Stock Market Data

```
library(ISLR)
library(GGally)
head(Smarket)
```

```
  Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
1 2001  0.381 -0.192 -2.624 -1.055  5.010 1.1913  0.959        Up
2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032        Up
3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623      Down
4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614        Up
5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213        Up
6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392        Up
```

```
tail(Smarket)
```

```
     Year   Lag1   Lag2   Lag3   Lag4   Lag5  Volume  Today Direction
1245 2005  0.252 -0.024 -0.584 -0.285 -0.141 2.06517  0.422        Up
1246 2005  0.422  0.252 -0.024 -0.584 -0.285 1.88850  0.043        Up
1247 2005  0.043  0.422  0.252 -0.024 -0.584 1.28581 -0.955      Down
1248 2005 -0.955  0.043  0.422  0.252 -0.024 1.54047  0.130        Up
1249 2005  0.130 -0.955  0.043  0.422  0.252 1.42236 -0.298      Down
1250 2005 -0.298  0.130 -0.955  0.043  0.422 1.38254 -0.489      Down
```

Logistic Regression

```
# ggpairs(Smarket)
glm.fit = glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket,
family=binomial(link="logit"))
summary(glm.fit)
```

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

Log-Odds, Odds, and Probability, 1

```
log.odds = coef(glm.fit)
log.odds
```

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

```
odds <- exp(coef(glm.fit))
odds
```

```
(Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5
  0.8816146   0.9295323   0.9585809   1.0111468   1.0094029   1.0103664
     Volume
  1.1450412
```

```
prob = odds/(1+odds)
prob
```

```
(Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5
  0.4685415   0.4817397   0.4894262   0.5027712   0.5023397   0.5025782
     Volume
  0.5338085
```

Log-Odds, Odds, and Probability, 2

```
confint(glm.fit)
```

```
                2.5 %     97.5 %
(Intercept) -0.59907825 0.34544102
Lag1        -0.17191967 0.02499864
Lag2        -0.14082062 0.05578504
Lag3        -0.08689851 0.10912653
Lag4        -0.08870645 0.10745613
Lag5        -0.08681511 0.10753778
Volume      -0.17448194 0.44696774
```

```
exp(confint(glm.fit))
```

```
               2.5 %   97.5 %
(Intercept) 0.5493177 1.412613
Lag1        0.8420468 1.025314
Lag2        0.8686451 1.057370
Lag3        0.9167701 1.115303
Lag4        0.9151142 1.113442
Lag5        0.9168466 1.113533
Volume      0.8398920 1.563564
```

```
exp(confint(glm.fit))/(1+exp(confint(glm.fit)))
```

```
               2.5 %    97.5 %
(Intercept) 0.3545546 0.5855116
```

```
Lag1        0.4571256 0.5062493
Lag2        0.4648529 0.5139426
Lag3        0.4782890 0.5272546
Lag4        0.4778379 0.5268382
Lag5        0.4783098 0.5268586
Volume      0.4564898 0.6099180
```

Log-Odds to Probabilities to Binary Predictions

```
glm.probs=predict(glm.fit,type="response")
glm.probs[1:10]
```

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

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

```
     Up
Down  0
Up    1
```

```
glm.pred = ifelse(glm.probs>0.5, "Up", "Down")
glm.pred[1:6]
```

```
    1      2      3      4      5      6 
 "Up" "Down" "Down"   "Up"   "Up"   "Up" 
```

Confusion Matrix

```
conf.mat <- table(glm.pred,Smarket$Direction)
conf.mat
```

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

Model Fit Statistics, 1

```
# Error rate: Proportion of misses (false positives & false negatives)
Error.Rate=(457+141)/1250
Error.Rate
```

```
[1] 0.4784
```

```
# or...
Error.Rate=(conf.mat[2,1]+conf.mat[1,2])/length(glm.pred)
Error.Rate
```

```
[1] 0.4784
```

```
# Sensitivity: Proportion of correct positives
glm.sensitivity=507/(141+507)
glm.sensitivity
```

```
[1] 0.7824074
```

```
# or...
glm.sensitivity=conf.mat[2,2]/(conf.mat[1,2]+conf.mat[2,2])
glm.sensitivity
```

```
[1] 0.7824074
```

```
# Specificity: Proportion of correct negatives
glm.specificity=145/(145+457)
glm.specificity
```

```
[1] 0.2408638
```

```
# or...
glm.specificity=conf.mat[1,1]/(conf.mat[1,1]+conf.mat[2,1])
glm.specificity
```

```
[1] 0.2408638
```

```
# False positive rate: Proportion of false positives
# FPR = 1 - Specificity
glm.false.pos <- 1 - glm.specificity
glm.false.pos
```

```
[1] 0.7591362
```

Model Fit Statistics, 2

```
glm.pred.cons <- ifelse(glm.probs>0.60, "Up", "Down")
glm.pred.cons[1:6]
```

```
     1      2      3      4      5      6
"Down" "Down" "Down" "Down" "Down" "Down"
```

```
conf.mat.cons <- table(glm.pred.cons,Smarket$Direction)
Error.Rate.cons <-
(conf.mat.cons[2,1]+conf.mat.cons[1,2])/length(glm.pred.cons)
glm.sensitivity.cons <-
conf.mat.cons[2,2]/(conf.mat.cons[1,2]+conf.mat.cons[2,2])
glm.specificity.cons <-
conf.mat.cons[1,1]/(conf.mat.cons[1,1]+conf.mat.cons[2,1])
glm.false.pos.cons <- 1 - glm.specificity.cons
glm.fit.stats.cons <- c(Error.Rate.cons, glm.sensitivity.cons,
glm.specificity.cons,glm.false.pos.cons)
names(glm.fit.stats.cons)=c("Error Rate", "Sensitivity", "Specificity",
"False Positives")

glm.fit.stats.cons
```

```
     Error Rate      Sensitivity      Specificity False Positives
    0.519200000      0.004629630      0.993355482      0.006644518
```

Multinomial Regression, 1

```
library(VGAM)
library(car)

head(Womenlf)
```

```
    partic hincome children  region
1 not.work      15  present Ontario
2 not.work      13  present Ontario
3 not.work      45  present Ontario
4 not.work      23  present Ontario
5 not.work      19  present Ontario
6 not.work       7  present Ontario
```

```
vglm.fit = vglm(partic ~ hincome+children, family=multinomial(refLevel=1),
data=Womenlf)
# levels(Womenlf$partic)
summary(vglm.fit)
```

```


Call:
vglm(formula = partic ~ hincome + children, family = multinomial(refLevel =
1),
    data = Womenlf)


Pearson residuals:
                      Min      1Q  Median      3Q   Max
log(mu[,2]/mu[,1]) -3.156 -0.7541  0.5405  0.6063 2.170
```

```
log(mu[,3]/mu[,1]) -2.440 -0.2928 -0.2520 -0.1768 4.495

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept):1     -1.98282    0.48418  -4.095 4.22e-05 ***
(Intercept):2     -3.41513    0.66552  -5.132 2.87e-07 ***
hincome:1          0.09723    0.02810   3.461 0.000539 ***
hincome:2          0.10412    0.03328   3.128 0.001759 **
childrenpresent:1  2.55860    0.36220   7.064 1.62e-12 ***
childrenpresent:2  2.58009    0.50972   5.062 4.15e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  2

Names of linear predictors: log(mu[,2]/mu[,1]), log(mu[,3]/mu[,1])

Residual deviance: 422.8819 on 520 degrees of freedom

Log-likelihood: -211.441 on 520 degrees of freedom

Number of iterations: 5

Reference group is level  1  of the response
```

Multinomial Regression, 2

```
coef.log.odds = coef(vglm.fit)
coef.log.odds[1:6]
```

```
    (Intercept):1     (Intercept):2          hincome:1          hincome:2
      -1.98282245       -3.41512944         0.09723067         0.10412282
childrenpresent:1 childrenpresent:2
       2.55859504        2.58008617
```

```
coef.odds = exp(coef.log.odds)
coef.odds[1:6]
```

```
    (Intercept):1     (Intercept):2          hincome:1          hincome:2
       0.13768009        0.03287215         1.10211457         1.10973674
childrenpresent:1 childrenpresent:2
      12.91765581       13.19827539
```

```
coef.prob = coef.odds/(1+coef.odds)
coef.prob[1:6]
```

```
    (Intercept):1     (Intercept):2          hincome:1          hincome:2
       0.12101828        0.03182596         0.52428854         0.52600721
childrenpresent:1 childrenpresent:2
       0.92814882        0.92956891
```

Multinomial Regression, 3

```
pred.log.odds = predictvglm(vglm.fit, newdata=NULL, se.fit = TRUE)
# pred.log.odds[1:6]

pred.log.odds$fitted.values[1:6]
```

```
[1] 2.034233 1.839771 4.951153 2.812078 2.423155 1.256387
```

```
pred.odds = exp(pred.log.odds$fitted.values)
pred.odds[1:6]
```

```
[1]   7.646382   6.295098 141.337785  16.644469  11.281399   3.512708
```

```
pred.prob = pred.odds/(1+pred.odds)
pred.prob[1:6]
```

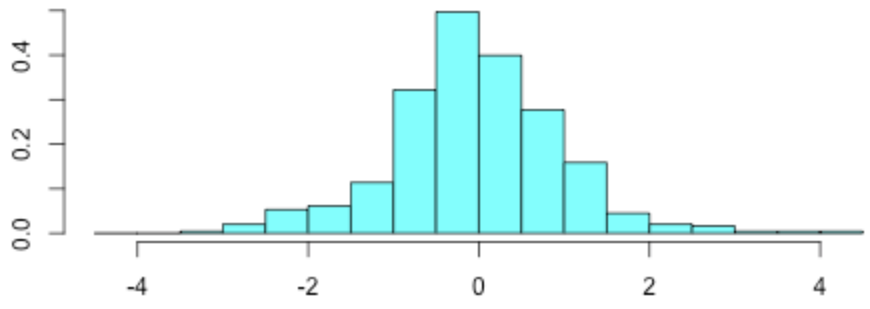
```
[1] 0.8843447 0.8629217 0.9929745 0.9433250 0.9185761 0.7784036
```

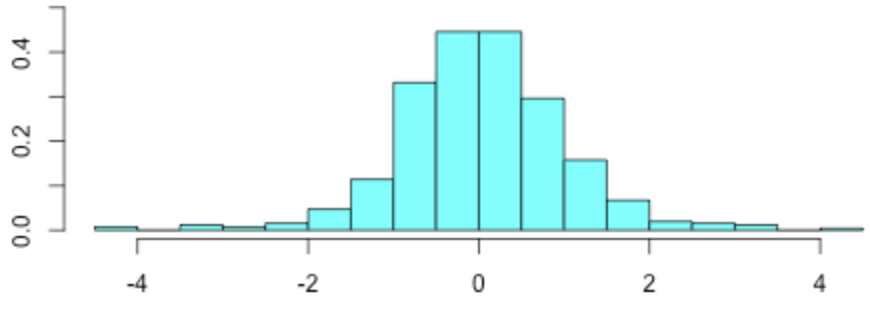
Linear Discriminant Analysis (LDA), 1

```
require(ISLR)
require(MASS)

train=(Smarket$Year<2005)
Smarket.2005=Smarket[!train,]
Direction.2005=Smarket$Direction[!train]

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

```
plot(lda.fit)
```


0.0
0.2
0.4
-4
-2
0
2
4
group Down


0.0
0.2
0.4
-4
-2
0
2
4
group Up

Linear Discriminant Analysis (LDA), 2

```
lda.pred=predict(lda.fit, Smarket.2005)
names(lda.pred)
```

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

```
lda.pred$class[1:6]
```

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

```
head(lda.pred$posterior)
```

```
          Down        Up
999  0.4901792 0.5098208
1000 0.4792185 0.5207815
1001 0.4668185 0.5331815
1002 0.4740011 0.5259989
1003 0.4927877 0.5072123
1004 0.4938562 0.5061438
```

```
head(lda.pred$x)
```

```
             LD1
999   0.08293096
```

```
1000  0.59114102
1001  1.16723063
1002  0.83335022
1003 -0.03792892
1004 -0.08743142
```

Linear Discriminant Analysis (LDA), 3

```
lda.class=lda.pred$class

lda.conf <- table(lda.class,Direction.2005)
lda.conf
```

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

```
lda.correct <- mean(lda.class==Direction.2005)
lda.error <- mean(lda.class!=Direction.2005)
lda.stats <- c(lda.correct, lda.error)
names(lda.stats) <- c("Correct", "error")
lda.stats
```

```
  Correct     error
0.5595238 0.4404762
```

Quadratic Discriminant Analysis (QDA)

```
train=(Smarket$Year<2005)
Smarket.2005=Smarket[!train,]
Direction.2005=Smarket$Direction[!train]

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

```
qda.class=predict(qda.fit,Smarket.2005)$class

qda.conf <- table(qda.class,Direction.2005)
qda.conf
```

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

QDA performs better here

```
qda.correct <- mean(qda.class==Direction.2005) # Correct rate
qda.error <- mean(qda.class!=Direction.2005) # Error rate
qda.stats <- c(qda.correct, qda.error)
names(qda.stats) <- c("Correct", "error")
qda.stats
```

```
  Correct     error
0.5992063 0.4007937
```

```
both.stats <- rbind(lda.stats, qda.stats)
both.stats
```

```
            Correct     error
lda.stats 0.5595238 0.4404762
qda.stats 0.5992063 0.4007937
```