STEVENS INSTITUTE OF TECHNOLOGY
FE582 Homework 3
Instructor: Dragos Bozdog

Assignment 3

Student: Aidana Bekboeva

Contents

1 Problem 1

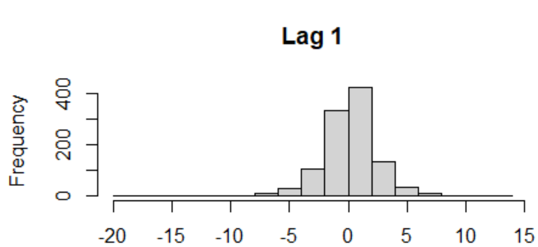
1.1 a) Produce some numerical and graphical summaries

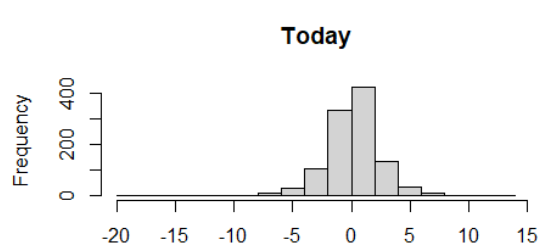
```r
weekly <- read.csv("C:\\Users\\Aidana Bekboeva\\Desktop\\STEVENS\\2. FA
   582 - Financial Data Science\\Assignment 3\\HW3_data\\Weekly.csv")

summary(weekly)

hist(weekly$Lag1, xlab = " ", main = "Lag 1")
hist(weekly$Lag2, xlab = " ", main = "Lag2 2")
hist(weekly$Today, xlab = " ", main = "Today")

ggplot(data = weekly, mapping = aes(x = Volume, y = Year), main = "Volume
   & Years ggplot") + geom_point()
```

Result:

```
> 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
 Length:1089
 Class :character
 Mode  :character
>
```

Listing 1: R output





Two graphs on the previous page are examples of distributions of variables "Lag1" through "Lag5" + the variable "Today", all of them follow the same distribution. Let's examine the correlations between all variables in the dataset:

```r
res <- cor(weekly[,-9])
round(res, 2)
```

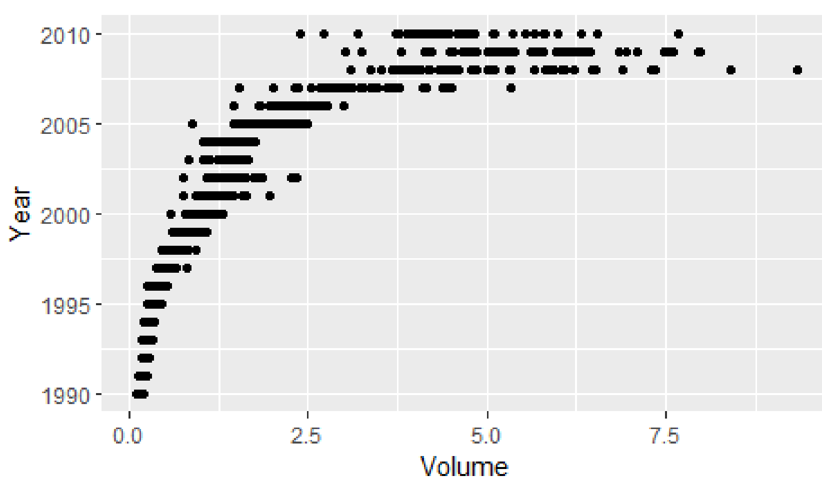
Listing 2: R code

Result:

```
> round(res, 2)
        Year  Lag1  Lag2  Lag3  Lag4  Lag5 Volume Today
Year    1.00 -0.03 -0.03 -0.03 -0.03 -0.03   0.84 -0.03
Lag1   -0.03  1.00 -0.07  0.06 -0.07 -0.01  -0.06 -0.08
Lag2   -0.03 -0.07  1.00 -0.08  0.06 -0.07  -0.09  0.06
Lag3   -0.03  0.06 -0.08  1.00 -0.08  0.06  -0.07 -0.07
Lag4   -0.03 -0.07  0.06 -0.08  1.00 -0.08  -0.06 -0.01
Lag5   -0.03 -0.01 -0.07  0.06 -0.08  1.00  -0.06  0.01
Volume  0.84 -0.06 -0.09 -0.07 -0.06 -0.06   1.00 -0.03
Today  -0.03 -0.08  0.06 -0.07 -0.01  0.01  -0.03  1.00
>
```

Listing 3: R output

We can see that that the only pair of variables that appear to have a correlation, are Year and Volume, with a positive correlation of 0.84.



By plotting the data we see that Volume is increasing over time. In other words, the average number of shares traded daily increased from 1990 to 2010.

1.2 b) Logistic regression

```r
weekly$Direction <- as.factor(weekly$Direction)
class(weekly$Direction)

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

Listing 4: R code

Result:

```
> summary(weekly_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

>
```

Listing 5: R output

The only variable that was statistically significant was Lag2, at the level of significance 0.05. The other variables fail to reject the null hypothesis.

1.3 c) Confusion matrix

```
weeklylogprob= predict(weekly_fit, type='response')
pred_weeklylog =rep("Down", length(weeklylogprob))
pred_weeklylog[weeklylogprob > 0.5] = "Up"
table(pred_weeklylog, Direction)
```

Listing 6: R code

Result:

```
pred_weeklylog Down  Up
          Down   54  48
          Up    430 557
```

Listing 7: R output

In order to see the mistakes made by the logistic regression, we can compute the percentage of correct predictions using the values from the result matrix.

$\frac{54+557}{54+48+430+557} = 0.5611 = 56\%$

We can further the investigation and calculate if the system makes mistakes when calculating Down or Up weekly trends:

$\frac{557}{48+557} = 0.9207 = 92\%$ Correctness for Up trends

$\frac{54}{54+430} = 0.1115 = 11\%$ Correctness for Down trends

1.4 d) Fitting the logistic regression model

```
t = (Year<2009)

weekly_t <-weekly[!t,]
weekly_fit<-glm(Direction~Lag2, data=weekly,family=binomial, subset=t)
weeklylogprob= predict(weekly_fit, weekly_t, type = "response")

pred_weeklylog = rep("Down", length(weeklylogprob))
pred_weeklylog[weeklylogprob > 0.5] = "Up"
Direction_t = Direction[!t]
table(pred_weeklylog, Direction_t)

mean(pred_weeklylog == Direction_t)
```

Listing 8: R code

Result:

This result shows us that the model correctly predicted weekly trends at a rate of 62% when the data is divided into two groups.

```
> table(pred_weeklylog, Direction_t)
              Direction_t
pred_weeklylog Down Up
          Down    9  5
          Up     34 56
> mean(pred_weeklylog == Direction_t)
[1] 0.625
```

Listing 9: R output

1.5 e) Repeat d) using LDA.

```
weeklylda.fit<-lda(Direction~Lag2, data=weekly,family=binomial, subset=t)
weeklylda.pred<-predict(weeklylda.fit, weekly_t)
table(weeklylda.pred$class, Direction_t)
mean(weeklylda.pred$class==Direction_t)
```

Listing 10: R code

```
> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down    9  5
  Up     34 56
> mean(weeklylda.pred$class==Direction_t)
[1] 0.625
```

Listing 11: R output

Results using LDA are the same as in d)

1.6 f) Repeat d) using QDA.

```
weeklylda.fit<-qda(Direction~Lag2, data=weekly,family=binomial, subset=t)
weeklylda.pred<-predict(weeklylda.fit, weekly_t)
table(weeklylda.pred$class, Direction_t)
mean(weeklylda.pred$class==Direction_t)
```

Listing 12: R code

```
> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down    0  0
  Up     43 61
> mean(weeklylda.pred$class==Direction_t)
[1] 0.5865385
```

Listing 13: R output

1.7 g) Repeat d) using KNN with K = 1

```
week_t = as.matrix(weekly$Lag2[t])
weekly_test = as.matrix(weekly$Lag2[!t])
train_Direction = weekly$Direction[t]
set.seed(1)
weekly_knnpred=knn(week_t, weekly_test, train_Direction, k=1)
table(weekly_knnpred, Direction_t)
mean(weekly_knnpred == Direction_t)
```

Listing 14: R code

```
> table(weekly_knnpred, Direction_t)
              Direction_t
weekly_knnpred Down Up
          Down   21 30
          Up     22 31
> mean(weekly_knnpred == Direction_t)
[1] 0.5
```

Listing 15: R output

1.8 h) Best results on this data?

The methods that have the highest accuracy rates are the Logistic Regression and LDA (Linear Discriminant Analysis), both having rates of 62.5%.

1.9 i) Experiment with different combinations of predictors

```
#1. Logistic regression
weekly_t <-weekly[!t,]
weekly_fit<-glm(Direction~Lag2:Lag1+Lag2, data=weekly,family=binomial,
   subset=t)
weeklylogprob= predict(weekly_fit, weekly_t, type = "response")
pred_weeklylog = rep("Down", length(weeklylogprob))
pred_weeklylog[weeklylogprob > 0.5] = "Up"
Direction_t = Direction[!t]
table(pred_weeklylog, Direction_t)
mean(pred_weeklylog == Direction_t)
```

Listing 16: R code

```
# Results for Logistic regression with interaction Lag2, Lag1

> table(pred_weeklylog, Direction_t)
              Direction_t
pred_weeklylog Down Up
          Down    3  3
          Up     40 58
> mean(pred_weeklylog == Direction_t)
```

```
[1] 0.5865385

# Results for Logistic regression with interaction Lag2, Lag4

> table(pred_weeklylog, Direction_t)
              Direction_t
pred_weeklylog Down Up
          Down    3  4
          Up     40 57
> mean(pred_weeklylog == Direction_t)
[1] 0.5769231
```

Listing 17: R output

```
#2. LDA
weeklylda.fit<-lda(Direction~Lag2:Lag1+Lag2, data=weekly,family=binomial,
   subset=t)
weeklylda.pred<-predict(weeklylda.fit, weekly_t)
table(weeklylda.pred$class, Direction_t)
mean(weeklylda.pred$class==Direction_t)
```

Listing 18: R code

```
# Results for LDA with interaction Lag2, Lag1

> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down    3  3
  Up     40 58
> mean(weeklylda.pred$class==Direction_t)
[1] 0.5865385

# Results for LDA with interaction Lag2, Lag3

> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down    9  5
  Up     34 56
> mean(weeklylda.pred$class==Direction_t)
[1] 0.625
```

Listing 19: R output

```
#3. QDA
weeklylda.fit<-qda(Direction~Lag2:Lag1+Lag2, data=weekly,family=binomial,
   subset=t)
weeklylda.pred<-predict(weeklylda.fit, weekly_t)
table(weeklylda.pred$class, Direction_t)
mean(weeklylda.pred$class==Direction_t)
```

Listing 20: R code

```
# Results for QDA with interaction Lag2, Lag1

> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down   24 37
  Up     19 24
> mean(weeklylda.pred$class==Direction_t)
[1] 0.4615385

# Results for QDA with interaction Lag2, Lag3

> table(weeklylda.pred$class, Direction_t)
      Direction_t
       Down Up
  Down    6  7
  Up     37 54
> mean(weeklylda.pred$class==Direction_t)
[1] 0.5769231
```

Listing 21: R output

```
#4. KNN with K = 5, 10, 15
week_t = as.matrix(Lag2[t])
weekly_test = as.matrix(Lag2[!t])
train_Direction = Direction[t]
set.seed(1)
weekly_knnpred=knn(week_t, weekly_test, train_Direction, k=10)
table(weekly_knnpred, Direction_t)
mean(weekly_knnpred == Direction_t)
```

Listing 22: R code

```
# K = 5

> table(weekly_knnpred, Direction_t)
              Direction_t
weekly_knnpred Down Up
          Down   16 21
          Up     27 40
> mean(weekly_knnpred == Direction_t)
[1] 0.5384615

# K = 10

> table(weekly_knnpred, Direction_t)
              Direction_t
weekly_knnpred Down Up
          Down   17 21
          Up     26 40
> mean(weekly_knnpred == Direction_t)
[1] 0.5480769
```

Listing 23: R output

2 Problem 2

2.1 a) Create a binary variable, mpg01, ...

```
dfauto <- read.csv("C:\\Users\\Aidana Bekboeva\\Desktop\\STEVENS\\2. FA 582 - Financial Data Science\\Assignment 3\\HW3_data\\Auto.csv")
summary(dfauto)

dfauto$mpg01 <- ifelse(dfauto$mpg > median(dfauto$mpg), "1", "0")
summary(dfauto)
```

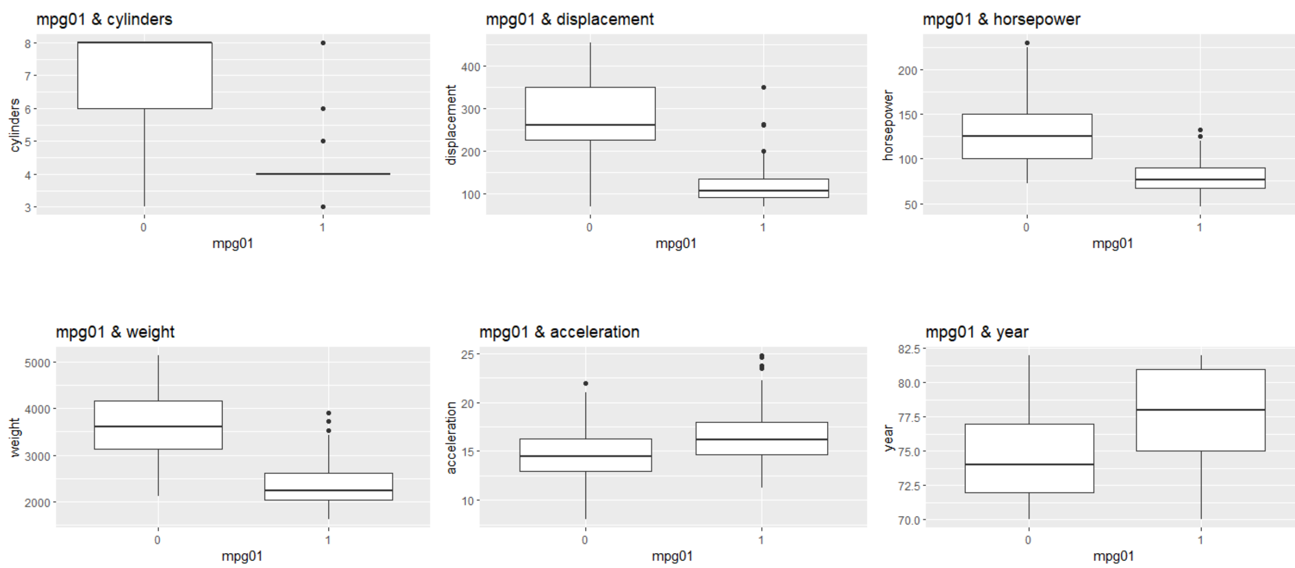
Listing 24: R code

2.2 b) Explore the data graphically

```
install.packages("ggpubr")
library("ggpubr")

ggplot(dfauto,
       aes(x = mpg01,
           y = cylinders)) +
  geom_boxplot() +
  labs(title = "mpg01 & cylinders")

ggplot(dfauto,
       aes(x = mpg01,
           y = displacement)) +
  geom_boxplot() +
  labs(title = "mpg01 & displacement")

ggplot(dfauto,
       aes(x = mpg01,
           y = horsepower)) +
  geom_boxplot() +
  labs(title = "mpg01 & horsepower")

ggplot(dfauto,
       aes(x = mpg01,
           y = weight)) +
  geom_boxplot() +
  labs(title = "mpg01 & weight")

ggplot(dfauto,
       aes(x = mpg01,
           y = acceleration)) +
  geom_boxplot() +
  labs(title = "mpg01 & acceleration")

ggplot(dfauto,
       aes(x = mpg01,
           y = year)) +
  geom_boxplot() +
```

```r
  labs(title = "mpg01 & year")

ggplot(dfauto,
       aes(x = mpg01,
           y = origin)) +
  geom_boxplot() +
  labs(title = "mpg01 & origin")

dfauto$mpg01 <- ifelse(dfauto$mpg > median(dfauto$mpg), 1, 0)
res <- cor(dfauto[,-9])
round(res, 2)
```

Listing 25: R code



Plotting the relationships between each of these variables, we can notice that the strongest relationships with a variable mpg01 are hold with variables Cylinders, Displacement, Horsepower, and Weight. In order to see the actual correlation indexes, we will refer to the correlation matrix:

```
> res <- cor(dfauto[,-9])
> round(res, 2)
...
                mpg01
mpg              0.84
cylinders       -0.76
displacement    -0.75
horsepower      -0.67
weight          -0.76
acceleration     0.35
year             0.43
origin           0.51
mpg01            1.00
```

Listing 26: R output

We can see that these variables appear to correlate negatively with this variable: given the values of -0.76, -0.75, -0.67, and -0.76 respectively.

2.3 c) Split the data into a training set and a test set.

```
train <- (dfauto$year %% 2 == 0)
train_set <- dfauto[train,]
test_set <- dfauto[-train,]
```

Listing 27: R code

2.4 d) Perform LDA on the training data

```
dfautolda_fit <- lda(mpg01~displacement+horsepower+weight+year+cylinders+
    origin, data = train_set)
dfautolda_pred <- predict(dfautolda_fit, test_set)
table(dfautolda_pred$class, test_set$mpg01)
mean(dfautolda_pred$class != test_set$mpg01)
```

Listing 28: R code

```
> table(dfautolda_pred$class, test_set$mpg01)

      0   1
  0 169   7
  1  26 189
> mean(dfautolda_pred$class != test_set$mpg01)
[1] 0.08439898
```

Listing 29: R output

2.5 e) Perform QDA on the training data

```
dfautolda_fit <- qda(mpg01~displacement+horsepower+weight+year+cylinders+
    origin, data = train_set)
dfautolda_pred <- predict(dfautolda_fit, test_set)
table(dfautolda_pred$class, test_set$mpg01)
mean(dfautolda_pred$class != test_set$mpg01)
```

Listing 30: R code

```
> table(dfautolda_pred$class, test_set$mpg01)

      0   1
  0 176  20
  1  19 176
> mean(dfautolda_pred$class != test_set$mpg01)
[1] 0.09974425
```

Listing 31: R output

2.6 f) Perform logistic regression on the training data

```
#f) Perform logistic regression on the training data
dfauto_fit <- glm(mpg01~displacement+horsepower+weight+year+cylinders+
    origin, data=train_set,family=binomial)
dfauto_probs = predict(dfauto_fit, test_set, type = "response")
dfauto_pred = rep(0, length(dfauto_probs))
dfauto_pred[dfauto_probs > 0.5] = 1
table(dfauto_pred, test_set$mpg01)
mean(dfauto_pred != test_set$mpg01)
```

Listing 32: R code

```
> table(dfauto_pred, test_set$mpg01)

dfauto_pred    0    1
          0  174   12
          1   21  184
> mean(dfauto_pred != test_set$mpg01)
[1] 0.08439898
```

Listing 33: R output

2.7 g) Perform KNN on the training data, K = 1, 5, 10, 15, 50

```
knn_training <- cbind(dfauto$displacement,dfauto$horsepower,
                      dfauto$weight,dfauto$cylinders,
                      dfauto$year, dfauto$origin)[train,]
knn_test=cbind(dfauto$displacement,dfauto$horsepower,
               dfauto$weight,dfauto$cylinders,
               dfauto$year, dfauto$origin)[-train,]
set.seed(1)
dfauto_knnpred=knn(knn_training,knn_test,train_set$mpg01, k=1) #k=5, k=10,
    k=15, k=50)
mean(dfauto_knnpred != test_set$mpg01)
```

Listing 34: R code

```
# K = 1            [1] 0.07161125
# K = 5            [1] 0.112532
# K = 10           [1] 0.1227621
# K = 15           [1] 0.1355499

# K = 50           [1] 0.1176471
# K = 100          [1] 0.1176471
```

Listing 35: R output

It appears that as the value of K increases, so does the error rate for this particular model. However, when the value of K reaches a large number the error rate stops exceeding the value of 0.1176471, which becomes a constant. The best performing value of K is 1.