Week 6 Assignment

David Russo

2/18/2017

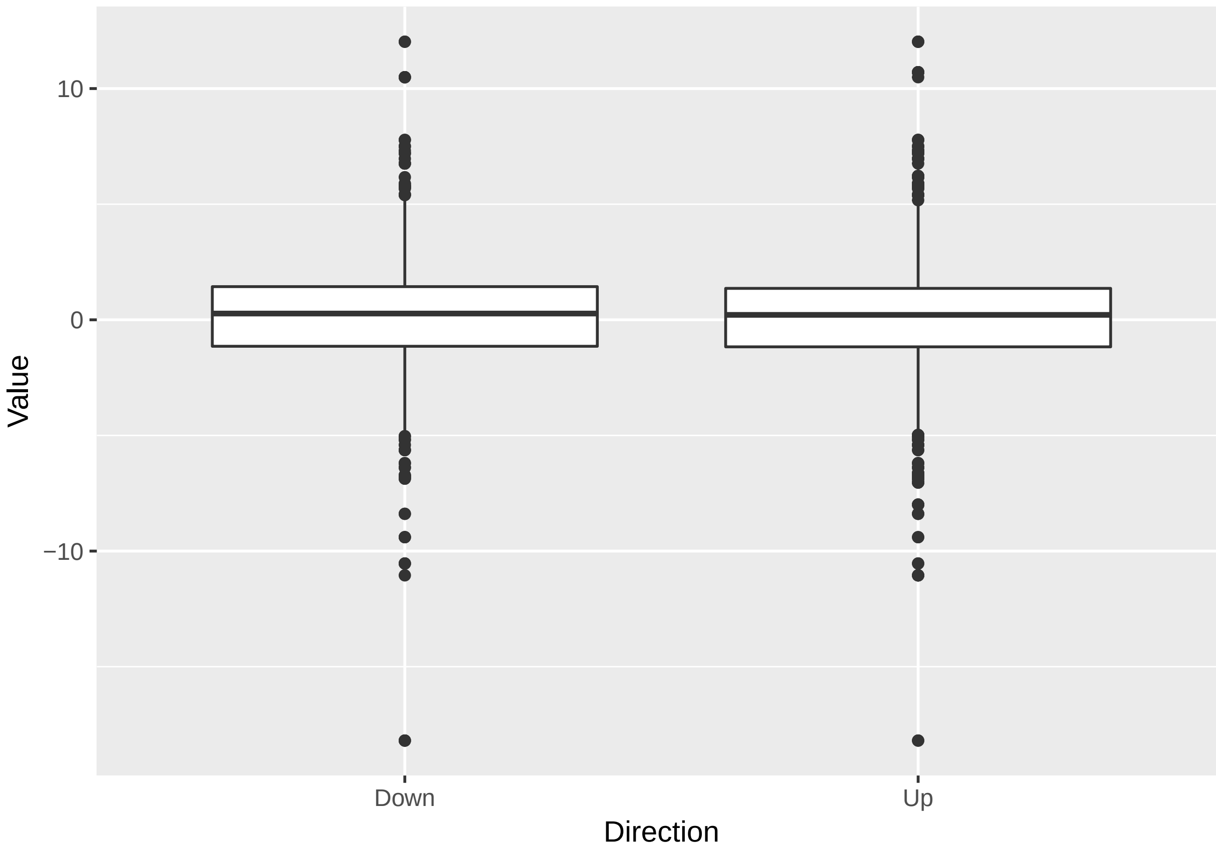
- 4.7 #10
- a.

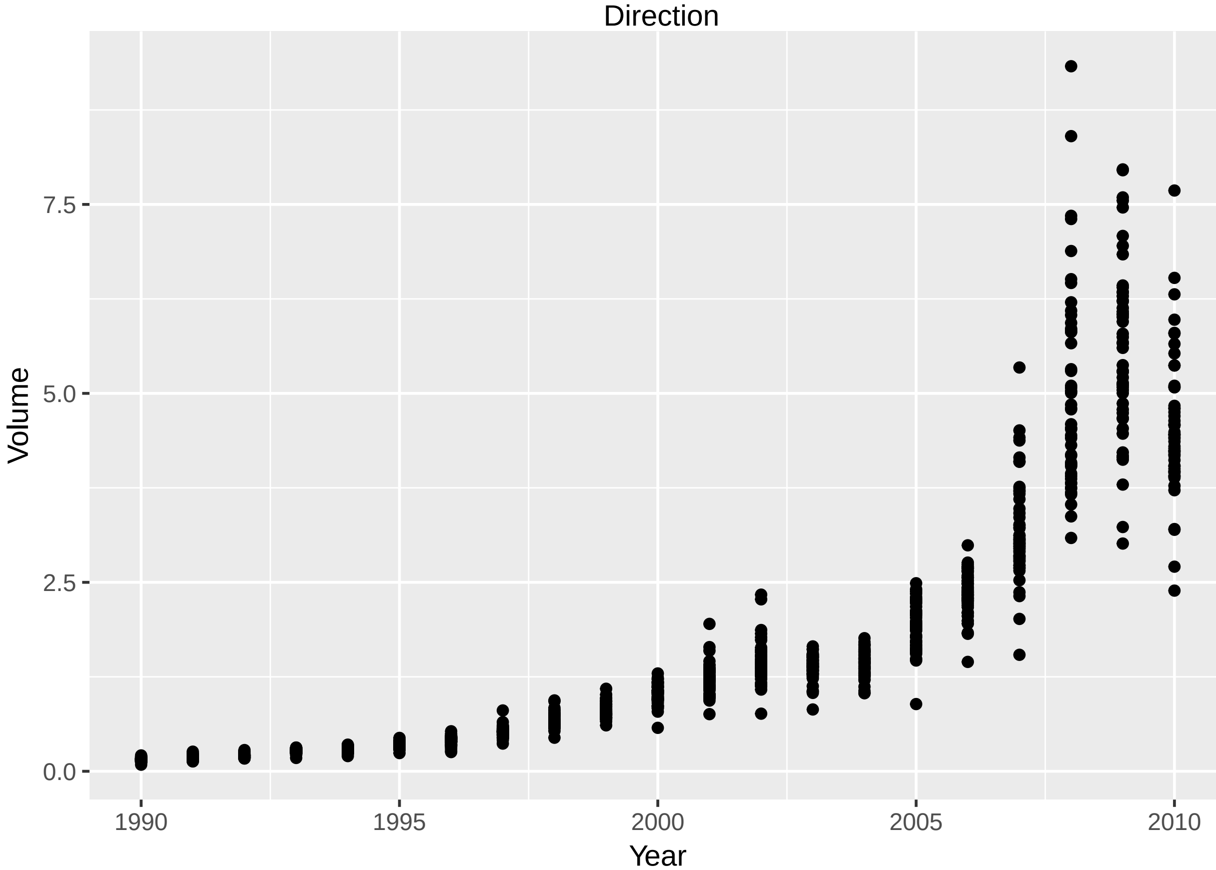
Direction	Count	Percent
Down	484	44%
Up	605	56%

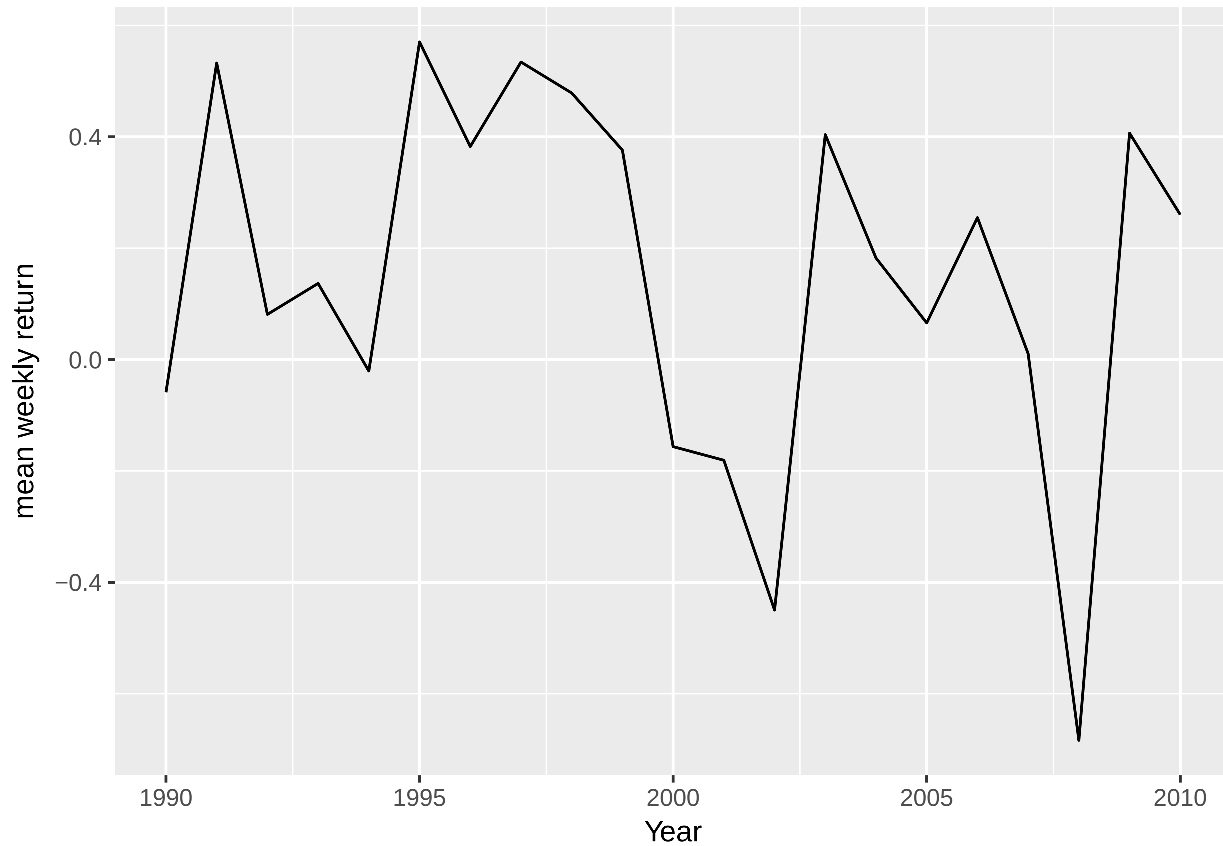
Direction	Mean Lag1	Mean Lag2	Mean Lag3	Mean Lag4	Mean Lag5	Mean Volume	Mean Today
Down	0.2823	-0.0404	0.2076	0.2000	0.1878	1.6085	-1.7466
Up	0.0452	0.3043	0.0989	0.1025	0.1015	1.5475	1.6671

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year	1.0000000	-0.0322893	-0.0333900	-0.0300065	-0.0311279	-0.0305191	0.8419416	-0.0324599
Lag1	-0.0322893	1.0000000	-0.0748531	0.0586357	-0.0712739	-0.0081831	-0.0649513	-0.0750318
Lag2	-0.0333900	-0.0748531	1.0000000	-0.0757209	0.0583815	-0.0724995	-0.0855131	0.0591667
Lag3	-0.0300065	0.0586357	-0.0757209	1.0000000	-0.0753959	0.0606572	-0.0692877	-0.0712436
Lag4	-0.0311279	-0.0712739	0.0583815	-0.0753959	1.0000000	-0.0756750	-0.0610746	-0.0078259
Lag5	-0.0305191	-0.0081831	-0.0724995	0.0606572	-0.0756750	1.0000000	-0.0585174	0.0110127
Volume	0.8419416	-0.0649513	-0.0855131	-0.0692877	-0.0610746	-0.0585174	1.0000000	-0.0330778
Today	-0.0324599	-0.0750318	0.0591667	-0.0712436	-0.0078259	0.0110127	-0.0330778	1.0000000

We see that for the market data, about 56% of the weeks had positive market performance while 44% of the weeks had negative market performance. The value of *today* does not appear to be highly correlated with any of the *lag* or *volume* covariates.


10
0
−10
Value
Down
Up
Direction


7.5
5.0
2.5
0.0
Volume
1990
1995
2000
2005
2010
Year



From the side-by-side boxplots, we see that markets finished up and down with relatively equal magnitudes. Furthermore, we can see that the number of trades has increased exponentially since 1990. Lastly, we can see that the average weekly return has varied from year to year, with several more down years between the years 2000 and 2010 vs 1990 and 2000.

- b.

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

summary(glm.b)
```

```
##
## 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
```

Only the variable Lag2 appears to be a significant predictor of direction.

- c.

As can be seen from the confusion matrix below, the model tends to predict that the market will go up, as 91% of the predicted values were for the market going up. When the model predicts that the market will go up, it is correct 56% of the time. When the model predicts that the market will go down, it is right 53% of the time. Overall, the model has a 56% accuracy rate.The model is overly optimisitc, as the stock market only went up around 56% of the time.

```r
# determine what is being modeled
contrasts(Weekly$Direction)
```

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

```r
# get predictions using 0.5 as the probability threshold
glm.b.preds <- ifelse(predict(glm.b, type = "response") < 0.5, "Down", "Up")
# get confusion matrix of predicted directions and actual directions
table(glm.b.preds, Weekly$Direction)
```

```
##
## glm.b.preds Down  Up
##        Down   54  48
##        Up    430 557
```

- d.

As can be seen from the confusion matrix below, the overall accuracy on the test data is 62%.

```r
# Create training and testing sets
Weekly_train <- dplyr::filter(Weekly, Year %in% 1990:2008)
Weekly_test <- dplyr::filter(Weekly, Year %in% 2009:2010)

# fit model
glm.d <- glm(Direction ~ Lag2,
             data = Weekly_train,
             family = binomial)
# get predicted responses
glm.d.preds <- ifelse(predict(glm.d, newdata = Weekly_test, type = "response") < 0.50,
                      "Down",
                      "Up")
```

```r
# create confusion matrix
table(glm.d.preds, Weekly_test$Direction)
```

```
##
## glm.d.preds Down Up
##        Down    9  5
##        Up     34 56
```

- e.

As can be seen from the confusion matrix below, the predictions for linear discriminant analysis mirror those of logistic regression. The overall accuracy rate is again 62%.

```r
# fit lda model
lda.e <- lda(Direction ~ Lag2,
             data = Weekly_train)

# extract predictions
lda.pred <- predict(lda.e, newdata = Weekly_test)$class

# create confusion matrix
table(lda.pred, Weekly_test$Direction)
```

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

- f.

As can be seen from the confusion matrix below, the predictions for quadratic discriminant analysis are always "Up". The overall accuracy is 59%. It is possible that a cut-off other than 0.50 will yield improved accuracy.

```r
qda.e <- qda(Direction ~ Lag2,
             data = Weekly_train)

# extract predictions
qda.pred <- predict(qda.e, newdata = Weekly_test)$class

# create confusion matrix
table(qda.pred, Weekly_test$Direction)
```

```
##
## qda.pred Down Up
##     Down    0  0
##     Up     43 61
```

- g.

As can be seen from the confusion matrix below, KNN with K = 1 has an overall accuracy of 50%.

```r
# set random number seed for tie breaking
set.seed(1)

# create training and testing sets
train.X <- dplyr::select(Weekly_train, Lag2)
test.X <- dplyr::select(Weekly_test, Lag2)
train.Y <- Weekly_train %>% dplyr::select(Direction) %>% collapse %>% .[[1]]
```

```r
# make predictions
knn.pred <- class::knn(train.X, test.X, train.Y, k = 1)

# create confusion matrix
table(knn.pred, Weekly_test$Direction)
```

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