Problem 4

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.

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

For each date, we have recorded the percentage returns for each of the five previous trading days, Lag1 through Lag5.

R-code:

library(ISLR)
names(Weekly)
summary(Weekly)

Result:

```
> names(Weekly)
[1] "Year"       "Lag1"       "Lag2"       "Lag3"
[5] "Lag4"       "Lag5"       "Volume"     "Today"
[9] "Direction"


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

>
```

R-code:

cor(Weekly[, -9])

Explanation:

The cor() function produces a matrix that contains all of the pairwise correlations among the predictors in a data set.

Result:

```
> cor(Weekly[, -9])
              Year         Lag1        Lag2        Lag3
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364
               Lag4         Lag5      Volume        Today
Year   -0.031127923 -0.030519101  0.84194162 -0.032459894
Lag1   -0.071273876 -0.008183096 -0.06495131 -0.075031842
Lag2    0.058381535 -0.072499482 -0.08551314  0.059166717
Lag3   -0.075395865  0.060657175 -0.06928771 -0.071243639
Lag4    1.000000000 -0.075675027 -0.06107462 -0.007825873
Lag5   -0.075675027  1.000000000 -0.05851741  0.011012698
Volume -0.061074617 -0.058517414  1.00000000 -0.033077783
Today  -0.007825873  0.011012698 -0.03307778  1.000000000

>
```

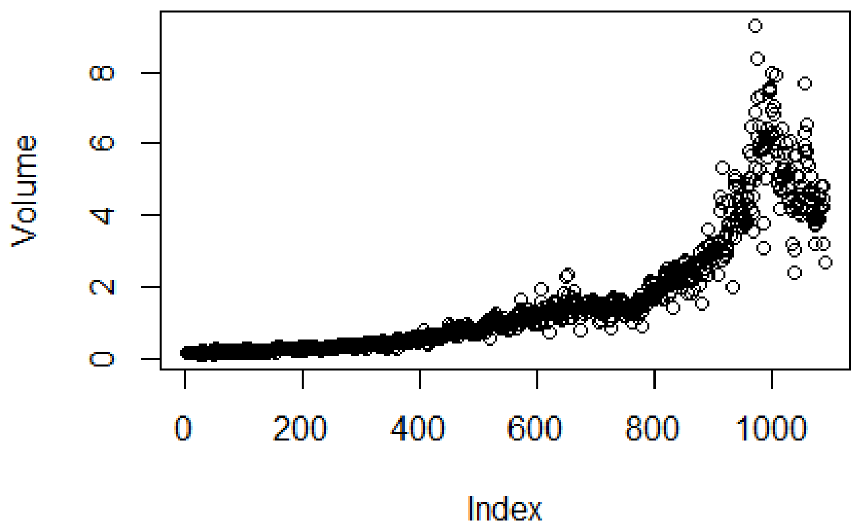
Explanation:

As in the previous example in the book (p.154, data about stock market), the correlations between the lag variables and today's returns are close to zero. So, there appears to be littlecorrelation between today's returns and previous days' returns. The only substantial correlation is between Year and Volume.

R-code:

attach(Weekly)
plot(Volume)

Result:



Explanation:

When we plot "Volume", we see that it is increasing over time.

(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?

R-code:

We fit a logistic regression model in order to predict Direction using Lag1 through Lag5 and Volume.
The glm() function fits *generalized linear models*,
The argument family=binomial run a logistic regression

fit.glm <- **glm**(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, **family = binomial)**
summary(fit.glm)

Result:

```
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
```

Explanation:

It would seem that **"Lag2"** is the only predictor statistically significant as its p-value is less than 0.05.

(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.

R-code:

probs <- predict(fit.glm, type = "response")
pred.glm <- rep("Down", length(probs))
pred.glm[probs > 0.5] <- "Up"
table(pred.glm, Direction)

The predict() function can be used to predict the probability that the market will go up, given values of the predictors. The type="response" option tells R to output probabilities of the form $P(Y = 1|X)$,
pred.glm <- rep("Down", length(probs)) creates a vector with length = length(probs).
pred.glm[probs > 0.5] <- "Up" transforms to Up all elements for which the predicted probability exceeds 0.5.
table(pred.glm, Direction) produce a confusion matrix to determine how many observations were correctly or incorrectly classified.

Result:
```
Direction
pred.glm Down  Up
    Down   54  48
    Up    430 557

>
```

Explanation:
We may conclude that **the percentage of correct predictions** on the training data is **(54+557)/1089** wich is equal to **56**.1065197**%**. In other words **43.8934803% is the training error rate**, which is often overly optimistic. We could also say that for weeks when the market goes **up, the model is right 92.0661157%** of the time (557/(48+557)). For weeks when the market goes **down, the model is right only 11.1570248%** of the time (54/(54+430)).

(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).

R-code:
```
//use data from the  years <2009 as training data
train <- (Year < 2009)
Weekly.20092010 <- Weekly[!train, ]
Direction.20092010 <- Direction[!train]
// !train is a vector similar to train, except that the elements that are TRUE
//in train get swapped to FALSE in !train, and the elements that are FALSE
//in train get swapped to TRUE in !train.
fit.glm2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
summary(fit.glm2)
```

Result:
```
Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
    subset = train)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.536  -1.264   1.021   1.091   1.368

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20326    0.06428   3.162  0.00157 **
Lag2         0.05810    0.02870   2.024  0.04298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.5  on 983  degrees of freedom
AIC: 1354.5

Number of Fisher Scoring iterations: 4
```

R-code:
```
probs2 <- predict(fit.glm2, Weekly.20092010, type = "response")
//rep(x) – replicates the valies of x.
pred.glm2 <- rep("Down", length(probs2))
pred.glm2[probs2 > 0.5] <- "Up"
table(pred.glm2, Direction.20092010)
```

Result:

```
pred.glm2 Down Up
     Down    9  5
     Up     34 56
```

Explanation:
In this case, we may conclude **that the percentage of correct predictions on the test data is *(9+56)/104* wich is equal to 62.5%.** In other words **37.5% is the test error rate**. We could also say that **for weeks when the market goes up, the model is right 91.8032787% of the time (*56/(56+5)*).** For weeks when the market goes **down**, the model is right only **20.9302326%** of the time (*9/(9+34)*).

(e) Repeat (d) using LDA (linear discriminant analysis).

R-code:
library(MASS)
fit.lda <- lda(Direction ~ Lag2, data = Weekly, subset = train)
fit.lda

Result:

```
Call:
lda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2
Down -0.03568254
Up    0.26036581

Coefficients of linear discriminants:
           LD1
Lag2 0.4414162
```

R-code:
pred.lda <- predict(fit.lda, Weekly.20092010)
table(pred.lda$class, Direction.20092010)

Result:

```
Direction.20092010
       Down Up
  Down    9  5
  Up     34 56

>
```

Explanation:
In this case, we may conclude that **the percentage of correct predictions on the test data is 62.5%.** In other words 37.5% is the test error rate. We could also say that for weeks when the market goes up, the model is right 91.8032787% of the time. For weeks when the market goes down, the model is right only 20.9302326% of the time. **These results are very close to those obtained with the logistic regression model which is not surpising.**

(f) Repeat (d) using QDA (Quadratic discriminant analysis).

R-code:
fit.qda <- qda(Direction ~ Lag2, data = Weekly, subset = train)
fit.qda

Result:
```
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2
Down -0.03568254
Up    0.26036581
```

R-code:
pred.qda <- predict(fit.qda, Weekly.20092010)
table(pred.qda$class, Direction.20092010)

Result:
```
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up 
0.4477157 0.5522843 

Group means:
            Lag2
Down -0.03568254
Up    0.26036581
```

Explanation:
In this case, we may conclude that the percentage **of correct predictions on the test data is 58.6538462%.** In other words **41.3461538% is the test error rate**. We could also say that for weeks when the market goes **up**, the model is right **100%** of the time. For weeks when the market goes down, the model is right only 0% of the time. **We may note, that QDA achieves a correctness of 58.6538462% even though the model chooses "Up" the whole time.**

(g) Repeat (d) using KNN (k-nearest neighbour classification) with *K* = 1.

R-code:
library(class)
train.X <- as.matrix(Lag2[train])
test.X <- as.matrix(Lag2[!train])
train.Direction <- Direction[train]
set.seed(1)
pred.knn <- knn(train.X, test.X, train.Direction, k = 1)
table(pred.knn, Direction.20092010)

Result:
```
        Direction.20092010
pred.knn Down Up
    Down   21 30
    Up     22 31
```

Explanation:
In this case, we may conclude that the percentage of **correct predictions on the test data is 50%.** In other words 50% is the test error rate. We could also say that for weeks when the market goes up, the model is right 50.8196721% of the time. For weeks when the market goes down, the model is right only 48.8372093% of the time.

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

If we compare the test error rates, we see **that logistic regression and LDA have the minimum error rates**, followed by QDA and KNN.

(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.

R-code:

```
# Logistic regression with Lag2:Lag1
fit.glm3 <- glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)
probs3 <- predict(fit.glm3, Weekly.20092010, type = "response")
pred.glm3 <- rep("Down", length(probs3))
pred.glm3[probs3 > 0.5] = "Up"
table(pred.glm3, Direction.20092010)
```

Result:

```
Direction.20092010
pred.glm3 Down Up
     Down    1  1
     Up     42 60
```

R-code:

```
mean(pred.glm3 == Direction.20092010)
```

Result:

```
[1] 0.5865385
```

R-code:

```
# QDA with sqrt(abs(Lag2))
fit.qda2 <- qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)
pred.qda2 <- predict(fit.qda2, Weekly.20092010)
table(pred.qda2$class, Direction.20092010)
```

Result:

```
Direction.20092010
       Down Up
  Down   12 13
  Up     31 48
```

R-code:

```
mean(pred.qda2$class == Direction.20092010)
```

Result:

```
[1] 0.5769231
```

R-code:

```
# KNN k =10
pred.knn2 <- knn(train.X, test.X, train.Direction, k = 10)
table(pred.knn2, Direction.20092010)
```

Result:

```
Direction.20092010
pred.knn2 Down Up
     Down   17 18
     Up     26 43
```

R-code:
mean(pred.knn2 == Direction.20092010)

Result:
```
[1] 0.5769231
```

R-code:
```
# KNN k = 100
pred.knn3 <- knn(train.X, test.X, train.Direction, k = 100)
table(pred.knn3, Direction.20092010)
```

Result:
```
Direction.20092010
pred.knn3 Down Up
     Down    9 12
     Up     34 49
```

R-code:
mean(pred.knn3 == Direction.20092010)

Result:
```
[1] 0.5576923
```

Conclusion:

Out of these combinations, the **original logistic regression and LDA** have the best perfor mance in terms of test error rates.