Bruce Campbell ST-617 Homework 2

Tue Jul 05 15:22:14 2016

Chapter 4

Problem 10

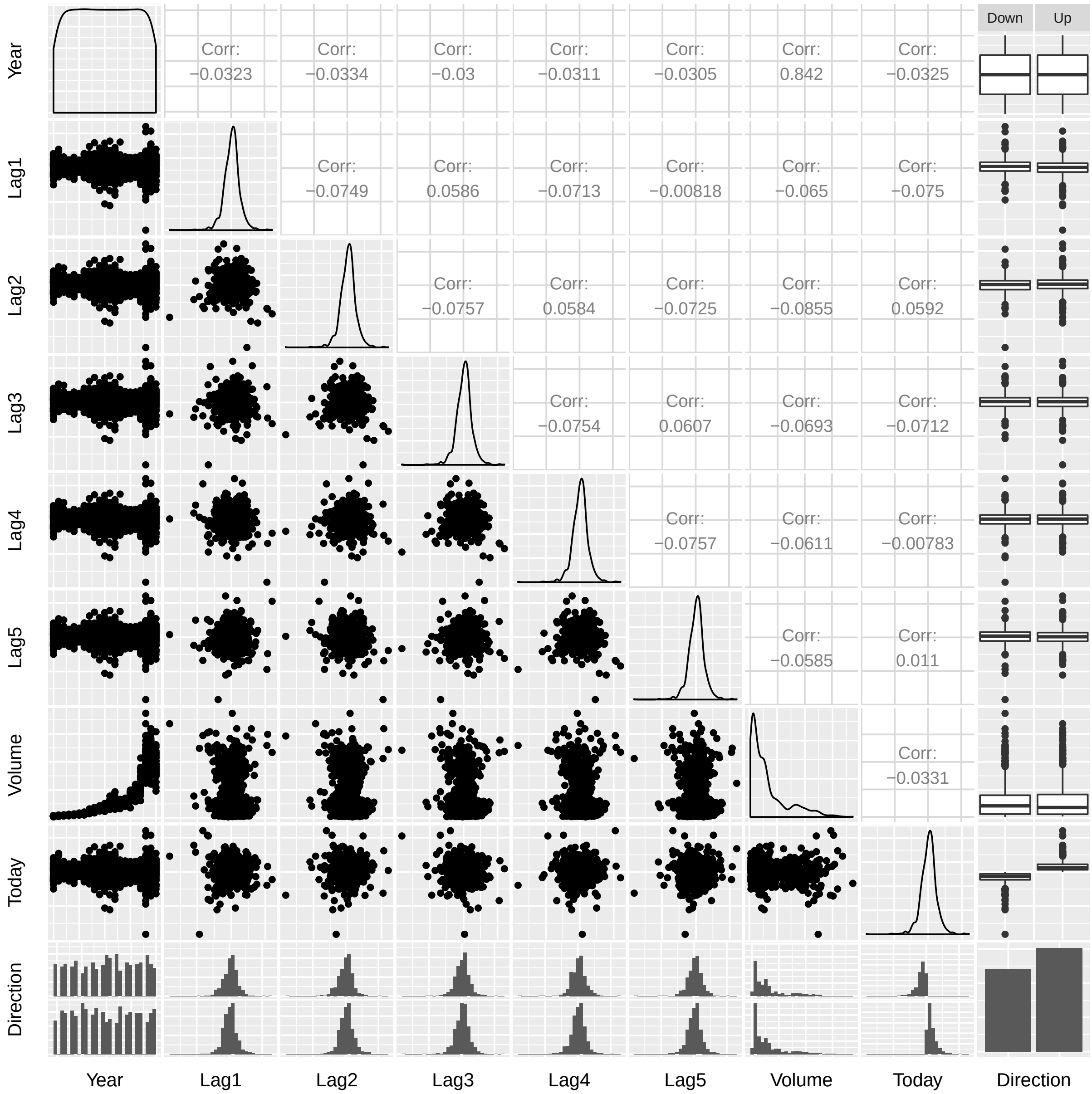
This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

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

```
r library(ISLR) attach(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 ## Min. :-18.1950 Min.
:-18.1950 Min. :0.08747 ## 1st Qu.: -1.1580 1st Qu.: -1.1660 1st Qu.:0.33202
## Median : 0.2380 Median : 0.2340 Median :1.00268 ## Mean : 0.1458 Mean
: 0.1399 Mean :1.57462 ## 3rd Qu.: 1.4090 3rd Qu.: 1.4050 3rd Qu.:2.05373
## Max. : 12.0260 Max. : 12.0260 Max. :9.32821 ## Today Direction
## Min. :-18.1950 Down:484 ## 1st Qu.: -1.1540 Up :605 ## Median : 0.2410
## Mean : 0.1499 ## 3rd Qu.: 1.4050 ## Max. : 12.0260
```

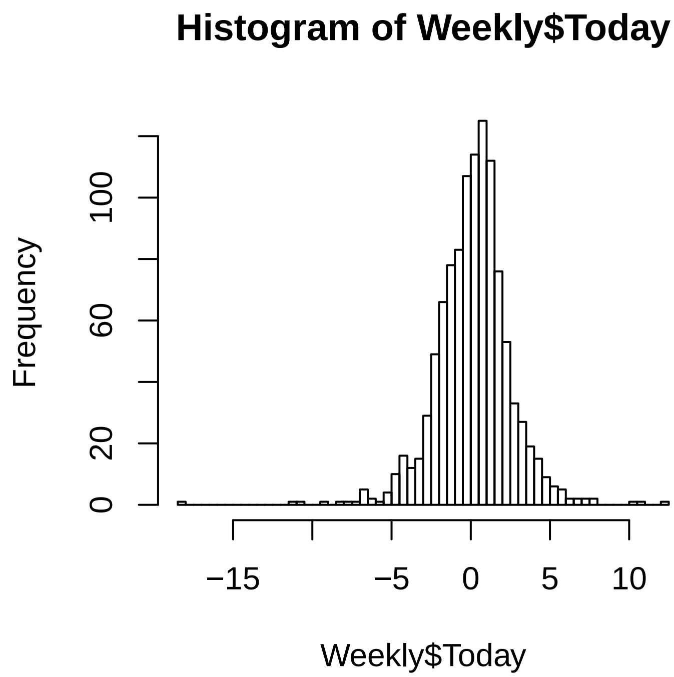
```
r library(ggplot2) require(GGally) ggpairs(Weekly) + theme(axis.line = element_blank(),
axis.text = element_blank(), axis.ticks = element_blank())
```



From the above we note that

- THere are more up than down weeks in the data set
- Volume is increasing over time
- Volume on up days has a longer tail that volumen on down days
- returns may have skew

`r  hist(Weekly$Today, 50)`




```
r  library(moments)  skewness(Weekly$Today)
```

```
## [1] -0.4805021
```

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  attach(Weekly)  DFWeekly = Weekly  glm.fit = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4
+ Lag5 + Volume, data = Weekly,       family = binomial)  summary(glm.fit)
```

```
##   ## 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
```

The lag2 variable is significant with a p-value of 0.0296

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
glm.probs = predict(glm.fit, type = "response")
library(pander)
contrasts(Direction)
```

```
##      Up
## Down  0
## Up    1
```

```r
glm.pred = rep("Down ", nrow(Weekly))
glm.pred[glm.probs > 0.5] = " Up"
table(glm.pred, Direction)
```

```
##         Direction
## glm.pred Down  Up
##     Up    430 557
##    Down    54  48
```

```r
pi_up = sum(Direction == "Up")
pi_down = sum(Direction == "Down")
```

We see that the accuracy is (557 + 54) / 1089 which is 56% and that the classifier does poorly on the down class where the accuracy is `0.1115702`

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
attach(Weekly)
DF <- Weekly
DFTrain <- DF[DF$Year <= 2008, ]
DFTest <- DF[DF$Year > 2008, ]
glm.fit = glm(Direction ~ Lag2, data = DFTrain, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = DFTrain)
##
## 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
```

```
glm.probs = predict(glm.fit, DFTest, type = "response")
library(pander)
glm.pred = rep("Down ", nrow(DFTest))
glm.pred[glm.probs > 0.5] = " Up"
table(glm.pred, DFTest$Direction)
```

```
##
## glm.pred Down Up
##     Up     34 56
##    Down     9  5
```

```
pi_up = sum(Direction == "Up")
pi_down = sum(Direction == "Down")
```

The accuracy for logistic regerssion on the test set is (9+56)/104 - or 62%.

e) Repeat (d) using LDA.

```
attach(Weekly)
library(MASS)
lda.fit = lda(Direction ~ Lag2, data = DFTrain)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag2, data = DFTrain)
##
## 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
lda.pred = predict(lda.fit, DFTest)
names(lda.pred)
```

```
## [1] "class"     "posterior" "x"
```

```r
lda.class = lda.pred$class
table(lda.class, DFTest$Direction)
```

```
##
## lda.class Down Up
##      Down    9  5
##      Up     34 56
```

The accuracy for LDA classification on the test set is (9+56)/104 - or 62%. Note this is identical to the logistic regression

f) Repeat (d) using QDA.

```r
attach(Weekly)
train = (Year < 2009)
Weekly.2009 = Weekly[!train, ]
Direction.2009 = Weekly$Direction[!train]
qda.fit = qda(Direction ~ Lag2, data = Weekly, subset = train)
qda.class = predict(qda.fit, Weekly.2009)$class
table(qda.class, Direction.2009)
```

```
##          Direction.2009
## qda.class Down Up
##      Down    0  0
##      Up     43 61
```

This classifier did not correctly classify any of the down test points. We diagnose the code a few ways below. First by adding the Lag1 variable and second by reproducing the results on the SMarket dataset.

```r
qda.fit = qda(Direction ~ Lag1 + Lag2, data = DFTrain)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = DFTrain)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##              Lag1        Lag2
## Down  0.289444444 -0.03568254
## Up   -0.009213235  0.26036581
```

```
qda.pred = predict(qda.fit, DFTest)
names(qda.pred)
```

```
## [1] "class"     "posterior"
```

```
qda.class = predict(qda.fit, DFTest)$class

qda.class = qda.pred$class
table(qda.class, DFTest$Direction)
```

```
##
## qda.class Down Up
##      Down    7 10
##      Up     36 51
```

The accuracy for this classifier is (7+51) / 104 - 56%

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

```
library(class)
attach(Weekly)
train = (Year < 2009)
train.X = data.frame(cbind(Lag2)[train, ])
test.X = data.frame(cbind(Lag2)[!train, ])
train.Direction = Direction[train]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.2009)
```

```
##         Direction.2009
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

The accuracy of KNN with k=1 is (32 + 18) / 104 - 48%.

h)

Which of these methods appears to provide the best results on this data? For this data set and model we see that the logistic regression and LDA are the top performers in terms of classification accuracy.

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
library(class)
attach(Weekly)
train = (Year < 2009)
Weekly.2009 = Weekly[!train, ]
Direction.2009 = Direction[!train]

message("KNN")
train.X = cbind(Lag1, Lag2)[train, ]
test.X = cbind(Lag1, Lag2)[!train, ]
train.Direction = Direction[train]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 2)
TB <- table(knn.pred, Direction.2009)
ACC_KNN = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- data.frame(model = "KNN(Direction~Lag1,Lag2) k=2", Accuracy = ACC_KNN)


train.X = cbind(Lag1, Lag2, Volume)[train, ]
test.X = cbind(Lag1, Lag2, Volume)[!train, ]
train.Direction = Direction[train]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)
TB <- table(knn.pred, Direction.2009)
ACC_KNN = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "KNN(Direction~Lag1+Lag2+Volume) k=1",
    Accuracy = ACC_KNN))


knn.pred = knn(train.X, test.X, train.Direction, k = 2)
TB <- table(knn.pred, Direction.2009)
ACC_KNN = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "KNN(Direction~Lag1+Lag2+Volume) k=2",
    Accuracy = ACC_KNN))


knn.pred = knn(train.X, test.X, train.Direction, k = 4)
TB <- table(knn.pred, Direction.2009)
ACC_KNN = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "KNN(Direction~Lag1+Lag2+Volume) k=4",
    Accuracy = ACC_KNN))

message("QDA")

attach(Weekly)
train = (Year < 2009)
Weekly.2009 = Weekly[!train, ]
Direction.2009 = Direction[!train]
qda.fit = qda(Direction ~ Lag1 + Lag2 + Volume, data = Weekly, subset = train)
qda.class = predict(qda.fit, Weekly.2009)$class
TB <- table(qda.class, Direction.2009)
ACC_QDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "QDA(Direction~Lag1+Lag2+Volume)",
    Accuracy = ACC_QDA))
```

```r
qda.fit = qda(Direction ~ Lag1 + Lag2 + +Volume + Lag1 * Lag2, data = Weekly,
    subset = train)
qda.class = predict(qda.fit, Weekly.2009)$class
TB <- table(qda.class, Direction.2009)
ACC_QDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "QDA(Direction~Lag1+Lag2+Volume+Direction + Lag1*Lag2)",
    Accuracy = ACC_QDA))

qda.fit = qda(Direction ~ Lag1 + Lag2 + Lag1 * Lag2, data = Weekly, subset = train)
qda.class = predict(qda.fit, Weekly.2009)$class
TB <- table(qda.class, Direction.2009)
ACC_QDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "QDA(Direction~Lag1+Lag2+Lag1 * Lag1)",
    Accuracy = ACC_QDA))

qda.fit = qda(Direction ~ Lag1 + Lag2, data = Weekly, subset = train)
qda.class = predict(qda.fit, Weekly.2009)$class
TB <- table(qda.class, Direction.2009)
ACC_QDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "QDA(Direction~Lag1+Lag2)", Accuracy = ACC_QDA))

message("LDA")

attach(Weekly)
train = (Year < 2009)
Weekly.2009 = Weekly[!train, ]
Direction.2009 = Direction[!train]
lda.fit = lda(Direction ~ Lag1 + Lag2 + Volume, data = Weekly, subset = train)
lda.class = predict(lda.fit, Weekly.2009)$class
TB <- table(lda.class, Direction.2009)
ACC_LDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "LDA(Direction~Lag1+Lag2+Volume)",
    Accuracy = ACC_LDA))

lda.fit = lda(Direction ~ Lag1 + Lag2 + +Volume + Lag1 * Lag2, data = Weekly,
    subset = train)
lda.class = predict(lda.fit, Weekly.2009)$class
TB <- table(lda.class, Direction.2009)
ACC_LDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "LDA(Direction~Lag1+Lag2+Volume+Direction + Lag1*Lag2)",
    Accuracy = ACC_LDA))

lda.fit = lda(Direction ~ Lag1 + Lag2 + Lag1 * Lag2, data = Weekly, subset = train)
lda.class = predict(lda.fit, Weekly.2009)$class
TB <- table(lda.class, Direction.2009)
ACC_LDA = (TB[1] + TB[4])/length(Direction.2009)
modelsDF <- rbind(modelsDF, data.frame(model = "LDA(Direction~Lag1+Lag2+Lag1 * Lag1)",
    Accuracy = ACC_LDA))

lda.fit = lda(Direction ~ Lag1 + Lag2, data = Weekly, subset = train)
lda.class = predict(lda.fit, Weekly.2009)$class
TB <- table(lda.class, Direction.2009)
ACC_LDA = (TB[1] + TB[4])/length(Direction.2009)
```

```r
modelsDF <- rbind(modelsDF, data.frame(model = "LDA(Direction~Lag1+Lag2)", Accuracy = ACC_LDA))

pander(modelsDF)
```

model	Accuracy
KNN(Direction~Lag1,Lag2) k=2	0.5288
KNN(Direction~Lag1+Lag2+Volume) k=1	0.5
KNN(Direction~Lag1+Lag2+Volume) k=2	0.5096
KNN(Direction~Lag1+Lag2+Volume) k=4	0.4712
QDA(Direction~Lag1+Lag2+Volume)	0.4615
QDA(Direction~Lag1+Lag2+Volume+Direction + Lag1*Lag2)	0.4519
QDA(Direction~Lag1+Lag2+Lag1 * Lag1)	0.4615
QDA(Direction~Lag1+Lag2)	0.5577
LDA(Direction~Lag1+Lag2+Volume)	0.5288
LDA(Direction~Lag1+Lag2+Volume+Direction + Lag1*Lag2)	0.5385
LDA(Direction~Lag1+Lag2+Lag1 * Lag1)	0.5769
LDA(Direction~Lag1+Lag2)	0.5769

We see the best prforming models from this set are QDA(Direction~Lag1+Lag2) and LDA(Direction~Lag1+Lag2)