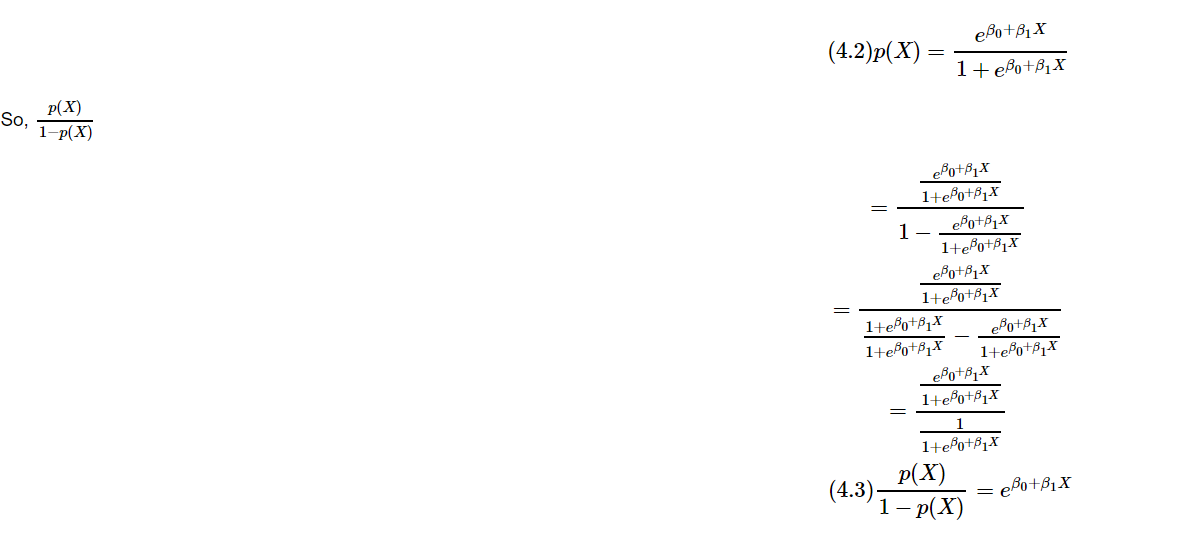
高级统计方法 第5次作业:

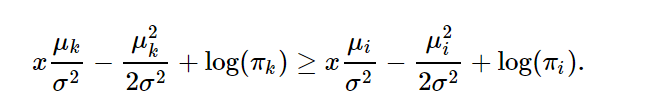
序号： 27 姓名：王琪瑞 学号：20202241014 班级：软2002

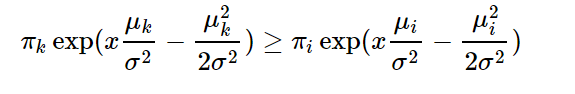
**概念**

1.问题（略）

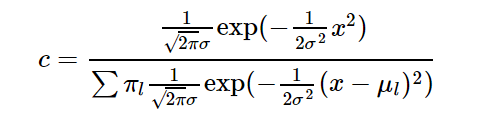


2.问题（略）

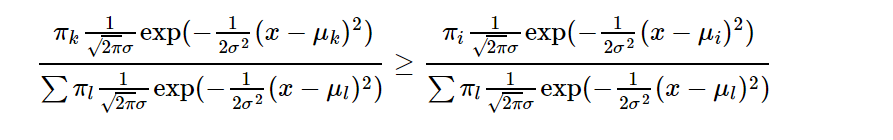




我们乘以常数c:



得到：



4.问题（略）

（a）问题（略）

10%。只需观测X附近10%的数据， < 0.05和 > 0.95的数据可以忽略。

（b）问题（略）

1%。与a类似，X1和X2各只需观测附近10%的数据，10%\*10%=1%。

（c）问题（略）

与b类似，为10%^100=10^-98%。

（d）问题（略）

当变量维数p以线性增长时，可观测的数据呈指数形式的减少，当p很大时，对任何给定的测试观测，附近都很少有训练观测。

（e）问题（略）

p=1, l=0.1

p=2, l=

……

p=n, l=

5.问题（略）

（a）问题（略）

在训练集上，QDA效果更好，因为具有更高的光滑度，有着更好的拟合效果。

在测试集上，LDA效果更好，因为LDA拟合更接近实际，而QDA会过拟合。

（b）问题（略）

在训练集和测试集上都是QDA效果更好。因为如果贝叶斯决策边界是非线性的，QDA既有更高的光滑度，也与实际更加接近。

（c）问题（略）

QDA的预测率会变得更好。因为 QDA 有着更高的光滑度，光滑度高的模型适合拟合样本量很大的情况，LDA会出现欠拟合的情况。

（d）问题（略）

错误。如果样本量很少，QDA的高光滑度容易产生过拟合的情况，会有更高的测试误差，无法表示一个线性决策边界。

6.问题（略）

（a）问题（略）

根据题意，得出多元逻辑斯蒂回归公式为：

其中，

将，

代入可得，p=37.75.

（b）问题（略）

即将，代入方程

得到，即需要学习50小时才能保证。

8.问题（略）

应该采取逻辑斯蒂回归。对于K=1的最近邻法，其训练错误率为0，对于观测值来说，最近邻就是其本身，所以其测试错误率为36%。而逻辑斯蒂回归的测试错误率为30%，相比最近邻法更低，因此采取逻辑斯蒂回归。

9.问题（略）

（a）问题（略）

发生比的公式为

代入0.37求得，p(X)=0.27

（b）问题（略）

将p(X)=0.16代入发生比的公式

求得发生比为0.19

**应用**

10.问题（略）

（a）问题（略）

脚本：

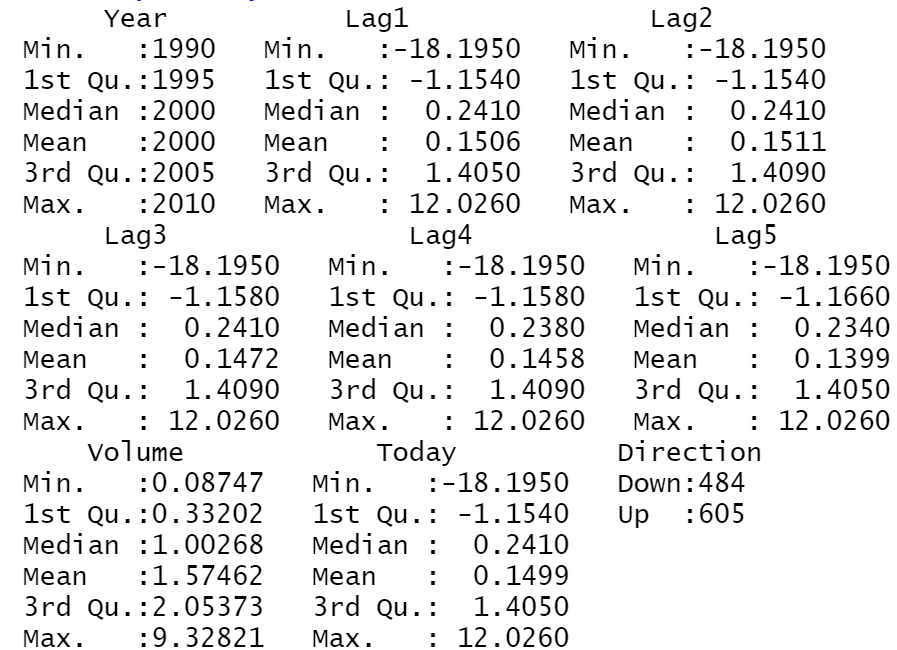
library(ISLR2)

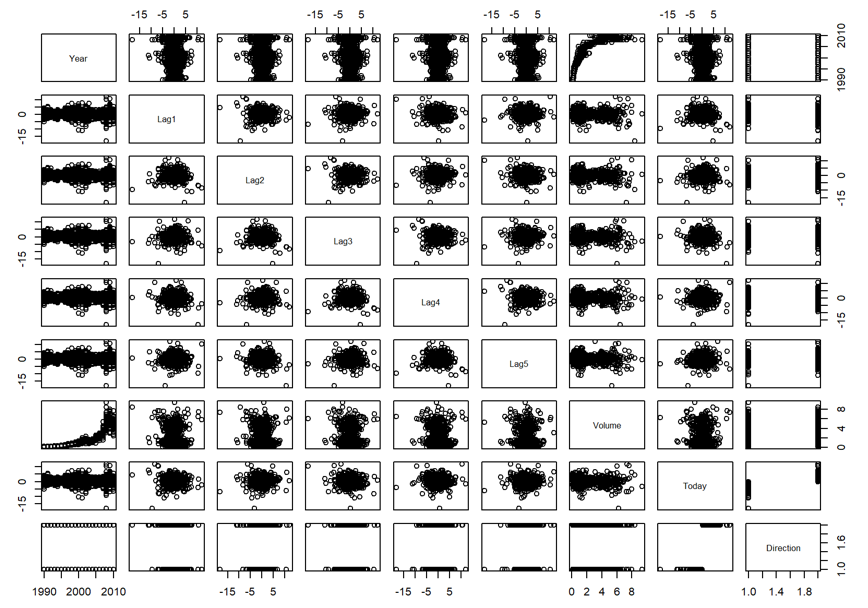
summary(Weekly)

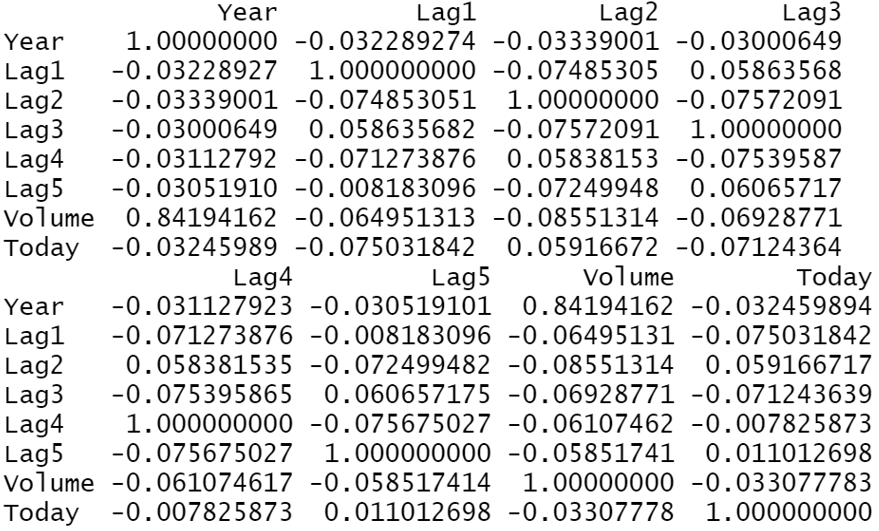
pairs(Weekly)

cor(Weekly[,-9])

截图：







从图中可以看出，Year和Volume有关系，其他模式都不怎么有关系。

（b）问题（略）

脚本：

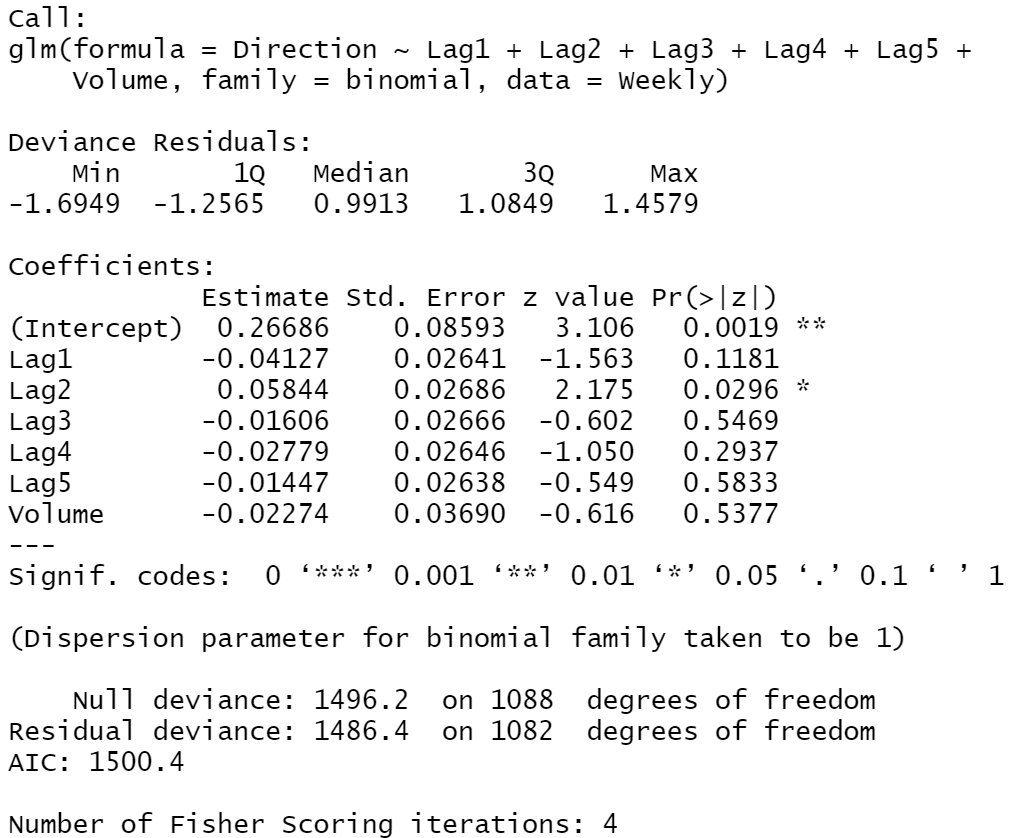
attach(Weekly)

glm.fit = glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Weekly,

family=binomial)

summary(glm.fit)

截图：



根据p值可以看出，Lag2具有统计上的显著性，因为其p值小于3%。

（c）问题（略）

脚本：

glm.probs = predict(glm.fit, type="response")

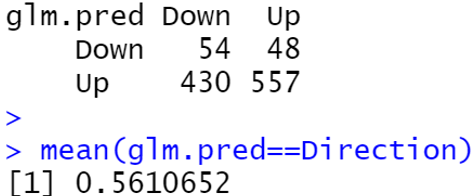
glm.pred = rep("Down", length(glm.probs))

glm.pred[glm.probs>.5] = "Up"

table(glm.pred, Direction)

mean(glm.pred==Direction)

截图：



预测准确率是0.561

第一类错误——假阳性率为430/(430+54)=88.84%。

第二类错误——假阴性率为48/(557+48)=7.93%。

错误应该为第一类错误。

在市场上升的时候，回归预测大部分是正确的：557/(557+48)= 92.1%。

在市场下跌的时候，回归预测大部分是错误的：54/(430+54)= 11.2%。

（d）问题（略）

脚本：

train = (Year < 2009)

Weekly.0910 = Weekly[!train,]

glm.fit = glm(Direction~Lag2,data=Weekly,family=binomial,subset=train)

glm.probs = predict(glm.fit, Weekly.0910,type="response")

glm.pred = rep("Down", length(glm.probs))

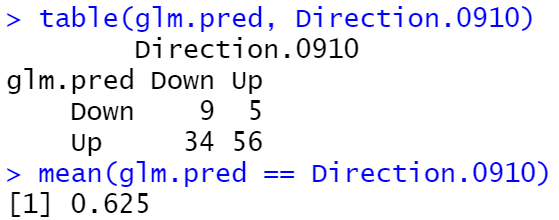
glm.pred[glm.probs>.5] = "Up"

Direction.0910 = Direction[!train]

table(glm.pred, Direction.0910)

mean(glm.pred == Direction.0910)

截图：



可以看到预测准确率是0.625

（e）问题（略）

脚本：

library(MASS)

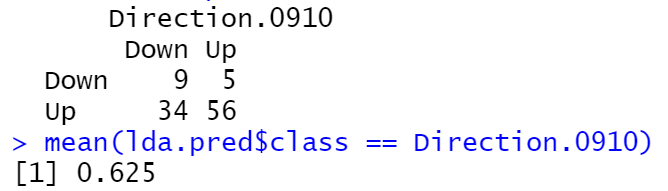
lda.fit = lda(Direction ~ Lag2, data=Weekly, subset=train)

lda.pred = predict(lda.fit, Weekly.0910)

table(lda.pred$class, Direction.0910)

mean(lda.pred$class == Direction.0910)

截图：



可以看到预测准确率是0.625

（f）问题（略）

脚本：

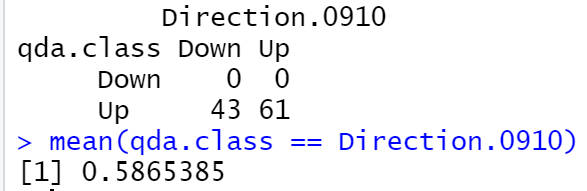
qda.fit = qda(Direction~Lag2, data=Weekly, subset=train)

qda.class = predict(qda.fit, Weekly.0910)$class

table(qda.class, Direction.0910)

mean(qda.class == Direction.0910)

截图：



可以看到预测准确率是0.587

（g）问题（略）

脚本：

library(class)

train.X = as.matrix(Lag2[train])

test.X = as.matrix(Lag2[!train])

train.Direction = Direction[train]

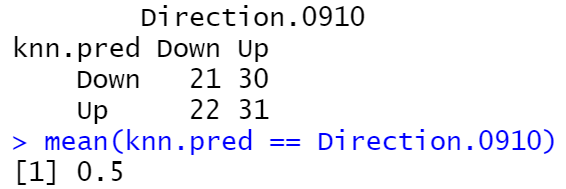
set.seed(1)

knn.pred = knn(train.X, test.X, train.Direction, k=1)

table(knn.pred, Direction.0910)

mean(knn.pred == Direction.0910)

截图：



可以看到预测准确率是0.5

（h）问题（略）

逻辑斯蒂回归和LDA有着相同且最好的准确率。

（i）问题（略）

脚本：

glm.fit=glm(Direction~Lag2+Lag1,data=Weekly,family=binomial, subset=train)

glm.probs = predict(glm.fit, Weekly.0910, type="response")

glm.pred = rep("Down", length(glm.probs))

glm.pred[glm.probs>.5] = "Up"

Direction.0910 = Direction[!train]

table(glm.pred, Direction.0910)

mean(glm.pred == Direction.0910)

lda.fit = lda(Direction ~ Lag2+Lag1, data=Weekly, subset=train)

lda.pred = predict(lda.fit, Weekly.0910)

table(lda.pred$class, Direction.0910)

mean(lda.pred$class == Direction.0910)

qda.fit = qda(Direction~ sqrt(abs(Lag2)), data=Weekly, subset=train)

qda.class = predict(qda.fit, Weekly.0910)$class

table(qda.class, Direction.0910)

mean(qda.class == Direction.0910)

knn.pred = knn(train.X, test.X, train.Direction, k=10)

table(knn.pred, Direction.0910)

mean(knn.pred == Direction.0910)

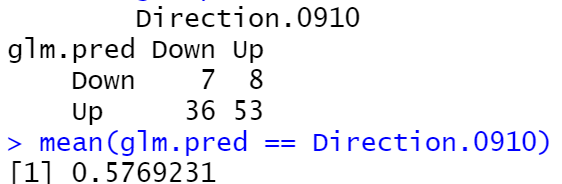
knn.pred = knn(train.X, test.X, train.Direction, k=100)

table(knn.pred, Direction.0910)

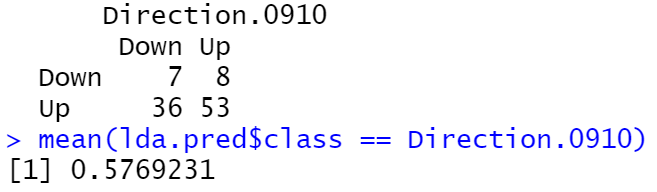
mean(knn.pred == Direction.0910)

截图：

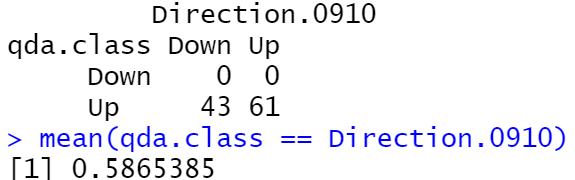
逻辑斯蒂回归，Lag1+Lag2



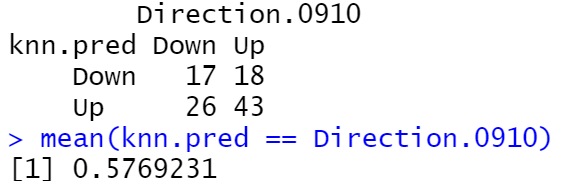
LDA,Lag1+Lag2



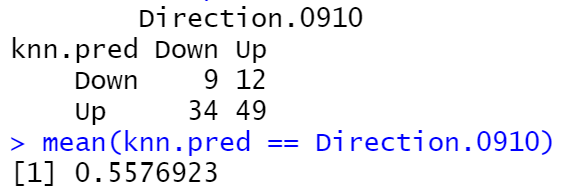
QDA, sqrt(abs(Lag2))



K=10



K=100



原逻辑斯蒂回归和LDA有着最好的准确率。

11.问题（略）

（a）问题（略）

脚本：

library(ISLR2)

attach(Auto)

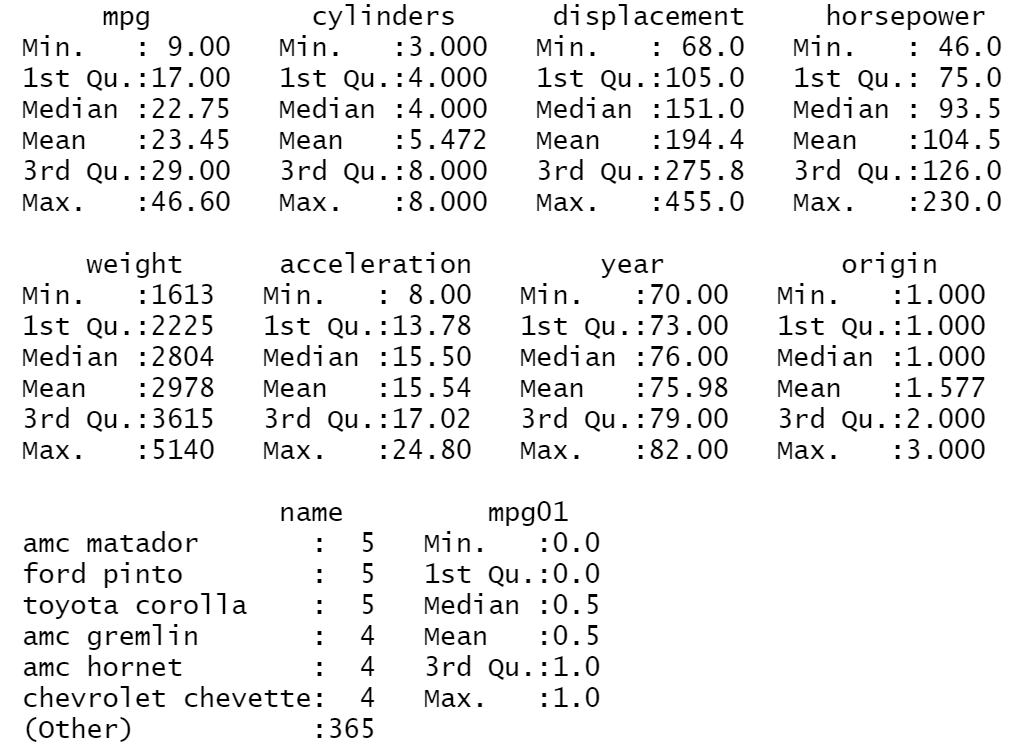
mpg01 = rep(0, length(mpg))

mpg01[mpg > median(mpg)] = 1

Auto = data.frame(Auto, mpg01)

summary(Auto)

截图：



（b）问题（略）

脚本：

cor(Auto[, -9])

pairs(Auto)

par(mfrow=c(2,4))

plot(factor(Auto$mpg01), Auto$mpg)

plot(factor(Auto$mpg01), Auto$cylinders)

plot(factor(Auto$mpg01), Auto$displacement)

plot(factor(Auto$mpg01), Auto$horsepower)

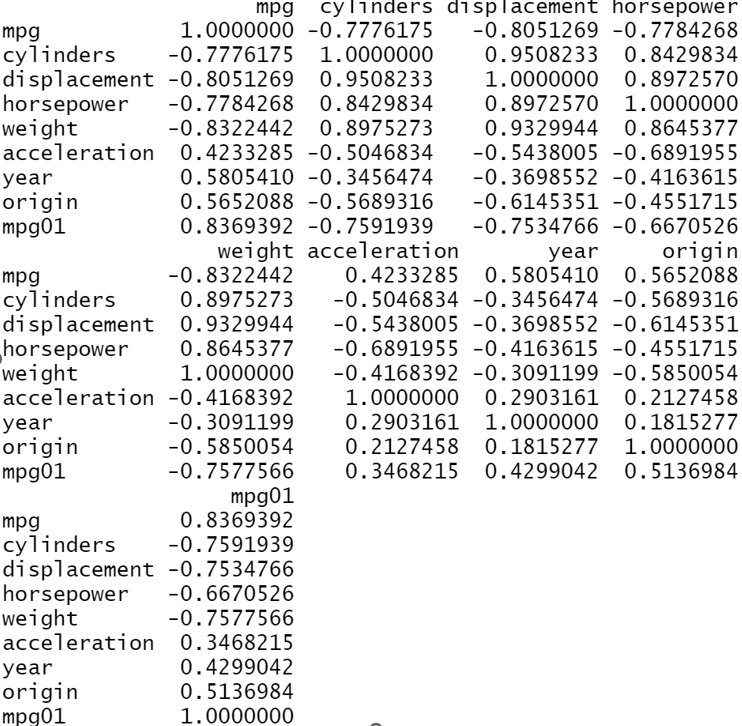
plot(factor(Auto$mpg01), Auto$weight)

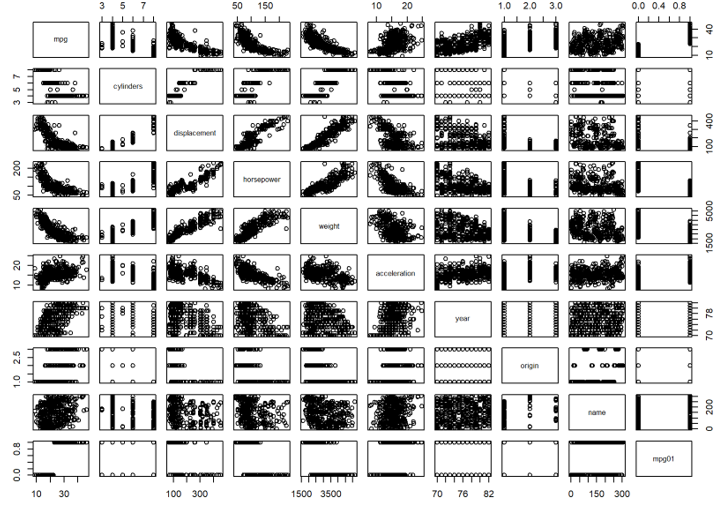
plot(factor(Auto$mpg01), Auto$acceleration)

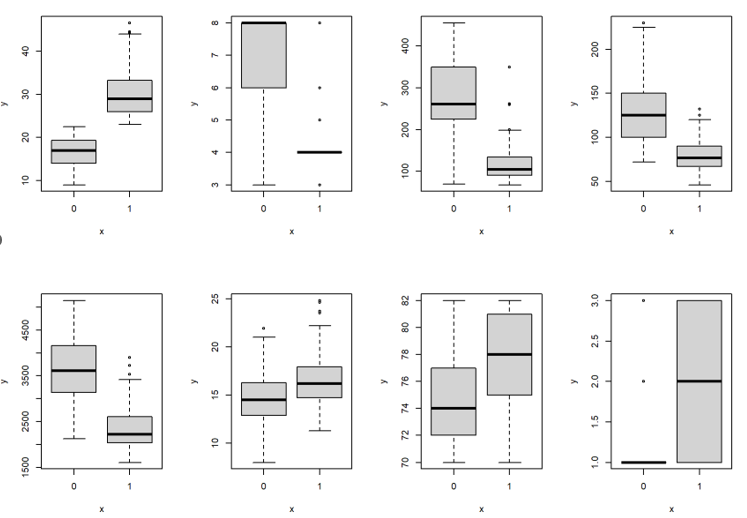
plot(factor(Auto$mpg01), Auto$year)

plot(factor(Auto$mpg01), Auto$origin)

截图：







由图可知，cylinders, weight, displacement, horsepower对mpg01有影响。

（c）问题（略）

脚本：

train = (year<80)

test = !train

Auto.train = Auto[train,]

Auto.test = Auto[test,]

mpg01.test = mpg01[test]

年份小于80的为训练集，年份大于等于80的为测试集

（d）问题（略）

脚本：

library(MASS)

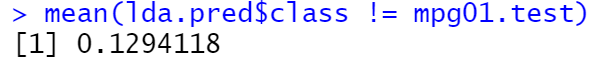
lda.fit = lda(mpg01~cylinders+weight+displacement+horsepower,

data=Auto, subset=train)

lda.pred = predict(lda.fit, Auto.test)

mean(lda.pred$class != mpg01.test)

截图：



测试误差为0.1294

（e）问题（略）

脚本：

qda.fit = qda(mpg01~cylinders+weight+displacement+horsepower,

data=Auto, subset=train)

qda.pred = predict(qda.fit, Auto.test)

mean(qda.pred$class != mpg01.test)

截图：



测试误差为0.1294

（f）问题（略）

脚本：

glm.fit = glm(mpg01~cylinders+weight+displacement+horsepower,

data=Auto,family=binomial,subset=train)

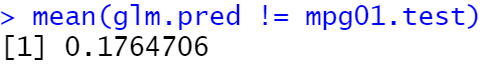
glm.probs = predict(glm.fit, Auto.test, type="response")

glm.pred = rep(0, length(glm.probs))

glm.pred[glm.probs > 0.5] = 1

mean(glm.pred != mpg01.test)

截图：



测试误差为0.1764

（g）问题（略）

脚本：

library(class)

train.X = cbind(cylinders, weight, displacement, horsepower)[train,]

test.X = cbind(cylinders, weight, displacement, horsepower)[test,]

train.mpg01 = mpg01[train]

set.seed(1)

knn.pred = knn(train.X, test.X, train.mpg01, k=1)

mean(knn.pred != mpg01.test)

knn.pred = knn(train.X, test.X, train.mpg01, k=10)

mean(knn.pred != mpg01.test)

knn.pred = knn(train.X, test.X, train.mpg01, k=100)

mean(knn.pred != mpg01.test)

截图：

K=1;



测试误差为0.2

K=10;



测试误差为0.2235

K=100;



测试误差为0.2353

K=1时效果最好。

12.问题（略）

（a）问题（略）

脚本：

Power = function() {

2^3

}

print(Power())

截图：



（b）问题（略）

脚本：

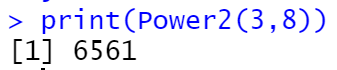
Power2 = function(x, a) {

x^a

}

Power2(3,8)

截图：



（c）问题（略）

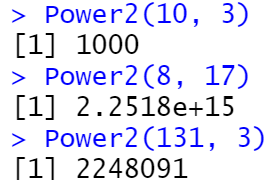
脚本：

Power2(10, 3)

Power2(8, 17)

Power2(131, 3)

截图：



（d）问题（略）

脚本：

Power3 = function(x, a) {

result = x^a

return(result)

}

（e）问题（略）

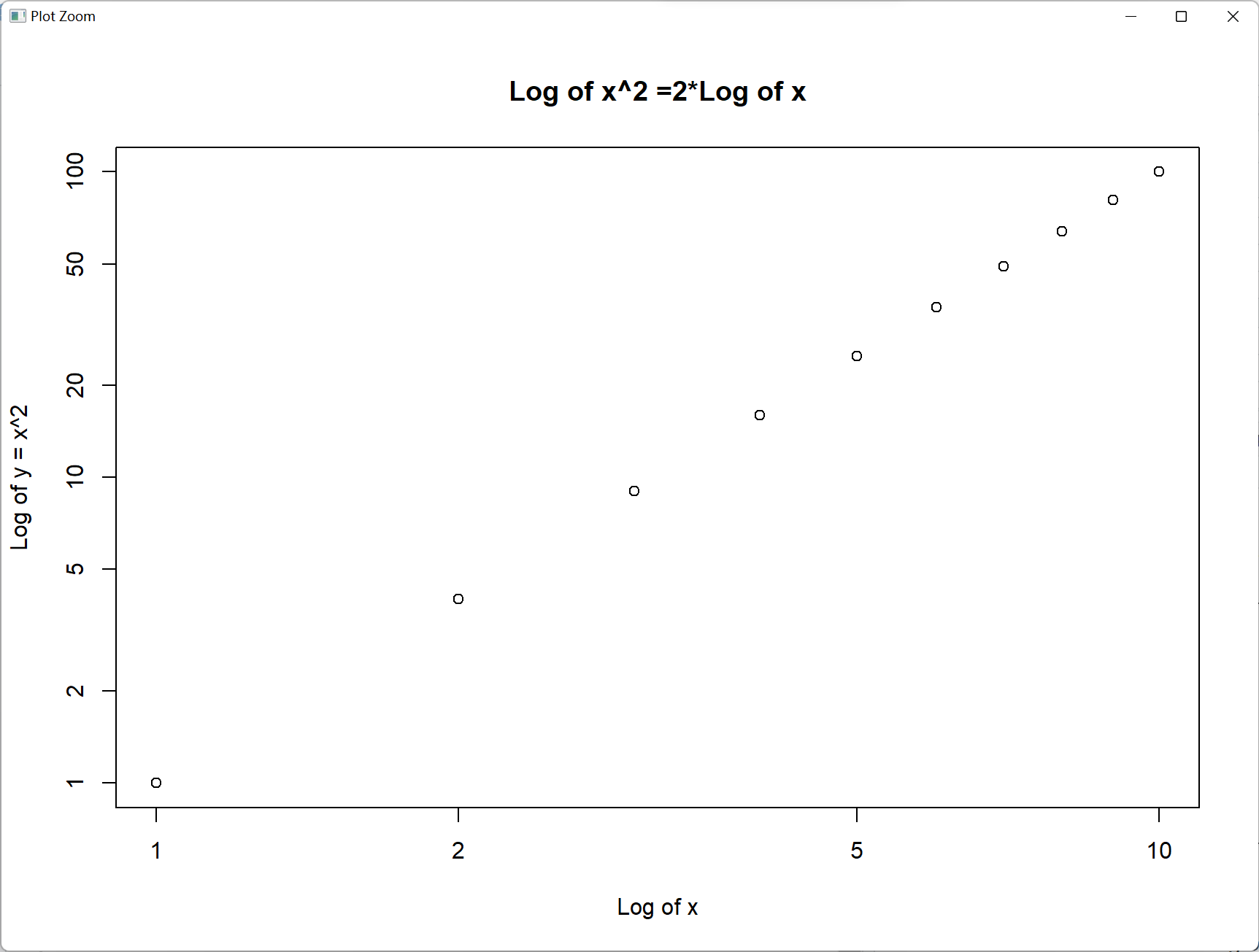
脚本：

x = 1:10

plot(x, Power3(x, 2), ylab="x^2", xlab="x",main="f(x)=x^2")

x = 1:10

plot(x, Power3(x, 2), log="xy", ylab="Log of y = x^2", xlab="Log of x",main="Log of x^2 =2\*Log of x")



（f）问题（略）

脚本：

PlotPower = function(x, a) {

plot(x, Power3(x, a))

}

PlotPower(1:10, 3)

截图：

