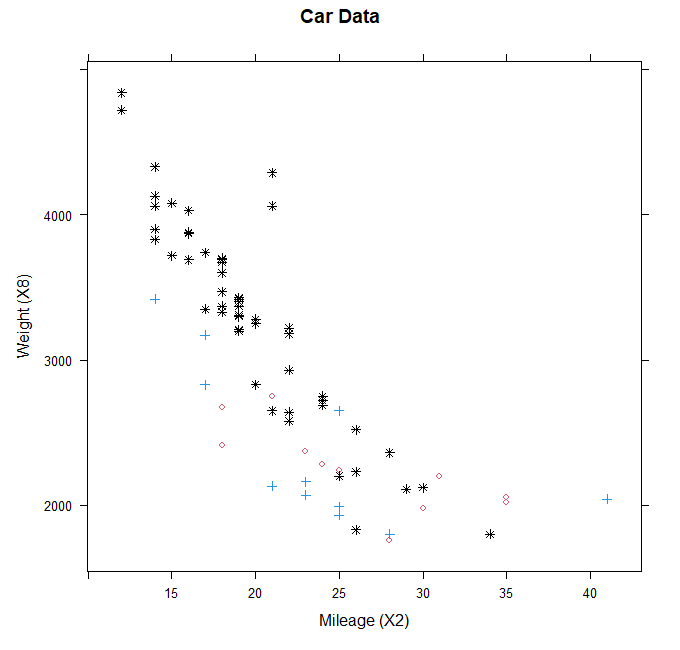
**Exercise 3.4** Compute the covariance between the variables

*X*2 = miles per gallon,

*X*8 = weight

from the car data set (Sect. 22.3). What sign do you expect the covariance to have?

Sol.

After calculate by R, it can be seen that the covariance of and is less than zero, it means that and have an inverse relationship. In other words, the higher weight of the car, the less mileage it can run (at the same gallon).

我們也可以藉由散佈圖驗證，當三種車款的有較大值時，對應會有較小值，形成負斜率的散佈圖，符合負的共變異數有負斜率的散佈圖這個特徵。

**Exercise 3.25** Compute the covariance matrix *S* = Cov*(X)* where *X* denotes the

matrix of observations on the counterfeit bank notes. Make a Jordan decomposition

of *S*. Why are all of the eigenvalues positive?

Sol.

**Eigenvalues:**

**Eigenvectors:**

**Jordan decomposition of S:**

Because the quadratic form of matrix S is positive, so it is positive definite matrix, it means that the eigenvalues will also be positive.

**Exercise 3.26** Compute the covariance of the counterfeit notes after they are linearly

transformed by the vector .

Sol.

Linear transformation of the matrix X leads to .

**R code:**

# clear all variables

rm(list = ls(all = TRUE))

graphics.off()

# 3.4

# cov

car <- read.table("C:/Users/user/Desktop/多變量11101/HW3\_1019/MVA-ToDo-master/QID-1532-MVAcareffect/carc.dat")

cov(car$V3,car$V9)

# install and load packages

libraries = c("lattice")

lapply(libraries, function(x) if (!(x %in% installed.packages())) {

install.packages(x)

})

lapply(libraries, library, quietly = TRUE, character.only = TRUE)

# scat

# load data

M = car[, 3]

W = car[, 9]

C = car[, 14]

# point definition

D = C

D[car[, 14] == 2] = 1

D[car[, 14] == 1] = 8

# color definition

P = C

P[car[, 14] == 3] = 4

P[car[, 14] == 2] = 2

P[car[, 14] == 1] = 1

leg = c(8, 1, 3)

# plot

xyplot(W ~ M, pch = D, col = P, xlab = "Mileage (X2)", ylab = "Weight (X8)", main = "Car Data")

# 3.25

bank <- read.table("C:/Users/user/Desktop/多變量11101/HW3\_1019/MVA-ToDo-master/QID-948-MVApcabankr/bank2.dat")

S <- cov(bank[101:200,])

S

Avalues <- eigen(S)$values

Avectors <- eigen(S)$vectors

Avalues

Avectors

t(Avectors)

J <- Avectors%\*%diag(Avalues)%\*%t(Avectors)

J

Q <- t(matrix(Avalues))%\*%S%\*%matrix(Avalues)

Q

# 3.26

a <- matrix(c(1, 1, 1, 1, 1, 1))

a

Sy <- t(a)%\*%S%\*%a

Sy