LDA QDA-Exercises

December 6, 2018

0.1 Exercise 1

- Try lda with all Lag variables.
- Make a quick comparison between logistic regression and lda.
- Try with qda and compare all three methods. Plot the results.

```
In [12]: library(MASS)
         library(ISLR)
```

Probaremos a ejecutar lda con todos los lag, aunque en primer lugar haremos un estudio para probar la normalidad de las muestras y si tienen varianzas similares.

```
In [13]: shapiro.test(Smarket$Lag1)
         shapiro.test(Smarket$Lag2)
         shapiro.test(Smarket$Lag3)
         shapiro.test(Smarket$Lag4)
         shapiro.test(Smarket$Lag5)
```

```
Shapiro-Wilk normality test

data:  Smarket$Lag1
W = 0.97219, p-value = 8.889e-15
```

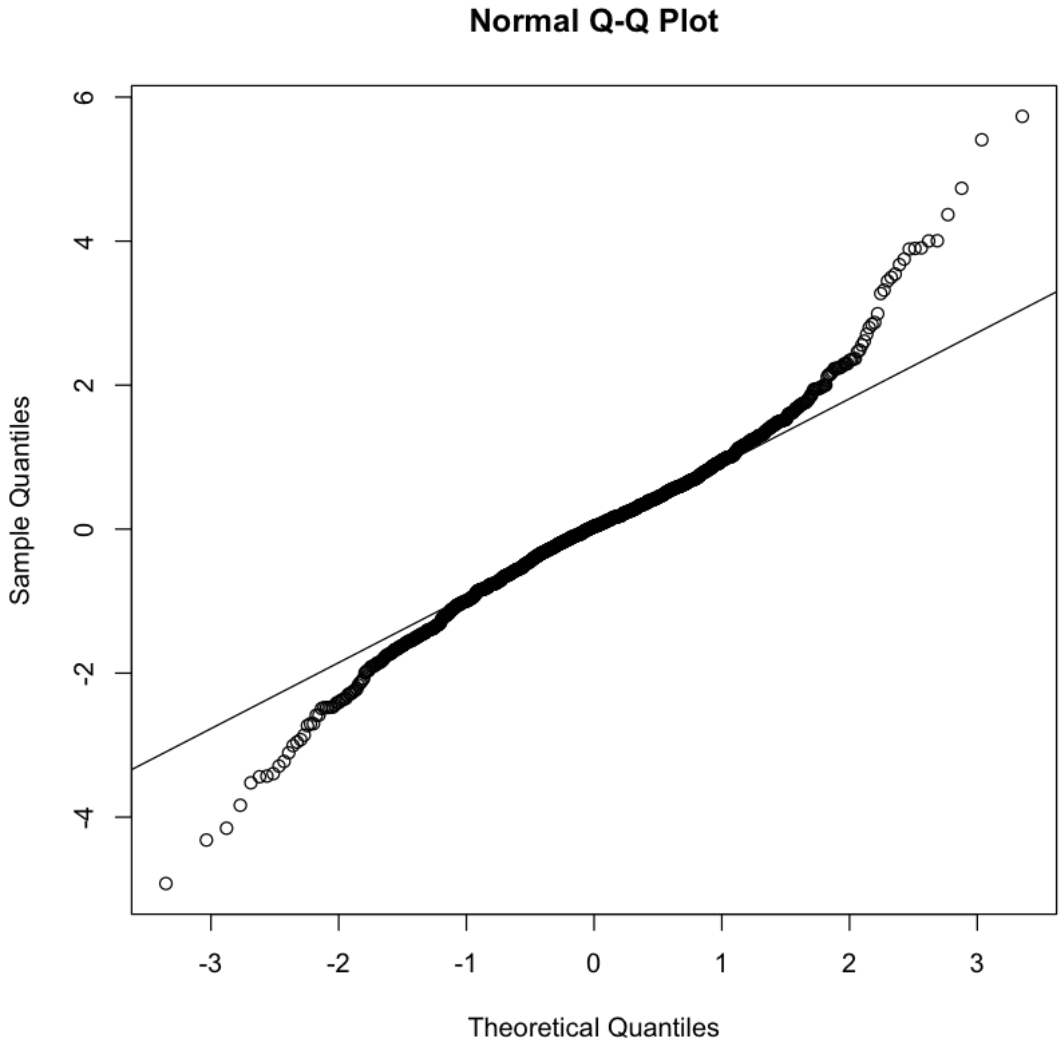
```
Shapiro-Wilk normality test

data:  Smarket$Lag2
W = 0.97217, p-value = 8.798e-15
```

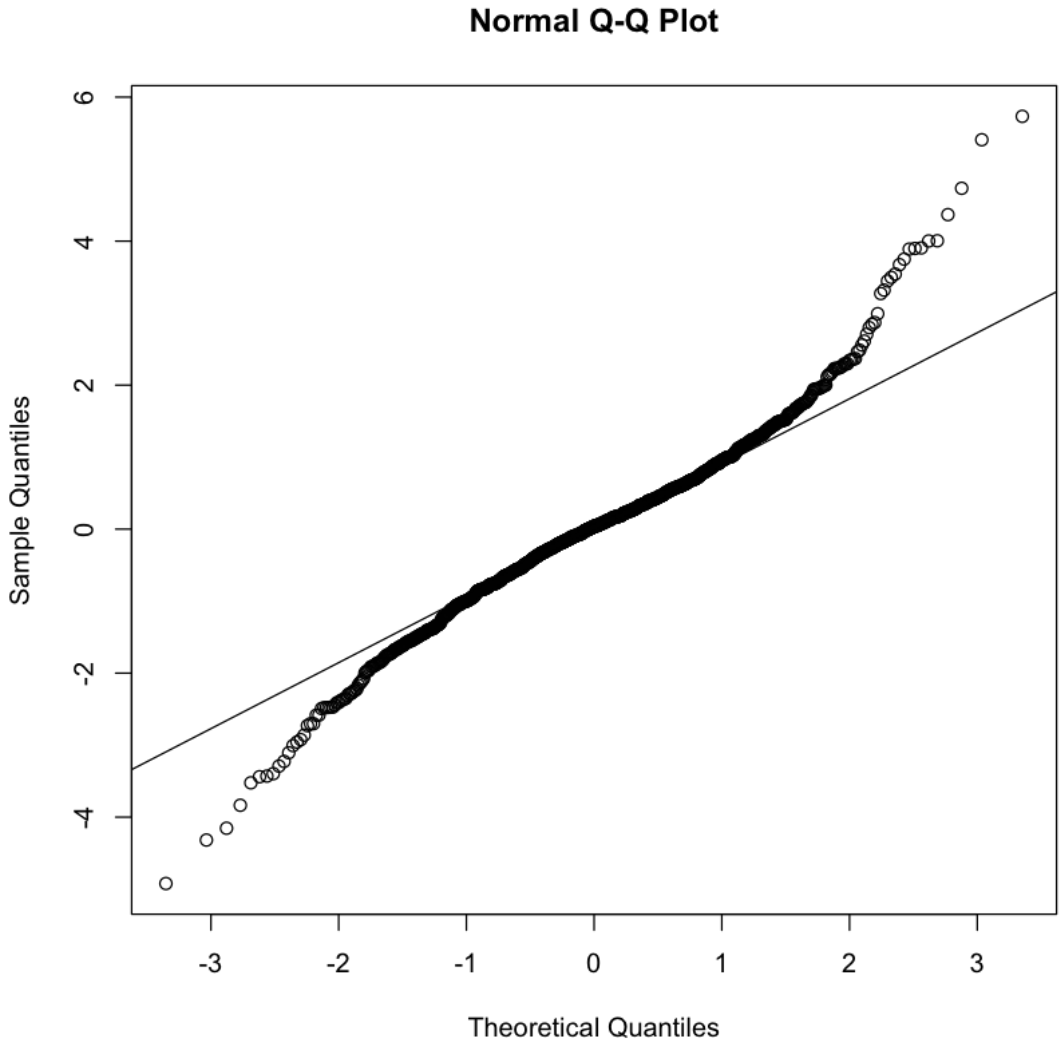
```
Shapiro-Wilk normality test

data:  Smarket$Lag3
W = 0.9724, p-value = 1.035e-14
```

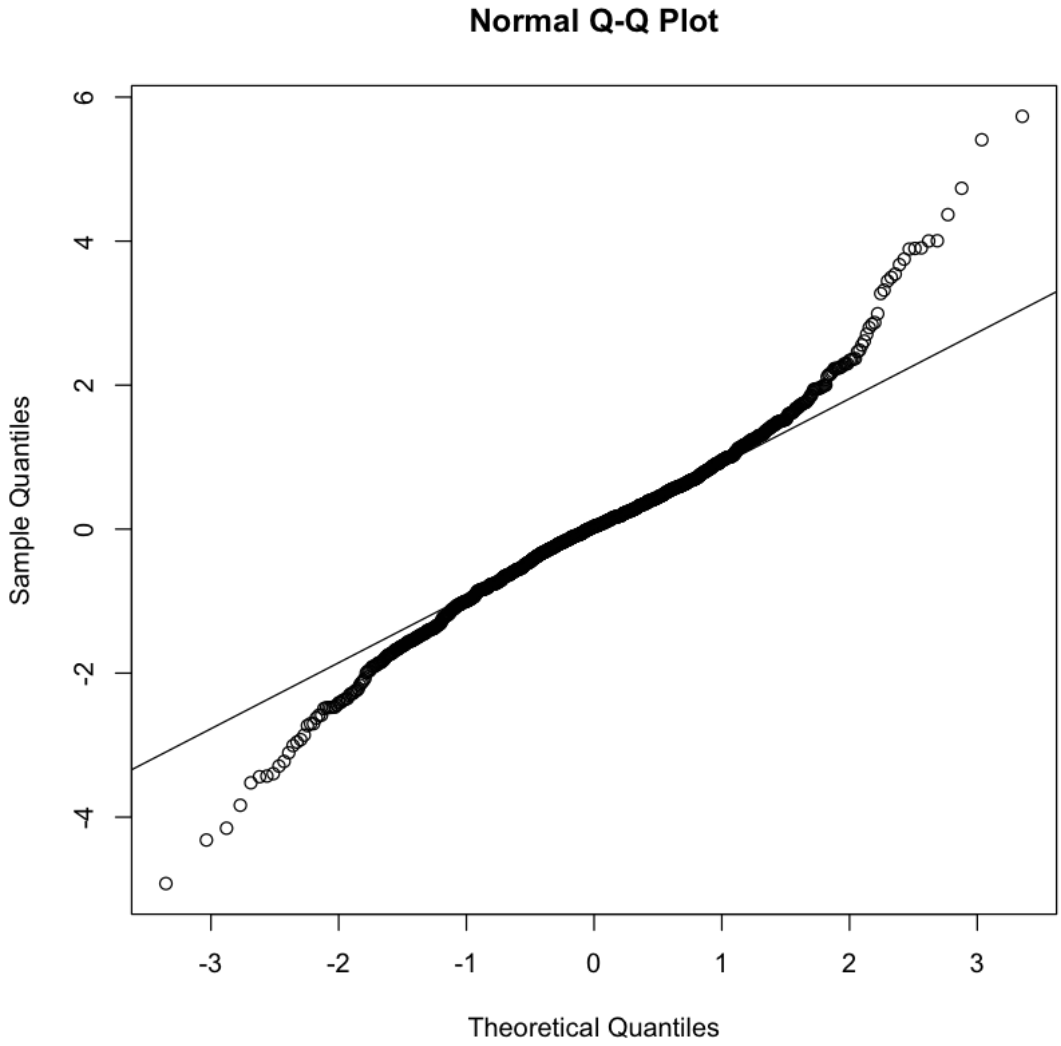
```
Shapiro-Wilk normality test

data:  Smarket$Lag4
W = 0.97242, p-value = 1.049e-14
```

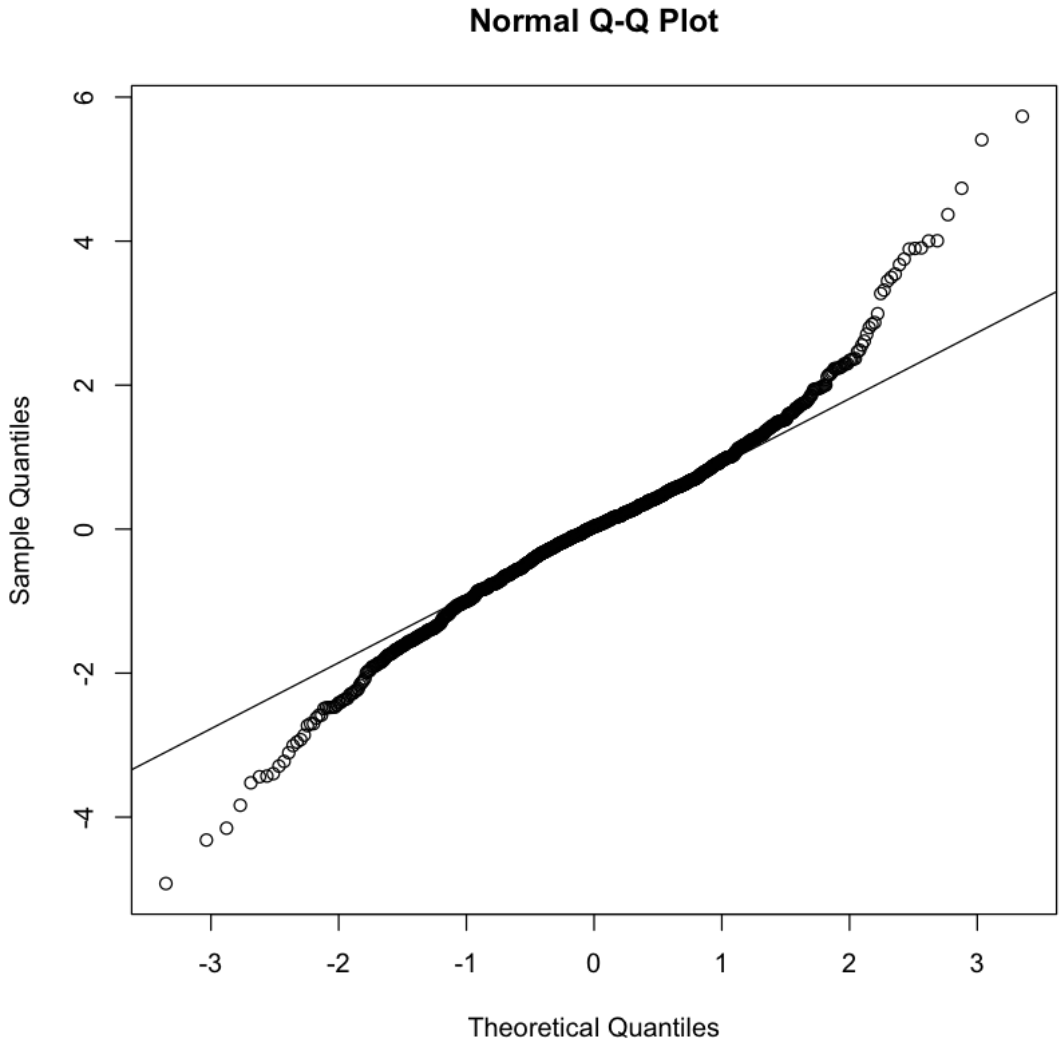
```
Shapiro-Wilk normality test

data:  Smarket$Lag5
W = 0.97011, p-value = 2.149e-15
```

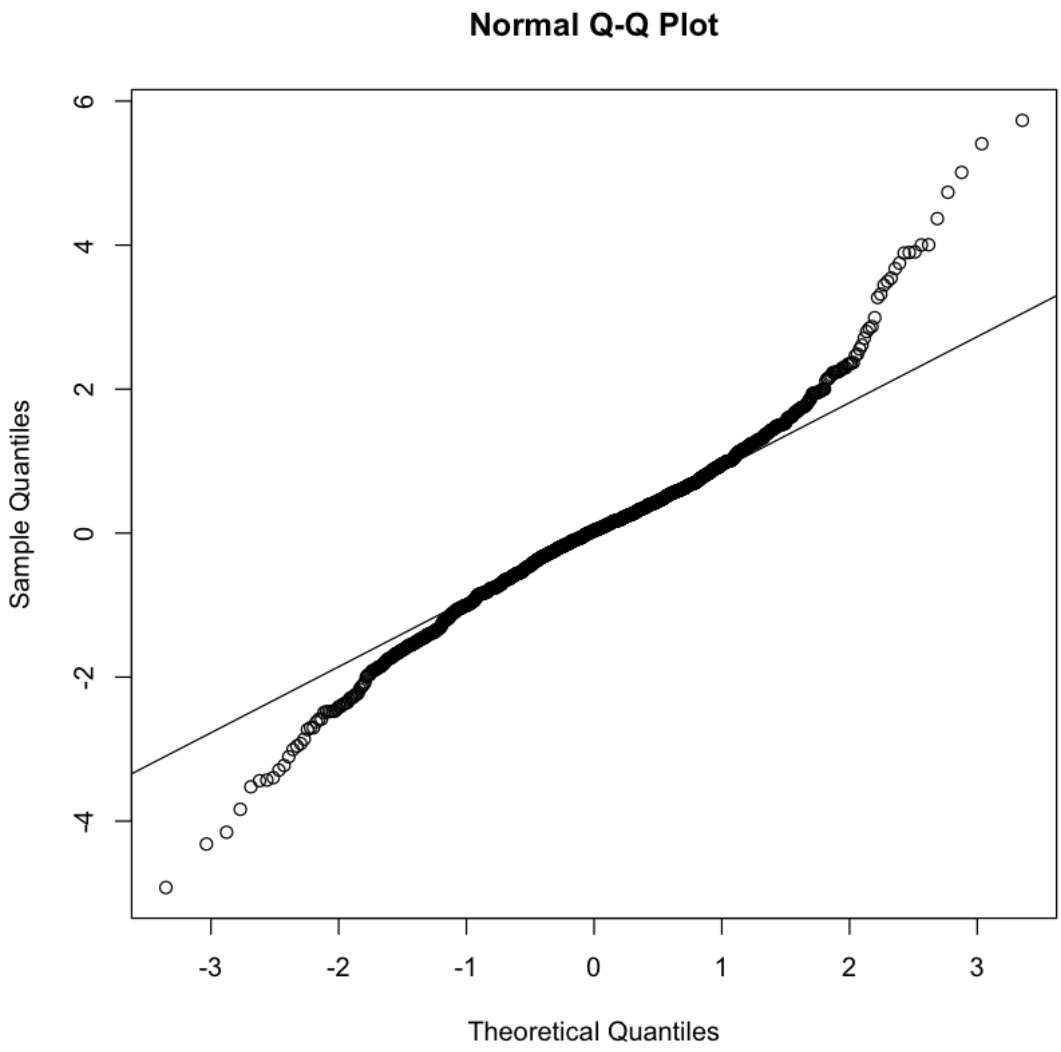
```
In [14]: qqnorm(y = Smarket$Lag1)
         qqline(y = Smarket$Lag1)
         qqnorm(y = Smarket$Lag2)
         qqline(y = Smarket$Lag2)
         qqnorm(y = Smarket$Lag3)
         qqline(y = Smarket$Lag3)
         qqnorm(y = Smarket$Lag4)
         qqline(y = Smarket$Lag4)
         qqnorm(y = Smarket$Lag5)
         qqline(y = Smarket$Lag5)
```


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

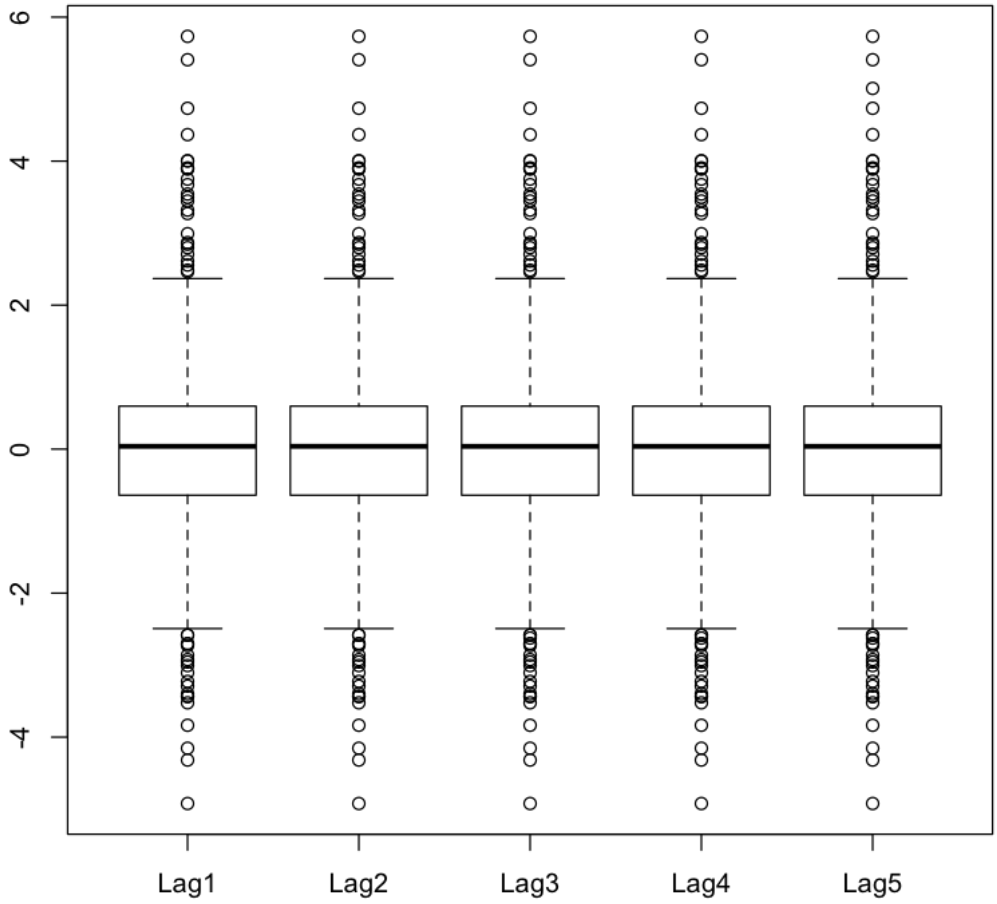


Más o menos podemos apreciar que todos siguen una distribución normal. Ahora comprobaremos las varianzas.

```
In [15]: # Predictors have a common variance
         boxplot(Smarket[,2:6])

         var(Smarket$Lag1)
         var(Smarket$Lag2)
         var(Smarket$Lag3)
         var(Smarket$Lag4)
         var(Smarket$Lag5)
```

1.29117506222642
1.2911328189265

1.29664442448359
1.29680562160128
1.31687148397502



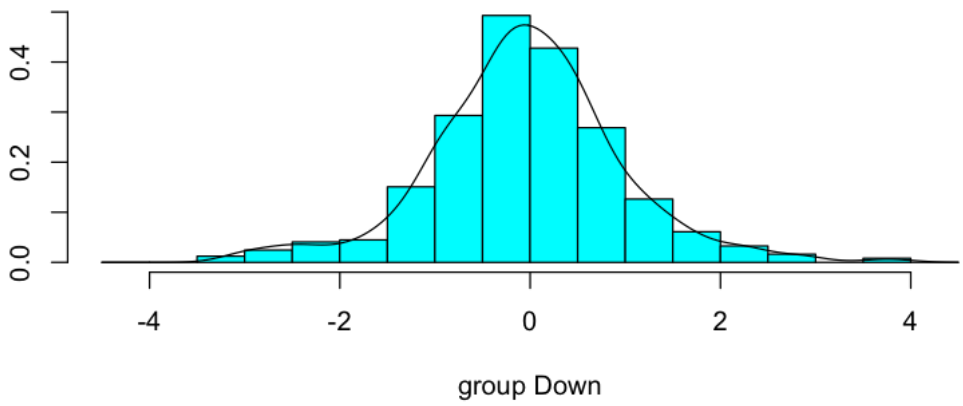
Como podemos observamos tienen una varianza similar los distintos predictores, aunque el Lag5 es el predictor en el que más difiere la varianza. Ahora aplicaremos LDA.

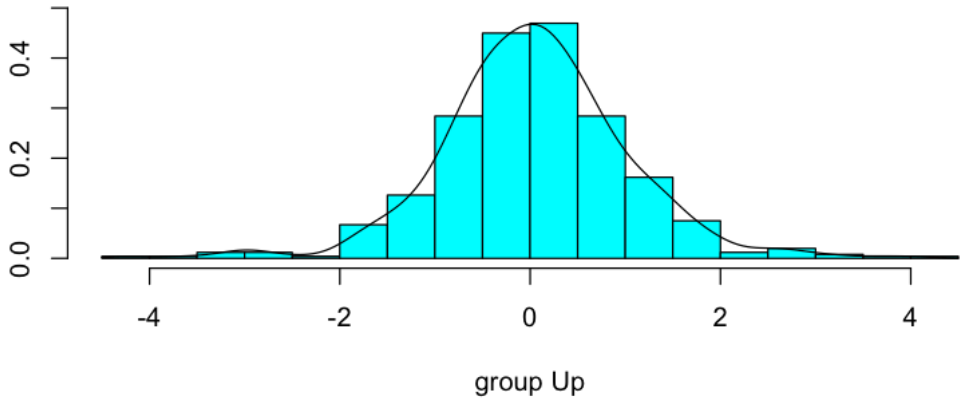
```
In [16]: # Linear Discriminant Analysis
         lda.fit <- lda(Direction~Lag1+Lag2+Lag3+Lag4+Lag5,data=Smarket, subset=Year<2005)
         lda.fit
```

```
Call:
lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = Smarket,
    subset = Year < 2005)
```

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

Group means:
            Lag1        Lag2         Lag3         Lag4          Lag5
Down  0.04279022  0.03389409 -0.009806517 -0.010598778  0.0043665988
Up   -0.03954635 -0.03132544  0.005834320  0.003110454 -0.0006508876

Coefficients of linear discriminants:
             LD1
Lag1 -0.63046918
Lag2 -0.50221745
Lag3  0.10142974
Lag4  0.09725317
Lag5 -0.03685767
```

```
In [17]: plot(lda.fit, type="both", xlab = "LD1", ylab = "Normalizedc frequency") # xlab and yla
```





Ahora predeciremos los datos a partir del modelo ajustado.

```
In [18]: Smarket.2005 <- subset(Smarket,Year==2005)
         lda.pred <- predict(lda.fit,Smarket.2005)
         class(lda.pred)
         lda.pred
```

'list'

$class 1. Up 2. Up 3. Up 4. Up 5. Up 6. Up 7. Up 8. Up 9. Up 10. Up 11. Up 12. Down 13. Up 14. Up 15. Up 16. Up 17. Up 18. Down 19. Up 20. Up 21. Up 22. Down 23. Down 24. Up 25. Down 26. Down 27. Up 28. Up 29. Up 30. Down 31. Down 32. Up 33. Up 34. Up 35. Up 36. Up 37. Up 38. Down 39. Down 40. Up 41. Up 42. Up 43. Up 44. Down 45. Down 46. Up 47. Up 48. Up 49. Up 50. Up 51. Up 52. Up 53. Up 54. Up 55. Up 56. Up 57. Up 58. Up 59. Up 60. Up 61. Down

62. Down 63. Up 64. Up 65. Down 66. Down 67. Down 68. Up 69. Up 70. Down 71. Up 72. Up 73. Up 74. Up 75. Down 76. Up 77. Down 78. Down 79. Up 80. Up 81. Up 82. Up 83. Up 84. Down 85. Up 86. Down 87. Down 88. Up 89. Up 90. Up 91. Up 92. Up 93. Up 94. Down 95. Down 96. Down 97. Down 98. Up 99. Up 100. Up 101. Up 102. Up 103. Down 104. Up 105. Up 106. Down 107. Up 108. Up 109. Up 110. Up 111. Up 112. Up 113. Up 114. Up 115. Up 116. Up 117. Down 118. Up 119. Up 120. Up 121. Up 122. Up 123. Up 124. Down 125. Down 126. Up 127. Up 128. Down 129. Up 130. Up 131. Down 132. Down 133. Up 134. Up 135. Up 136. Up 137. Up 138. Up 139. Down 140. Up 141. Up 142. Up 143. Up 144. Up 145. Down 146. Up 147. Up 148. Down 149. Down 150. Up 151. Up 152. Up 153. Down 154. Up 155. Up 156. Up 157. Up 158. Up 159. Up 160. Up 161. Up 162. Up 163. Up 164. Up 165. Up 166. Up 167. Up 168. Up 169. Down 170. Down 171. Up 172. Down 173. Down 174. Up 175. Up 176. Down 177. Up 178. Up 179. Up 180. Down 181. Up 182. Up 183. Up 184. Up 185. Up 186. Up 187. Up 188. Up 189. Down 190. Down 191. Up 192. Up 193. Up 194. Up 195. Up 196. Up 197. Up 198. Up 199. Up 200. Down 201. Down 202. Up 203. Up 204. Up 205. Up 206. Down 207. Down 208. Up 209. Up 210. Down 211. Down 212. Up 213. Up 214. Down 215. Up 216. Up 217. Up 218. Up 219. Down 220. Down 221. Up 222. Up 223. Up 224. Down 225. Down 226. Down 227. Down 228. Down 229. Up 230. Up 231. Up 232. Up 233. Down 234. Down 235. Up 236. Up 237. Up 238. Up 239. Up 240. Up 241. Up 242. Down 243. Up 244. Up 245. Up 246. Up 247. Up 248. Down 249. Up 250. Up 251. Up 252. Up

Levels: 1. 'Down' 2. 'Up'

	Down	Up
999	0.4883105	0.5116895
1000	0.4798093	0.5201907
1001	0.4671632	0.5328368
1002	0.4761685	0.5238315
1003	0.4970100	0.5029900
1004	0.4964641	0.5035359
1005	0.4941381	0.5058619
1006	0.4865417	0.5134583
1007	0.4905456	0.5094544
1008	0.4848967	0.5151033
1009	0.4914164	0.5085836
1010	0.5124976	0.4875024
1011	0.4902899	0.5097101
1012	0.4666312	0.5333688
1013	0.4750463	0.5249537
1014	0.4845293	0.5154707
1015	0.4958958	0.5041042
1016	0.5045606	0.4954394
1017	0.4971783	0.5028217
1018	0.4864088	0.5135912
1019	0.4998384	0.5001616
1020	0.5116944	0.4883056
1021	0.5026554	0.4973446
1022	0.4880486	0.5119514
1023	0.5026592	0.4973408
1024	0.5030859	0.4969141
1025	0.4897883	0.5102117
1026	0.4782685	0.5217315
1027	0.4893221	0.5106779
1028	0.5078775	0.4921225

$posterior

1221	0.4903737	0.5096263
1222	0.5081924	0.4918076
1223	0.5087487	0.4912513
1224	0.5012844	0.4987156
1225	0.5019254	0.4980746
1226	0.5009805	0.4990195
1227	0.4967487	0.5032513
1228	0.4810421	0.5189579
1229	0.4817958	0.5182042
1230	0.4848896	0.5151104
1231	0.5037268	0.4962732
1232	0.5064982	0.4935018
1233	0.4876756	0.5123244
1234	0.4879247	0.5120753
1235	0.4878553	0.5121447
1236	0.4849732	0.5150268
1237	0.4951269	0.5048731
1238	0.4976356	0.5023644
1239	0.4997661	0.5002339
1240	0.5029600	0.4970400
1241	0.4934127	0.5065873

	LD1
999	0.16677488
1000	0.55461363
1001	1.13260064
1002	0.72085843
1003	-0.22979082
1004	-0.20490817
1005	-0.09889344
1006	0.24743666
1007	0.06486872
1008	0.32246953
1009	0.02516951
1010	-0.93574001
1011	0.07652807
1012	1.15695011
1013	0.77212640
1014	0.33922812
1015	-0.17900833
1016	-0.57390470
1017	-0.23745710
1018	0.25349881
1019	-0.35868961
1020	-0.89911177
1021	-0.48707341
1022	0.17871946
1023	-0.48724516
1024	-0.50669331
1025	0.09939583
1026	0.62495544
1027	0.12065029
1028	-0.72509169

$x

1221	0.07270378
1222	-0.73944813
1223	-0.76480663
1224	-0.42459094
1225	-0.45380206
1226	-0.41073939
1227	-0.21787808
1228	0.49834065
1229	0.46394617
1230	0.32279164
1231	-0.53590499
1232	-0.66222229
1233	0.19572969
1234	0.18436844
1235	0.18753506
1236	0.31897808
1237	-0.14396271
1238	-0.25830048
1239	-0.35539601
1240	-0.50095431
1241	-0.06582911

```
In [19]: data.frame(lda.pred)
         table(lda.pred$class,Smarket.2005$Direction)
         mean(lda.pred$class==Smarket.2005$Direction)
```

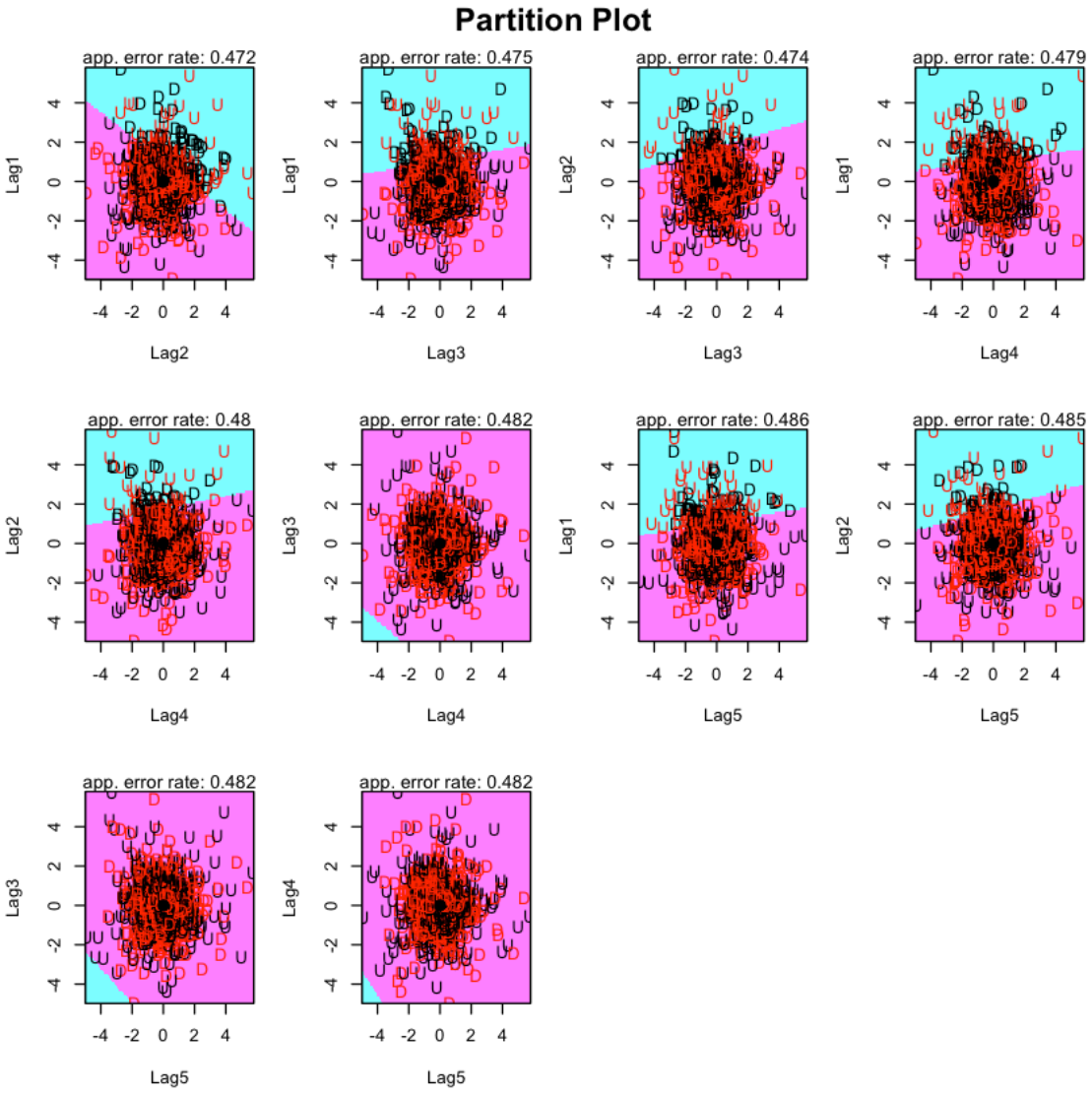
	class	posterior.Down	posterior.Up	LD1
999	Up	0.4883105	0.5116895	0.16677488
1000	Up	0.4798093	0.5201907	0.55461363
1001	Up	0.4671632	0.5328368	1.13260064
1002	Up	0.4761685	0.5238315	0.72085843
1003	Up	0.4970100	0.5029900	-0.22979082
1004	Up	0.4964641	0.5035359	-0.20490817
1005	Up	0.4941381	0.5058619	-0.09889344
1006	Up	0.4865417	0.5134583	0.24743666
1007	Up	0.4905456	0.5094544	0.06486872
1008	Up	0.4848967	0.5151033	0.32246953
1009	Up	0.4914164	0.5085836	0.02516951
1010	Down	0.5124976	0.4875024	-0.93574001
1011	Up	0.4902899	0.5097101	0.07652807
1012	Up	0.4666312	0.5333688	1.15695011
1013	Up	0.4750463	0.5249537	0.77212640
1014	Up	0.4845293	0.5154707	0.33922812
1015	Up	0.4958958	0.5041042	-0.17900833
1016	Down	0.5045606	0.4954394	-0.57390470
1017	Up	0.4971783	0.5028217	-0.23745710
1018	Up	0.4864088	0.5135912	0.25349881
1019	Up	0.4998384	0.5001616	-0.35868961
1020	Down	0.5116944	0.4883056	-0.89911177
1021	Down	0.5026554	0.4973446	-0.48707341
1022	Up	0.4880486	0.5119514	0.17871946
1023	Down	0.5026592	0.4973408	-0.48724516
1024	Down	0.5030859	0.4969141	-0.50669331
1025	Up	0.4897883	0.5102117	0.09939583
1026	Up	0.4782685	0.5217315	0.62495544
1027	Up	0.4893221	0.5106779	0.12065029
1028	Down	0.5078775	0.4921225	-0.72509169
1221	Up	0.4903737	0.5096263	0.07270378
1222	Down	0.5081924	0.4918076	-0.73944813
1223	Down	0.5087487	0.4912513	-0.76480663
1224	Down	0.5012844	0.4987156	-0.42459094
1225	Down	0.5019254	0.4980746	-0.45380206
1226	Down	0.5009805	0.4990195	-0.41073939
1227	Up	0.4967487	0.5032513	-0.21787808
1228	Up	0.4810421	0.5189579	0.49834065
1229	Up	0.4817958	0.5182042	0.46394617
1230	Up	0.4848896	0.5151104	0.32279164
1231	Down	0.5037268	0.4962732	-0.53590499
1232	Down	0.5064982	0.4935018	-0.66222229
1233	Up	0.4876756	0.5123244	0.19572969
1234	Up	0.4879247	0.5120753	0.18436844
1235	Up	0.4878553	0.5121447	0.18753506
1236	Up	0.4849732	0.5150268	0.31897808
1237	Up	0.4951269	0.5048731	-0.14396271
1238	Up	0.4976356	0.5023644	-0.25830048
1239	Up	0.4997661	0.5002339	-0.35539601
1240	Down	0.5029600	0.4970400	-0.50095431
1241	Up	0.4934127	0.5065873	-0.06582911

```
      Down  Up
Down    37  30
Up      74 111
```

0.587301587301587

Aquí podemos observar la matriz de confusión la media de aciertos, con lo que fijándonos en el lda usando solo los predictores Lag1 y Lag2 vemos que ha mejorado un poco. Ahora procederemos a pintar los resultados.

```
In [20]: library(klaR)
         partimat(Direction~Lag1+Lag2+Lag3+Lag4+Lag5, data=Smarket, method="lda")
```



0.2 Comparación LDA con Regresión Logística

Ahora vamos a comparar los resultados con los de regresión logística, la matriz de confusión y la media de acierto que habíamos obtenido fueron los siguientes:

```
glmProb
      Down  Up
Down  221   381
Up    190   458
```

0.5432

Con lo cual podemos concluir que en este caso lda se comporta un poco mejor que regresión logística para este caso, ya que los predictores tienen una varianza similar y los datos siguen aproximadamente una distribución normal.

0.2.1 QDA

En primer lugar vamos a comprobar que la varianza es similar pero esta vez para cada clase

```
In [11]: var(Smarket[Smarket$Direction == "Up",]$Lag1)
         var(Smarket[Smarket$Direction == "Up",]$Lag2)
         var(Smarket[Smarket$Direction == "Up",]$Lag3)
         var(Smarket[Smarket$Direction == "Up",]$Lag4)
         var(Smarket[Smarket$Direction == "Up",]$Lag5)

         var(Smarket[Smarket$Direction == "Down",]$Lag1)
         var(Smarket[Smarket$Direction == "Down",]$Lag2)
         var(Smarket[Smarket$Direction == "Down",]$Lag3)
         var(Smarket[Smarket$Direction == "Down",]$Lag4)
         var(Smarket[Smarket$Direction == "Down",]$Lag5)
```

1.27913706688038
1.24717074806801
1.27785074789389
1.22092377886303
1.36348271540061
1.30204143704291
1.33905195969342
1.31893272233984
1.38060525375758
1.26880333331491

Todos los Lag tienen varianza similar para la clase Up y para la clase Down, excepto de nuevo para el Lag5. Aplicaremos ahora QDA.

```
In [12]: # QDA
         qda.fit <- qda(Direction~Lag1+Lag2+Lag3+Lag4+Lag5, data=Smarket, subset=Year<2005)
         qda.fit
```

```
Call:
qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = Smarket,
    subset = Year < 2005)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2         Lag3         Lag4          Lag5
Down  0.04279022  0.03389409 -0.009806517 -0.010598778  0.0043665988
Up   -0.03954635 -0.03132544  0.005834320  0.003110454 -0.0006508876
```

Ahora predeciremos siguiendo el modelo ajustado y a continuación mostraremos los resultados de la matriz de confusión y la media de acierto.

```
In [13]: qda.pred <- predict(qda.fit,Smarket.2005)
         class(qda.pred)
         data.frame(qda.pred)
```

'list'

	class	posterior.Down	posterior.Up
999	Up	0.4940099	0.5059901
1000	Up	0.4592991	0.5407009
1001	Up	0.4611609	0.5388391
1002	Up	0.4685393	0.5314607
1003	Down	0.5157051	0.4842949
1004	Up	0.4923885	0.5076115
1005	Up	0.4673655	0.5326345
1006	Up	0.4882192	0.5117808
1007	Up	0.4893342	0.5106658
1008	Up	0.4736901	0.5263099
1009	Up	0.4941605	0.5058395
1010	Down	0.5112872	0.4887128
1011	Up	0.4936002	0.5063998
1012	Down	0.5170779	0.4829221
1013	Up	0.4325890	0.5674110
1014	Up	0.4985044	0.5014956
1015	Up	0.4772451	0.5227549
1016	Up	0.4861023	0.5138977
1017	Up	0.4886557	0.5113443
1018	Up	0.4958629	0.5041371
1019	Down	0.5061148	0.4938852
1020	Down	0.5243518	0.4756482
1021	Up	0.4897090	0.5102910
1022	Down	0.5093799	0.4906201
1023	Down	0.5091812	0.4908188
1024	Down	0.5028376	0.4971624
1025	Up	0.4817993	0.5182007
1026	Up	0.4855008	0.5144992
1027	Up	0.4869710	0.5130290
1028	Down	0.5183652	0.4816348
1221	Up	0.4832489	0.5167511
1222	Down	0.5218038	0.4781962
1223	Down	0.5024929	0.4975071
1224	Up	0.4896412	0.5103588
1225	Down	0.5132830	0.4867170
1226	Up	0.4903046	0.5096954
1227	Up	0.4953512	0.5046488
1228	Up	0.4694407	0.5305593
1229	Up	0.4768343	0.5231657
1230	Up	0.4658755	0.5341245
1231	Down	0.5131627	0.4868373
1232	Up	0.4970743	0.5029257
1233	Up	0.4881669	0.5118331
1234	Down	0.5033895	0.4966105
1235	Up	0.4580498	0.5419502
1236	Up	0.4861135	0.5138865
1237	Up	0.4932770	0.5067230
1238	Down	0.5001412	0.4998588
1239	Up	0.4862707	0.5137293
1240	Down	0.5030516	0.4969484
1241	Up	0.4904194	0.5095806

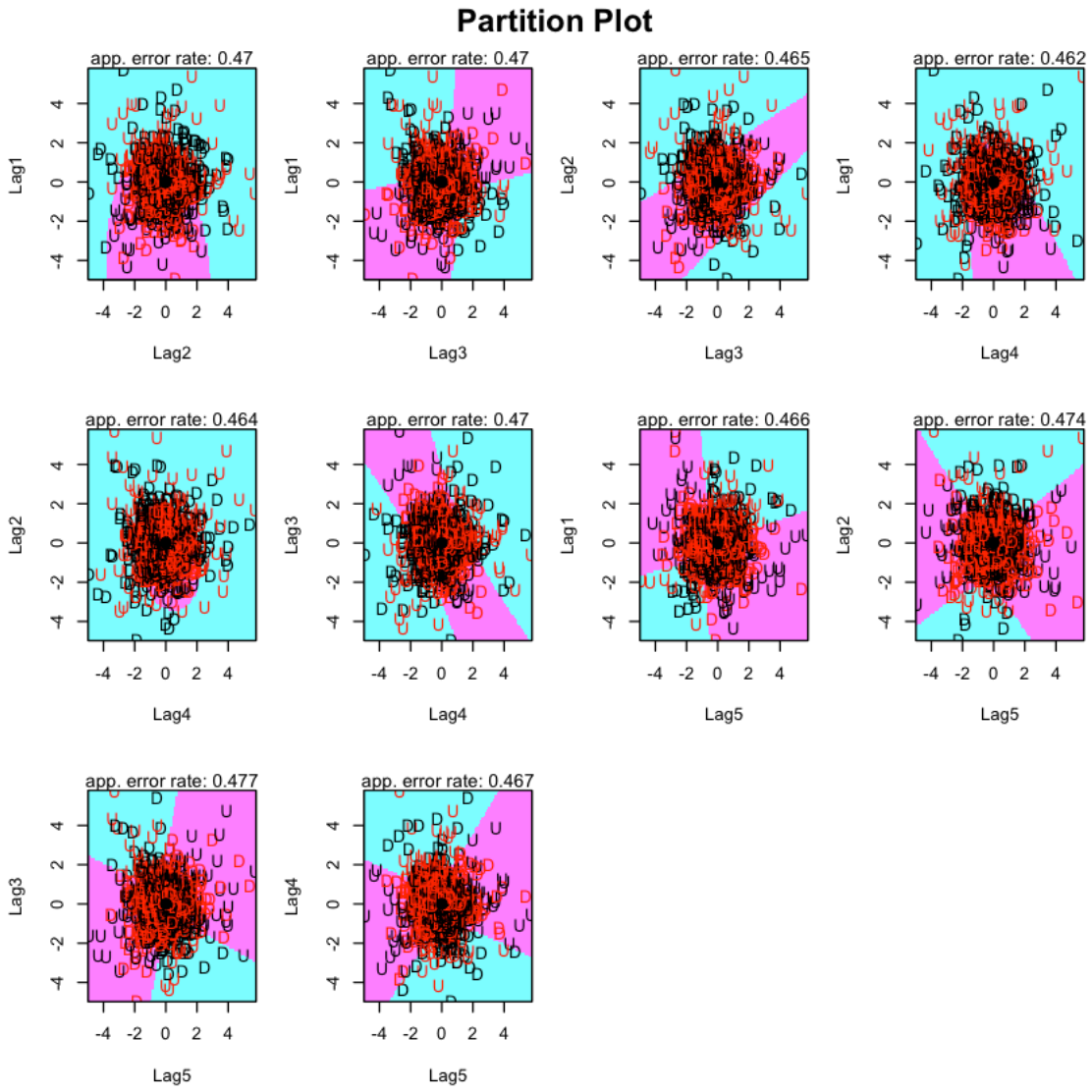
```
In [14]: table(qda.pred$class,Smarket.2005$Direction)
         mean(qda.pred$class==Smarket.2005$Direction)
```

```
      Down  Up
Down    37  35
Up      74 106
```

0.567460317460317

Como podemos observar QDA nos da peor resultado que LDA pero mejor que regresión logística para este dataset, sin embargo sigue sin ser demasiado bueno. Además, usando solo los predictores L1 Y L2 nos da una mejor tasa de acierto que usando todas los predictores.

```
In [21]: partimat(Direction~Lag1+Lag2+Lag3+Lag4+Lag5, data=Smarket ,method="qda")
```



Observando las gráficas y los resultados podemos deducir que nos encontramos ante una clasificación que no es para nada lineal, de ahí que tanto regresión logística como LDA no funcionen bien. Además, tampoco podríamos aplicar QDA por lo que podemos ver en la distribución de las muestras y en los resultados, por lo que podemos deducir que el método de clasificación de los vistos hasta ahora que mejor resultado nos podría dar a priori sería kNN.

0.3 Exercise 2

Using only the information in file clasif_train_alumnos.csv: * Compare lda and qda using Wilcoxon. * Perform a multiple comparison using Friedman. * Using Holm see if there is a winning algorithm (even if Friedman says there is no chance...).

```
In [1]: #Cargamos el archivo
        setwd("/Users/jramongomez/Master/IntroduccionDataScience/CLASIFICACION")
        resultados <- read.csv("clasif_train_alumnos.csv")
        tablatst <- cbind(resultados[,2:dim(resultados)[2]])
        colnames(tablatst) <- names(resultados)[2:dim(resultados)[2]]
        rownames(tablatst) <- resultados[,1]
        tablatst
```

	out_train_knn	out_train_lda	out_train_qda
appendicitis	0.8834602	0.8815461	0.8690241
australian	0.7277419	0.8605475	0.8072464
balance	0.9072122	0.8791122	0.9167999
bupa	0.7405521	0.7024224	0.6447628
contraceptive	0.6168944	0.5236485	0.5314180
haberman	0.7795116	0.7519934	0.7567115
hayes-roth	0.6475524	0.5604167	0.7361111
heart	0.7342975	0.8576132	0.8777778
iris	0.9791045	0.9800000	0.9814815
led7digit	0.7636971	0.7635556	0.7680556
mammographic	0.8160856	0.8274465	0.8196843
monk-2	0.9793684	0.7821826	0.9303010
newthyroid	0.9158409	0.9183457	0.9700283
pima	0.7791914	0.7792266	0.7633125
tae	0.5263460	0.5584858	0.5688072
titanic	0.7892319	0.7760111	0.7732851
vehicle	0.7213300	0.7989229	0.9123989
vowel	0.8378652	0.6457912	0.9701459
wine	0.7745126	1.0000000	0.9956250
wisconsin	0.9739304	0.9614471	0.9588436

0.3.1 Wilcoxon test

Wilcoxon ordena por las diferencias para los rankings. En clasificación los errores están en el mismo rango, así que no tenemos que normalizar el error. Realizaremos la comparativa por pares:

```
In [2]: LdavsQdatst <- wilcox.test(tablatst$out_train_lda, tablatst$out_train_qda,
                                   alternative = "two.sided", paired=TRUE)
        Rmas <- LdavsQdatst$statistic
        pvalue <- LdavsQdatst$p.value
        LdavsQdatst <- wilcox.test(tablatst$out_train_qda, tablatst$out_train_lda,
                                   alternative = "two.sided", paired=TRUE)
        Rmenos <- LdavsQdatst$statistic

        Rmas
        Rmenos
        pvalue
```

V: 68
V: 142
0.176853179931641
Podemos deducir que si hay diferencias significativas entre ambos algoritmos:

```
In [18]: confianza = (1-pvalue) * 100
         confianza
```

82.3146820068359
Tenemos un 82.3% de confianza de que ambos algoritmos sean distintos.

0.3.2 Friedman

```
In [19]: test_friedman <- friedman.test(as.matrix(tablatst))
         test_friedman
```

```
Friedman rank sum test

data:  as.matrix(tablatst)
Friedman chi-squared = 1.3, df = 2, p-value = 0.522
```

No podemos decir según Friedman si existe un ganador claro ya que la confianza de que existan diferencias significativas es del 48%, lo cual no es un porcentaje muy alto.

0.3.3 Holm

```
In [20]: tam <- dim(tablatst)
         groups <- rep(1:tam[2], each=tam[1])
         pairwise.wilcox.test(as.matrix(tablatst), groups, p.adjust = "holm", paired = TRUE)
```

```
Pairwise comparisons using Wilcoxon signed rank test

data:  as.matrix(tablatst) and groups
```

```
  1    2
2 0.65 -
3 0.59 0.53

P value adjustment method: holm
```

Según Holm podemos decir que los 3 métodos pueden ser considerados equivalentes para este conjunto y que no existe un ganador claro.