Chapter 4:Classification- problem 10

Kartheek Raj

12/22/2019

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.

Packages required :ISLR,caret,class and MASS

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

Answer

Weekly dataset pulling from ISLR pacakage

```r
require(ISLR)
```

```
## Loading required package: ISLR
```

```r
require(caret)
```

```
## Loading required package: caret

## Loading required package: lattice

## Loading required package: ggplot2
```

```r
require(MASS)
```

```
## Loading required package: MASS
```

```r
require(class)
```

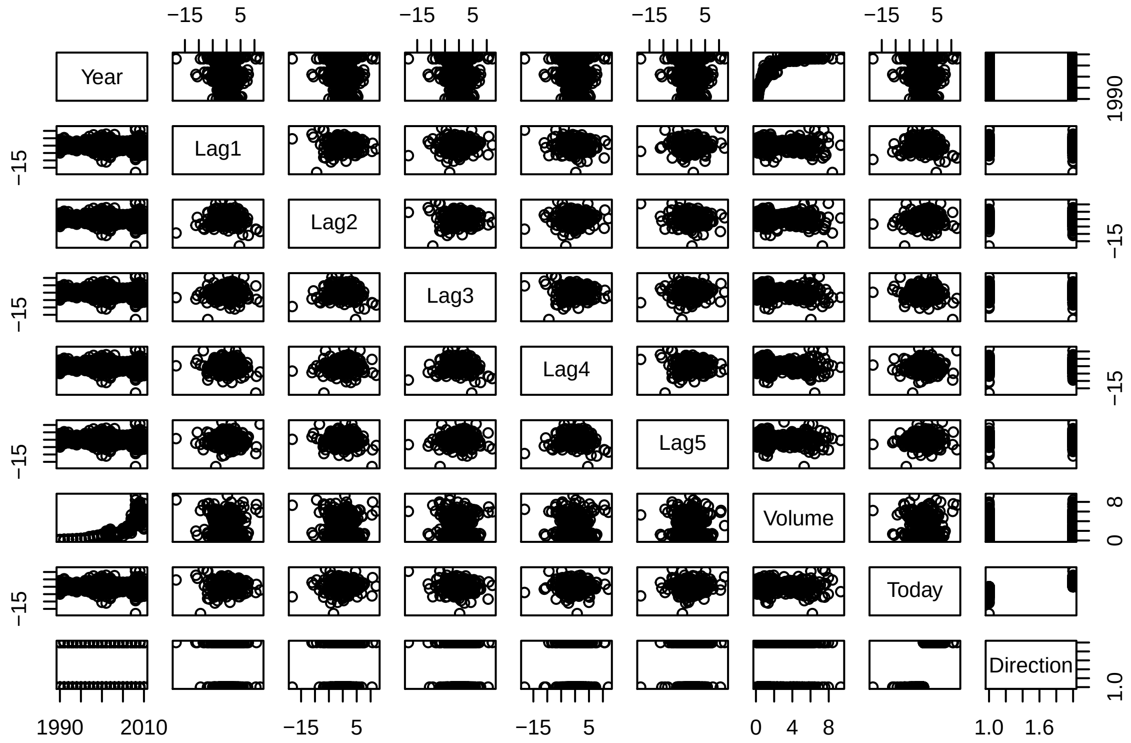
```
## Loading required package: class
```

```r
library(ISLR)
library(caret)
library(MASS)
library(class)
weekly<-data.frame(Weekly)
head(weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

Dataset numerical summary and garphical summary

```r
pairs(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  
##            
##            
##            
## 
```

```
head(weekly)
```

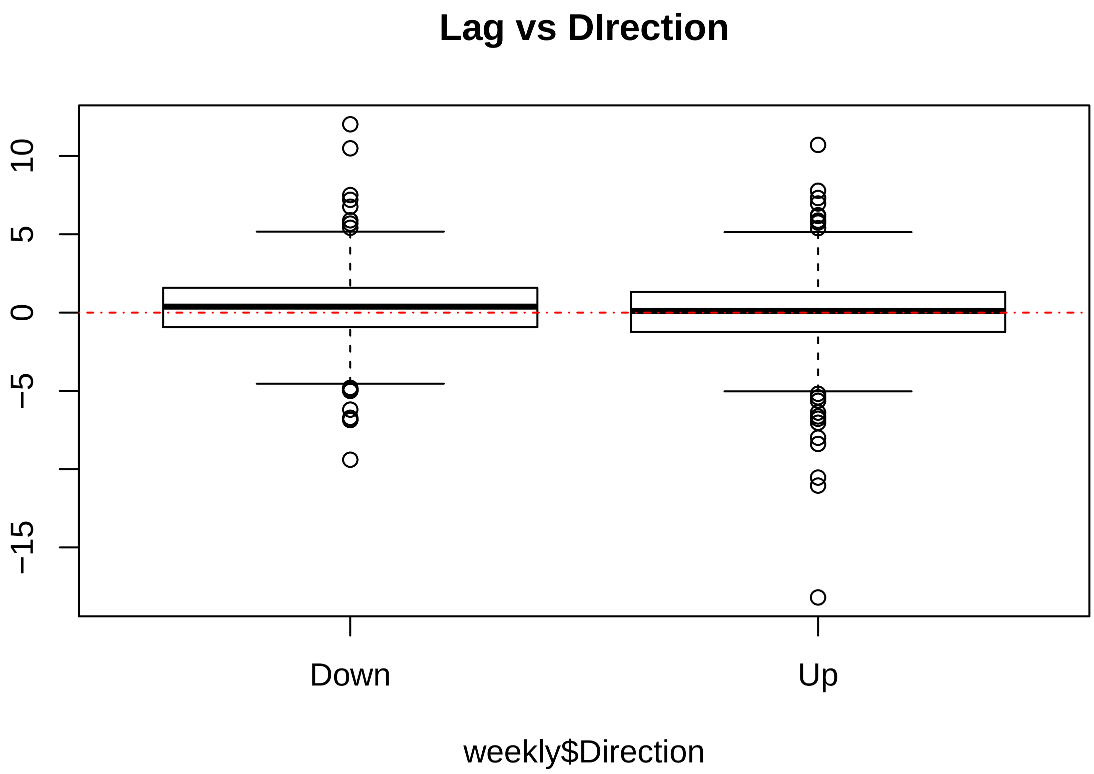
```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

```
print(names(weekly))
```

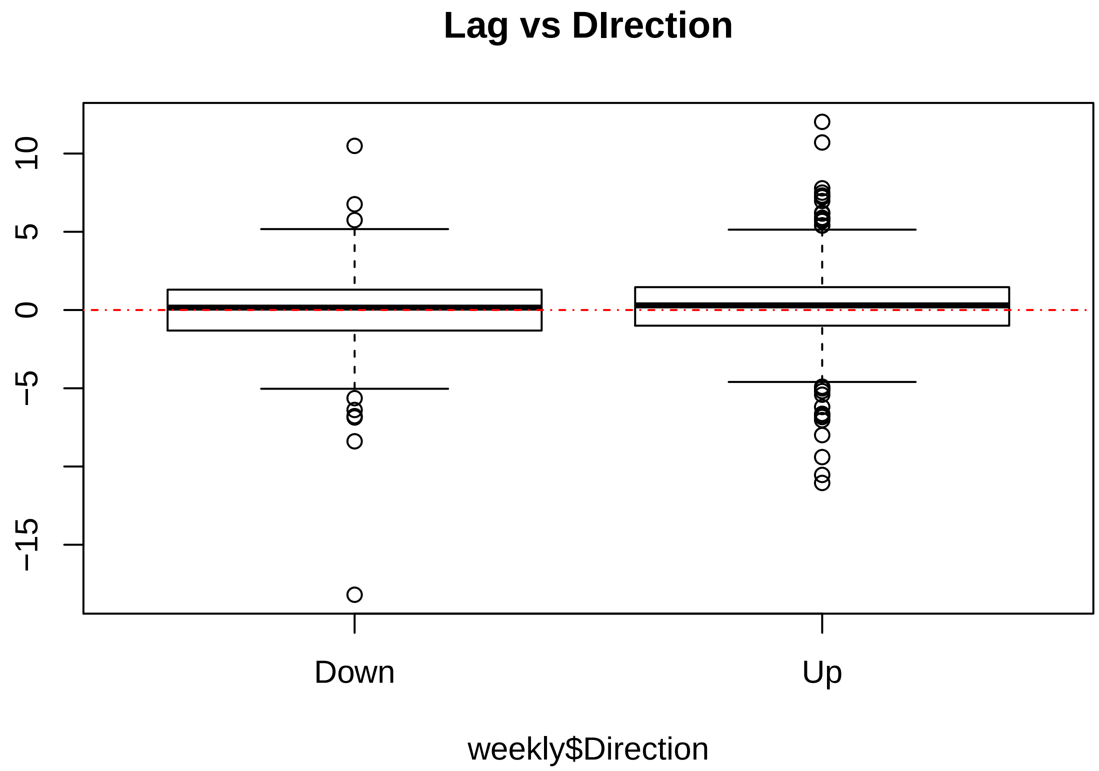
```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

EDA insight 1: Volume vs year shown relation i.e upward trend.

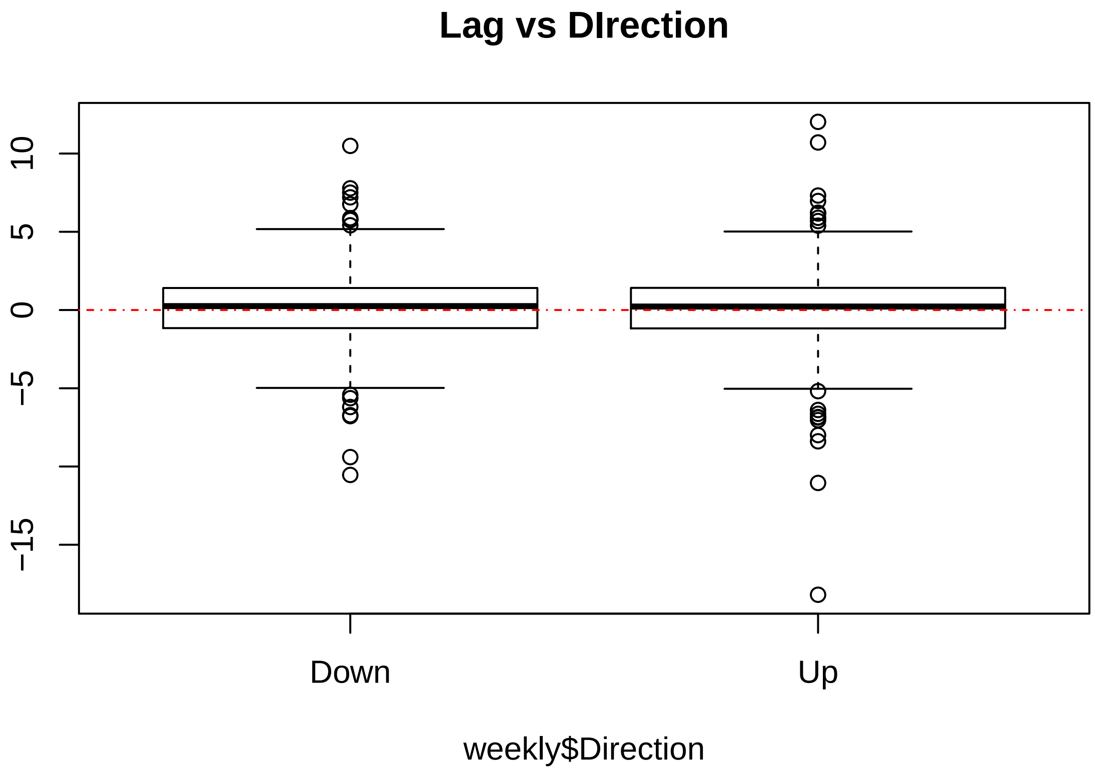
```
dplot<-function(x){
  h=x

  boxplot(h~weekly$Direction,ylab=colnames(x),main="Lag vs DIrection")
  abline(h=0,col="red",lty="dotdash")
}
dplot(weekly$Lag1)
```



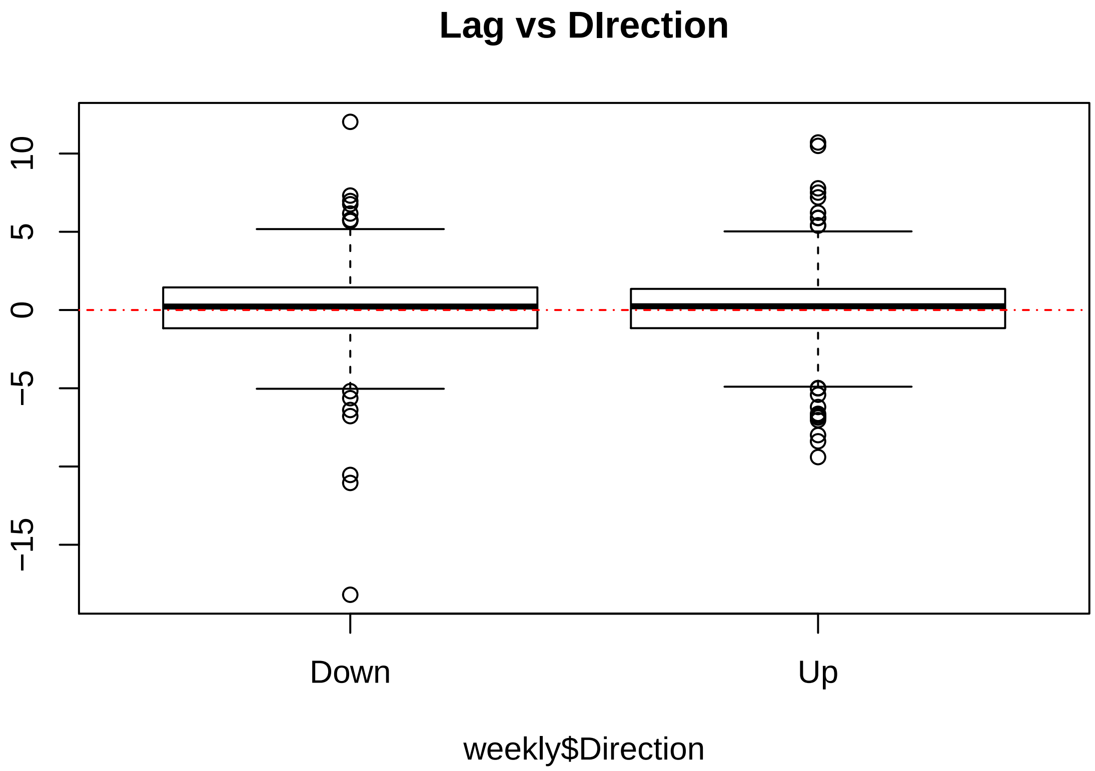
```
dplot(weekly$Lag2)
```



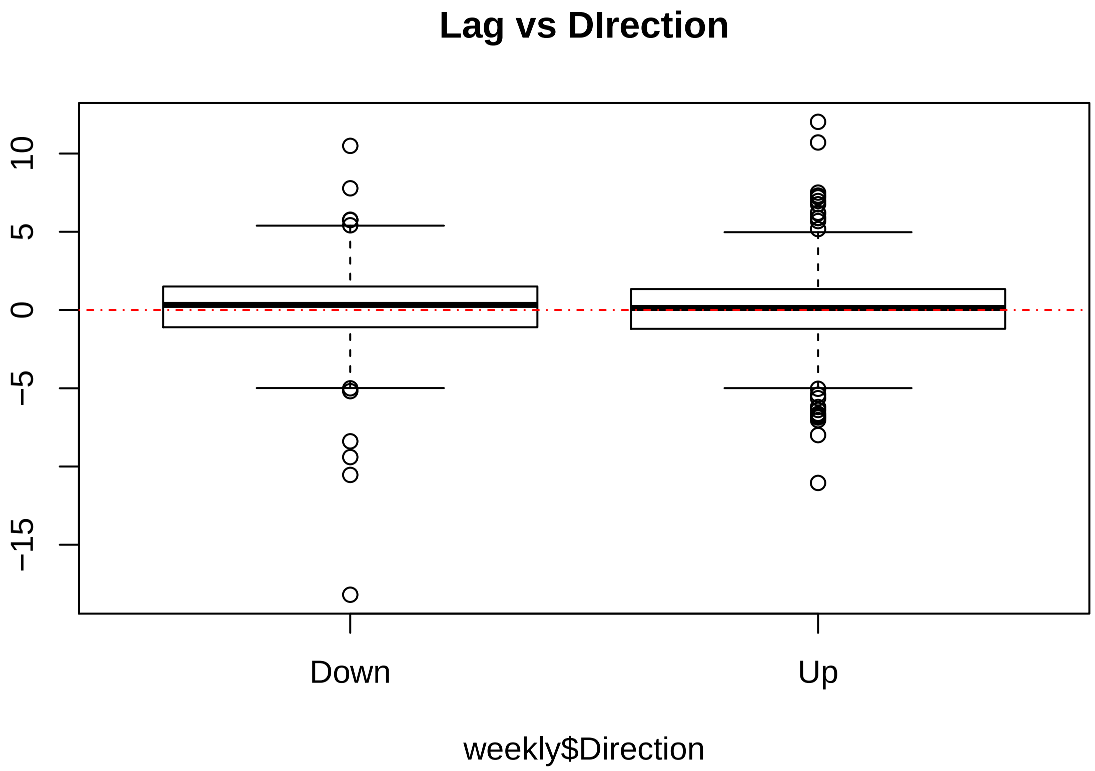
```r
dplot(weekly$Lag3)
```



```
dplot(weekly$Lag4)
```



```
dplot(weekly$Lag5)
```



EDA insight 2 : All lags up and down reflects similar data disturbition not the same to each other.

(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?

Answer

Logi Model building all columns except **Today**.

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

```
##
## 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
```

Model Insight 1: Lag2 which is the highest signaficance predcitor of the model and the rest of the predictors not much.

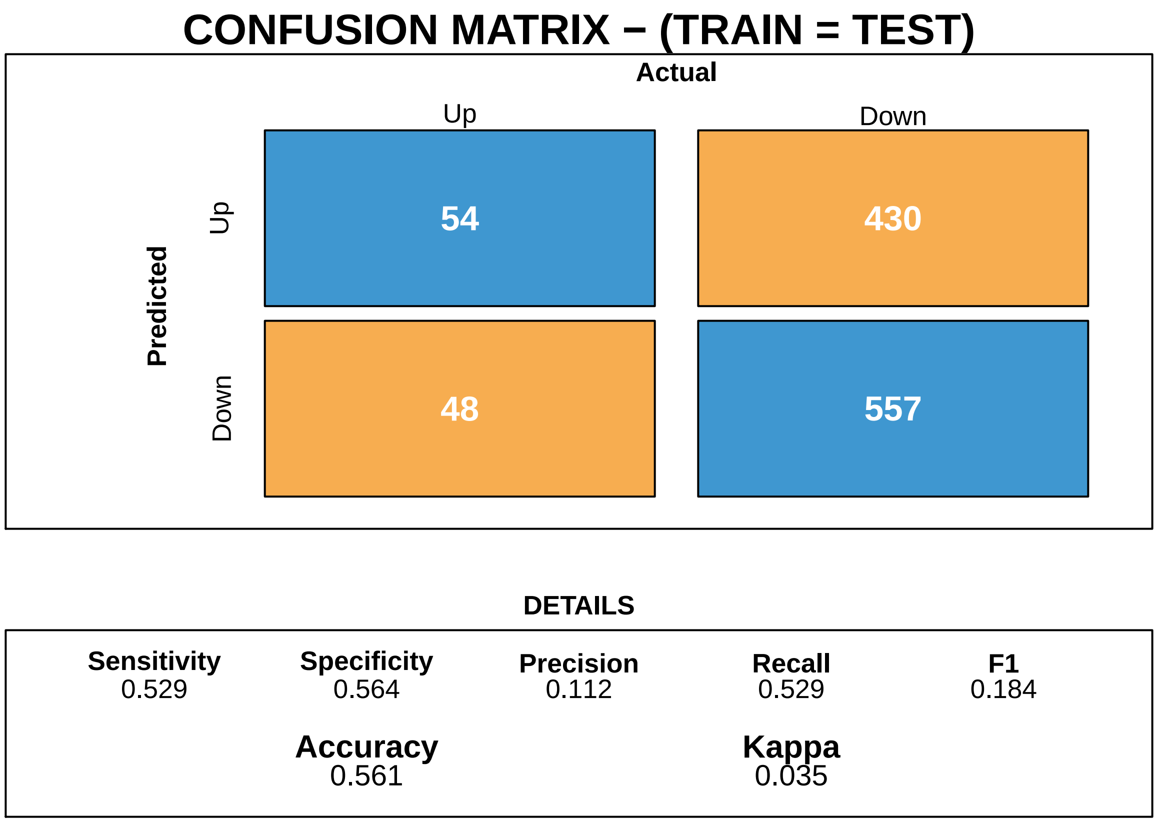
(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

Answer

Generating the Confusion matrix.

```r
logip<-predict(logi,newdata = weekly,type="response")

logipc<-as.factor(ifelse(logip>0.5,"Up","Down"))
confusionMatrix(weekly$Direction,logipc,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down  Up
##       Down   54 430
##       Up     48 557
##
##                Accuracy : 0.5611
##                  95% CI : (0.531, 0.5908)
##     No Information Rate : 0.9063
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.035
##
##  Mcnemar's Test P-Value : <2e-16
##
##               Precision : 0.11157
##                  Recall : 0.52941
##                      F1 : 0.18430
##              Prevalence : 0.09366
##          Detection Rate : 0.04959
##    Detection Prevalence : 0.44444
##       Balanced Accuracy : 0.54687
##
##        'Positive' Class : Down
##
```



CONFUSION MATRIX − (TRAIN = TEST)

Predicted \ Actual	Up	Down
Up	54	430
Down	48	557

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.529	0.564	0.112	0.529	0.184

Accuracy	Kappa
0.561	0.035

(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).

Answer

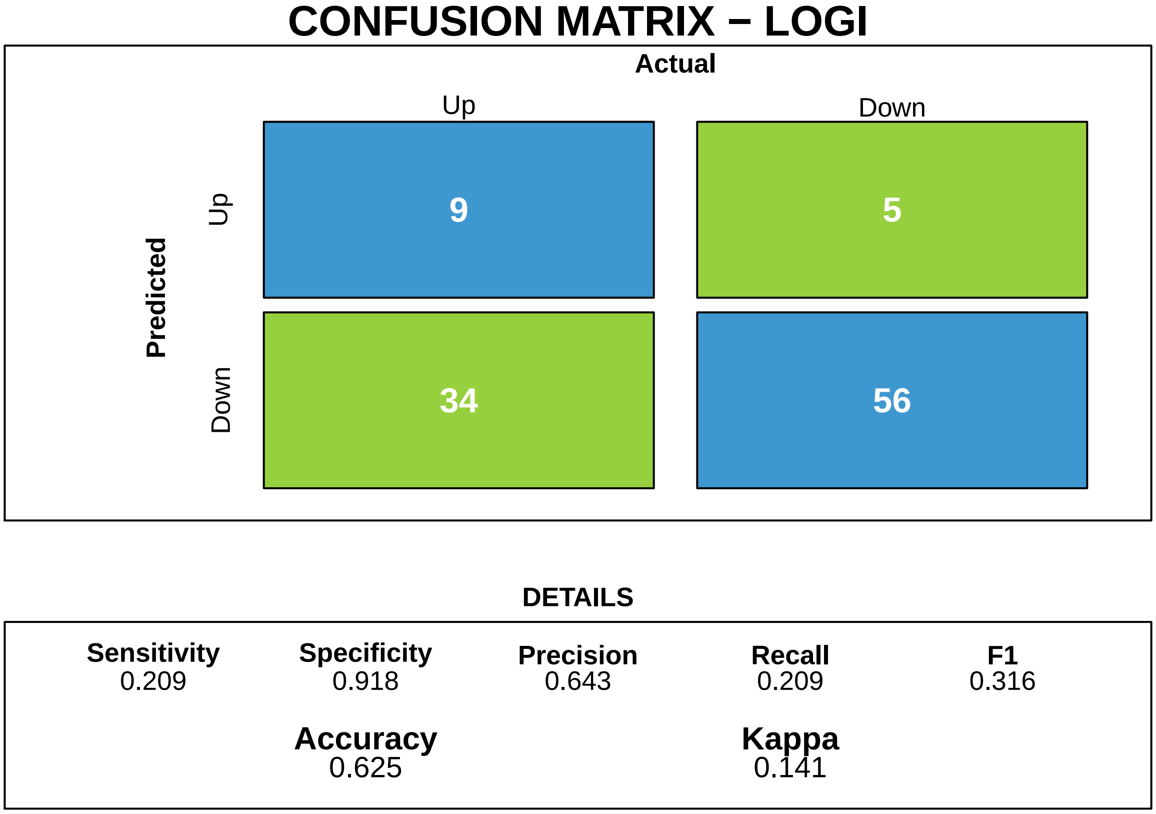
Test and train data split: Train dataset which include all rows before 2008 and after 2008 is test data

```
train<-weekly[weekly$Year<=2008,]
test<-weekly[weekly$Year>2008,]
```

Building the logi model **Lag2** as solo predictor onto Direction and test the data

```
logi1<-glm(Direction~Lag2,data = train,family = binomial )
logip1<-predict(logi1,newdata=test,type="response")
logipc1<-as.factor(ifelse(logip1>0.5,"Up","Down"))
confusionMatrix(logipc1,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    9  5
##       Up     34 56
##
##                Accuracy : 0.625
##                  95% CI : (0.5247, 0.718)
##     No Information Rate : 0.5865
##      P-Value [Acc > NIR] : 0.2439
##
##                    Kappa : 0.1414
##
##  Mcnemar's Test P-Value : 7.34e-06
##
##                Precision : 0.64286
##                   Recall : 0.20930
##                       F1 : 0.31579
##               Prevalence : 0.41346
##           Detection Rate : 0.08654
##     Detection Prevalence : 0.13462
##        Balanced Accuracy : 0.56367
##
##         'Positive' Class : Down
##
```



Model Insight 2: Accuracy is 62% and No information rate is 0.5865.

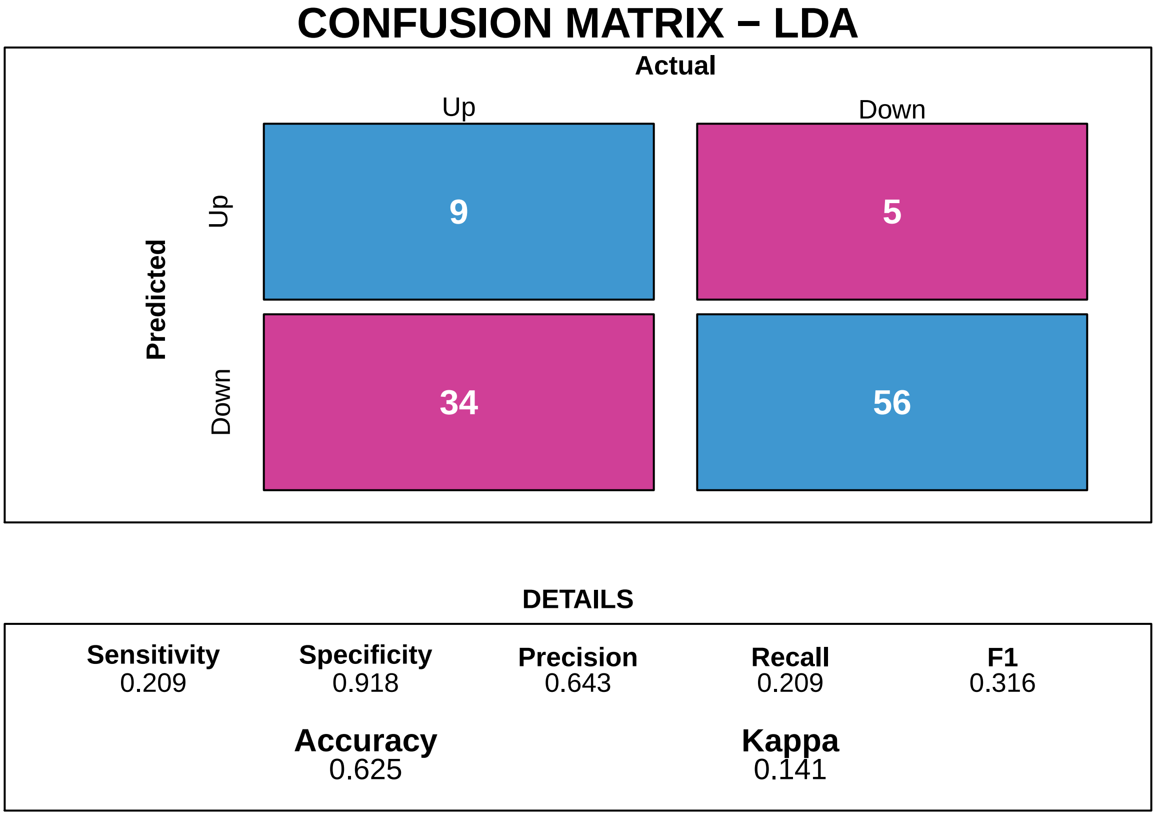
(e) Repeat (d) using LDA.

Answer

```r
ldad<-lda(Direction~Lag2,data = train)
ldadp<-predict(ldad,newdata= test)
confusionMatrix(ldadp$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
```

```
##
##            Reference
## Prediction Down Up
##       Down    9  5
##       Up     34 56
##
##                Accuracy : 0.625
##                  95% CI : (0.5247, 0.718)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.2439
##
##                   Kappa : 0.1414
##
##  Mcnemar's Test P-Value : 7.34e-06
##
##               Precision : 0.64286
##                  Recall : 0.20930
##                      F1 : 0.31579
##              Prevalence : 0.41346
##          Detection Rate : 0.08654
##    Detection Prevalence : 0.13462
##       Balanced Accuracy : 0.56367
##
##        'Positive' Class : Down
##
```



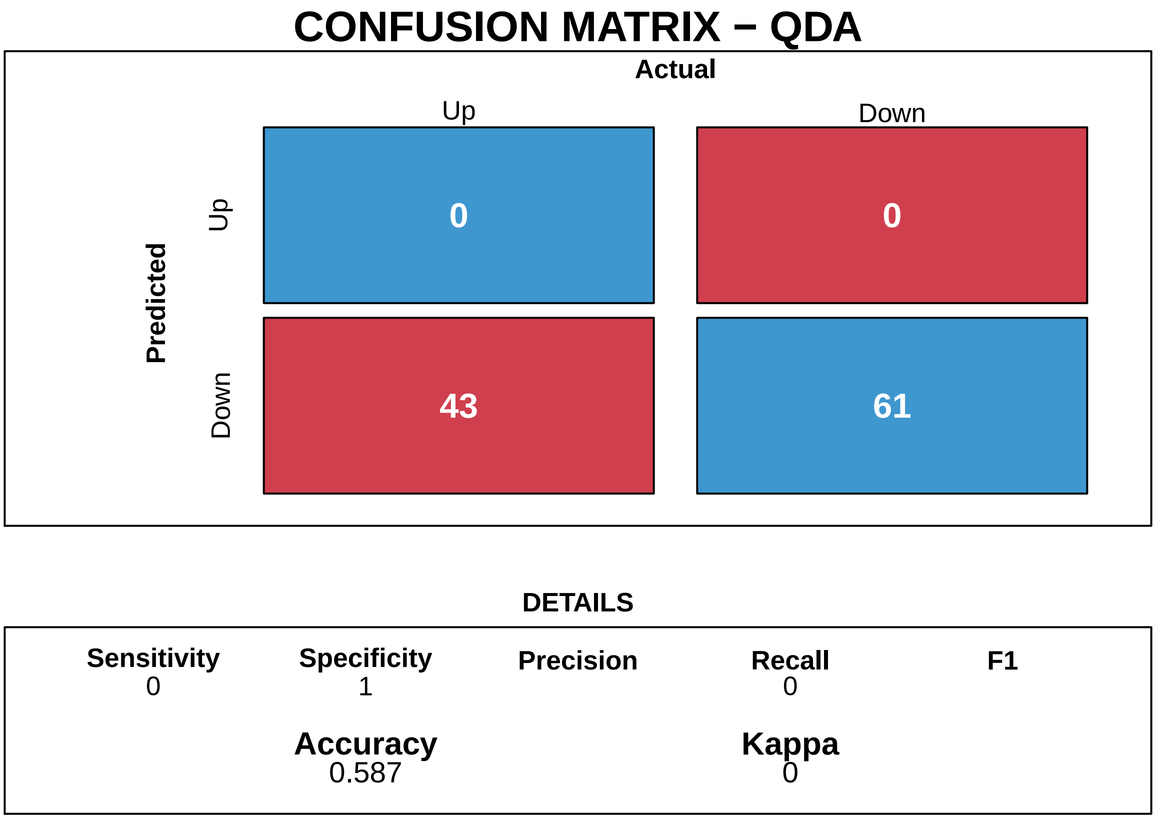
Model Insight 3 : LDQ model parameters same as logistic regression model.

(f) Repeat (d) using QDA.

Answer

```
qdad<-qda(Direction~Lag2,data = train)
qdadp<-predict(qdad,newdata= test)
confusionMatrix(qdadp$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    0  0
##       Up     43 61
##
##                Accuracy : 0.5865
##                  95% CI : (0.4858, 0.6823)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.5419
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 1.504e-10
##
##               Precision :     NA
##                  Recall : 0.0000
##                      F1 :     NA
##              Prevalence : 0.4135
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Down
##
```

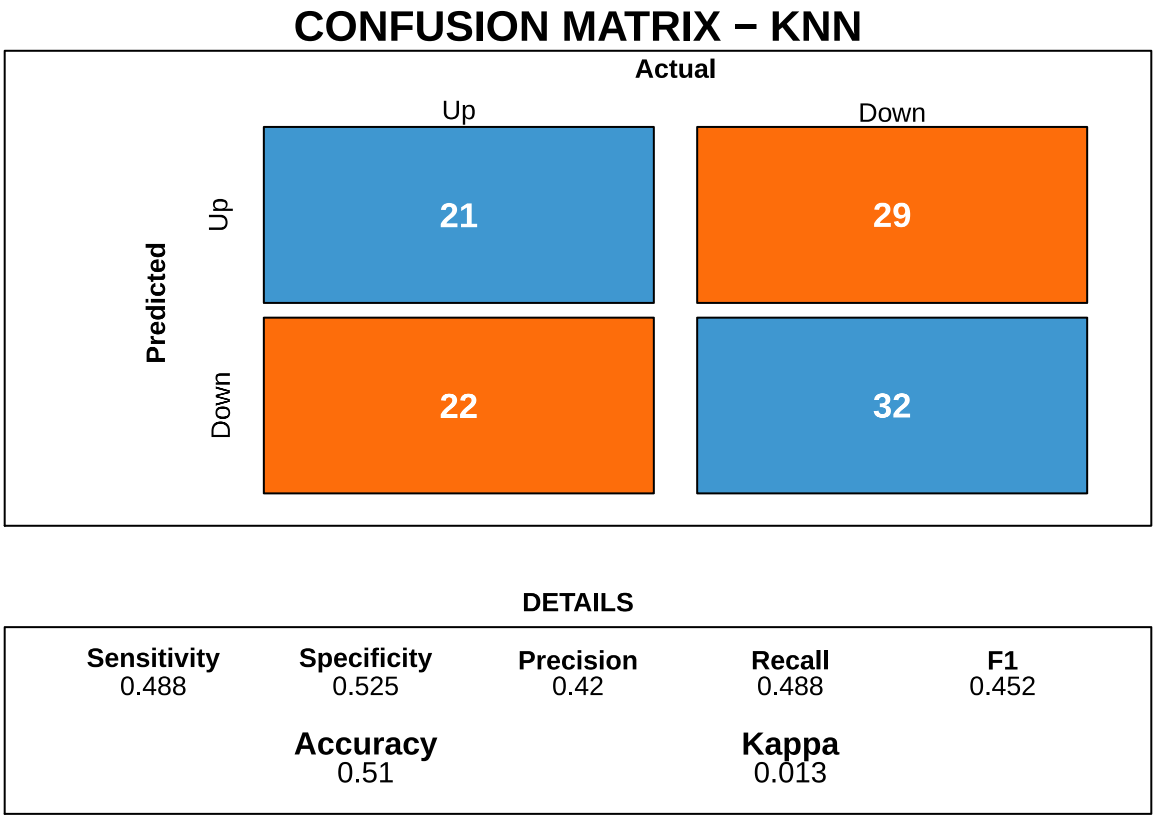


(g) Repeat (d) using KNN with K = 1.

Answer

```
knnp<-knn(as.matrix(train$Lag2),as.matrix(test$Lag2),train$Direction,k=1)
confusionMatrix(knnp,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down   21 29
##       Up     22 32
##
##                Accuracy : 0.5096
##                  95% CI : (0.4097, 0.609)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.9540
##
##                   Kappa : 0.0127
##
##  Mcnemar's Test P-Value : 0.4008
##
##               Precision : 0.4200
##                  Recall : 0.4884
##                      F1 : 0.4516
##              Prevalence : 0.4135
##          Detection Rate : 0.2019
##    Detection Prevalence : 0.4808
##       Balanced Accuracy : 0.5065
##
##        'Positive' Class : Down
##
```



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

Answer: logi and LDA model yoelds high accuracy but KNN model has highest F1 score amon all models.

(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.

Answer

Building models with polynomial transformations and combinations.

```r
logi_comb<-glm(Direction ~ Lag1 + Lag2, family = binomial, data = train)
lda_comb<-lda(Direction~Lag1+Lag2+Lag3,data = train)
qda_comb<-qda(Direction~Lag2+Lag3,data = train)
logi_trans<-glm(Direction ~ poly(Lag2,2), family = binomial, data = train)
lda_trans<-lda(Direction~poly(Lag2,2),data = train)
qda_trans<-qda(Direction~poly(Lag2,2),data = train)
knn_comb<-knn(as.matrix(train$Lag1+train$Lag2),as.matrix(test$Lag1+test$Lag2),train$Direction,k=1)
```

Generated predictions on previously generated models.

```
pred_logi_comb<-predict(logi_comb,newdata= test,type="response")
pred_lda_comb<-predict(lda_comb,newdata= test)
pred_qda_comb<-predict(qda_comb,newdata= test)
pred_logi_trans<-predict(logi_trans,newdata= test,type="response")
pred_lda_trans<-predict(lda_trans,newdata= test)
pred_qda_trans<-predict(qda_trans,newdata= test)
l_combo<-as.factor(ifelse(pred_logi_comb>0.5,"Up","Down"))
l_trans<-as.factor(ifelse(pred_logi_trans>0.5,"Up","Down"))
```

Confusion matrix summaries all tranformation and combination models.

```
confusionMatrix(l_combo,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    7  8
##       Up     36 53
##
##                Accuracy : 0.5769
##                  95% CI : (0.4761, 0.6732)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.6193
##
##                   Kappa : 0.035
##
##  Mcnemar's Test P-Value : 4.693e-05
##
##               Precision : 0.46667
##                  Recall : 0.16279
##                      F1 : 0.24138
##              Prevalence : 0.41346
##          Detection Rate : 0.06731
##    Detection Prevalence : 0.14423
##       Balanced Accuracy : 0.51582
##
##        'Positive' Class : Down
##
```

```
confusionMatrix(l_trans,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    8  4
##       Up     35 57
##
##                Accuracy : 0.625
##                  95% CI : (0.5247, 0.718)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.2439
##
```

```
##                   Kappa : 0.1348
##
##  Mcnemar's Test P-Value : 1.556e-06
##
##               Precision : 0.66667
##                  Recall : 0.18605
##                      F1 : 0.29091
##              Prevalence : 0.41346
##          Detection Rate : 0.07692
##    Detection Prevalence : 0.11538
##       Balanced Accuracy : 0.56024
##
##        'Positive' Class : Down
##
```

```
confusionMatrix(pred_lda_comb$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    8  9
##       Up     35 52
##
##                Accuracy : 0.5769
##                  95% CI : (0.4761, 0.6732)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.619326
##
##                   Kappa : 0.0423
##
##  Mcnemar's Test P-Value : 0.000164
##
##               Precision : 0.47059
##                  Recall : 0.18605
##                      F1 : 0.26667
##              Prevalence : 0.41346
##          Detection Rate : 0.07692
##    Detection Prevalence : 0.16346
##       Balanced Accuracy : 0.51925
##
##        'Positive' Class : Down
##
```

```
confusionMatrix(pred_qda_comb$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    4  2
##       Up     39 59
##
##                Accuracy : 0.6058
##                  95% CI : (0.5051, 0.7002)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.3847
##
##                   Kappa : 0.069
##
##  Mcnemar's Test P-Value : 1.885e-08
##
##               Precision : 0.66667
##                  Recall : 0.09302
##                      F1 : 0.16327
##              Prevalence : 0.41346
##          Detection Rate : 0.03846
##    Detection Prevalence : 0.05769
##       Balanced Accuracy : 0.53012
##
##        'Positive' Class : Down
##
```

```
confusionMatrix(pred_lda_trans$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    7  4
##       Up     36 57
##
##                Accuracy : 0.6154
##                  95% CI : (0.5149, 0.7091)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.311
##
##                   Kappa : 0.1092
##
##  Mcnemar's Test P-Value : 9.509e-07
##
##               Precision : 0.63636
##                  Recall : 0.16279
##                      F1 : 0.25926
##              Prevalence : 0.41346
##          Detection Rate : 0.06731
##    Detection Prevalence : 0.10577
##       Balanced Accuracy : 0.54861
##
##        'Positive' Class : Down
##
```

```
confusionMatrix(pred_qda_trans$class,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down    7  3
##       Up     36 58
```

```
##
##                Accuracy : 0.625
##                  95% CI : (0.5247, 0.718)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.2439
##
##                   Kappa : 0.1281
##
##  Mcnemar's Test P-Value : 2.99e-07
##
##               Precision : 0.70000
##                  Recall : 0.16279
##                      F1 : 0.26415
##              Prevalence : 0.41346
##          Detection Rate : 0.06731
##    Detection Prevalence : 0.09615
##       Balanced Accuracy : 0.55681
##
##        'Positive' Class : Down
##
```

```r
confusionMatrix(knn_comb,test$Direction,mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
##       Down   15 28
##       Up     28 33
##
##                Accuracy : 0.4615
##                  95% CI : (0.3633, 0.562)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.9962
##
##                   Kappa : -0.1102
##
##  Mcnemar's Test P-Value : 1.0000
##
##               Precision : 0.3488
##                  Recall : 0.3488
##                      F1 : 0.3488
##              Prevalence : 0.4135
##          Detection Rate : 0.1442
##    Detection Prevalence : 0.4135
##       Balanced Accuracy : 0.4449
##
##        'Positive' Class : Down
##
```