Chapter 2 R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Chapter 2

This is an R Notebook with the code from Machine Learning with R, Lantz.

Blog post about Projects and Notebooks

Prime Hings for Running A Data Project in R

Good website for learning about R.

Quick-R

Chapter 2: Managing and Understanding Data

Libraries

```
library(here)
```

```
## here() starts at /home/esuess/classes/2017-2018/03 - Spring2018/Stat6620/Projects/Chap02
```

```
library(lattice)
library(corrgram)
library(gmodels)
```

```
here::here()
```

```
## [1] "/home/esuess/classes/2017-2018/03 - Spring2018/Stat6620/Projects/Chap02"
```

R data structures

Vectors

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
```

examples of accessing items in vector

include items in the range 2 to 3

```
temperature[2:3]
```

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

exclude item 2 using the minus sign

```
temperature[-2]
```

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

use a vector to indicate whether to include item

```
temperature[c(TRUE, TRUE, FALSE)]
```

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

Factors

add gender factor

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

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

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
```

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
```

check for symptoms greater than moderate

```
symptoms > "MODERATE"
```

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

Lists

display information for a patient

```
subject_name[1]
```

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

```
temperature[1]
```

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

```
flu_status[1]
```

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

```
gender[1]
```

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

```
blood[1]
```

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

```
symptoms[1]
```

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

create list for a patient and display the patient

```
subject1 <- list(fullname = subject_name[1],
                 temperature = temperature[1],
                 flu_status = flu_status[1],
                 gender = gender[1],
                 blood = blood[1],
                 symptoms = symptoms[1])
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
```

methods for accessing a list

get a single list value by position (returns a sub-list)

```
subject1[2]
```

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

get a single list value by position (returns a numeric vector)

```
subject1[[2]]
```

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

get a single list value by name

```
subject1$temperature
```

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

get several list items by specifying a vector of names

```
subject1[c("temperature", "flu_status")]
```

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

access a list like a vector get values 2 and 3

```
subject1[2:3]
```

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

Data frames

create a data frame from medical patient data and display the data frame

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

accessing a data frame

get a single column

```
pt_data$subject_name
```

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

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
```

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
```

accessing by row and column

```
pt_data[1, 2]
```

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

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

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

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

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

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

```
# 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
```

the following are equivalent

```r
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

create a 2x2 matrix

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

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

equivalent to the above

```r
m <- matrix(c(1, 2, 3, 4), ncol = 2)
m
```

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

create a 2x3 matrix

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

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

create a 3x2 matrix

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

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

extract values from matrixes

```r
m[1, 1]
```

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

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

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

extract rows

```
m[1, ]
```

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

extract columns

```
m[, 1]
```

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

Managing data with R

saving, loading, and removing R data structures

show all data structures in memory

```
ls()
```

```
## [1] "blood"        "flu_status"   "gender"       "m"
## [5] "pt_data"      "subject_name" "subject1"     "symptoms"
## [9] "temperature"
```

remove the m and subject1 objects

```
rm(m, subject1)
ls()
```

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

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

Exploring and understanding data

data exploration example using used car data

```
usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
```

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" ...
```

Exploring numeric variables

summarize numeric variables

```
summary(usedcars$year)
```

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

```
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.: 55124
##  Max.   :21992   Max.   :151479
```

calculate the mean income

```
(36000 + 44000 + 56000) / 3
```

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

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

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

the median income

```
median(c(36000, 44000, 56000))
```

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

the min/max of used car prices

```
range(usedcars$price)
```

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

the difference of the range

```
diff(range(usedcars$price))
```

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

IQR for used car prices

```
IQR(usedcars$price)
```

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

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
```

the 99th percentile

```
quantile(usedcars$price, probs = c(0.01, 0.99))
```

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

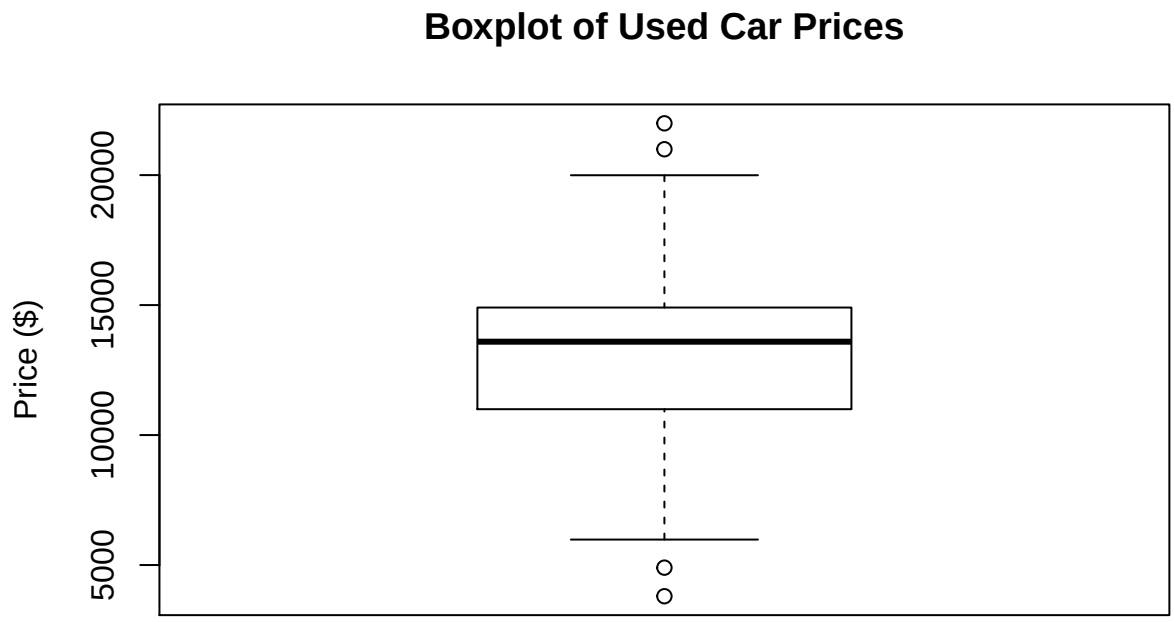
quintiles

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

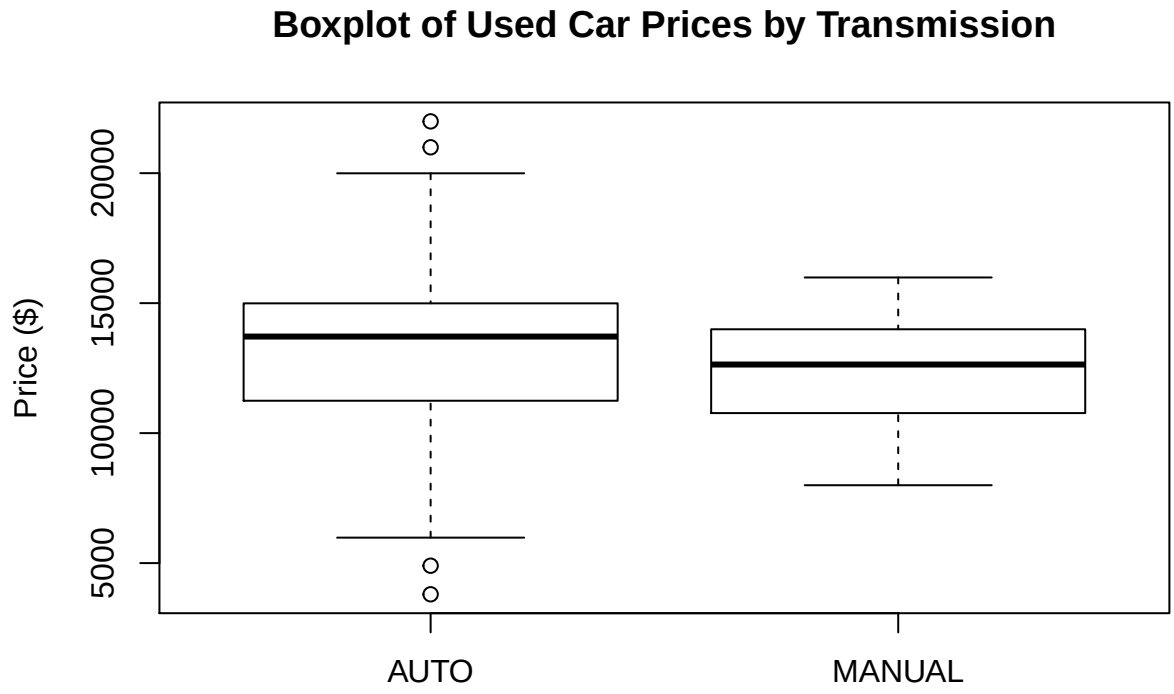
```
##       0%      20%      40%      60%      80%     100%
##  3800.0 10759.4 12993.8 13992.0 14999.0 21992.0
```

boxplot of used car prices and mileage

```
boxplot(usedcars$price, main="Boxplot of Used Car Prices",
      ylab="Price ($)")
```

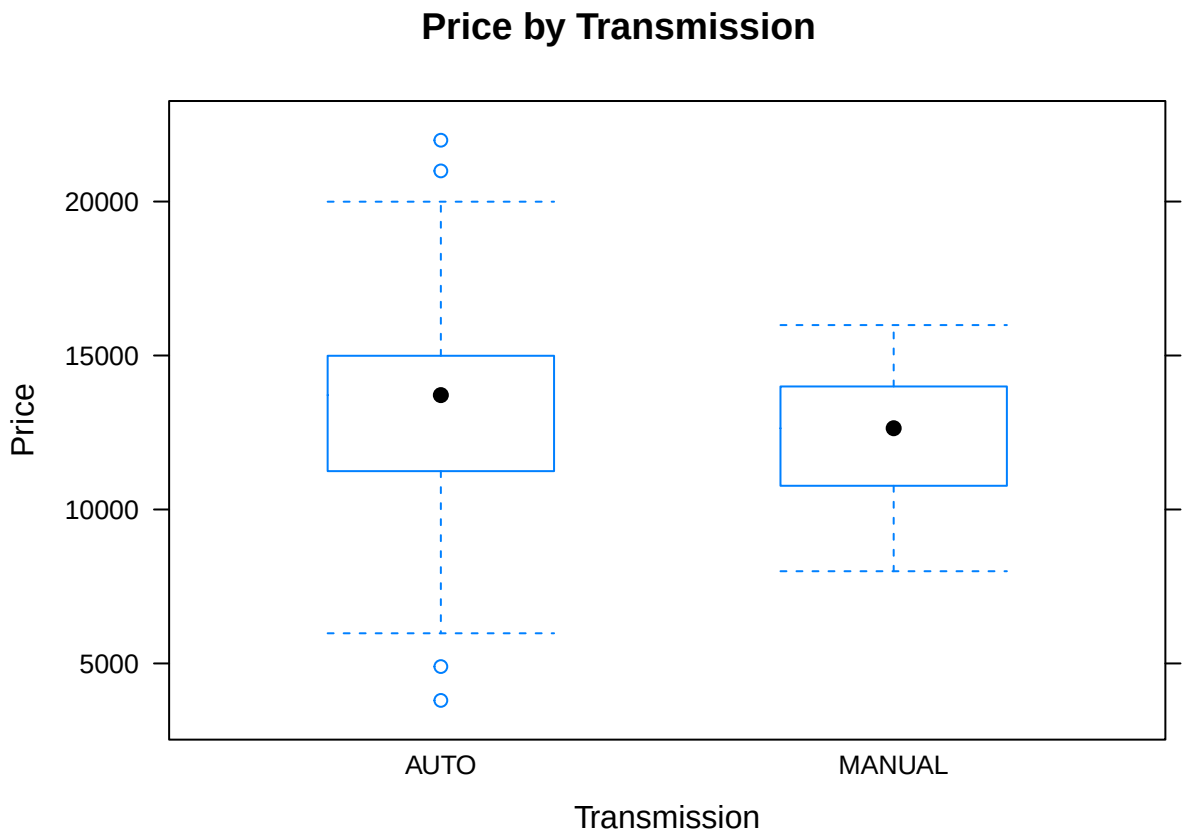



```
boxplot(usedcars$price ~ usedcars$transmission, main="Boxplot of Used Car Prices by Transmission",
      ylab="Price ($)")
```

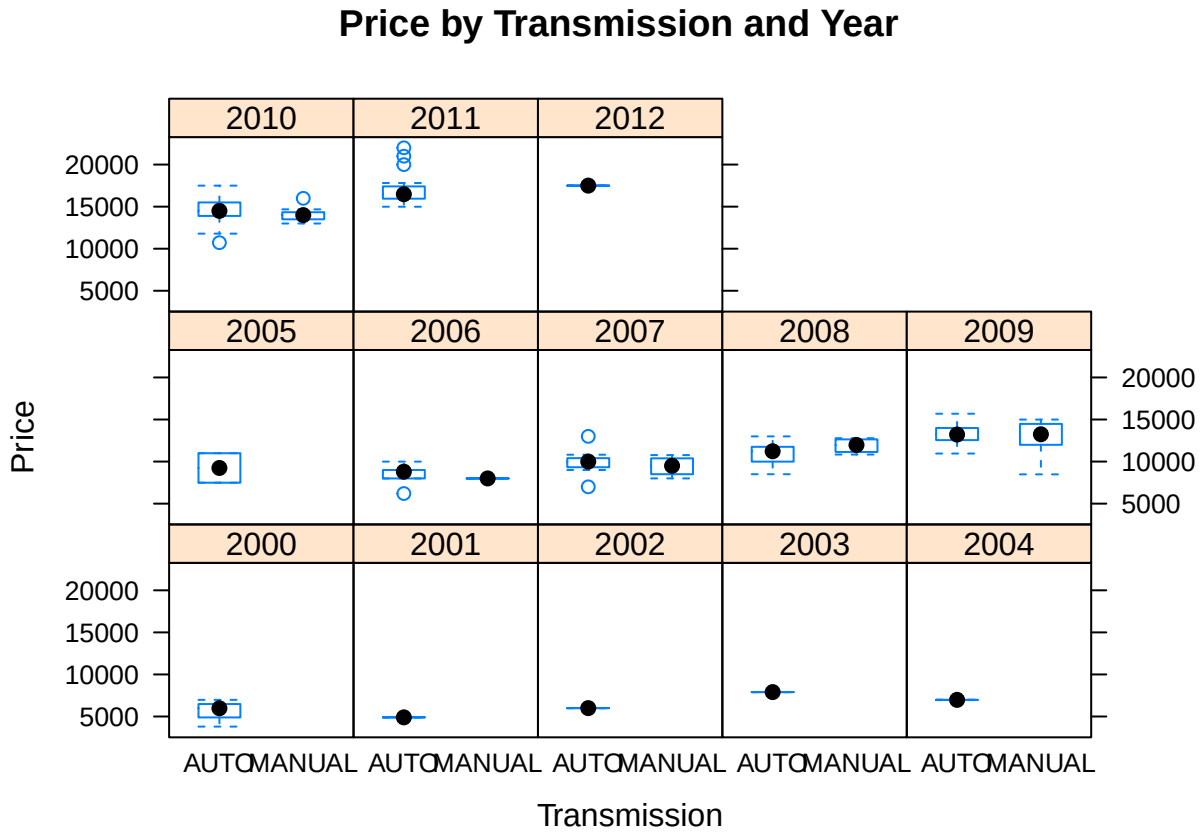


using the lattice package

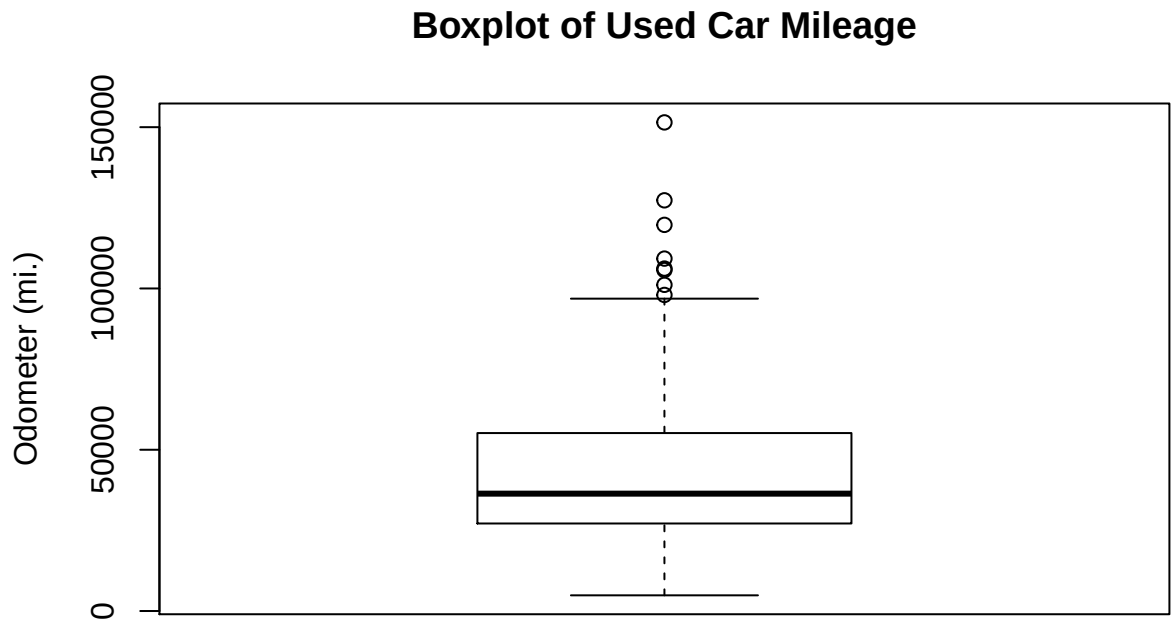
```
lattice::bwplot(usedcars$price~usedcars$transmission,
   ylab="Price", xlab="Transmission",
   main="Price by Transmission")
```



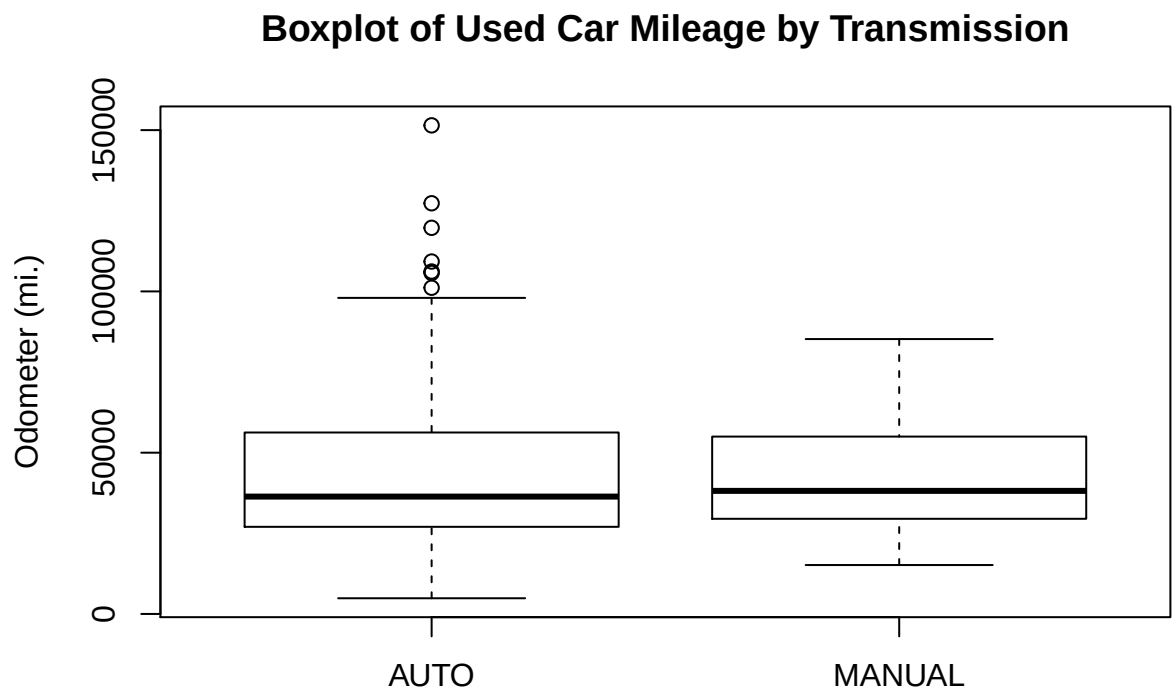

```
usedcars$year <- as.character(usedcars$year)

lattice::bwplot(usedcars$price~usedcars$transmission|usedcars$year,
   ylab="Price", xlab="Transmission",
   main="Price by Transmission and Year", layout=(c(5,3)))
```




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

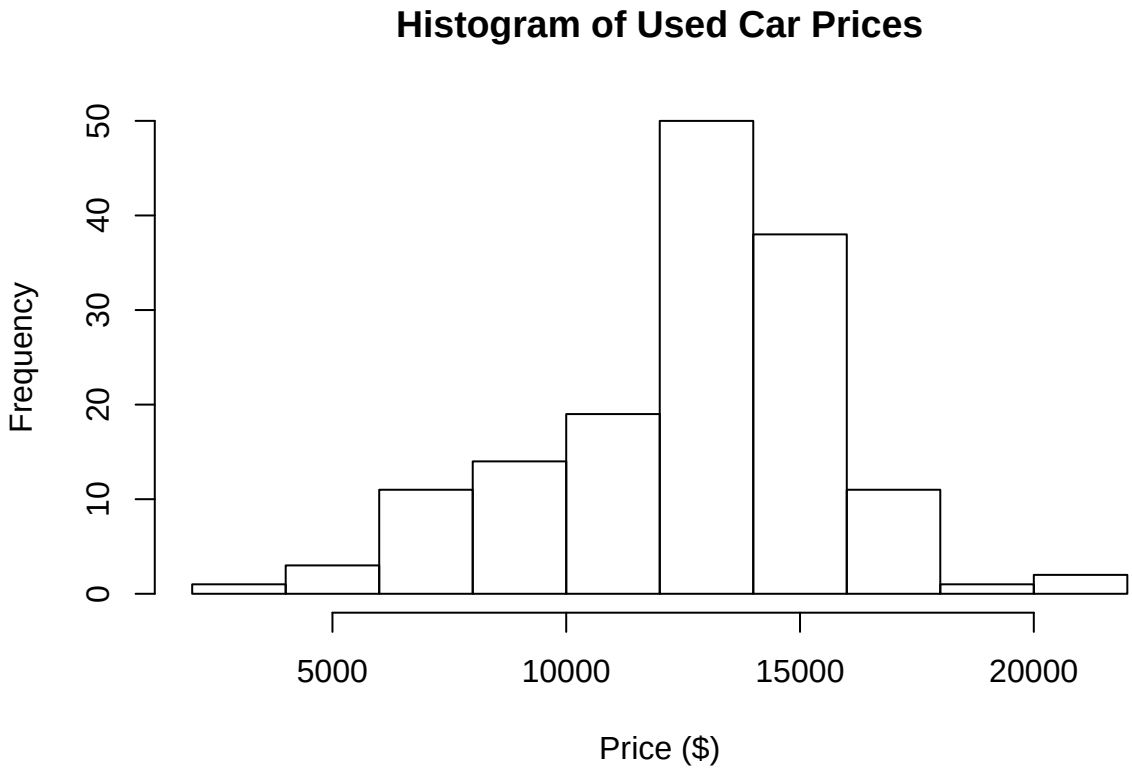



```
boxplot(usedcars$mileage ~ usedcars$transmission, main="Boxplot of Used Car Mileage by Transmission", yl
```

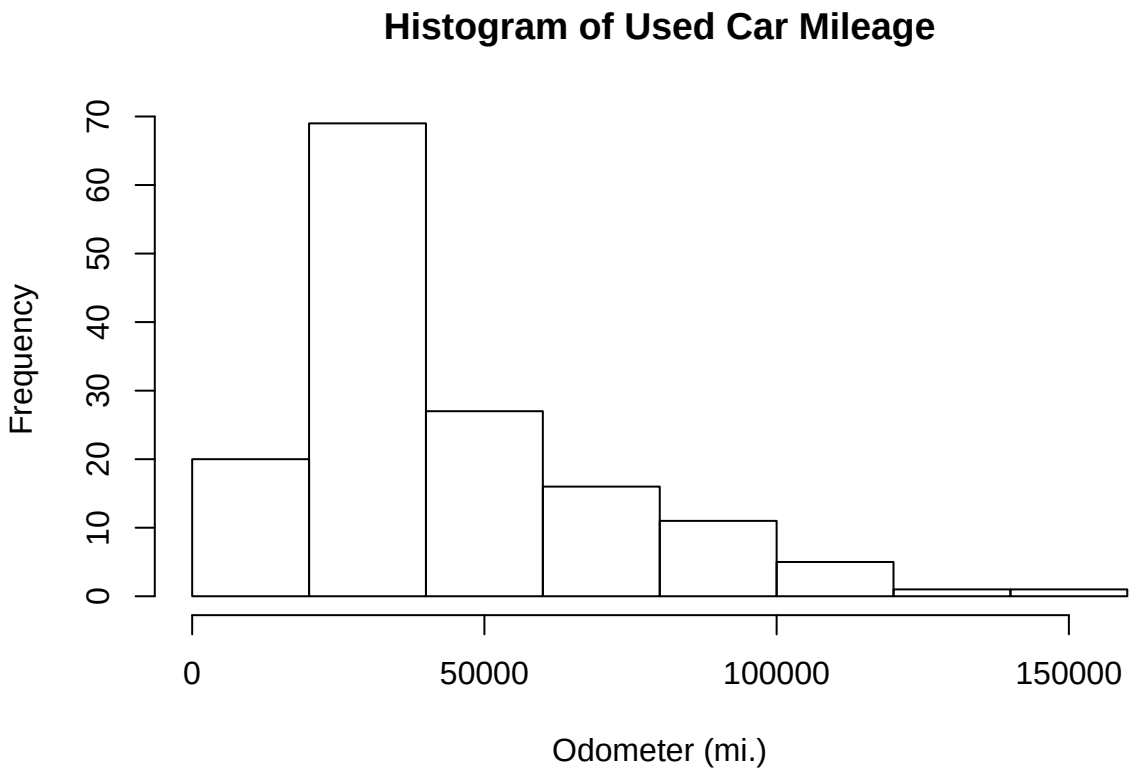


histograms of used car prices and mileage

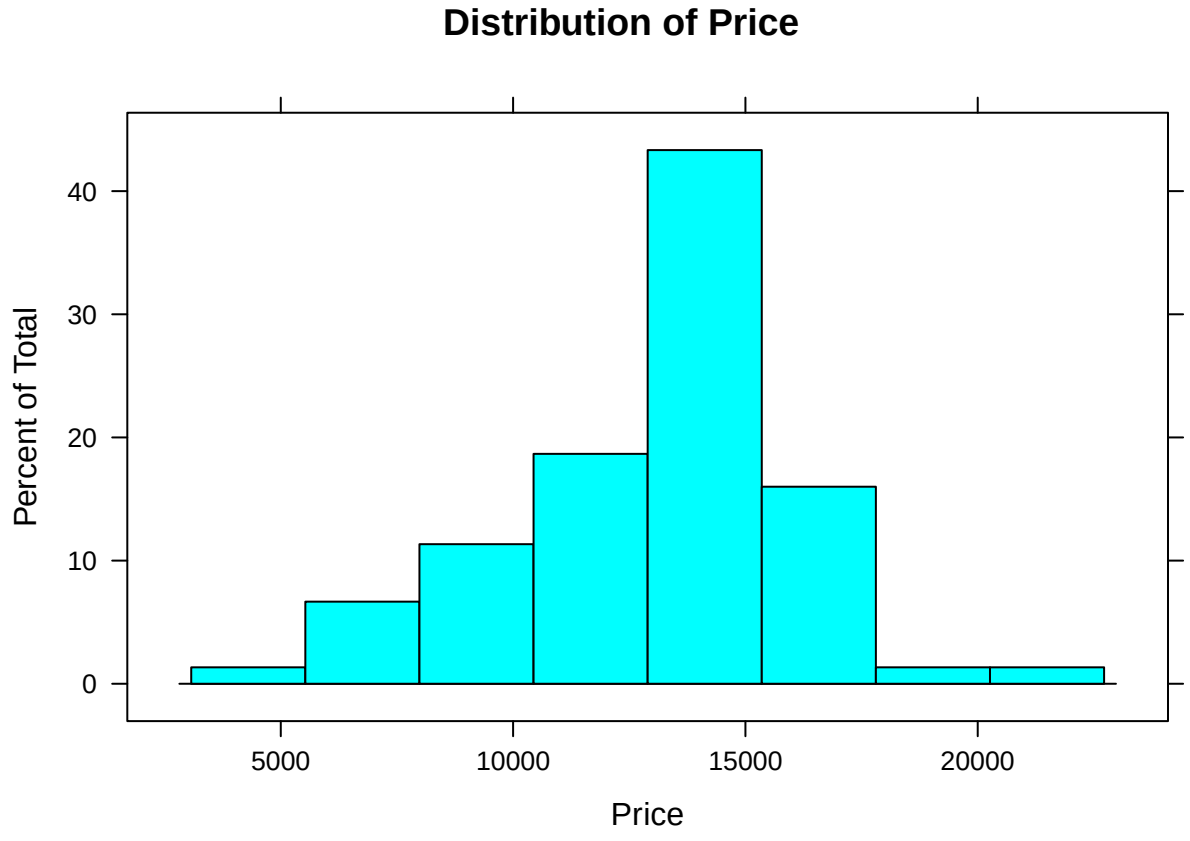
```
hist(usedcars$price, main = "Histogram of Used Car Prices",
     xlab = "Price ($)")
```



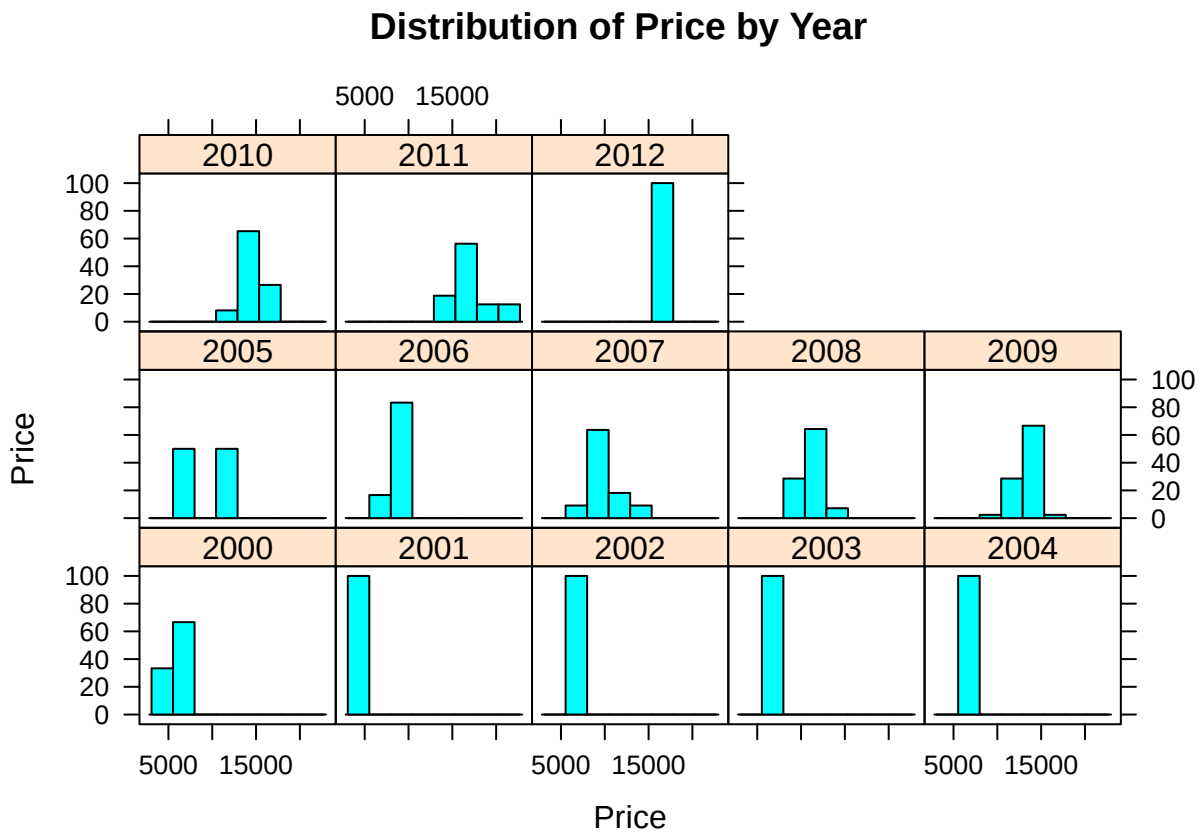

```
hist(usedcars$mileage, main = "Histogram of Used Car Mileage",
     xlab = "Odometer (mi.)")
```



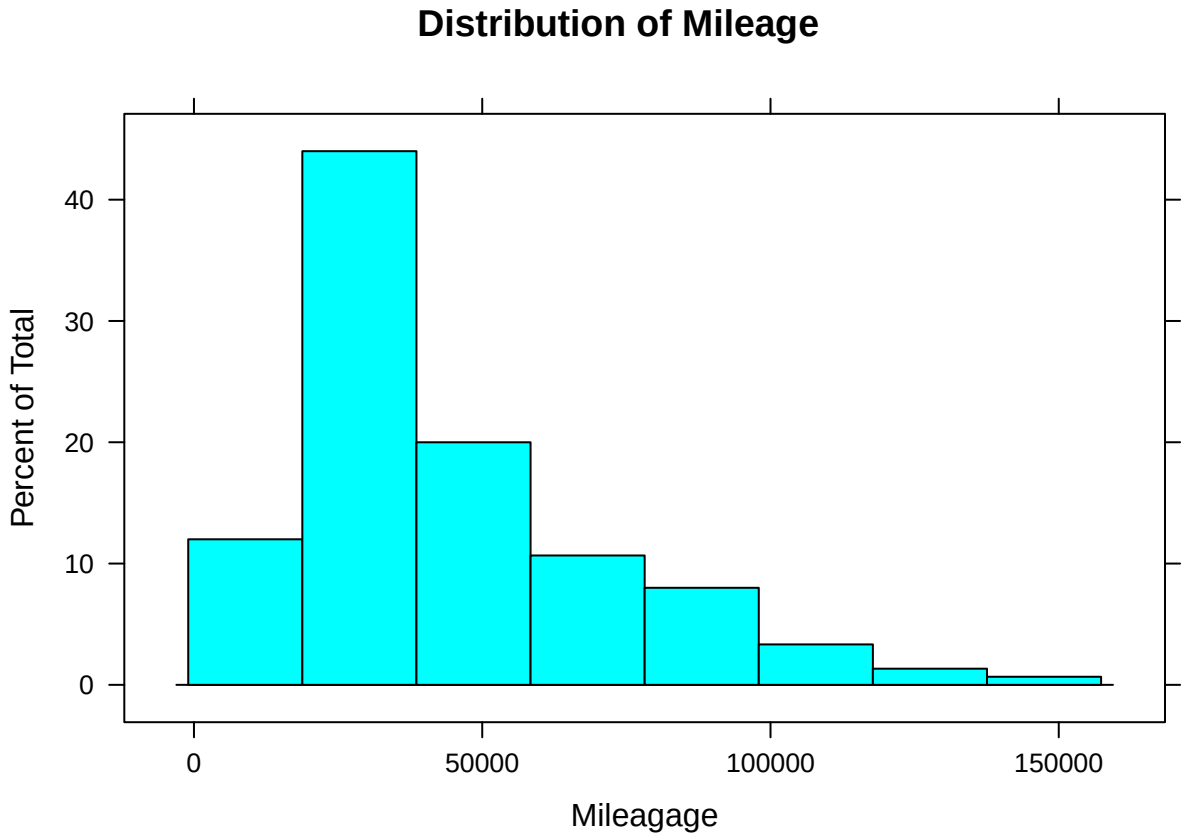

```
lattice::histogram(~ usedcars$price,
   xlab="Price",
   main="Distribution of Price")
```



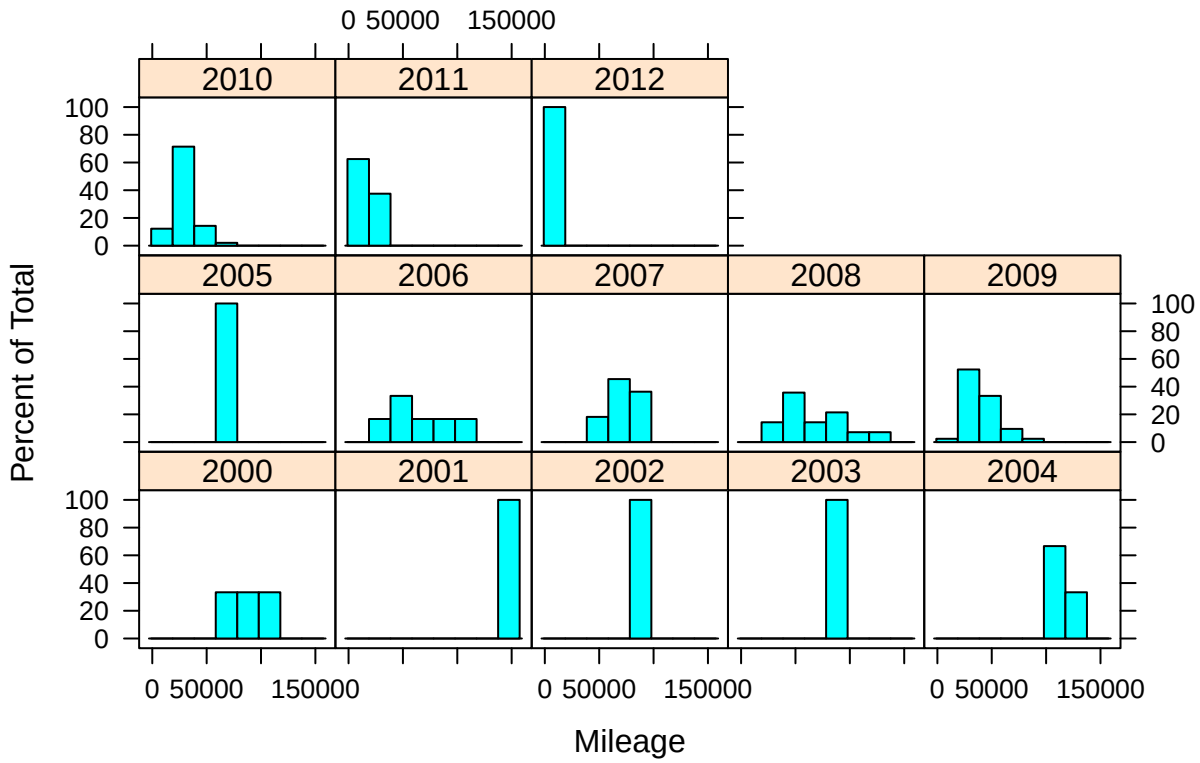
```
usedcars$year <- as.character(usedcars$year)

lattice::histogram(~ usedcars$price | usedcars$year,
   ylab="Price", xlab="Price",
   main="Distribution of Price by Year", layout=(c(5,3)))
```




```
lattice::histogram(~ usedcars$mileage,
   xlab="Mileagage",
   main="Distribution of Mileage")
```




```
usedcars$year <- as.character(usedcars$year)

lattice::histogram(~ usedcars$mileage | usedcars$year,
   xlab="Mileage",
   main="Distribution of Mileage by Year", layout=(c(5,3)))
```



variance and standard deviation of the used car data

```
var(usedcars$price)
```

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

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

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

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

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

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

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

Exploring numeric variables

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
```

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

```
##
```

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

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

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

compute table proportions

```
model_table <- table(usedcars$model)
prop.table(model_table)
```

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

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

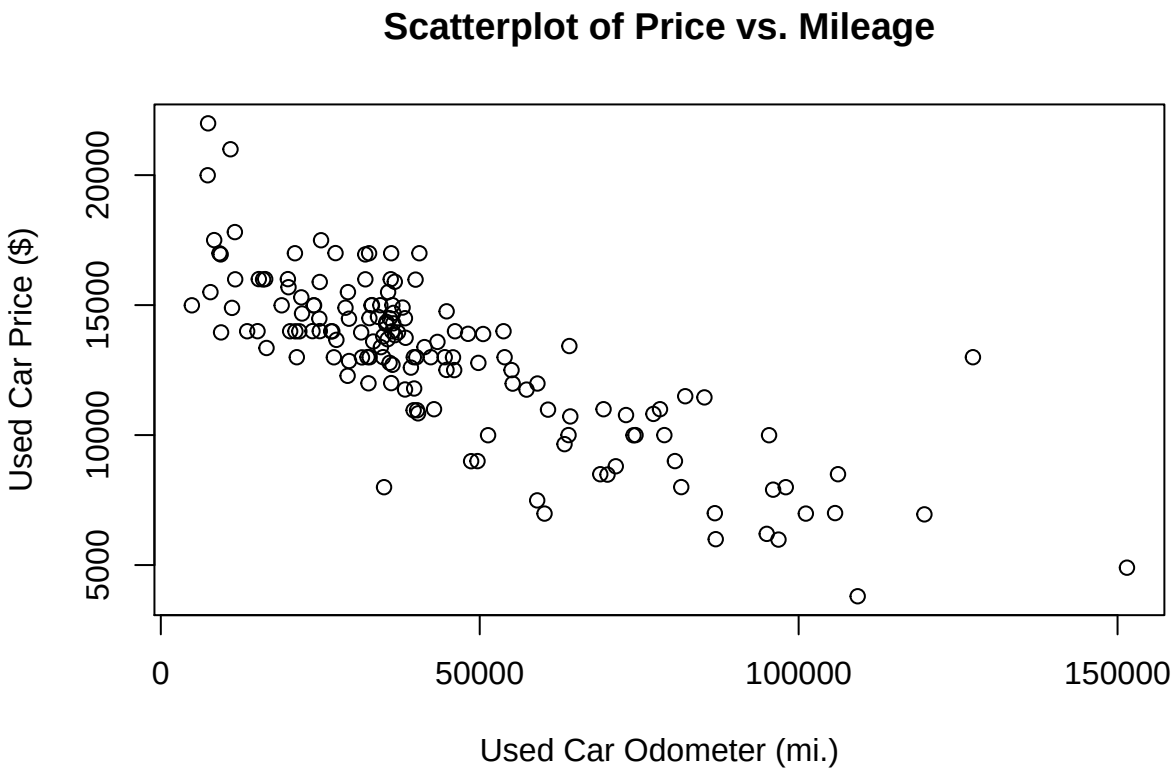
correlation

```
cor(x = usedcars$mileage, y = usedcars$price)
```

```
## [1] -0.8061494
```

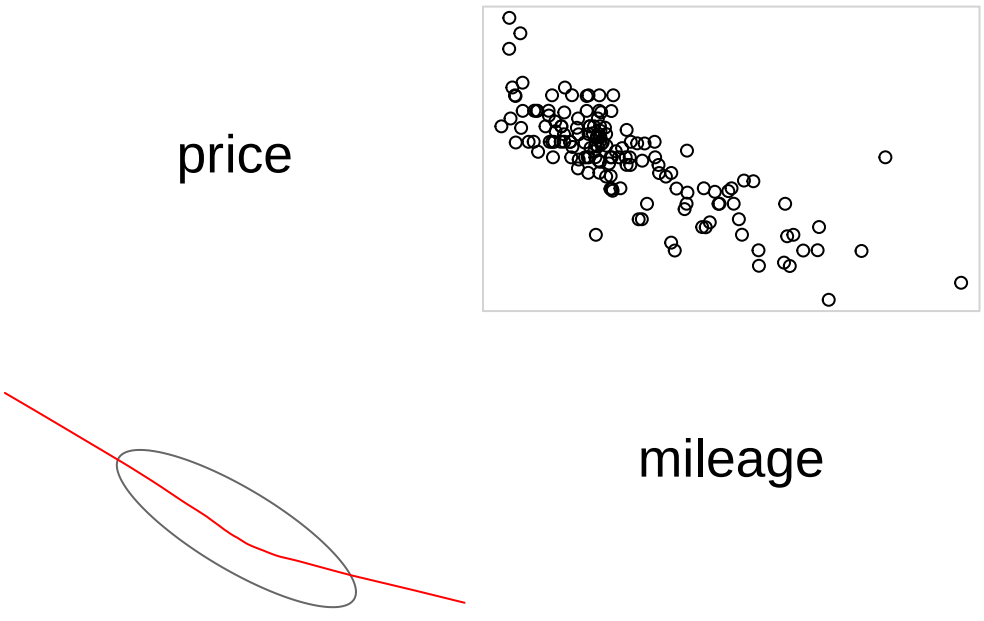
scatterplot of price vs. mileage

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



The corrgram package has the corrgram function that is nice for looking at relationships between numeric variable.

```
corrgram::corrgram(usedcars,lower.panel=panel.ellipse,
  upper.panel=panel.pts)
```



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
```

Crosstab of conservative by model

```
gmodels::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 |           |
## ---------------|-----------|-----------|-----------|
##
##
```