IDS575 HW3

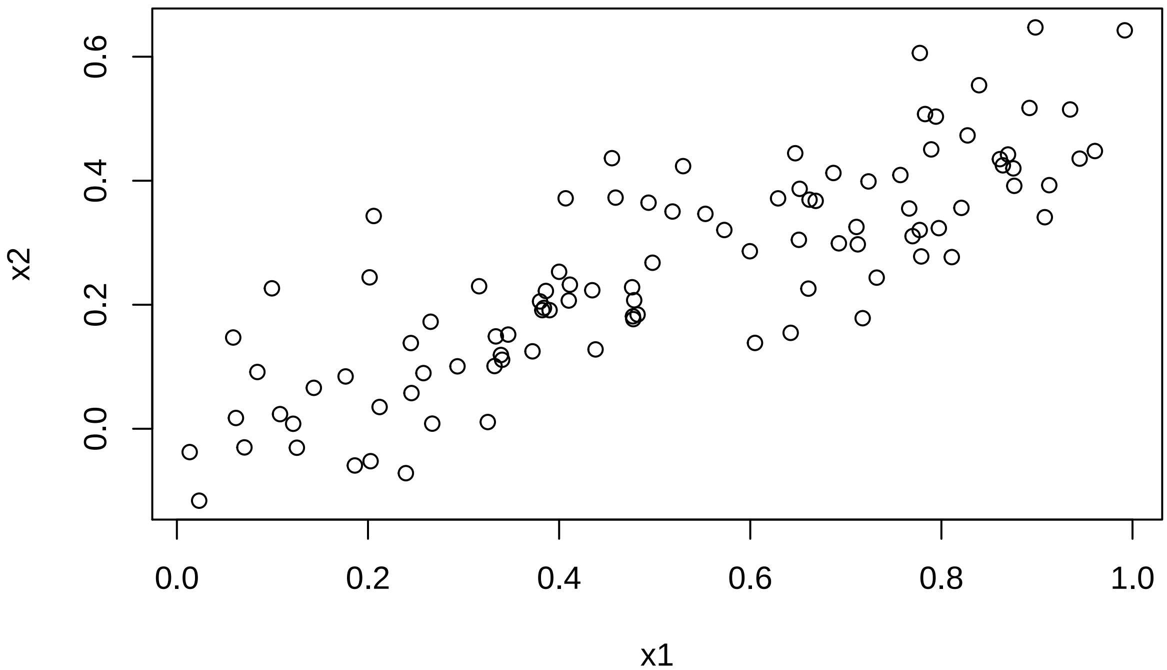
```r
#Ex 14 Page 125
#a
library(stats)
set.seed(1)
x1=runif(100)
x2=0.5*x1+rnorm(100)/10
y=2+2*x1+0.3*x2+rnorm(100)

# in that particular case for model "y", the linear coefficients would be:
# beta0 = 2, beta1=2, beta2 = 0.3

#b
cor(x1,x2)
```

```
## [1] 0.8351212
```

```r
plot(x1,x2)
```




```r
#the correlation between x1 and x2 is 0.8351.
#from the plot we can observe that x1 and x2 are highly correlated.
#the data points are spread and look like a upwards diagonal line

#c
linear_regression = lm(y~x1+x2)
summary(linear_regression)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

```r
# beta0 = 2.13 , beta1 = 1.44 , beta2 = 1.01
#the least square regression model looks as: y = 2.13 + 1.44 * X1 + 1.01 * X2
# we may reject the null hypothesis for B1, however we can´t do that for
#B2, since the p-value is greater than 0.05
#d
linear_regressionx1 = lm(y~x1)
summary(linear_regressionx1)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

```r
# the model using only x1 as a predictor would look: y = 2.112 + 1.976 * X1
# we see that the coefficient for x1 is very different from the previous
#model. Based on the p-value we see that x1 is highly significant
# (p-value is smaller than 0.05)

#e
linear_regressionx2 = lm(y~x2)
summary(linear_regressionx2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

```
#the model using only x2 as a predictors woul look: y = 2.39 + 2.90 * X2
#In this case we see that the p-value is also very small (lower than 0.05).
# That means that x2 is higly significnat. That differs from the first model
# in which we included both predictors together.

#f
#No, the results not contradict each other. Both predictors x1 and x2
#are hoghly correlated, and this is a case of collinearity. It is quite
#difficult to determine how each predictor separetely is associated with
#the responde variable. Since collinearity reduces the accuracy, it causes
#the standard error of beta1 to grow.
```

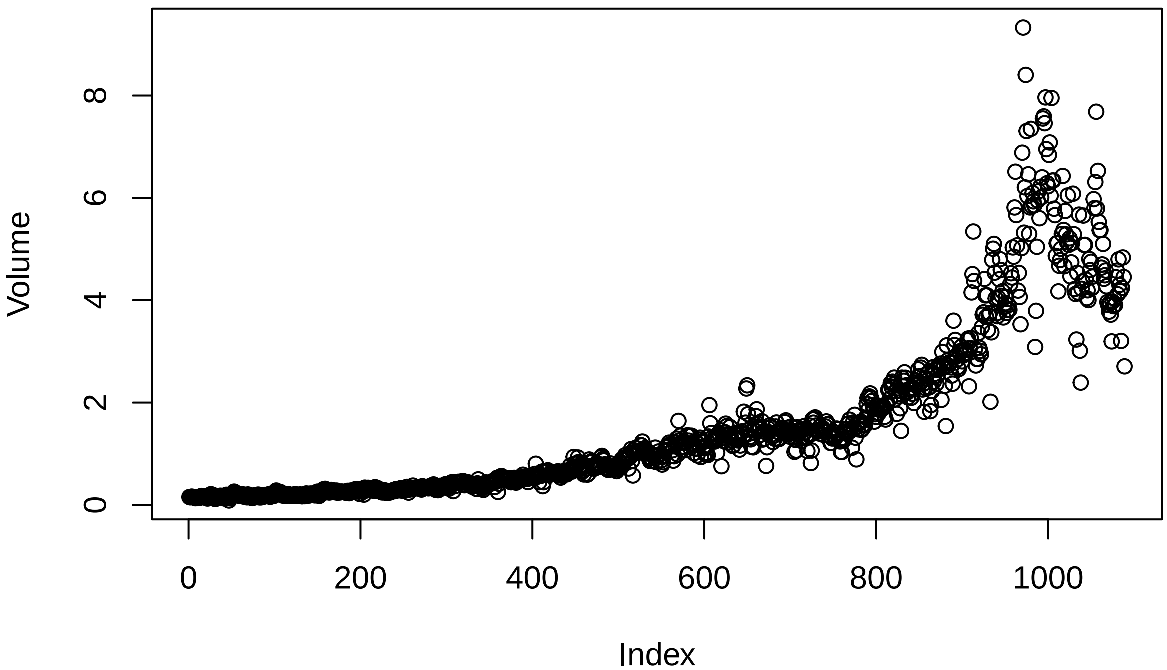
```
#Ex 10 Page 171
#a
library(ISLR)
attach(Weekly)
summary(Weekly)
```

```
##       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
cor(Weekly[, -9])
```

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

```r
#we do not appreciate any correlation between lags, most of them are close to 0
#we just see some correlation between volume and year
plot(Volume)
```




```r
#b
log_model = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)
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
```

```r
#we see that Lag2 is the only predictor that is statistcally significant since
# p value is smaller than 0.05

#c
probabilities <- predict(log_model, type = "response")
predict_logmodel <- rep("Down", length(probabilities))
predict_logmodel[probabilities > 0.5] <- "Up"
table(predict_logmodel, Direction)
```

```
##                 Direction
## predict_logmodel Down  Up
##             Down   54  48
##             Up    430 557
```

```r
#from the confusion matrix we can calculate the percentage of accuracy in the
# training dataset. That is (54+557)/1089 = 56.106%
#When predicting down the model is right 54/(54+48) = 52.94%
#When predicting up the model is right 430(430+557) = 43.56%

#d
train <- (Year < 2009)
log_model2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
summary(log_model2)
```

```
##
## 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
Weekly_20092010 <- Weekly[!train, ]
Direction_20092010 <- Direction[!train]
probabilities2<- predict(log_model2, Weekly_20092010, type = "response")
predict_logmodel2 <- rep("Down", length(probabilities2))
predict_logmodel2[probabilities2 > 0.5] <- "Up"
table(predict_logmodel2, Direction_20092010)
```

```
##                  Direction_20092010
## predict_logmodel2 Down Up
##              Down    9  5
##              Up     34 56
```

```r
#total accuracy: (9+56)/104 = 62.5%
#accuracy when predicting up: 56/(56+34) = 62.22%
#accuracy when predicting down: 9/(9+5) = 64.28%

#e
library(MASS)
lda_model <- lda(Direction ~ Lag2, data = Weekly, subset = train)
lda_model
```

```
## 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
predict_lda <- predict(lda_model, Weekly_20092010)
table(predict_lda$class, Direction_20092010)
```

```
##       Direction_20092010
##        Down Up
##   Down    9  5
##   Up     34 56
```

```
#total accuracy: (9+56)/104 = 62.5%
#accuracy when predicting up: 56/(56+34) = 62.22%
#accuracy when predicting down: 9/(9+5) = 64.28%
#accuracy is the same as for logistic regression

#f
qda_model <- qda(Direction ~ Lag2, data = Weekly, subset = train)
qda_model
```

```
## 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
```

```
predict_qda <- predict(qda_model, Weekly_20092010)
table(predict_qda$class, Direction_20092010)
```

```
##       Direction_20092010
##        Down Up
##   Down    0  0
##   Up     43 61
```

```
#this model only predicts up movement
#total accuracy 61/(43+61) = 58.65%

#g
library(class)
```

```
## Warning: package 'class' was built under R version 3.5.2
```

```
train.X <- as.matrix(Lag2[train])
test.X <- as.matrix(Lag2[!train])
train.Direction <- Direction[train]
set.seed(1)
predict_knn <- knn(train.X, test.X, train.Direction, k = 1)
table(predict_knn, Direction_20092010)
```

```
##            Direction_20092010
## predict_knn Down Up
##        Down   21 30
##        Up     22 31
```

```
#total accuracy: (21+31)/104 = 50%
#total accuracy predicting up: 31/54 = 58.49%
#total accuracy predicting down: 21/51 = 41.17%
```

```r
#h
#Comparing the accuracy of the model we observe that logistic regression and
#LDA have the highest one.

#i
# Logistic regression with Lag2 and Lag1
log_model3 <- glm(Direction ~ Lag2 + Lag1, data = Weekly, family = binomial, subset = train)
probabilitites3 <- predict(log_model3, Weekly_20092010, type = "response")
pred_logmodel3 <- rep("Down", length(probabilitites3))
pred_logmodel3[probabilitites3 > 0.5] = "Up"
table(pred_logmodel3, Direction_20092010)
```

```
##                Direction_20092010
## pred_logmodel3 Down Up
##           Down    7  8
##           Up     36 53
```

```r
mean(pred_logmodel3 == Direction_20092010)
```

```
## [1] 0.5769231
```

```r
lda_model2<- lda(Direction ~ Lag2 + Lag1, data = Weekly, subset = train)
pred_ldamodel2 <- predict(lda_model2, Weekly_20092010)
mean(pred_ldamodel2$class == Direction_20092010)
```

```
## [1] 0.5769231
```

```r
qda_model2 <- qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)
pred_qda2 <- predict(qda_model2, Weekly_20092010)
table(pred_qda2$class, Direction_20092010)
```

```
##        Direction_20092010
##         Down Up
##   Down    12 13
##   Up      31 48
```

```r
mean(pred_qda2$class == Direction_20092010)
```

```
## [1] 0.5769231
```

```r
# KNN k =10
pred_knn2 <- knn(train.X, test.X, train.Direction, k = 10)
table(pred_knn2, Direction_20092010)
```

```
##           Direction_20092010
## pred_knn2 Down Up
##      Down   17 18
##      Up     26 43
```

```r
mean(pred_knn2 == Direction_20092010)
```

```
## [1] 0.5769231
```

```r
pred_knn3 <- knn(train.X, test.X, train.Direction, k = 100)
table(pred_knn3, Direction_20092010)
```

```
##           Direction_20092010
## pred_knn3 Down Up
```

```
##      Down    9 12
##      Up     34 49
```

```r
mean(pred_knn3== Direction_20092010)
```

```
## [1] 0.5576923
```

```r
#from the last applied models we observe that logistic regression and LDA
#are the best performers
```

```r
#Ex 5 Page 169
#a
#If the Bayes decision boundary is linear, we expect QDA to perform better on the training set because
#its higher flexiblity may yield a closer fit. On the test set, we expect LDA to perform better than QD
#because QDA could overfit the linearity on the Bayes decision boundary.

#b
#If the Bayes decision bounary is non-linear,
#we expect QDA to perform better both on the training and test sets.

#c
#QDA (which is more flexible than LDA and so has higher variance) is recommended
#if the training set is very large, so that the variance of the classifier is not a major concern.

#d
#False. With fewer sample points, the variance from using a more flexible method such as QDA, may lead
#which in turns may lead to an inferior test error rate.
```

```r
#Ex 6 Page 169
#In order to solve this problem we will make use of the sigmoid function
#The probability of getting an A for a student who studied 40 hours and has a
#GPA of 3.5 is:
p1 = exp(-6+0.05*40+1*3.5)/(1+exp(-6+0.05*40+1*3.5))

# 0.5 = exp(-6+0.05*Hours+1*3.5)/(1+exp(-6+0.05*Hours+1*3.5))
#Solving the equation for Hours:
#exp(-6+0.05*Hours+3.5) = 1
Hours = 2.5/0.05
#the number of hours the previous student would have to study to have
# 50% chance of getting an A is 50.
```