Chapter 4

Book Work

Simple Logistic Regression

```r
data.default <- data.table(ISLR::Default)[, dflt := ifelse(default == "Yes", 1, 0)]

summary(model1 <- glm(dflt ~ balance, data = data.default, family = "binomial"))
```

```

Call:
glm(formula = dflt ~ balance, family = "binomial", data = data.default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

```r
predict(model1, newdata = data.frame( balance = c(1000, 2000) ), type = "response")
```

```
          1           2
0.005752145 0.585769370
```

```r
data.default[, is_student := ifelse(student == "Yes", 1, 0)]

summary(model2 <- glm(dflt ~ is_student, data = data.default, family = "binomial"))
```

```

Call:
glm(formula = dflt ~ is_student, family = "binomial", data = data.default)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2970  -0.2970  -0.2434  -0.2434   2.6585

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
is_student   0.40489    0.11502    3.52 0.000431 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 2908.7  on 9998  degrees of freedom
AIC: 2912.7

Number of Fisher Scoring iterations: 6
```

```r
predict(model2, newdata = data.frame( is_student = c(1, 0) ), type = "response")
```

```
         1          2 
0.04313859 0.02919501 
```

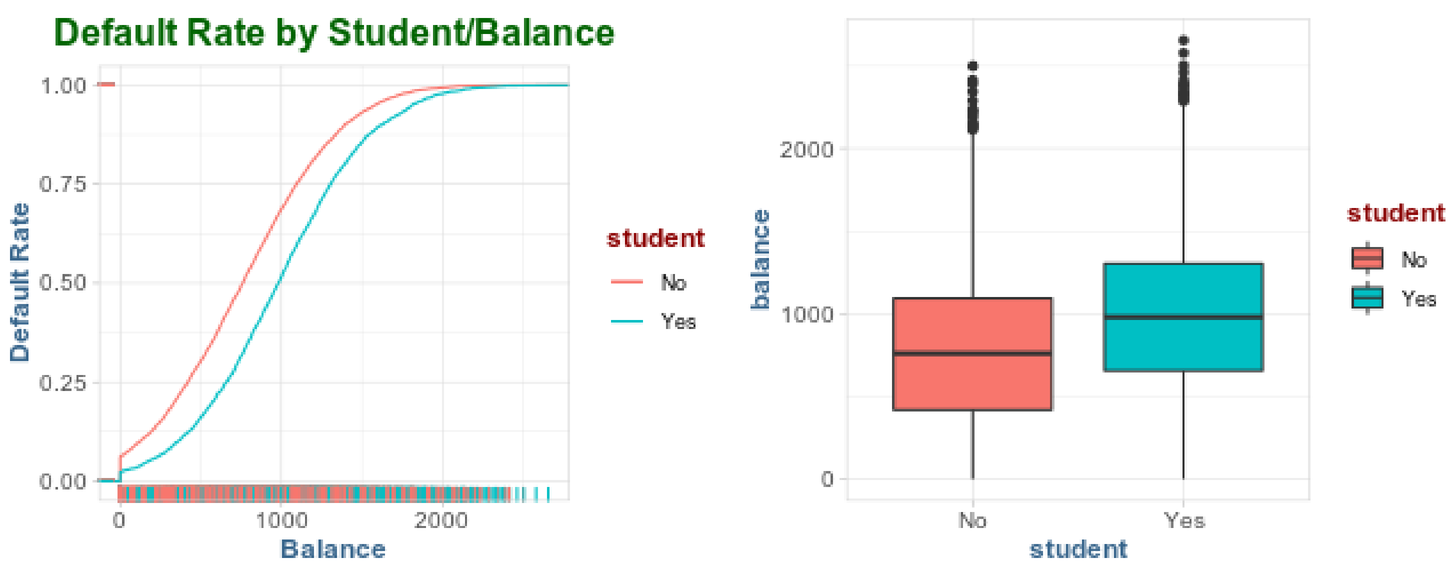
Multiple Logistic Regression

```r
summary(model3 <- glm(dflt ~ balance + is_student, data = data.default, family = "binomial"))
```

```
Call:
glm(formula = dflt ~ balance + is_student, family = "binomial",
    data = data.default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4578  -0.1422  -0.0559  -0.0203   3.7435

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
is_student  -7.149e-01  1.475e-01  -4.846 1.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.7  on 9997  degrees of freedom
AIC: 1577.7

Number of Fisher Scoring iterations: 8
```

```r
p1 <- ggplot(data.default, aes(balance, dflt, color = student)) +
   stat_ecdf() +
   geom_rug(aes(balance, dflt)) +
   labs(x = "Balance", y = "Default Rate", title = "Default Rate by Student/Balance")

p2 <- ggplot(data.default, aes(student, balance, fill = student)) +
   geom_boxplot()

grid.arrange(p1, p2, nrow = 1)
```




```r
predict(model3, newdata =
          data.frame( balance = c(1500, 1500),
                      is_student = c(1, 0) ),
        type = "response")
```

```
         1          2
0.05430945 0.10504923
```

R Lab

```r
Smarket <- as.data.table(ISLR::Smarket)

names(Smarket)
```

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

```r
dim(Smarket)
```

```
[1] 1250    9
```

```r
summary(Smarket)
```

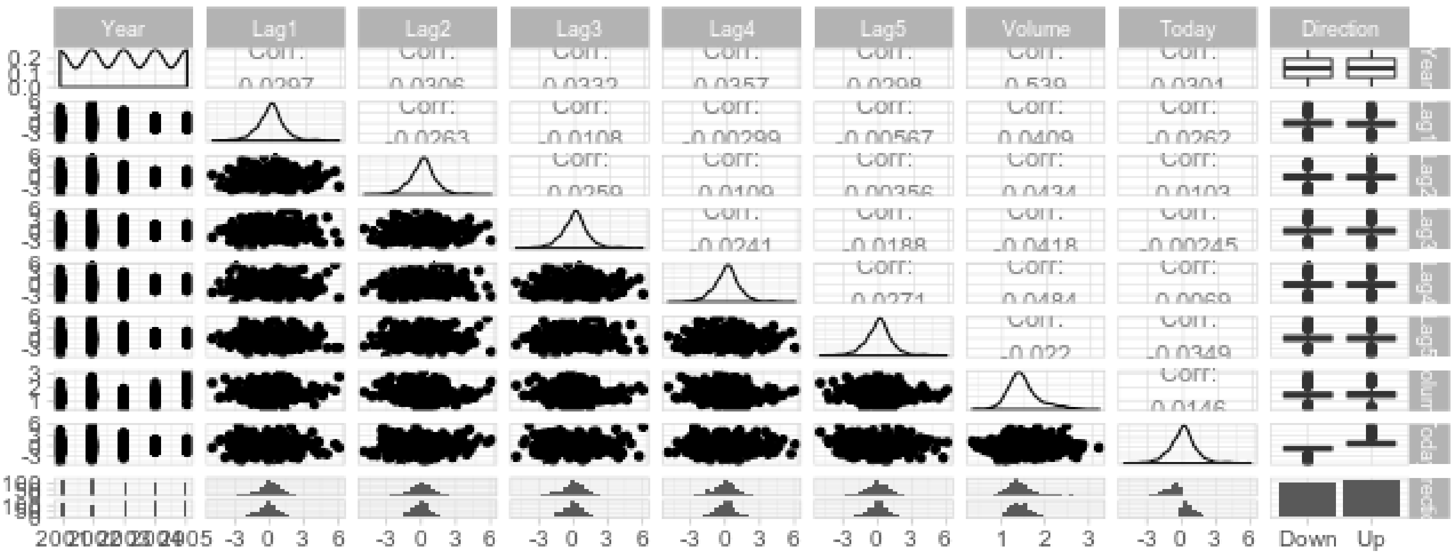
```
      Year           Lag1                Lag2                Lag3
 Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000
 1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000
 Median :2003   Median : 0.039000   Median : 0.039000   Median : 0.038500
 Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716
 3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750
 Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000
      Lag4                Lag5              Volume           Today
 Min.   :-4.922000   Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000
 1st Qu.:-0.640000   1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500
 Median : 0.038500   Median : 0.03850   Median :1.4229   Median : 0.038500
 Mean   : 0.001636   Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138
 3rd Qu.: 0.596750   3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750
 Max.   : 5.733000   Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000
 Direction
 Down:602
 Up  :648
```

Pairs

```r
ggpairs(Smarket) %>%
   print(progress = F)
```

```
Warning in ggmatrix_gtable(x, ...): Please use the 'progress' parameter in
your ggmatrix-like function call. See ?ggmatrix_progress for a few examples.
ggmatrix_gtable 'progress' and 'progress_format' will soon be deprecated.TRUE

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

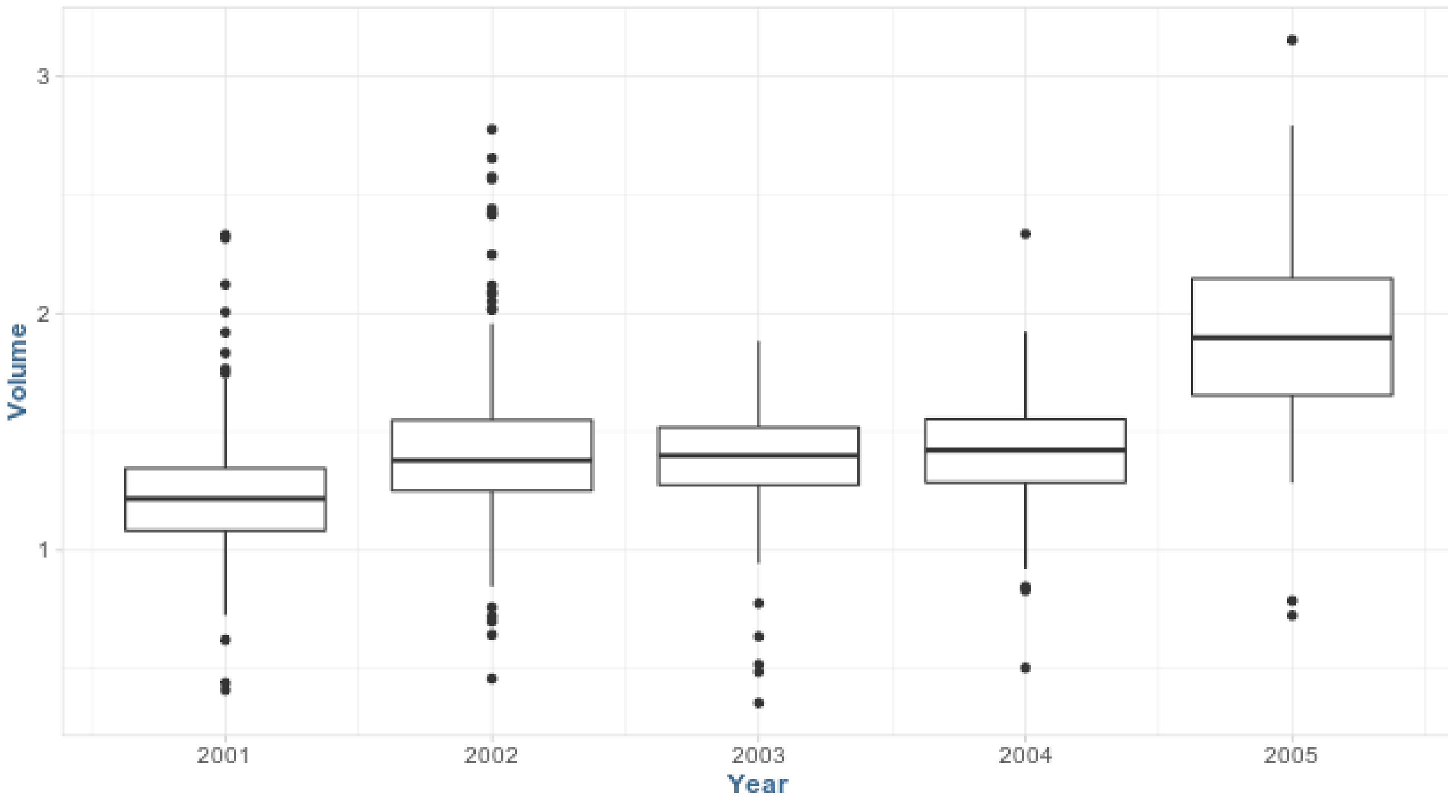
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
cor(Smarket %>% select(-Direction))
```

```
              Year        Lag1         Lag2         Lag3         Lag4
Year    1.00000000  0.029699649  0.030596422  0.033194581  0.035688718
Lag1    0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911
Lag2    0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533
Lag3    0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036
Lag4    0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000
Lag5    0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641
Volume  0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246
Today   0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527
                Lag5      Volume        Today
Year     0.029787995  0.53900647  0.030095229
Lag1    -0.005674606  0.04090991 -0.026155045
Lag2    -0.003557949 -0.04338321 -0.010250033
Lag3    -0.018808338 -0.04182369 -0.002447647
Lag4    -0.027083641 -0.04841425 -0.006899527
Lag5     1.000000000 -0.02200231 -0.034860083
Volume  -0.022002315  1.00000000  0.014591823
Today   -0.034860083  0.01459182  1.000000000
```

```r
ggplot(Smarket) +
  geom_boxplot(aes(Year, Volume, group = Year))
```



Logistic Regression

```r
summary(glm.fits <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                 data = Smarket, family = binomial))
```

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
```

```
Lag5          0.010313   0.049511   0.208    0.835
Volume        0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6

Number of Fisher Scoring iterations: 3
```

```
coef(glm.fits)
```

```
 (Intercept)         Lag1         Lag2         Lag3         Lag4         Lag5
-0.126000257 -0.073073746 -0.042301344  0.011085108  0.009358938  0.010313068
      Volume
 0.135440659
```

Probabilites of going up (first 10 trading days)

```
glm.probs <- predict(glm.fits, type = "response")
head(glm.probs, 10)
```

```
        1         2         3         4         5         6         7         8
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 0.5092292
        9        10
0.5176135 0.4888378
```

```
contrasts(Smarket$Direction)
```

```
     Up
Down  0
Up    1
```

Predictions

```
glm.pred <- rep("Down", nrow(Smarket))
glm.pred[glm.probs > 0.5] <- "Up"
```

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

```

glm.pred Down  Up
    Down  145 141
    Up    457 507
```

```
mean(glm.pred == Smarket$Direction)
```

```
[1] 0.5216
```

Validation

Get the holdout set.

```
train <- (Smarket$Year < 2005)
Smarket.2005 <- Smarket[!train]
dim(Smarket.2005)
```

```
[1] 252   9
```

```
Direction.2005 <- Smarket$Direction[!train]
```

Train the logistic regression model.

```
glm.fits <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                data = Smarket, family = binomial, subset = train)

glm.probs <- predict(glm.fits, Smarket.2005, type = "response")
```

Test

```
glm.pred <- rep("Down", 252)
glm.pred[glm.probs > 0.5] <- "Up"

table(glm.pred, Direction.2005)
```

```
        Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
```

```
mean(glm.pred == Direction.2005)
```

```
[1] 0.4801587
```

Model 2

```
summary(glm.fits <- glm(Direction ~ Lag1 + Lag2, data = Smarket, family = binomial, subset = tr
```

```

Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Smarket,
    subset = train)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.345   -1.188    1.074    1.164    1.326

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  0.03222    0.06338   0.508    0.611
Lag1        -0.05562    0.05171  -1.076    0.282
Lag2        -0.04449    0.05166  -0.861    0.389

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.4  on 995  degrees of freedom
AIC: 1387.4

Number of Fisher Scoring iterations: 3
```

```
glm.probs <- predict(glm.fits, Smarket.2005, type = "response")
glm.pred <- rep("Down", nrow(Smarket.2005))

glm.pred[glm.probs >= 0.5] <- "Up"

table(glm.pred, Direction.2005)
```

```
        Direction.2005
glm.pred Down  Up
    Down   35  35
    Up     76 106
```

```
mean(glm.pred == Direction.2005)
```

```
[1] 0.5595238
```

```
predict(glm.fits, newdata = data.table(Lag1 = c(1.2, 1.5),
                                       Lag2 = c(1.1, -0.8)),
        type = "response")
```

```
        1         2 
0.4791462 0.4960939 
```

Linear Discriminant Analysis

LDA is from **MASS** package.

```
summary(lda.fit <- lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train))
```

```
        Length Class  Mode     
prior   2      -none- numeric  
counts  2      -none- numeric  
means   4      -none- numeric  
scaling 2      -none- numeric  
lev     2      -none- character
```

```
svd     1      -none- numeric
N       1      -none- numeric
call    4      -none- call
terms   3      terms  call
xlevels 0      -none- list
```

```r
lda.fit
```

```
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Prior probabilities of groups:
    Down       Up 
0.491984 0.508016 

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544

Coefficients of linear discriminants:
            LD1
Lag1 -0.6420190
Lag2 -0.5135293
```

```r
lda.pred <- predict(lda.fit, Smarket.2005)

names(lda.pred)
```

```
[1] "class"     "posterior" "x"
```

Predictions:

```r
lda.class <- lda.pred$class
table(lda.class, Direction.2005)
```

```
         Direction.2005
lda.class Down  Up
     Down   35  35
     Up     76 106
```

Note: almost identical to logistic regression.

```r
mean(lda.class == Direction.2005)
```

```
[1] 0.5595238
```

```r
sum(lda.pred$posterior[, 1] >= 0.5)
```

```
[1] 70
```

```r
sum(lda.pred$posterior[, 1] < 0.5)
```

```
[1] 182
```

The posterior probabilites output by the model corresponds to the probability that the market will decrease.

```r
lda.pred$posterior[1:20, 1]
```

```
        1         2         3         4         5         6         7         8 
0.4901792 0.4792185 0.4668185 0.4740011 0.4927877 0.4938562 0.4951016 0.4872861 
        9        10        11        12        13        14        15        16 
0.4907013 0.4844026 0.4906963 0.5119988 0.4895152 0.4706761 0.4744593 0.4799583 
       17        18        19        20 
0.4935775 0.5030894 0.4978806 0.4886331 
```

```r
lda.class[1:20]
```

```
 [1] Up   Up   Up   Up   Up   Up   Up   Up   Up   Up   Up   Down Up   Up   Up  
[16] Up   Up   Down Up   Up  
Levels: Down Up
```

Apply a threshold of 90% to predictions:

```r
sum(lda.pred$posterior[, 1] > .9)
```

```
[1] 0
```

Quadratic Discriminant Analysis

```r
summary(qda.fit <- qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train))
```

```
        Length Class  Mode     
prior   2      -none- numeric  
counts  2      -none- numeric  
means   4      -none- numeric  
scaling 8      -none- numeric  
ldet    2      -none- numeric  
lev     2      -none- character
N       1      -none- numeric  
call    4      -none- call     
terms   3      terms  call     
xlevels 0      -none- list     
```

```r
qda.fit
```

```
Call:
qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
```

```
Prior probabilities of groups:
    Down       Up 
0.491984 0.508016 

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544
```

Predictions

```
qda.class <- predict(qda.fit, Smarket.2005)$class
table(qda.class, Direction.2005)
```

```
         Direction.2005
qda.class Down  Up
     Down   30  20
     Up     81 121
```

```
mean(qda.class == Direction.2005)
```

```
[1] 0.5992063
```

K-Nearest Neighbors

Data Setup

```
train.X <- with(Smarket, cbind(Lag1, Lag2))[train, ]
test.X <- with(Smarket, cbind(Lag1, Lag2))[!train, ]
train.Direction <- Smarket$Direction[train]
```

KNN

```
set.seed(1)

knn.pred <- knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.2005)
```

```
        Direction.2005
knn.pred Down Up
    Down   43 58
    Up     68 83
```

```
set.seed(1)

knn.pred <- knn(train.X, test.X, train.Direction, k = 3)
table(knn.pred, Direction.2005)
```

```
        Direction.2005
knn.pred Down Up
    Down   48 55
    Up     63 86
```

Caravan Insurance Data

```r
caravan <- Caravan

dim(caravan)
```

```
[1] 5822   86
```

```r
summary(caravan$Purchase)
```

```
  No  Yes
5474  348
```

```r
table(caravan$Purchase) %>% prop.table()
```

```

        No        Yes
0.94022673 0.05977327
```

```r
standardized.X <- scale(caravan[, -86])

var(caravan[, 1])
```

```
[1] 165.0378
```

```r
var(caravan[, 2])
```

```
[1] 0.1647078
```

```r
var(standardized.X[, 1])
```

```
[1] 1
```

```r
var(standardized.X[, 2])
```

```
[1] 1
```

- K=1

```r
test <- 1:1000
train.X <- standardized.X[-test,]
test.X <- standardized.X[test,]

train.Y <- caravan$Purchase[-test]
test.Y <- caravan$Purchase[test]
```

```r
set.seed(1)

knn.pred <- knn(train.X, test.X, train.Y, k = 1)
mean(test.Y != knn.pred)
```

```
[1] 0.118
```

```r
mean(test.Y != "No")
```

```
[1] 0.059
```

```r
result <- table(knn.pred, test.Y)
result
```

```
        test.Y
knn.pred  No Yes
     No  873  50
     Yes  68   9
```

```r
result %>% prop.table()
```

```
        test.Y
knn.pred    No   Yes
     No  0.873 0.050
     Yes 0.068 0.009
```

- K=3

```r
set.seed(1)

knn.pred <- knn(train.X, test.X, train.Y, k = 3)
mean(test.Y != knn.pred)
```

```
[1] 0.074
```

```r
mean(test.Y != "No")
```

```
[1] 0.059
```

```r
result <- table(knn.pred, test.Y)
result
```

```
        test.Y
knn.pred  No Yes
     No  921  54
     Yes  20   5
```

```r
result %>% prop.table()
```

```
        test.Y
```

```
knn.pred    No   Yes
     No  0.921 0.054
     Yes 0.020 0.005
```

- K=5

```
set.seed(1)

knn.pred <- knn(train.X, test.X, train.Y, k = 5)
mean(test.Y != knn.pred)
```

```
[1] 0.066
```

```
mean(test.Y != "No")
```

```
[1] 0.059
```

```
result <- table(knn.pred, test.Y)
result
```

```
        test.Y
knn.pred  No Yes
     No  930  55
     Yes  11   4
```

```
result %>% prop.table()
```

```
        test.Y
knn.pred    No   Yes
     No  0.930 0.055
     Yes 0.011 0.004
```

Logistic Regression Alternative

```
glm.fits <- glm(Purchase ~ ., data = caravan, family = binomial,
                subset = -test)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
glm.probs <- predict(glm.fits, caravan[test,], type = "response")

# .5 cut-off
glm.pred <- rep("No", 1000)
glm.pred[glm.probs > .5] <- "Yes"

results <- table(glm.pred, test.Y)
results %>% prop.table()
```

```
        test.Y
glm.pred    No   Yes
     No  0.934 0.059
```

```
     Yes 0.007 0.000
```

```r
# .25 cut-off
glm.pred <- rep("No", 1000)
glm.pred[glm.probs > .25] <- "Yes"

results <- table(glm.pred, test.Y)
results %>% prop.table()
```

```
        test.Y
glm.pred    No   Yes
     No  0.919 0.048
     Yes 0.022 0.011
```

```r
# Quiz

bal <- 1936.75

exp(-10.6513 + 0.0055 * bal) / ( 1 + exp(-10.6513 + 0.0055*bal))
```

```
[1] 0.5002062
```

```r
b0 <- -6; b1 <- 0.05; b2 <- 1
x1 <- 50; x2 <- 3.5
exp(b0 + b1 * x1 + b2 * x2) / ( 1 + exp(b0 + b1 * x1 + b2 * x2))
```

```
[1] 0.5
```

Conceptual

1.) Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representaion and logit representation for the logistic regression models are equivalent

2.) It was stated in the text that classifying an observation to the class for which (4.13) is largest. Prove that this is the case. In other words, under the assumption that the observations in the *k*th class are drawn from a N (□, □^2) distribution, the Bayes' clssifier assigns an observation to the class for which the discriminant function is maximized.

3.) This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class-specific mean vector and a class specific covariance matrix. We consider the simple case where p=1; i.e. there is only one feature.

Suppose that we have K classes, and if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, $X \sim N(\mu k, \sigma 2k)$. Recall that the density function for the one-dimensional normal distribution is given in (4.11). Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.

Applied

8.)