
## [1]  98.1 101.4
```

```r
# use a vector to indicate whether to include item
temperature[c(TRUE, TRUE, FALSE)]
```

```
## [1] 98.1 98.6
```

```r
## Factors -----

# add gender factor
gender <- factor(c("MALE", "FEMALE", "MALE"))
gender
```

```
## [1] MALE   FEMALE MALE
## Levels: FEMALE MALE
```

```r
# add blood type factor
blood <- factor(c("O", "AB", "A"),
                levels = c("A", "B", "AB", "O"))
blood
```

```
## [1] O  AB A
## Levels: A B AB O
```

```r
# add ordered factor
symptoms <- factor(c("SEVERE", "MILD", "MODERATE"),
                   levels = c("MILD", "MODERATE", "SEVERE"),
                   ordered = TRUE)
symptoms
```

```
## [1] SEVERE   MILD     MODERATE
## Levels: MILD < MODERATE < SEVERE
```

```r
# check for symptoms greater than moderate
symptoms > "MODERATE"
```

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

Lists

```r
# display information for a patient
subject_name[1]
```

```
## [1] "John Doe"
```

```r
temperature[1]
```

```
## [1] 98.1
```

```r
flu_status[1]
```

```
## [1] FALSE
```

```r
gender[1]
```

```
## [1] MALE
## Levels: FEMALE MALE
```

```r
blood[1]
```

```
## [1] O
## Levels: A B AB O
```

```r
symptoms[1]
```

```
## [1] SEVERE
## Levels: MILD < MODERATE < SEVERE
```

```r
# create list for a patient
subject1 <- list(fullname = subject_name[1],
                 temperature = temperature[1],
                 flu_status = flu_status[1],
                 gender = gender[1],
                 blood = blood[1],
                 symptoms = symptoms[1])

# display the patient
subject1
```

```
## $fullname
## [1] "John Doe"
##
## $temperature
## [1] 98.1
##
## $flu_status
## [1] FALSE
##
## $gender
## [1] MALE
## Levels: FEMALE MALE
##
## $blood
## [1] O
## Levels: A B AB O
##
## $symptoms
## [1] SEVERE
## Levels: MILD < MODERATE < SEVERE
```

```r
## methods for accessing a list

# get a single list value by position (returns a sub-list)
subject1[2]
```

```
## $temperature
## [1] 98.1
```

```r
# get a single list value by position (returns a numeric vector)
subject1[[2]]
```

```
## [1] 98.1
```

```r
# get a single list value by name
subject1$temperature
```

```
## [1] 98.1
```

```r
# get several list items by specifying a vector of names
subject1[c("temperature", "flu_status")]
```

```
## $temperature
## [1] 98.1
##
## $flu_status
## [1] FALSE
```

```r
## access a list like a vector
# get values 2 and 3
subject1[2:3]
```

```
## $temperature
## [1] 98.1
##
## $flu_status
## [1] FALSE
```

Data frames

```r
# create a data frame from medical patient data

pt_data <- data.frame(subject_name, temperature, flu_status, gender,
                      blood, symptoms, stringsAsFactors = FALSE)

# display the data frame
pt_data
```

```
##   subject_name temperature flu_status gender blood symptoms
## 1     John Doe        98.1      FALSE   MALE     O   SEVERE
## 2     Jane Doe        98.6      FALSE FEMALE    AB     MILD
## 3 Steve Graves       101.4       TRUE   MALE     A MODERATE
```

```r
## accessing a data frame

# get a single column
pt_data$subject_name
```

```
## [1] "John Doe"     "Jane Doe"     "Steve Graves"
```

```r
# get several columns by specifying a vector of names
pt_data[c("temperature", "flu_status")]
```

```
##   temperature flu_status
## 1        98.1      FALSE
## 2        98.6      FALSE
## 3       101.4       TRUE
```

```r
# this is the same as above, extracting temperature and flu_status
pt_data[2:3]
```

```
##   temperature flu_status
## 1        98.1      FALSE
## 2        98.6      FALSE
## 3       101.4       TRUE
```

```r
# accessing by row and column
pt_data[1, 2]
```

```
## [1] 98.1
```

```r
# accessing several rows and several columns using vectors
pt_data[c(1, 3), c(2, 4)]
```

```
##   temperature gender
## 1        98.1   MALE
## 3       101.4   MALE
```

Leave a row or column blank to extract all rows or columns

```r
# column 1, all rows
pt_data[, 1]
```

```
## [1] "John Doe"     "Jane Doe"     "Steve Graves"
```

```r
# row 1, all columns
pt_data[1, ]
```

```
##   subject_name temperature flu_status gender blood symptoms
## 1     John Doe        98.1      FALSE   MALE     O   SEVERE
```

```r
# all rows and all columns
pt_data[ , ]
```

```
##   subject_name temperature flu_status gender blood symptoms
## 1     John Doe        98.1      FALSE   MALE     O   SEVERE
## 2     Jane Doe        98.6      FALSE FEMALE    AB     MILD
## 3 Steve Graves       101.4       TRUE   MALE     A MODERATE
```

```r
# the following are equivalent
pt_data[c(1, 3), c("temperature", "gender")]
```

```
##   temperature gender
## 1        98.1   MALE
## 3       101.4   MALE
```

```r
pt_data[-2, c(-1, -3, -5, -6)]
```

```
##   temperature gender
## 1        98.1   MALE
## 3       101.4   MALE
```

Matrixes

```r
# create a 2x2 matrix
m <- matrix(c(1, 2, 3, 4), nrow = 2)
m
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```r
# equivalent to the above
m <- matrix(c(1, 2, 3, 4), ncol = 2)
m
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```r
# create a 2x3 matrix
m <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```r
# create a 3x2 matrix
m <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2)
m
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```r
# extract values from matrixes
m[1, 1]
```

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

```r
m[3, 2]
```

```
## [1] 6
```

```r
# extract rows
m[1, ]
```

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

```r
# extract columns
m[, 1]
```

```
## [1] 1 2 3
```

Managing data with R

saving, loading, and removing R data structures

```r
# show all data structures in memory
ls()
```

```
##  [1] "blood"        "flu_status"   "gender"       "m"            "pt_data"
##  [6] "subject_name" "subject1"     "symptoms"     "temperature"  "usedcars"
```

```r
# remove the m and subject1 objects
rm(m, subject1)
ls()
```

```
## [1] "blood"        "flu_status"   "gender"       "pt_data"      "subject_name"
## [6] "symptoms"     "temperature"  "usedcars"
```

```r
rm(list=ls())
```

Exploring and understanding data

```r
## data exploration example using used car data
#usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
usedcars <- read.csv("C:/Users/kuris/OneDrive/Escritorio/machine learning/Unidad1 (1)/Unidad1/usedcars(
# get structure of used car data
str(usedcars)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ year        : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
##  $ model       : chr  "SEL" "SEL" "SEL" "SEL" ...
##  $ price       : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
##  $ mileage     : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
##  $ color       : chr  "Yellow" "Gray" "Silver" "Gray" ...
##  $ transmission: chr  "AUTO" "AUTO" "AUTO" "AUTO" ...
```

```r
## Exploring numeric variables -----

# summarize numeric variables
summary(usedcars$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2000    2008    2009    2009    2010    2012
```

```r
summary(usedcars[c("price", "mileage")])
```

```
##      price          mileage
##  Min.   : 3800   Min.   :  4867
##  1st Qu.:10995   1st Qu.: 27200
##  Median :13592   Median : 36385
##  Mean   :12962   Mean   : 44261
##  3rd Qu.:14904   3rd Qu.: 55125
##  Max.   :21992   Max.   :151479
```

```r
# calculate the mean income
(36000 + 44000 + 56000) / 3
```

```
## [1] 45333.33
```

```r
mean(c(36000, 44000, 56000))
```

```
## [1] 45333.33
```

```r
# the median income
median(c(36000, 44000, 56000))
```

```
## [1] 44000
```

```r
# the min/max of used car prices
range(usedcars$price)
```

```
## [1]  3800 21992
```

```r
# the difference of the range
diff(range(usedcars$price))
```

```
## [1] 18192
```

```r
# IQR for used car prices
IQR(usedcars$price)
```

```
## [1] 3909.5
```

```r
# use quantile to calculate five-number summary
quantile(usedcars$price)
```

```
##      0%     25%     50%     75%    100% 
##  3800.0 10995.0 13591.5 14904.5 21992.0
```

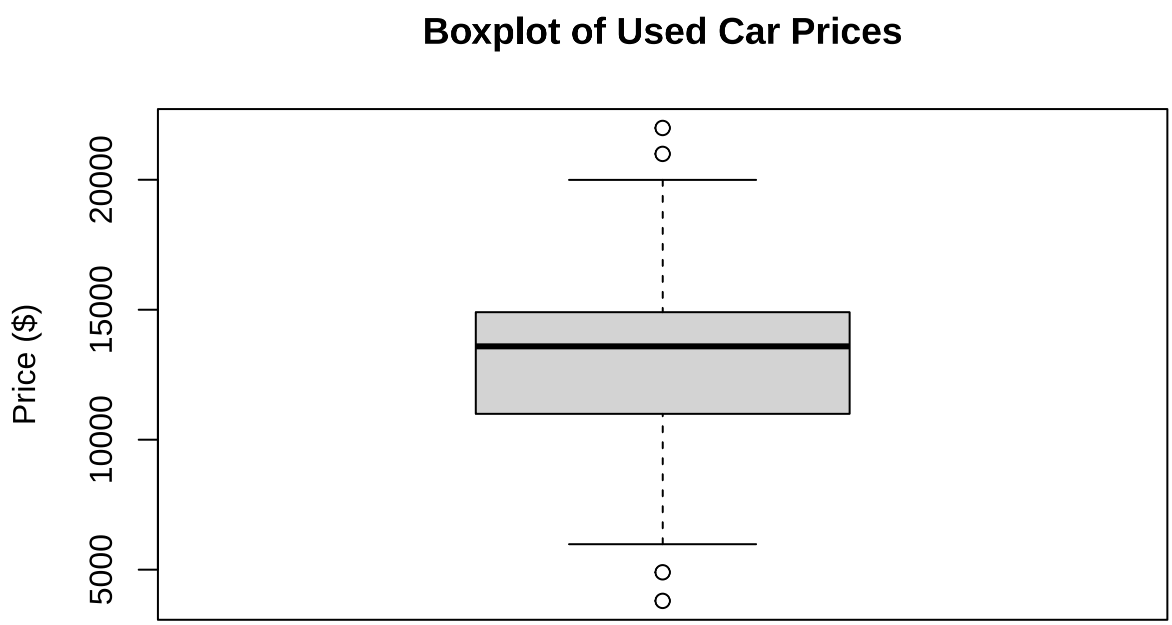
```r
# the 99th percentile
quantile(usedcars$price, probs = c(0.01, 0.99))
```

```
##       1%      99% 
##  5428.69 20505.00
```

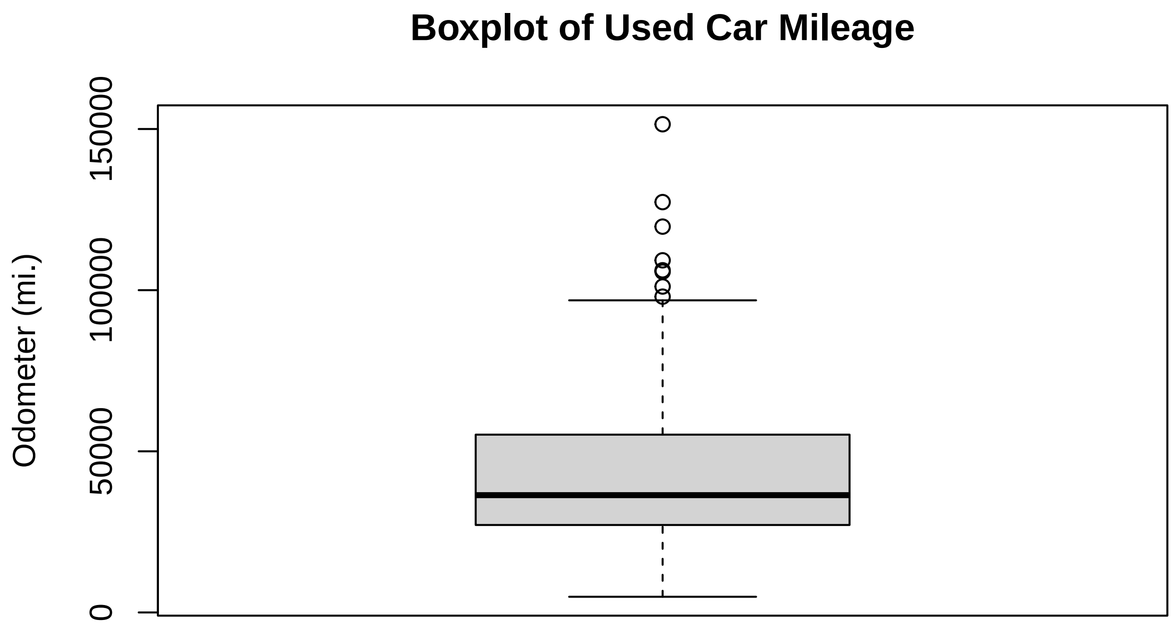
```r
# quintiles
quantile(usedcars$price, seq(from = 0, to = 1, by = 0.20))
```

```
##      0%     20%     40%     60%     80%    100% 
##  3800.0 10759.4 12993.8 13992.0 14999.0 21992.0
```

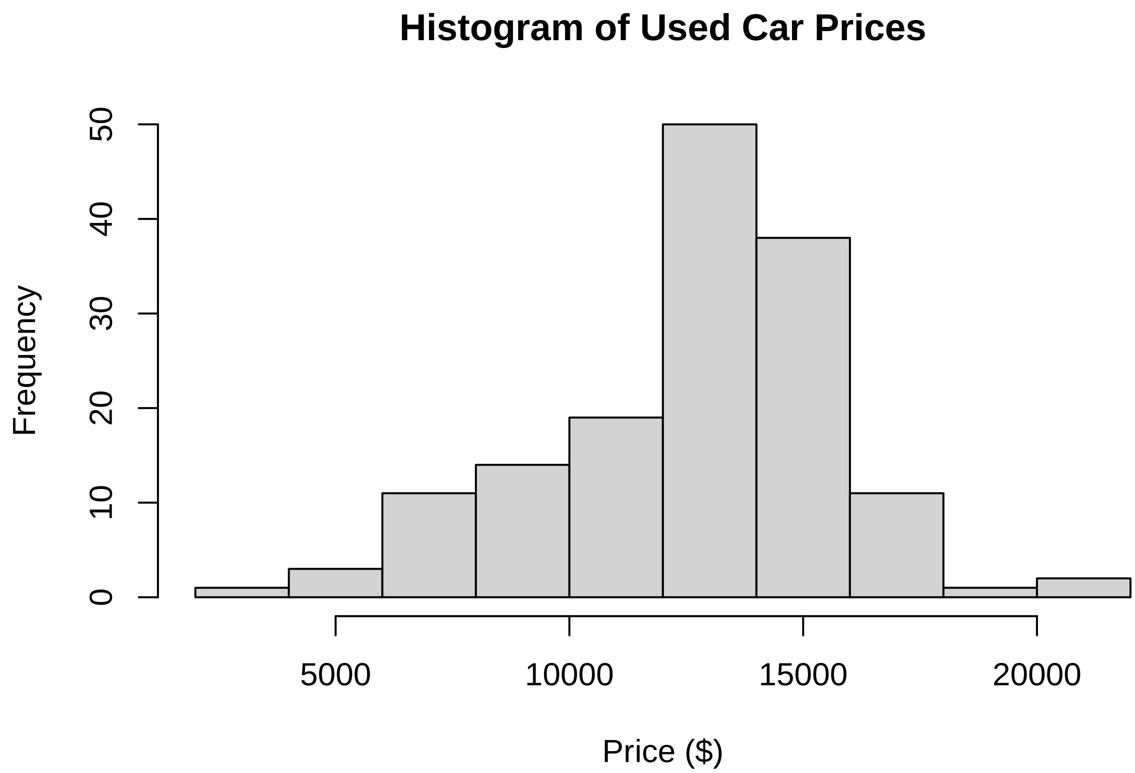
```r
# boxplot of used car prices and mileage
boxplot(usedcars$price, main="Boxplot of Used Car Prices",
      ylab="Price ($)")
```



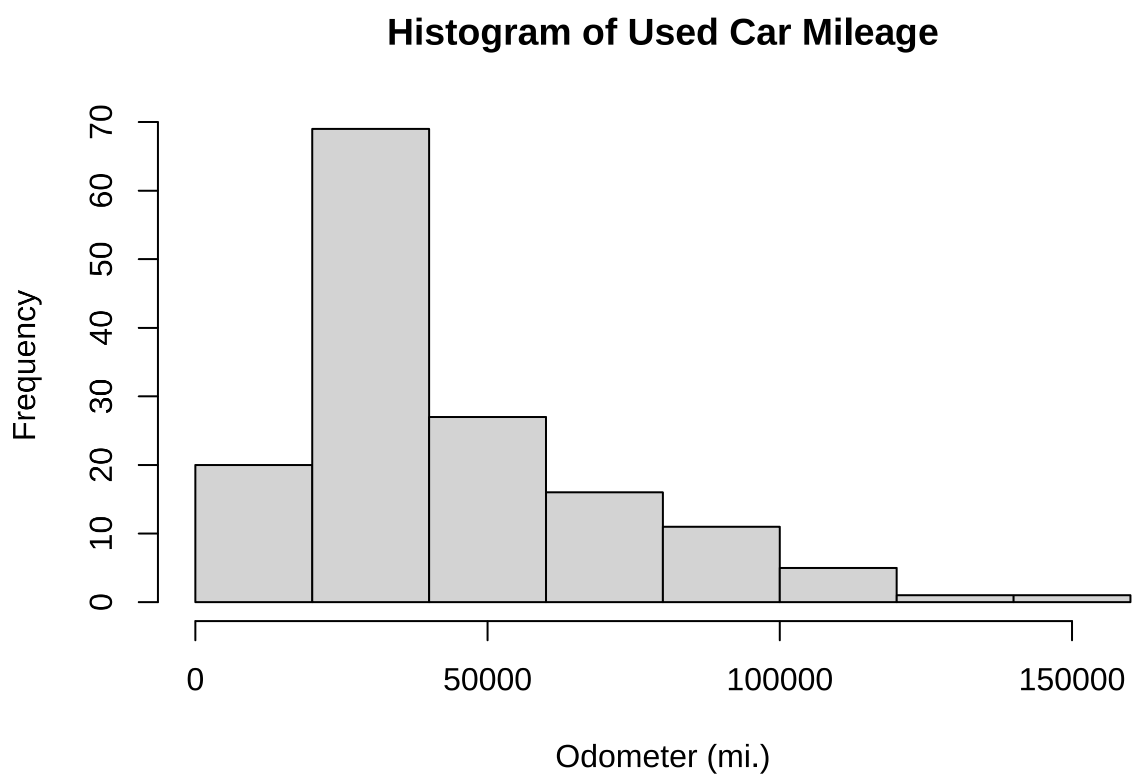

```r
boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage",
      ylab="Odometer (mi.)")
```




```r
# histograms of used car prices and mileage
hist(usedcars$price, main = "Histogram of Used Car Prices",
     xlab = "Price ($)")
```




```r
hist(usedcars$mileage, main = "Histogram of Used Car Mileage",
     xlab = "Odometer (mi.)")
```




```r
# variance and standard deviation of the used car data
var(usedcars$price)
```

```
## [1] 9749892
```

```r
sd(usedcars$price)
```

```
## [1] 3122.482
```

```r
var(usedcars$mileage)
```

```
## [1] 728033954
```

```r
sd(usedcars$mileage)
```

```
## [1] 26982.1
```

Exploring numeric variables

```r
# one-way tables for the used car data
table(usedcars$year)
```

```
##
## 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
##    3    1    1    1    3    2    6   11   14   42   49   16    1
```

```r
table(usedcars$model)
```

```
##
##  SE SEL SES
##  78  23  49
```

```r
table(usedcars$color)
```

```
##
##  Black   Blue   Gold   Gray  Green    Red Silver  White Yellow
##     35     17      1     16      5     25     32     16      3
```

```r
# compute table proportions
model_table <- table(usedcars$model)
prop.table(model_table)
```

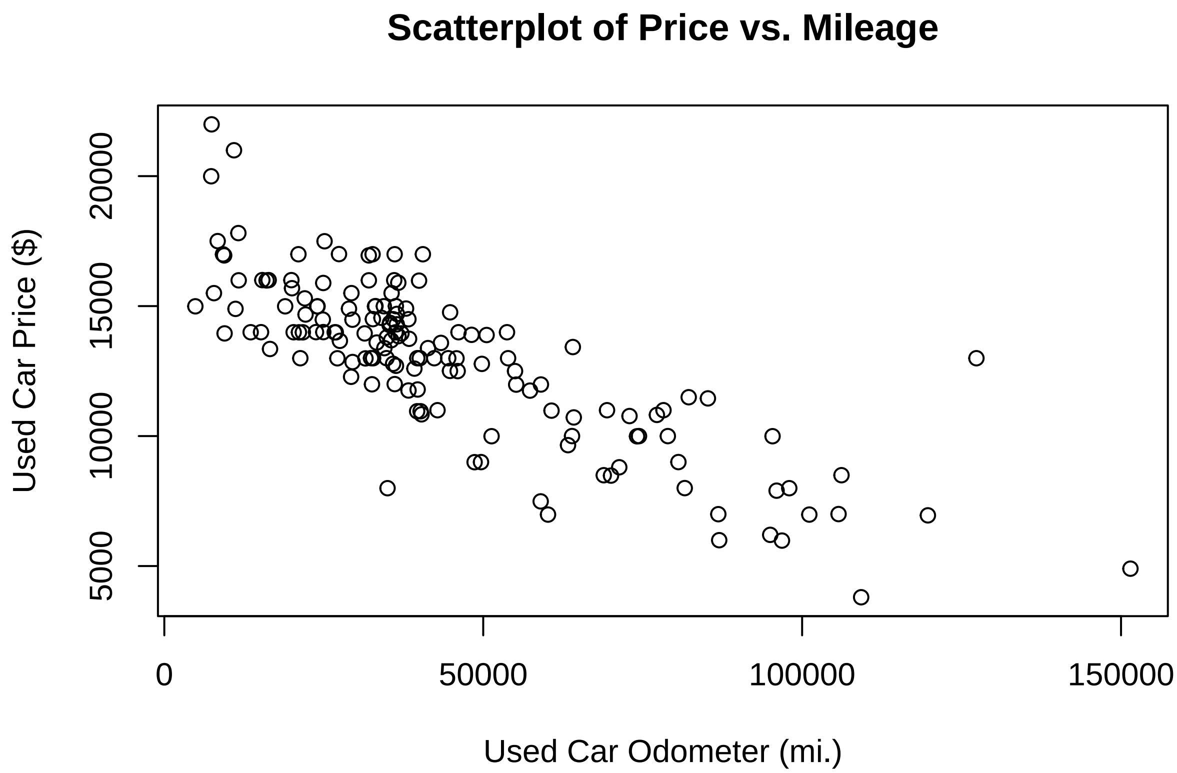
```
##
##        SE       SEL       SES
## 0.5200000 0.1533333 0.3266667
```

```r
# round the data
color_table <- table(usedcars$color)
color_pct <- prop.table(color_table) * 100
round(color_pct, digits = 1)
```

```
##
##  Black   Blue   Gold   Gray  Green    Red Silver  White Yellow
##   23.3   11.3    0.7   10.7    3.3   16.7   21.3   10.7    2.0
```

Exploring relationships between variables

```r
# scatterplot of price vs. mileage
plot(x = usedcars$mileage, y = usedcars$price,
     main = "Scatterplot of Price vs. Mileage",
     xlab = "Used Car Odometer (mi.)",
     ylab = "Used Car Price ($)")
```




```r
# new variable indicating conservative colors
usedcars$conservative <-
  usedcars$color %in% c("Black", "Gray", "Silver", "White")

# checking our variable
table(usedcars$conservative)
```

```
##
## FALSE  TRUE
##    51    99
```

```r
# Crosstab of conservative by model
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.3.3
```

```r
CrossTable(x = usedcars$model, y = usedcars$conservative)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
```

```
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  150
##
##
##                 | usedcars$conservative
## usedcars$model |     FALSE |      TRUE | Row Total |
## ---------------|-----------|-----------|-----------|
##             SE |        27 |        51 |        78 |
##                |     0.009 |     0.004 |           |
##                |     0.346 |     0.654 |     0.520 |
##                |     0.529 |     0.515 |           |
##                |     0.180 |     0.340 |           |
## ---------------|-----------|-----------|-----------|
##            SEL |         7 |        16 |        23 |
##                |     0.086 |     0.044 |           |
##                |     0.304 |     0.696 |     0.153 |
##                |     0.137 |     0.162 |           |
##                |     0.047 |     0.107 |           |
## ---------------|-----------|-----------|-----------|
##            SES |        17 |        32 |        49 |
##                |     0.007 |     0.004 |           |
##                |     0.347 |     0.653 |     0.327 |
##                |     0.333 |     0.323 |           |
##                |     0.113 |     0.213 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |        51 |        99 |       150 |
##                |     0.340 |     0.660 |           |
## ---------------|-----------|-----------|-----------|
##
##
```

Measuring spread - variance and standard deviation

In order to calculate the standard deviation, we must first obtain the variance, which is defined as the average of the squared differences between each value and the mean value. In mathematical notation, the variance of a set of n values of x is defined by the following formula. The Greek letter mu ( ) (similar in appearance to an m) denotes the mean of the values, and the variance itself is denoted by the Greek letter sigma ( ) squared (similar to a b turned sideways):

$Var(X) = sigma^2 = 1/n \sum_{i=1}^{n} (x_1 - \ )^2$

The standard deviation is the square root of the variance, and is denoted by sigma as shown in the following formula:

$StdDev(X) = \sigma = \sqrt{\frac{1}{n} \sum_{i=n}^{n}} (x_1 - \mu)^2$

Reference

Andrieu, Christophe, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. 2003. *Machine Learning* 50 (1/2): 5–43. https://doi.org/10.1023/a:1020281327116.

Goldberg, David E., and John H. Holland. 1988. *Machine Learning* 3 (2/3): 95–99. https://doi.org/10.1023/a:1022602019183.

Lantz, Brett. 2023. *Machine Learning with r: Learn Techniques for Building and Improving Machine Learning Models, from Data Preparation to Model Tuning, Evaluation, and Working with Big Data.* Packt Publishing Ltd.