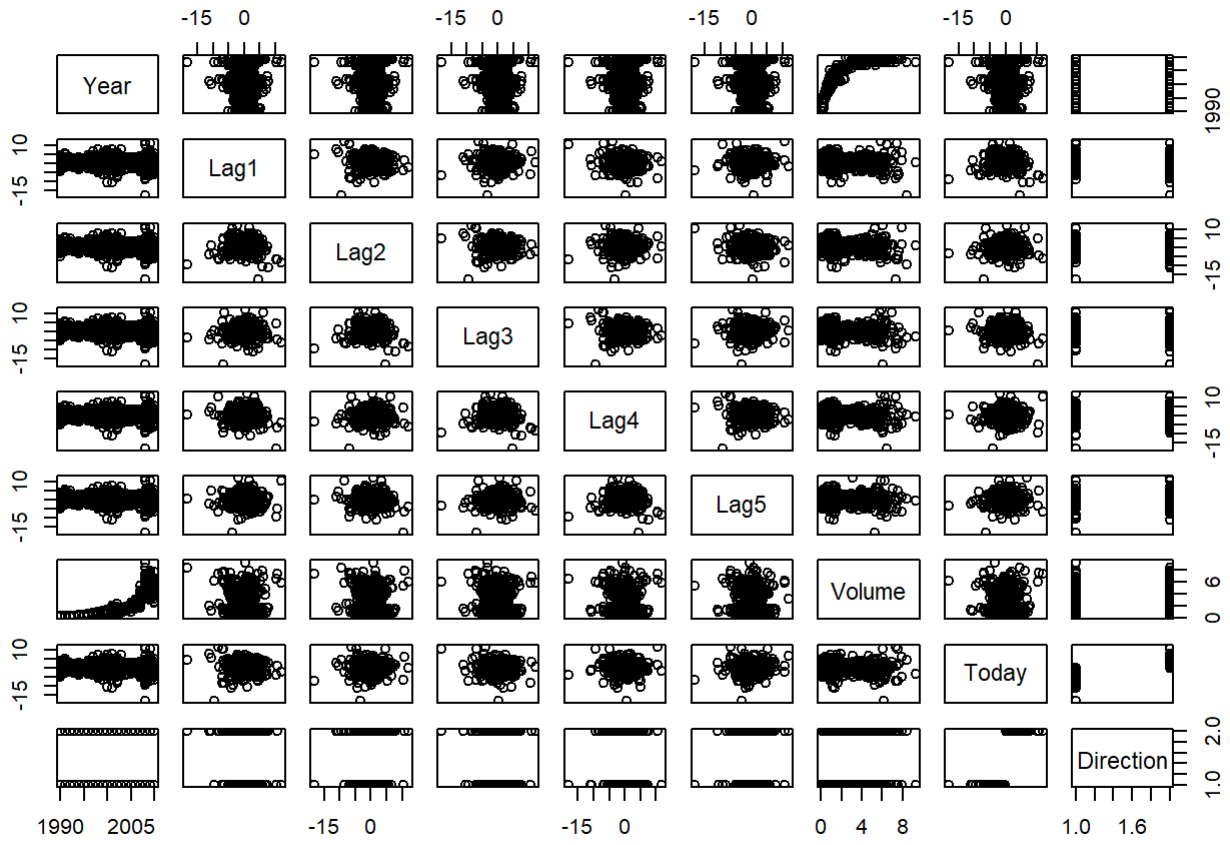
Homework3

```
#a
names(Weekly)
```

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

```
pairs(Weekly)
```




```
Weekly %>% group_by(Direction) %>% summarize(mean = mean(Volume))
```

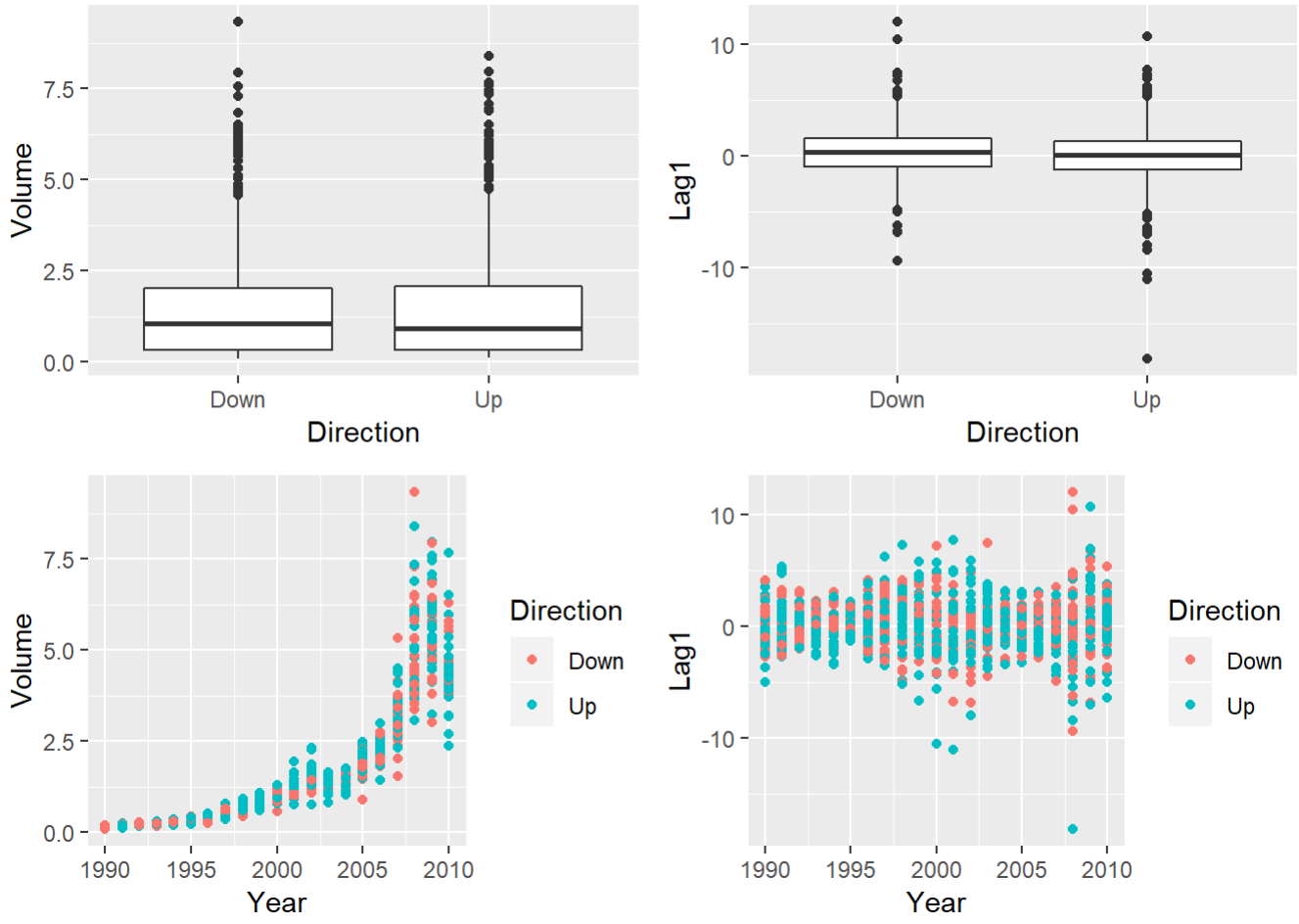
```
## # A tibble: 2 x 2
##   Direction  mean
##   <fct>     <dbl>
## 1 Down       1.61
## 2 Up         1.55
```

```
Weekly %>% group_by(Direction) %>% summarize(mean = mean(Lag1))
```

```
## # A tibble: 2 x 2
##   Direction   mean
##   <fct>      <dbl>
## 1 Down      0.282
## 2 Up        0.0452
```

```
p1 <- Weekly %>% ggplot(aes(Direction, Volume)) + geom_boxplot()
p2 <- Weekly %>% ggplot(aes(Direction, Lag1)) + geom_boxplot()

p3 <- Weekly %>% ggplot(aes(Year, Volume, color = Direction)) + geom_point()
p4 <- Weekly %>% ggplot(aes(Year, Lag1, color = Direction)) + geom_point()
grid.arrange(p1, p2, p3, p4)
```



a. We can see that there does not appear to be much of a difference in the mean of Volume based on whether the market had a positive or negative return as the mean of Volume is 1.61 when Direction is “Down” and 1.55 when Direction is “Up.” A similar pattern is seen for Lag1, which is also confirmed by looking at the boxplots. The scatterplot of Year against Volume shows that the standard devation and mean of Volume appears to increase with Year. Lag1 does not appear to have a relationship with Year, based on the scatterplot. Neither one of the scatterplots shows any relationships that these variables have with direction. The pairs plot shows that volume and year have a curvilinear relationship . There do not appear to be strong relationships among the rest of the variables, but there appears to be an outlier in Lag1.

```
#b
names(Weekly)
```

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

```
a <- table(Weekly$Direction)
a
```

```
##
## Down   Up
##  484  605
```

```
log_model = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family= "b
inomial")
summary(log_model)
```

```
##
## 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
```

b. Only Lag2 has a significant relationship with Direction (z = 2.18, p = 0.03 < 0.05).

```
#c
glm.probs = predict(log_model, type="response")
glm.pred = rep("Down",1089)
glm.pred[glm.probs >.5]=" Up"


table(glm.pred, Weekly$Direction )
```

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

```
correct = (54+557)/1089
correct
```

```
## [1] 0.5610652
```

```
# correct when actually up
557/(557 + 48)
```

```
## [1] 0.9206612
```

```
# correct when actually down
(54)/(54 + 430)
```

```
## [1] 0.1115702
```

c. The logistic regression model predicted 56.1% of all of the data's direction values correctly. It was able to predict "up" values more accurately than "down" values with accuracy rates of 92.1% and 11.2%, respectively. This means that when the market actually did go up, the model predicted this accuraely 92.1% of the time, but only predicted the market going down when it actually did 11.2% of the time.

```
#d
names(Weekly)
```

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

```
train_set = Weekly %>% filter(Year <= 2008)


train_lm = glm(Direction ~ Lag2, data=train_set, family=binomial )
summary(train_lm)
```

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

```
test_set = Weekly %>% filter(Year > 2008)
#up up down down up up up down down na
test.probs = predict (train_lm, newdata = test_set, type="response")
test.pred = rep("Down", 104)
test.pred[test.probs > .5] = "Up"
table(test.pred, test_set$Direction)
```

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

```
correct = (9 + 56)/104
correct
```

```
## [1] 0.625
```

d. The logistic regression model has a 62.5% correct rate for the test data that was composed of 2009 and 2010 data.

```
#e

train_lda = lda(Direction ~ Lag2, data=train_set, family=binomial )
train_lda
```

```
## Call:
## lda(Direction ~ Lag2, data = train_set, family = binomial)
##
## 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
```

```
lda.pred = predict (train_lda, test_set)
lda.class = lda.pred$class
table(lda.class, test_set$Direction)
```

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

```
correct = (9 + 56)/104
correct
```

```
## [1] 0.625
```

```
# predicting up
56/(56+5)
```

```
## [1] 0.9180328
```

```
#predicting down
9/(9+34)
```

```
## [1] 0.2093023
```

e. The LDA predicted 62.5% of the test data's directions correctly. When the market actually went down, the model predicted this 91.8% of the time, but only 20.9% of "down" predictions were correct when the market truly went down.

```
#f
train_qda = qda(Direction ~ Lag2, data=train_set, family=binomial )
train_qda
```

```
## Call:
## qda(Direction ~ Lag2, data = train_set, family = binomial)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
qda.pred = predict (train_qda, test_set)
qda.class = qda.pred$class
table(qda.class, test_set$Direction)
```

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

```
correct = (0 + 61)/104
correct
```

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

f. The QDA predicted 58.7% of the test data's directions correctly.

```
#g
train.X = cbind(train_set$Lag2)
test.X = cbind(test_set$Lag2)
train.Direction = train_set$Direction

set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, test_set$Direction)
```

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

```
correct = (21 + 31)/104
correct
```

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

g. The k-nearest neighbors method only predicted 50% of the test data's directions correctly.

h. LDA and logistic regression provided the best results, as both had an accuracy of 62.5%, followed by QDA at 58.7%.

```
#i
## logistic
train_lm = glm(Direction ~ Lag2 + log(Volume)*Year , data=train_set, family=binomial )
summary(train_lm)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2 + log(Volume) * Year, family = binomial,
##     data = train_set)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.414  -1.260   1.017   1.086   1.441
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -29.349772 131.683953  -0.223   0.8236
## Lag2               0.052996   0.029305   1.808   0.0705 .
## log(Volume)        6.630492  25.830541   0.257   0.7974
## Year               0.014770   0.065817   0.224   0.8224
## log(Volume):Year  -0.003398   0.012898  -0.263   0.7922
## ---
## 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: 1348.9  on 980  degrees of freedom
## AIC: 1358.9
##
## Number of Fisher Scoring iterations: 4
```

```
test.probs = predict (train_lm, newdata = test_set, type="response")
test.pred = rep("Down", 104)
test.pred[test.probs > .5] = "Up"
table(test.pred, test_set$Direction)
```

```
##
## test.pred Down Up
##      Down   18 18
##      Up     25 43
```

```
correct = (18 + 43)/104
correct
```

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

```
# lda
train_lda = lda(Direction ~ Volume + Lag3 + Year, data=train_set, family=binomial )
train_lda
```

```
## Call:
## lda(Direction ~ Volume + Lag3 + Year, data = train_set, family = binomial)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##        Volume       Lag3     Year
## Down 1.266966 0.17080045 1999.295
## Up   1.156529 0.08404044 1998.853
##
## Coefficients of linear discriminants:
##                LD1
## Volume -0.61161805
## Lag3   -0.21468697
## Year   -0.03529623
```

```
lda.pred = predict (train_lda, test_set)
lda.class = lda.pred$class
table(lda.class, test_set$Direction)
```

```
##
## lda.class Down Up
##      Down   36 45
##      Up      7 16
```

```
correct = (36 + 16)/104
correct
```

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

```
## qda
train_qda = qda(Direction ~ log(Volume)*Year, data=train_set, family=binomial )
train_qda
```

```
## Call:
## qda(Direction ~ log(Volume) * Year, data = train_set, family = binomial)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##      log(Volume)     Year log(Volume):Year
## Down  -0.2386605 1999.295        -471.7261
## Up    -0.3281402 1998.853        -650.6346
```

```
qda.pred = predict(train_qda, test_set)
qda.class = qda.pred$class
table(qda.class, test_set$Direction)
```

```
##
## qda.class Down Up
##      Down   42 57
##      Up      1  4
```

```
correct = (42+4)/104
correct
```

```
## [1] 0.4423077
```

```
## knearest

train.X = cbind(sqrt(train_set$Volume),train_set$Lag1)
test.X = cbind(sqrt(test_set$Volume),test_set$Lag1)
train.Direction = train_set$Direction

set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 3)
table(knn.pred, test_set$Direction)
```

```
##
## knn.pred Down Up
##     Down   22 29
##     Up     21 32
```

```
correct = (22+32)/104
correct
```

```
## [1] 0.5192308
```

i. None of these models outperformed the logistic and LDA models from parts (d) - (e) on the test data. The model that worked the best was the logistic regression with predictors log(Volume), Lag2, and Year with an interaction for log(Volume) and Year. However, this was not a significant interaction and none of these predictors were significant. This model had an accuracy rate of 58.7%, followed by the k-nearest with predictors log(Volume) and Year with k = 3 at 51.9%. The QDA model with predictors log(Volume), Year, and their interaction had the worst accuracy rating of 44.2%.