Chapter 4 HW

Nhi Vu

March 21,2021

Conceptual Questions

3. This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class- specific mean vector and a class specific covariance matrix. We consider the simple case where $p = 1$; i.e. there is only one feature.

Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal dis- tribution, $X \sim N(\mu_k, \sigma_k^2)$. Recall that the density function for the one-dimensional normal distribution is given in (4.11).$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k}} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.

***Hint:** For this problem, you should follow the arguments laid out in Section 4.4.2, but without making the assumption that $\sigma_1^2 = \ldots\ldots = \sigma_k^2$.*

Bayes theorem:

$$P(Y = k|X = x) = \frac{P(X = x|Y = k)P(Y = k)}{\sum_{i=1}^{k} P\,(X = x|Y = i)P(Y = i)}$$

Let $\pi_k = P(Y = k)$, then:

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^{k} \pi_i\, P(X = x|Y = i)}$$

Applying Bayer Classifier: We find the class k which maximize $P(Y = k|X = x)$. However, maximizing $P(Y = k|X = x)$ is equivalent to maximizing $log(P(Y = k|X = x))$, so we take the natural logarithm of both sides.

$$log(P(Y = k|X = x)) = log(f_k(x)) + log(\pi_k) - log(\sum_{i=1}^{k} \pi_i\, P(X = x|Y = i))$$

Because $X \sim N(\mu_k, \sigma_k^2)$, then

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k}} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

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

Then, we replace $log(f_k(x))$ into $log(P(Y = k|X = x))$, we will get:

$$log(P(Y = k|X = x)) = -\frac{1}{2\sigma_k^2}(x - \mu_k)^2 - \frac{1}{2}log(2\pi\sigma_k) + log(\pi_k) - log(\sum_{i=1}^{k} \pi_i P(X = x|Y = i))$$

This expression is obviously not linear in x and it contains x^2 which is is quadratic in x.

6. Suppose we collect data for a group of students in a statistics class with variables X_1 = hours studied, X_2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, $\beta_0 = -6, \beta_1 = 0.05, \beta_2 = 1$.

From given information, we get a logistic function:

$$P(\text{Recieved an A}|X) = \frac{e^{-6+0.05X_1+X_2}}{1 + e^{-6+0.05X_1+X_2}}$$

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

With $X_1 = 40, X_2 = 3.5$,

$$P(\text{Recieved an A}|X) = \frac{e^{-6+0.05(40)+3.5}}{1 + e^{-6+0.05(40)+3.5}} = 0.3775$$

The probability that this student gets an A in the class is 37.75 %

(b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

The equation for predicted probability tells us that

$$\frac{e^{-6+0.05X_1+3.5}}{1 + e^{-6+0.05X_1+3.5}} = 0.5$$

$$e^{-6+0.05X_1+3.5} = 1$$

$$X_1 = \frac{2.5}{50} = 50$$

With an undergrad GPA of 3.5, to get a 50 % chance of getting an A in the class, this student needs spend 50 hours per week. ## Applied question:

10.This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

```
library(ISLR)     # for data
data(Weekly)
```

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
summary(Weekly)
```

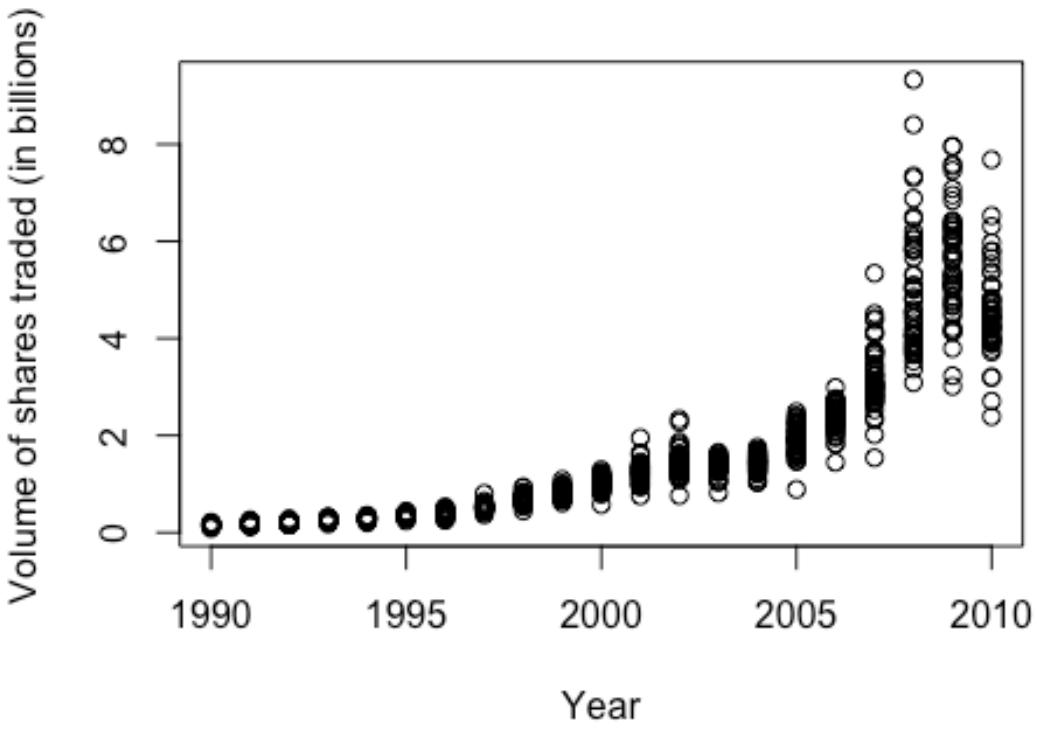
```
##       Year           Lag1                Lag2                Lag3          
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  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
##
##
##
##
```

```
cor(Weekly [,-9])
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

Comment: correlations are all basically zero, the only substantial correlation is between Year and Volume (r=0.84194162),which shows an increase in volume over time.

```
plot(Weekly$Year,Weekly$Volume, xlab="Year", ylab="Volume of shares traded
(in billions)")
```



(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so,which ones?

```
logistc.F = glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,family =
"binomial", data=Weekly)
summary(logistc.F)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = "binomial", data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## 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
```

Lag2 with the smallest p-value is the only statistically significant variable to predict the Direction, while p-value of other predictors are quite large. The positive coefficient for this predictor suggests that if the market had a positive return yesterday,then it is more likely to go up today. Also, the estimate coefficient is of 0.058, which an increase of 1 in Lag2 represents an increase of $e^{0.058} = 1.06$ in the odds of direction going up.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
probs = predict(logistc.F, type="response")
preds = rep("Down", 1089)
preds[probs > 0.5] = "Up"
table(preds, Weekly$Direction)
```

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

The overall error rate:

$$\frac{430 + 48}{1089} = 0.4389348$$

The training error rate of model is 43.89% which logistic regression correctly predicted the movement of the market 56.11% % of the time.However, the training error tends to underestimate the test error rate, because while the model correctly predicted the Up weekly trends 55748+557=0.9207; 92.07% correct. In contrast Down weekly trends were predicted at a lower rate, 54430+54=0.1115; or only 11.15% correctly predicted.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
train <- (Weekly$Year < 2009)
Weekly_train <- Weekly[train,]
```

```
Weekly_test <- Weekly[!train,]
Direction_train <- Weekly_train$Direction
Direction_test <- Weekly_test$Direction

logistc.fit <- glm(Direction ~ Lag2,
                   data = Weekly_train,
                   family = binomial)
logistic_probs <- predict(logistc.fit, Weekly_test, type = "response")
logistic_pred = rep("Down", length(Direction_test))
logistic_pred[logistic_probs > 0.5] <- "Up"
table(logistic_pred, Direction_test)

##              Direction_test
## logistic_pred Down Up
##          Down    9  5
##          Up     34 56

mean(logistic_pred == Direction_test)

## [1] 0.625
```

When spliting up the whole Weekly dataset into a training and test dataset, the model correctly predicted weekly trends at rate of 62.5%, which is a moderate improvement from the model that utilized the whole dataset. Also this model such as the previous one did better at predicting upward trends (91.80%) compared to downward trends (20.93%); although this model was able to improve significantly on correctly predicting downward trends.

(e) Repeat (d) using LDA.

```
library(MASS)     # for LDA
lda.fit=lda(Direction~Lag2,data = Weekly, subset = train)
lda.pred<-predict(lda.fit, Weekly_test)
table(lda.pred$class, Direction_test)

##       Direction_test
##        Down Up
##   Down    9  5
##   Up     34 56

mean(lda.pred$class==Direction_test)

## [1] 0.625
```

The error rate stays the same as it was with the Logistic Regression, 37,5% and the correct rate is still 62.25%.

(f) Repeat (d) using QDA.

```
qda.fit = qda(Direction~Lag2, data= Weekly_train)
qda.pred=predict(qda.fit,Weekly_test)
table(qda.pred$class,Direction_test)
```

```
##       Direction_test
##        Down Up
##   Down    0  0
##   Up     43 61
```

```
mean(qda.pred$class==Direction_test)
```

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

Quadratic Linear Analysis created a model with an accuracy of 58.65%, which is lower than the previous methods.

(g) Repeat (d) using KNN with K = 1.

```
library(class)    # for KNN
train_X <- as.matrix(Weekly$Lag2[train])
test_X <- as.matrix(Weekly$Lag2[!train])

set.seed(1)
knn_pred <- knn(train_X, test_X, Direction_train, k = 1)
table(knn_pred,Direction_test)
```

```
##         Direction_test
## knn_pred Down Up
##     Down   21 30
##     Up     22 31
```

```
mean(knn_pred==Direction_test)
```

```
## [1] 0.5
```

The error rate seems awful, 50%. However, it is better than the other models to identify True Negatives, since it identifies roughly a 49% of them properly.

(h) Which of these methods appears to provide the best results on this data?

The methods that have the highest accuracy rates are the Logistic Regression and Linear Discriminant Analysis; both having rates of 62.5%.

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

```
# Different Logistic Models with Lag2 and Lag1.
logistc2.fit=glm(Direction ~  Lag2 + Lag1, data = Weekly_train, family =
binomial)
logistic2_probs <- predict(logistc2.fit, Weekly_test, type = "response")
logistic2_pred = rep("Down", length(Direction_test))
logistic2_pred[logistic2_probs > 0.5] <- "Up"
table(logistic2_pred, Direction_test)
```

```
##                Direction_test
## logistic2_pred Down Up
##           Down    7  8
##           Up     36 53

mean(logistic2_pred == Direction_test)

## [1] 0.5769231
```

The accuracy rate is 57.69 which is lower than the Logistic Models with only Lag2.

```
#Different LDA method with Lag2 and Lag1.
lda2.fit=lda(Direction~Lag2+Lag1, data=Weekly_train)
lda2.pre=predict(lda2.fit,Weekly_test)
table(lda2.pre$class,Direction_test)

##       Direction_test
##        Down Up
##   Down    7  8
##   Up     36 53

mean(lda2.pre$class==Direction_test)

## [1] 0.5769231
```

Similarly to the Logistic Models, the accuracy rate is 57.69 which is lower than the LDA method with only Lag2.

```
#Different QDA method with Lag2 and Lag1.
qda2.fit=qda(Direction~Lag2+Lag1, data=Weekly_train)
qda.pre=predict(qda2.fit,Weekly_test)
table(qda.pre$class,Direction_test)

##       Direction_test
##        Down Up
##   Down    7 10
##   Up     36 51

mean(qda.pre$class==Direction_test)

## [1] 0.5576923
```

The accuracy rate is 55.77% which is lower than the QDA method with only Lag2 and absolutely is lower than previous method with predictors Lag2 and Lag1.

```
#Different KNN with K = 10.
train_X <- as.matrix(Weekly$Lag2[train])
test_X <- as.matrix(Weekly$Lag2[!train])

set.seed(1)
knn_pred <- knn(train_X, test_X, Direction_train, k = 10)
table(knn_pred,Direction_test)
```

```
##          Direction_test
## knn_pred Down Up
##     Down   17 21
##     Up     26 40

mean(knn_pred==Direction_test)

## [1] 0.5480769
```

With bigger k, the accuracy rate is higher.

12. This problem involves writing functions.

(a) Write a function, Power(), that prints out the result of raising 2 to the 3rd power. In other words, your function should compute 2^3 and print out the results.

Hint: Recall that x^a raises x to the power a. Use the print() function to output the result.

```
Power=function(){
  print(2^3);
}
Power()

## [1] 8
```

(b) Create a new function, Power2(), that allows you to pass any two numbers, x and a, and prints out the value of x^a. You can do this by beginning your function with the line > Power2=function (x,a){ You should be able to call your function by entering, for instance,> Power2 (3,8) on the command line. This should output the value of 3^8, namely, 6,561.

```
Power2=function(x,a){
  print(x^a)
}
Test=Power2(3,8)

## [1] 6561
```

(c) Using the Power2() function that you just wrote, compute 10^3,8^{17}, and 131^3.

```
Power2(10, 3)

## [1] 1000

Power2(8, 17)

## [1] 2.2518e+15

Power2(131, 3)

## [1] 2248091
```

(d) Now create a new function, Power3(), that actually returns the result x^a as an R object, rather than simply printing it to the screen. That is, if you store the value x^a in an object called result within your function, then you can simply return() this return() result, using the following line: return(result)

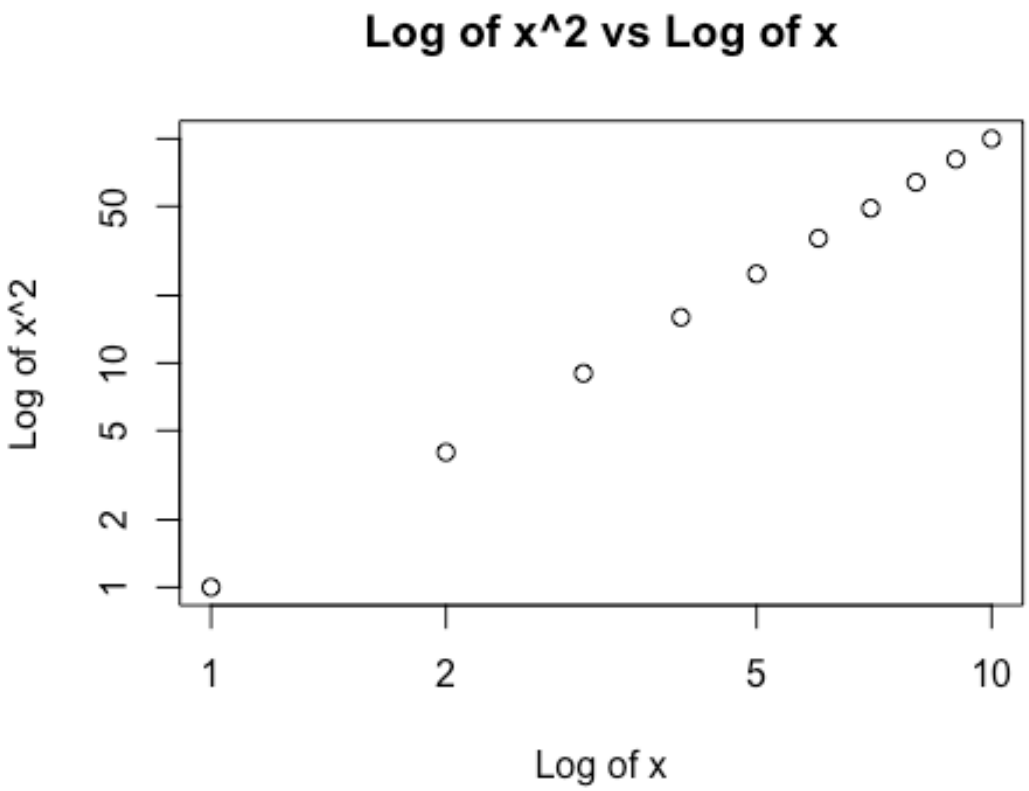
The line above should be the last line in your function, before the } symbol.

```
Power3=function(x,a){
  result=x^a
  return(result)
}
test=Power3(3,4)
test
```

```
## [1] 81
```

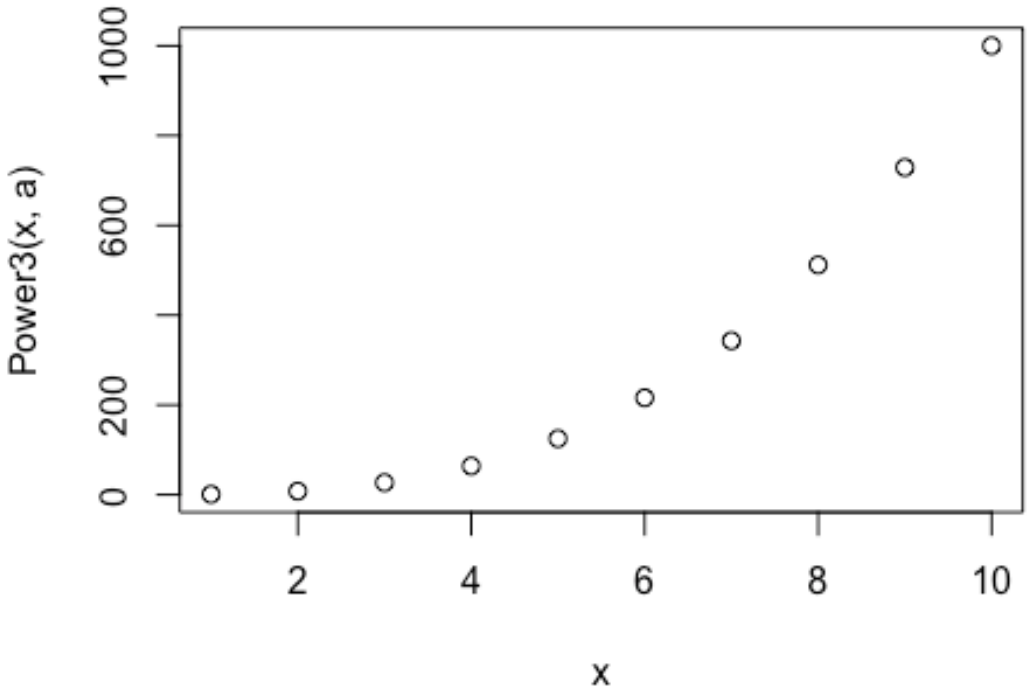
(e) Now using the Power3() function, create a plot of $f(x) = x^2$. The x-axis should display a range of integers from 1 to 10, and the y-axis should display x2. Label the axes appropriately, and use an appropriate title for the figure. Consider displaying either the x-axis, the y-axis, or both on the log-scale. You can do this by using log="x", log="y", or log="xy" as arguments to the plot() function.

```
x<-seq(1,10,by=1)
y<-Power3(x,2)
plot(x, y, log = "xy", xlab = "Log of x", ylab = "Log of x^2", main = "Log of
x^2 vs Log of x")
```



(f) Create a function, PlotPower(), that allows you to create a plot of x against x^a for a fixed a and for a range of values of x. For instance, if you call > PlotPower (1:10,3) then a plot should be created with an x-axis taking on values 1, 2,…, 10, and a y-axis taking on values $1^3, 2^3, \ldots, 10^3$

```
PlotPower=function(x,a){
  plot(x,Power3(x,a))
}
PlotPower(1:10,3)
```


Power3(x, a)
0
200
600
1000
2
4
6
8
10
x