Statistical hw3

全金

2025-03-24

3

$$p_k(x) = \frac{\pi_k \sigma_k^{-1} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}}{\sum_{l=1}^K \pi_l \sigma_l^{-1} e^{-\frac{(x-\mu_l)^2}{2\sigma_l^2}}}$$

取对数去最大值得（忽略常数项）：

$$\log p_k(x) \propto \log \pi_k - \frac{1}{2}\log \sigma_k^2 - \frac{(x-\mu_k)^2}{2\sigma_k^2}$$

展开得：

$$f_k(x) = \log \pi_k - \frac{1}{2}\log \sigma_k^2 - \frac{x^2}{2\sigma_k^2} + \frac{x\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2}$$

因含 x^2 项，故得证。

5

(a) QDA 训练集更优，LDA 测试集更优

(b) QDA 在训练和测试集均可能更优

(c) 更好，数据量更多，QDA 能更好地拟合数据。

(d) 错误：QDA 可能过拟合

12

(a) 对数几率：$\hat{\beta}_0 + \hat{\beta}_1 x$

(b) 友人模型对数几率：$(\alpha_0-\alpha_0)+(\alpha_1-\alpha_1)x$

(c) $\alpha_0-\alpha_0=2$, $\alpha_1-\alpha_1=-1$

(d) $\hat{\beta}_0=-1.8$, $\hat{\beta}_1=-2.6$

(e) 模型等价，预测一致

13

(a)

```
library(ISLR2)
data(Weekly)
summary(Weekly)
```

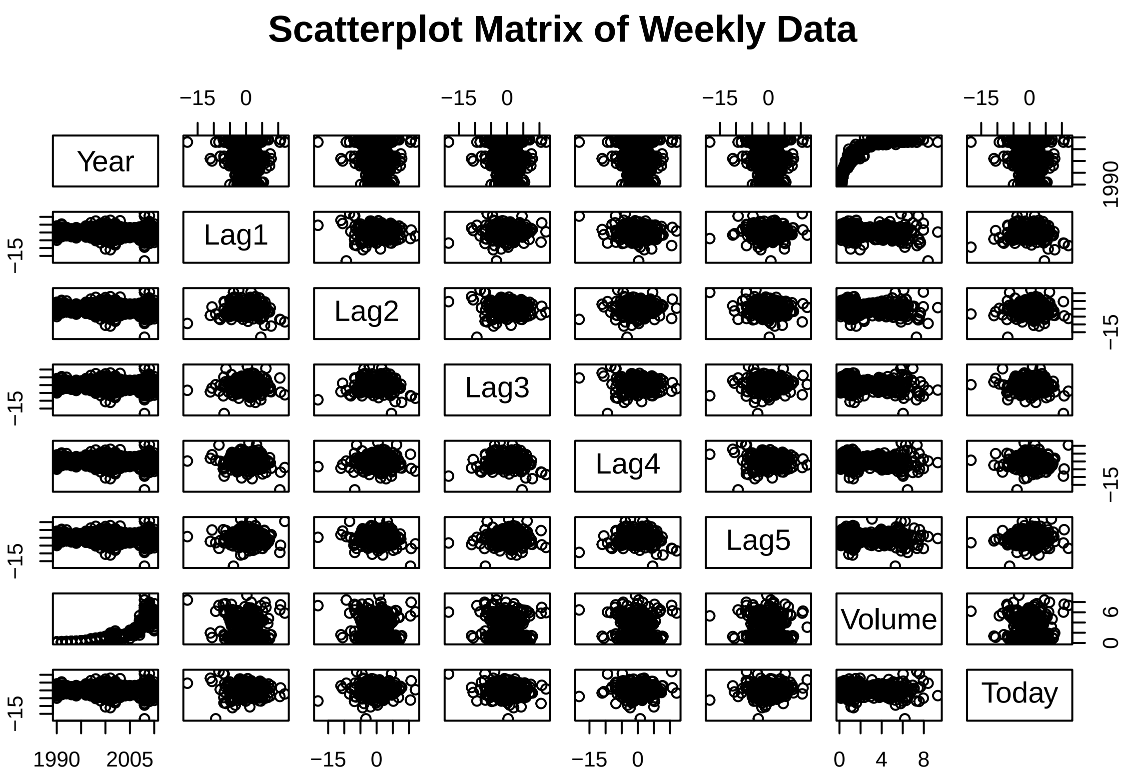
```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

```
str(Weekly)
```

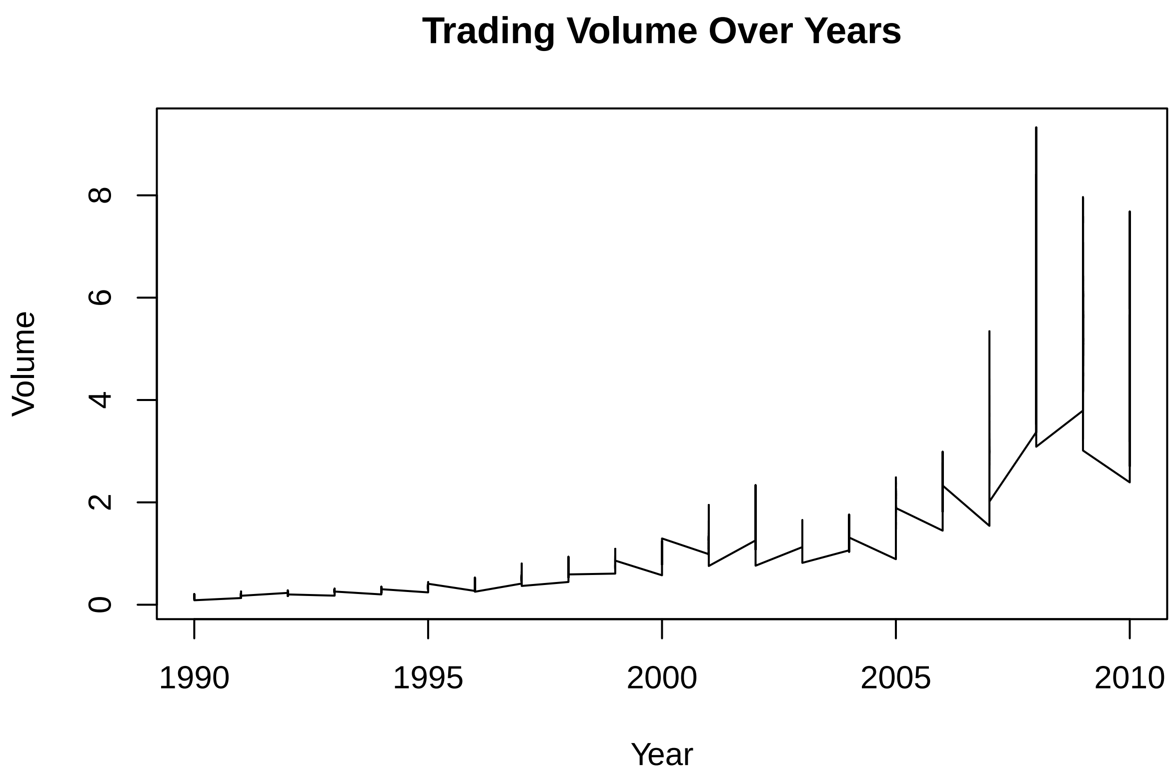
```
## 'data.frame':    1089 obs. of  9 variables:
##  $ Year     : num  1990 1990 1990 1990 1990 1990 1990 1990 1990 1990 ...
```

```
##  $ Lag1     : num  0.816 -0.27 -2.576 3.514 0.712 ...
##  $ Lag2     : num  1.572 0.816 -0.27 -2.576 3.514 ...
##  $ Lag3     : num  -3.936 1.572 0.816 -0.27 -2.576 ...
##  $ Lag4     : num  -0.229 -3.936 1.572 0.816 -0.27 ...
##  $ Lag5     : num  -3.484 -0.229 -3.936 1.572 0.816 ...
##  $ Volume   : num  0.155 0.149 0.16 0.162 0.154 ...
##  $ Today    : num  -0.27 -2.576 3.514 0.712 1.178 ...
##  $ Direction: Factor w/ 2 levels "Down","Up": 1 1 2 2 2 1 2 2 2 1 ...
```

```r
pairs(Weekly[, 1:8], main = "Scatterplot Matrix of Weekly Data")
```




```r
plot(Weekly$Year, Weekly$Volume, type = "l",
     xlab = "Year", ylab = "Volume",
     main = "Trading Volume Over Years")
```



图中可看出，成交量与年份呈强正相关。

(b)

```r
glm_full <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                data = Weekly,
                family = binomial)
summary(glm_full)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
```

```
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

图中可看出，Lag2 为显著变量.

(c)

```r
glm_probs <- predict(glm_full, type = "response")
glm_pred <- ifelse(glm_probs > 0.5, "Up", "Down")
conf_mat <- table(glm_pred, Weekly$Direction)
accuracy <- mean(glm_pred == Weekly$Direction)

conf_mat
```

```
##
## glm_pred Down  Up
##     Down   54  48
##     Up    430 557
```

```r
accuracy
```

```
## [1] 0.5610652
```

该模型在市场实际下跌时有很大的错误预测率。总体准确率优于随机猜测。

(d)

```r
train <- Weekly$Year <= 2008
test_data <- Weekly[!train, ]
glm_lag2 <- glm(Direction ~ Lag2,
                data = Weekly,
                family = binomial,
                subset = train)
glm_probs_test <- predict(glm_lag2, test_data, type = "response")
```

```r
glm_pred_test <- ifelse(glm_probs_test > 0.5, "Up", "Down")
conf_mat_d <- table(glm_pred_test, test_data$Direction)
accuracy_d <- mean(glm_pred_test == test_data$Direction)

conf_mat_d
```

```
##
## glm_pred_test Down Up
##          Down    9  5
##          Up     34 56
```

```r
accuracy_d
```

```
## [1] 0.625
```

(e)

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```r
lda_fit <- lda(Direction ~ Lag2, data = Weekly, subset = train)
lda_pred <- predict(lda_fit, test_data)$class
conf_mat_e <- table(lda_pred, test_data$Direction)
accuracy_e <- mean(lda_pred == test_data$Direction)

conf_mat_e
```

```
##
## lda_pred Down Up
##     Down    9  5
##     Up     34 56
```

```r
accuracy_e
```

```
## [1] 0.625
```

(f)

```
qda_fit <- qda(Direction ~ Lag2, data = Weekly, subset = train)
qda_pred <- predict(qda_fit, test_data)$class
conf_mat_f <- table(qda_pred, test_data$Direction)
accuracy_f <- mean(qda_pred == test_data$Direction)
conf_mat_f
```

```
##
## qda_pred Down Up
##     Down    0  0
##     Up     43 61
```

```
accuracy_f
```

```
## [1] 0.5865385
```

(g)

```
library(class)
train_X <- as.matrix(Weekly$Lag2[train])
test_X <- as.matrix(Weekly$Lag2[!train])
train_dir <- Weekly$Direction[train]
set.seed(123)
knn_pred <- knn(train_X, test_X, train_dir, k = 1)
conf_mat_g <- table(knn_pred, test_data$Direction)
accuracy_g <- mean(knn_pred == test_data$Direction)
conf_mat_g
```

```
##
## knn_pred Down Up
##     Down   21 29
##     Up     22 32
```

```
accuracy_g
```

```
## [1] 0.5096154
```

(h)

```
library(e1071)
nb_fit <- naiveBayes(Direction ~ Lag2, data = Weekly, subset = train)
nb_pred <- predict(nb_fit, test_data)
```

```r
conf_mat_h <- table(nb_pred, test_data$Direction)
accuracy_h <- mean(nb_pred == test_data$Direction)
conf_mat_h
```

```
##
## nb_pred Down Up
##    Down    0  0
##    Up     43 61
```

```r
accuracy_h
```

```
## [1] 0.5865385
```

(i)

逻辑回归和 LDA 最好

(j)

```r
fit <- glm(Direction ~ Lag1, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5673077
```

```r
fit <- glm(Direction ~ Lag3, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag1 * Lag2 * Lag3 * Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5961538
```

```r
fit <- lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5769231
```

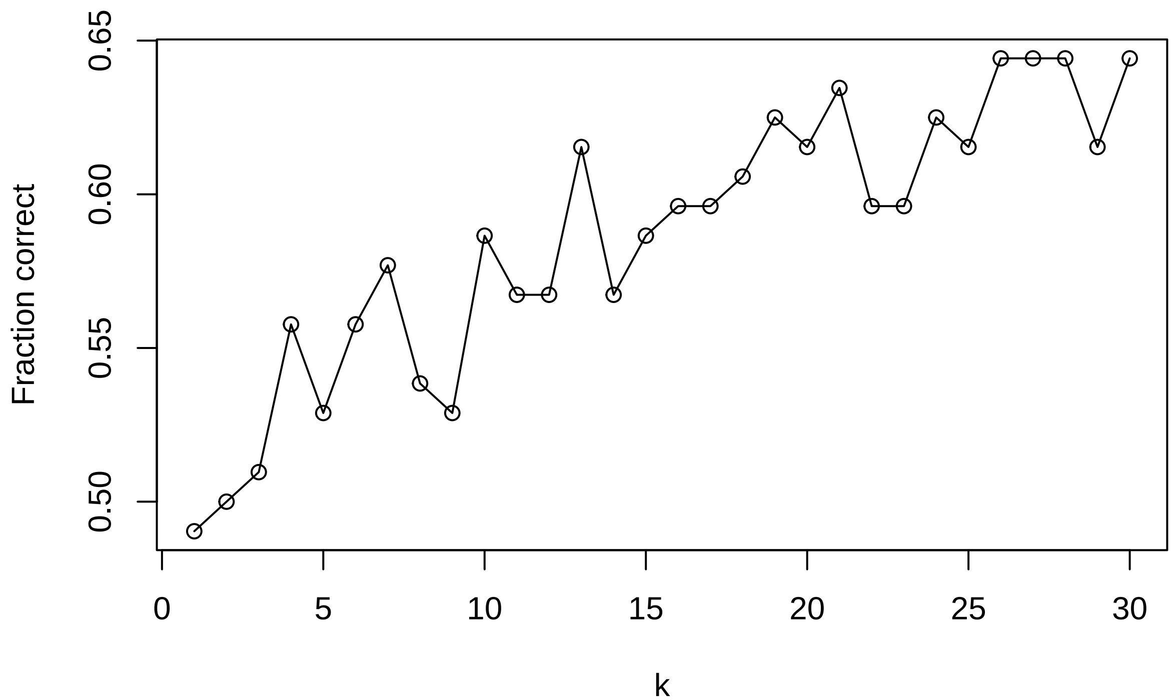
```r
fit <- qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5192308
```

```r
fit <- naiveBayes(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "class")
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5096154
```

```r
set.seed(1)
res <- sapply(1:30, function(k) {
  fit <- knn(
    Weekly[train, 2:4, drop = FALSE],
    Weekly[!train, 2:4, drop = FALSE],
    Weekly$Direction[train],
    k = k
  )
  mean(fit == Weekly[!train, ]$Direction)
})
plot(1:30, res, type = "o", xlab = "k", ylab = "Fraction correct")
```




```
(k <- which.max(res))
```

```
## [1] 26
```

```
fit <- knn(
  Weekly[train, 2:4, drop = FALSE],
  Weekly[!train, 2:4, drop = FALSE],
  Weekly$Direction[train],
  k = k
)
table(fit, Weekly[!train, ]$Direction)
```

```
##
## fit    Down Up
##   Down   23 18
##   Up     20 43
```

```
mean(fit == Weekly[!train, ]$Direction)
```

```
## [1] 0.6346154
```

最佳结果：使用前三个滞后变量的 KNN，k=26

15

(a)

```
Power <- function() {
  print(2^3)
}
Power()
```

```
## [1] 8
```

(b)

```
Power2 <- function(x, a) {
  print(x^a)
}
```

(c)

```
Power2(10, 3)
```

```
## [1] 1000
```

```
Power2(8, 17)
```

```
## [1] 2.2518e+15
```

```
Power2(131, 3)
```

```
## [1] 2248091
```

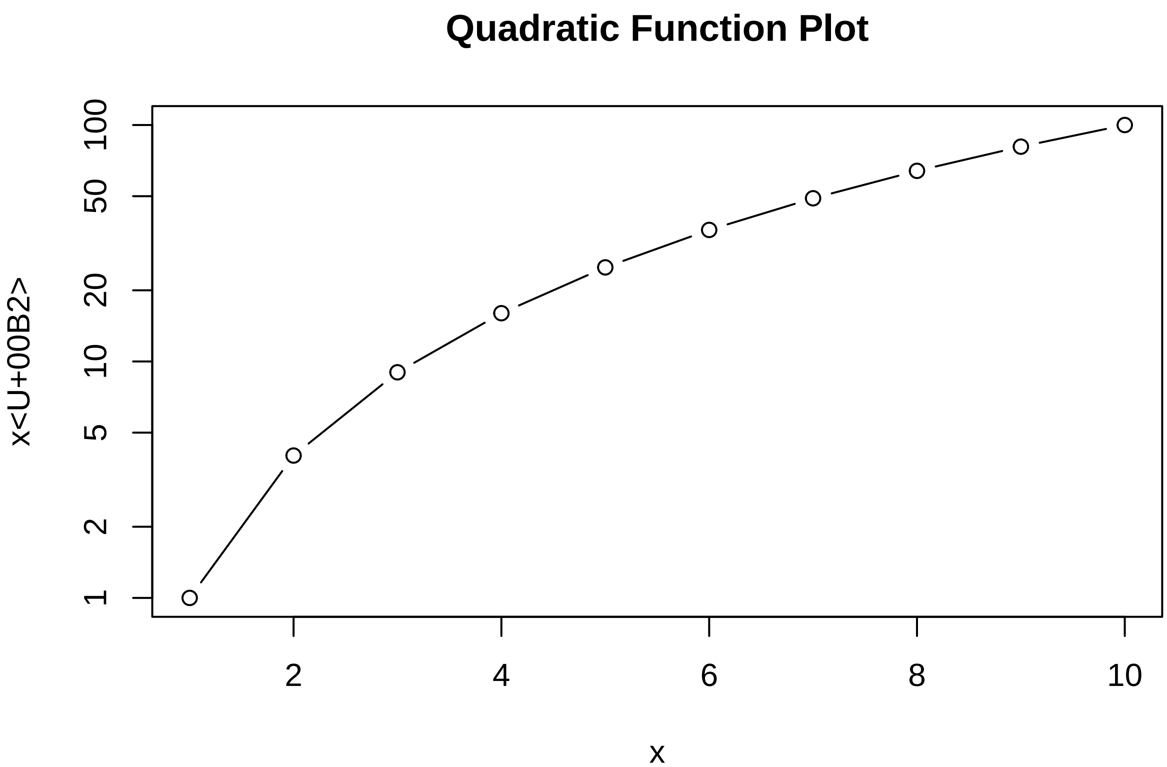
(d)

```
Power3 <- function(x, a) {
  return(x^a)
}
```

(e)

```
x <- 1:10
y <- Power3(x, 2)
plot(x, y, type = "b",
     xlab = "x", ylab = "x²",
```

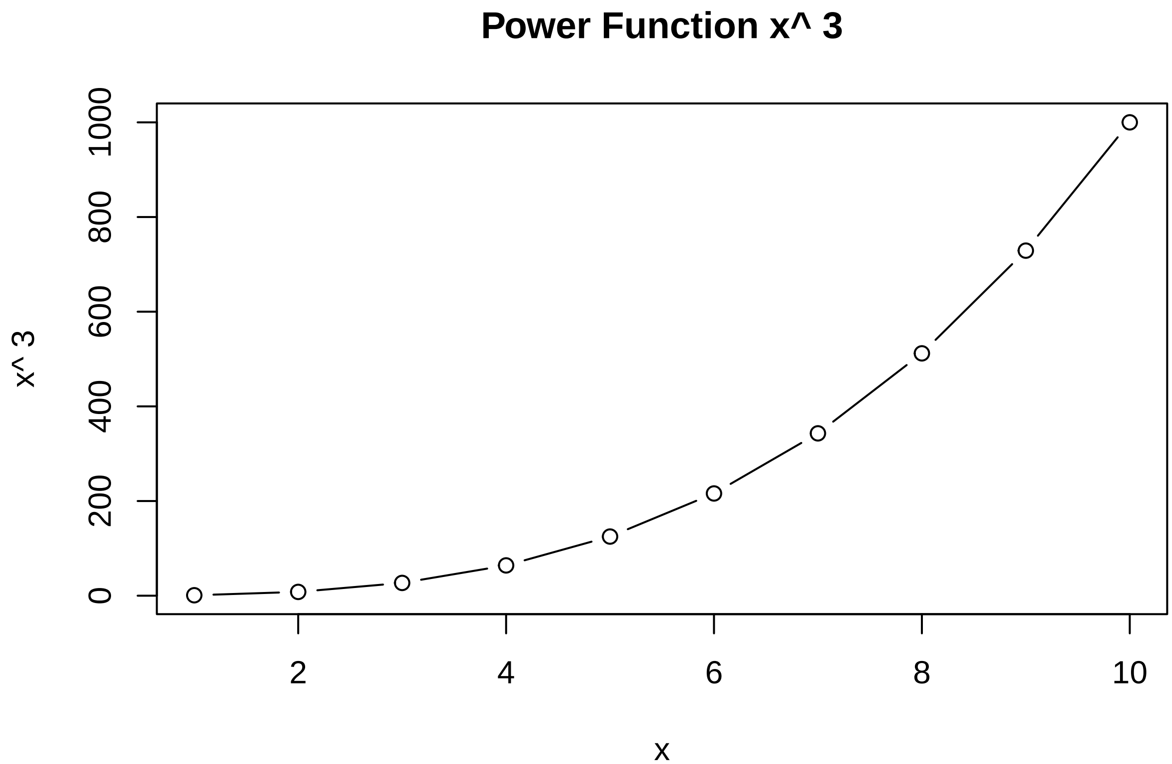
```
    main = "Quadratic Function Plot",
    log = "y")
```



##

(f)

```
PlotPower <- function(x_values, a) {
  y_values <- Power3(x_values, a)
  plot(x_values, y_values,
       xlab = "x", ylab = paste("x^", a),
       main = paste("Power Function x^", a),
       type = "b")
}
PlotPower(1:10, 3)
```


Power Function x^ 3
x^ 3
0
200
400
600
800
1000
2
4
6
8
10
x