Análisis de acciones SP500

Gómez Jiménez Aaron Mauricio

2023-05-25

La base de datos SP500.txt contiene el porcentaje de retornos desde inicios del 2001 a finales de 2005. Para cada fecha se tiene el porcentaje de retornos record para cada uno de los 5 días previos, el volumen de transacciones del día previo, el porcentaje de retorno del día actual y un indicador binario de si el mercado iba hacia arriba o hacia abajo en esa fecha.

Realizaremos un análisis de discriminante para desarrollar predicciones sobre las acciones de acuerdo a los datos disponibles.

Análisis Exploratorio

```
head(datos)
```

```
##   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
```

```
describe(datos)
```

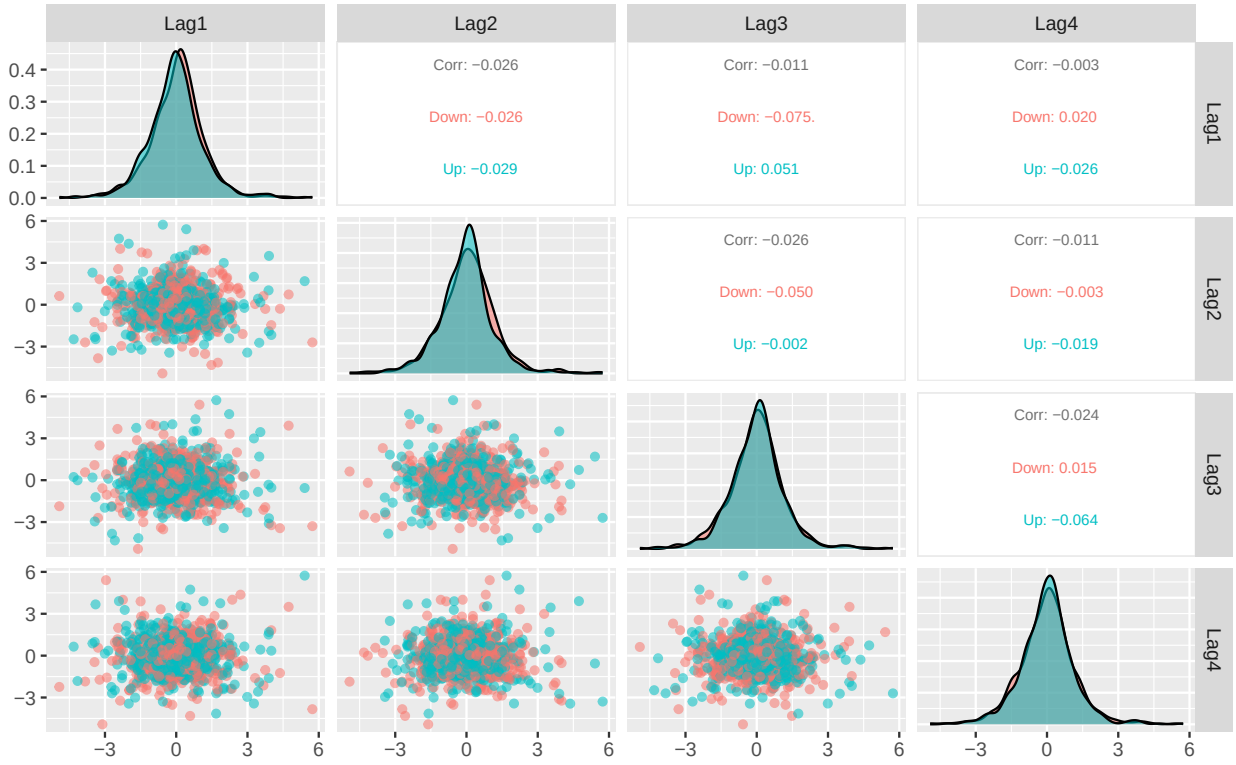
```
##            vars    n    mean   sd  median trimmed  mad     min     max range
## Year          1 1250 2003.02 1.41 2003.00 2003.02 1.48 2001.00 2005.00  4.00
## Lag1          2 1250    0.00 1.14    0.04    0.00 0.91   -4.92    5.73 10.65
## Lag2          3 1250    0.00 1.14    0.04    0.00 0.91   -4.92    5.73 10.65
## Lag3          4 1250    0.00 1.14    0.04    0.00 0.91   -4.92    5.73 10.65
## Lag4          5 1250    0.00 1.14    0.04    0.00 0.91   -4.92    5.73 10.65
## Lag5          6 1250    0.01 1.15    0.04    0.00 0.92   -4.92    5.73 10.65
## Volume        7 1250    1.48 0.36    1.42    1.45 0.28    0.36    3.15  2.80
## Today         8 1250    0.00 1.14    0.04    0.00 0.91   -4.92    5.73 10.65
## Direction*    9 1250    1.52 0.50    2.00    1.52 0.00    1.00    2.00  1.00
##             skew kurtosis   se
## Year       -0.01    -1.29 0.04
## Lag1        0.20     2.38 0.03
## Lag2        0.20     2.38 0.03
## Lag3        0.19     2.36 0.03
## Lag4        0.19     2.36 0.03
## Lag5        0.25     2.48 0.03
```

```
## Volume      0.82     1.44 0.01
## Today       0.20     2.38 0.03
## Direction* -0.07    -2.00 0.01
```

Podemos notar que para las variables Lag 1, Lag2, Lag 3, Lag 4 y Lag 5 las estadísticas obtenidas son muy parecidas y en la mayoría de los casos iguales para estas variables.

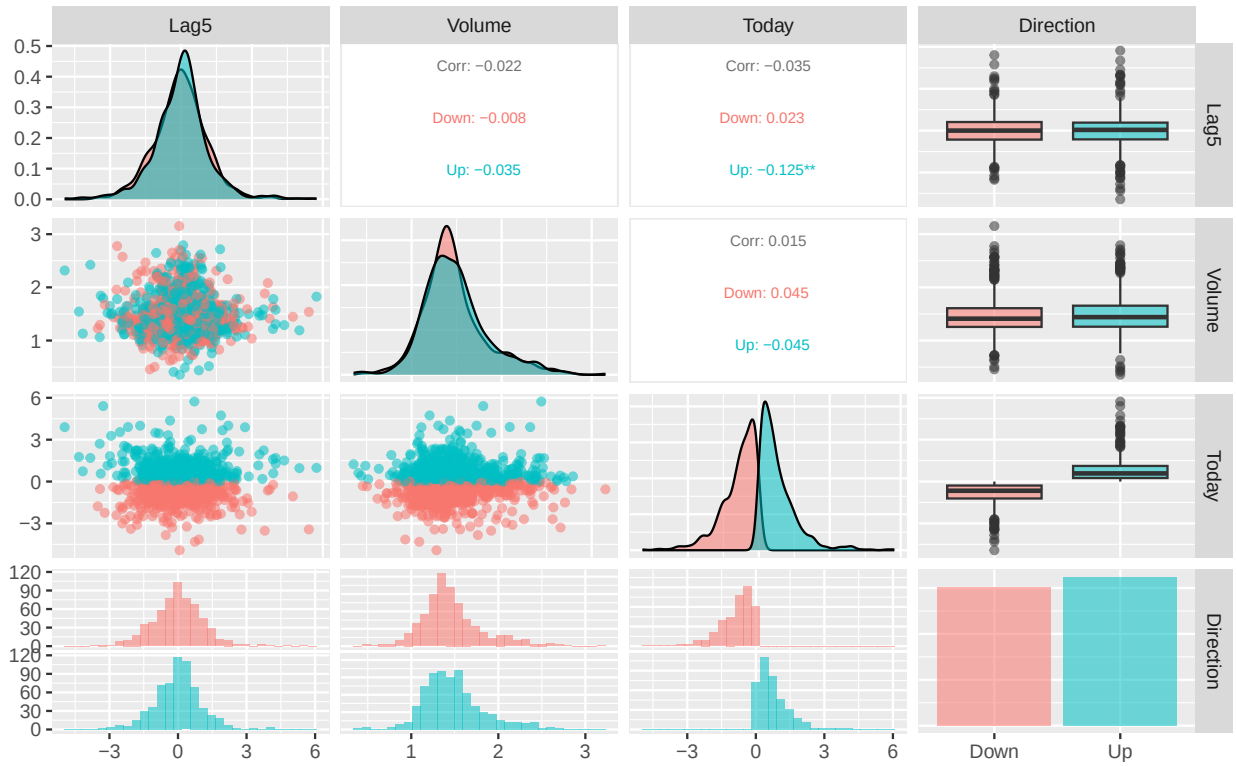
Visualizaremos estas variables entre sí para ver si estan relacionadas

```
ggpairs(datos,columns=2:5,aes(color=Direction,alpha=0.5),
upper= list(continuous= wrap("cor",size=2.5)))
```



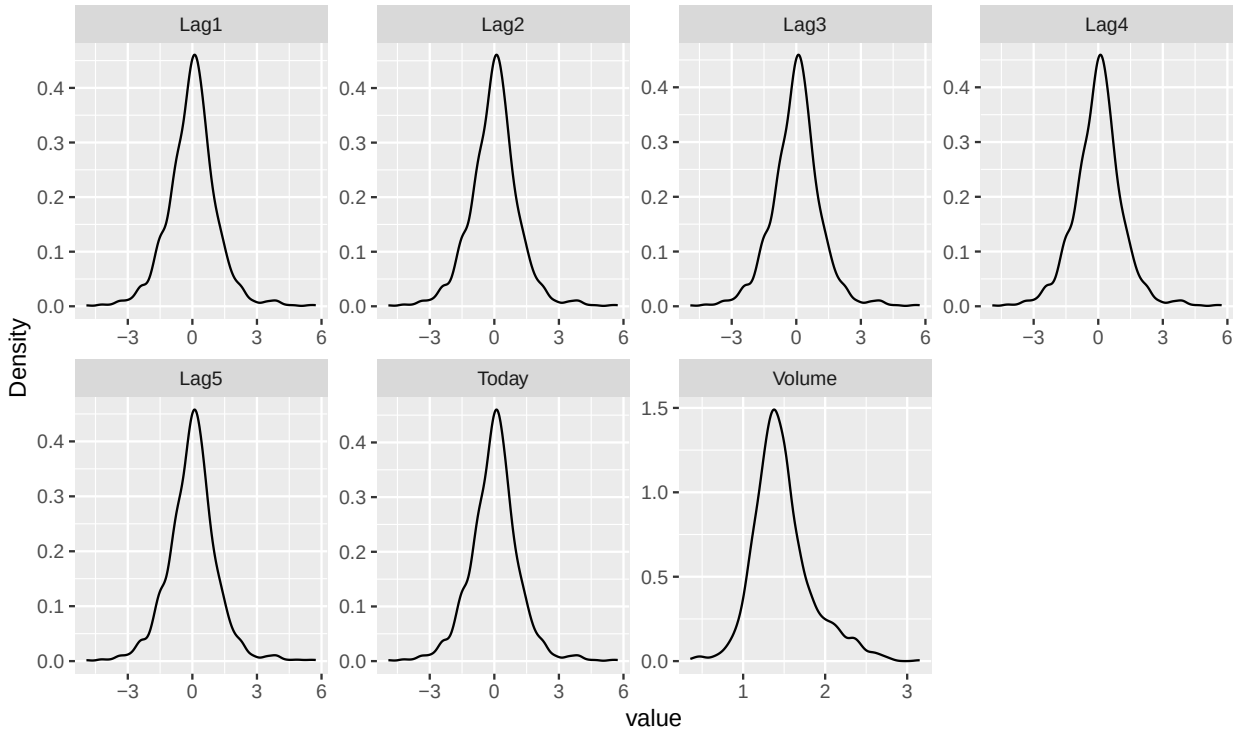

```
ggpairs(datos, columns = 6:9, aes(color = Direction, alpha = 0.5),
        upper = list(continuous = wrap("cor", size = 2.5)))
```

```
## `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`.
```



Graficaremos su densidad para ver como se comporta y sí podemos deducir normalidad.

```
plot_density(datos[, -1])
```



No podemos asegurar Normalidad ya que las gráficas indican que no se cumple este criterio, por lo tanto haremos prueba de hipotesis para concluir nuestra infenrencia.

Normalidad de las variables

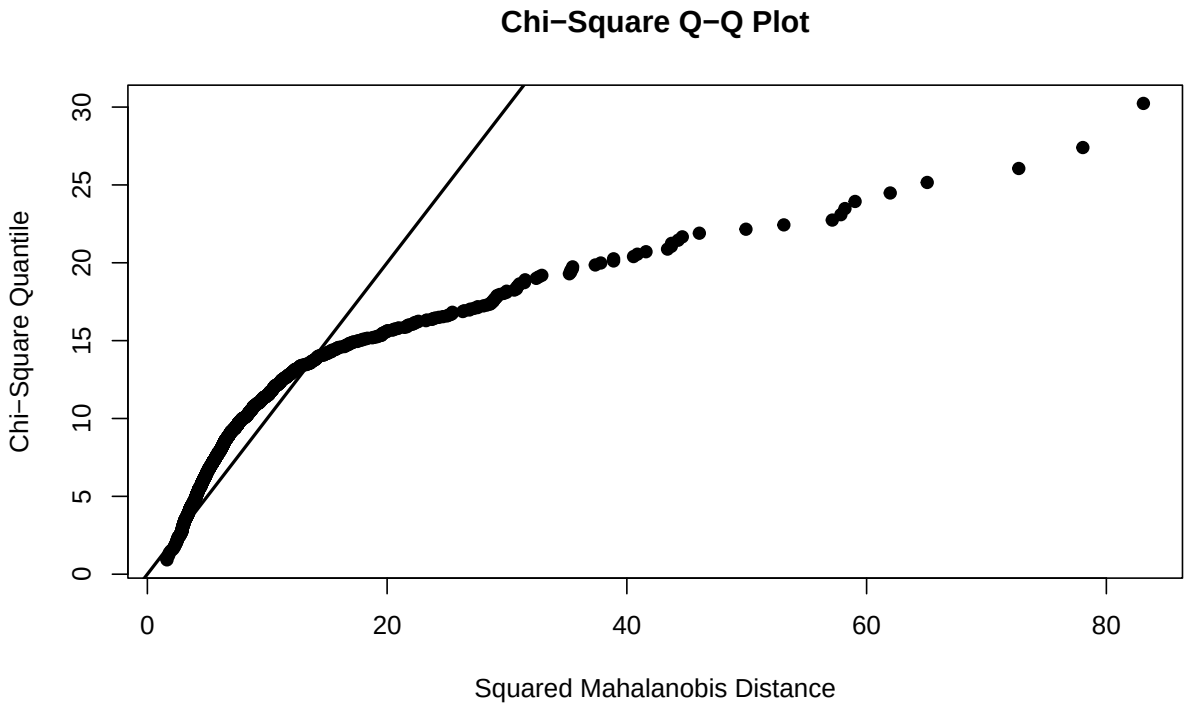
Verificaremos la Normalidad de las variables ya que este es un suspuesto en el análisis de discriminante lineal, como podemos observar la ultima variable no es númerica, asi que la convertiremos en una variable binaria de 0 y 1

```
datos_1=dplyr::select(datos, - c(Direction, valor_Down))
datos_1=datos_1%>%
  rename(Direction= valor_Up)
head(datos_1)
```

```
##   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         1
## 2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032         1
## 3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623         0
## 4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614         1
## 5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213         1
## 6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392         1
```

Donde el valor de la variable Direction es 1 si el valor subió y 0 si el valor bajo.

```
royston_test <- mvn(data = datos_1, mvnTest = "royston", multivariatePlot = "qq")
```



```
royston_test$univariateNormality
```

```
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling   Year      44.3997  <0.001      NO
## 2 Anderson-Darling   Lag1       7.3400  <0.001      NO
## 3 Anderson-Darling   Lag2       7.3504  <0.001      NO
## 4 Anderson-Darling   Lag3       7.3387  <0.001      NO
## 5 Anderson-Darling   Lag4       7.3326  <0.001      NO
## 6 Anderson-Darling   Lag5       7.7458  <0.001      NO
## 7 Anderson-Darling  Volume     18.8928  <0.001      NO
## 8 Anderson-Darling   Today      7.3073  <0.001      NO
## 9 Anderson-Darling Direction  224.7929  <0.001      NO
```

```
royston_test$multivariateNormality
```

```
##      Test        H       p value MVN
## 1 Royston 775.1462 4.345398e-161  NO
```

Podemos concluir que no existe normalidad univariada ni multivariada, por lo cual podemos intuir que el discriminante lineal no hará una buena clasificación de los datos ya que este método no es robusto en ese sentido.

Bases de Entrenamiento y Prueba

Primero entrenaremos nuestro modelo con datos del 2001-2004 y nuestros datos de prueba serán los del año 2005.

Solo utilizaremos 3 variables para nuestro modelo, ya que la variable Direction es una clasificación binaria de la variable Today además que para las variables Lag4 y Lag5 los datos son muy parecidos

```
datos_entrena=filter(datos_1, Year <= 2004  )
datos_prueba=filter(datos_1, Year > 2004)
```

Creación del modelo lineal

Creamos el modelo lineal con las variables Lag1, Lag2 y Lag3 para la clasificación de la variable Dirección, es decir si sbe o baja el valor de la acción.

```
modelo_lin=lda(Direction ~ Lag1 + Lag2 + Lag3, data= datos_entrena)
modelo_lin
```

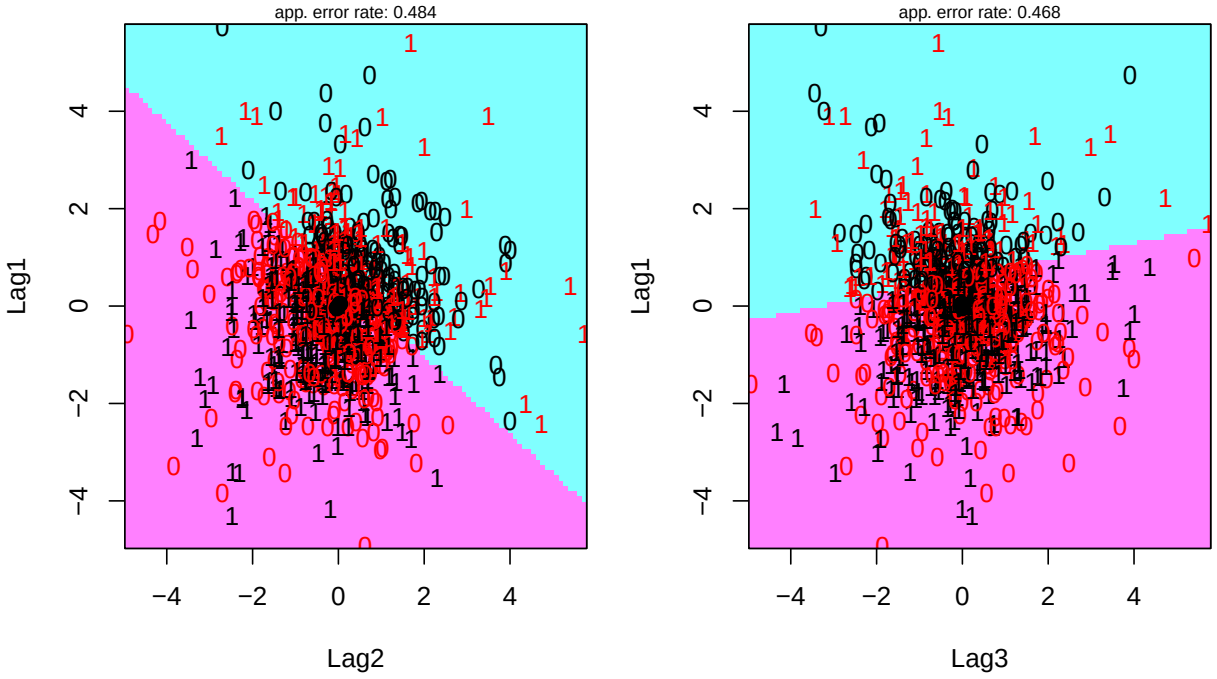
```
## Call:
## lda(Direction ~ Lag1 + Lag2 + Lag3, data = datos_entrena)
##
## Prior probabilities of groups:
##        0        1
## 0.491984 0.508016
##
## Group means:
##          Lag1        Lag2         Lag3
## 0  0.04279022  0.03389409 -0.009806517
## 1 -0.03954635 -0.03132544  0.005834320
##
```

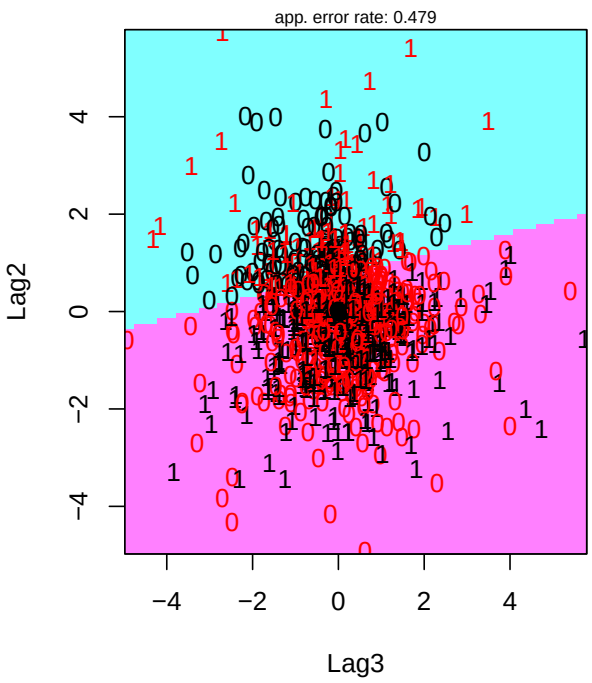
```
## Coefficients of linear discriminants:
##              LD1
## Lag1 -0.6359074
## Lag2 -0.5074032
## Lag3  0.1011166
```

Al interpertar el modelo obtenemos que los datos de la variable Direction tiene casi la misma probabilidad de estar en el grupo donde subió o bajo la acción

Graficaremos el modelo

```
partimat(factor(Direction) ~ Lag1+Lag2+Lag3, data = datos_entrena, method= "lda", plot.matrix=FALSE)
```





Predicción

Realizamos la predicción de los valores con los datos de prueba

```
predicción= predict(modelo_lin, datos_prueba)
predicción$class[1:20]
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1
## Levels: 0 1
```

Como podemos observar la predicción es que los valores suban en los primeros 10 observaciones, para visualizarlo de mejor manera lo hacemos un data frame

```
as.data.frame(predicción)[1:20,]
```

```
##    class posterior.0 posterior.1         LD1
## 1      1   0.4901932   0.5098068  0.08163079
## 2      1   0.4792223   0.5207777  0.58634400
## 3      1   0.4671730   0.5328270  1.14175192
## 4      1   0.4758412   0.5241588  0.74205347
## 5      1   0.4953589   0.5046411 -0.15583293
## 6      1   0.4946377   0.5053623 -0.12268282
## 7      1   0.4943235   0.5056765 -0.10824352
## 8      1   0.4875951   0.5124049  0.20109821
## 9      1   0.4899635   0.5100365  0.09219289
## 10     1   0.4857416   0.5142584  0.28634677
```

```
## 11     1   0.4898421   0.5101579  0.09777512
## 12     0   0.5138411   0.4861589 -1.00544169
## 13     1   0.4881697   0.5118303  0.17467344
## 14     1   0.4686145   0.5313855  1.07522750
## 15     1   0.4765891   0.5234109  0.70760293
## 16     1   0.4817012   0.5182988  0.47223830
## 17     1   0.4949900   0.5050100 -0.13887911
## 18     0   0.5038316   0.4961684 -0.54524834
## 19     1   0.4969730   0.5030270 -0.23002247
## 20     1   0.4875692   0.5124308  0.20228994
```

Obteniendo la matriz de confusión

```
table(predicción$class, datos_prueba$Direction)
```

```
##
##       0   1
##   0  38  31
##   1  73 110
```

Sacando el promedio de exactitud de la predicción

```
mean(predicción$class== datos_prueba$Direction)
```

```
## [1] 0.5873016
```

Podemos notar que se acerca al 60% de exactitud en la predicción, por lo tanto nuestro modelo no es muy bueno, es algo de esperarse ya que no se cumplen los supuestos de Normalidad en las variables.

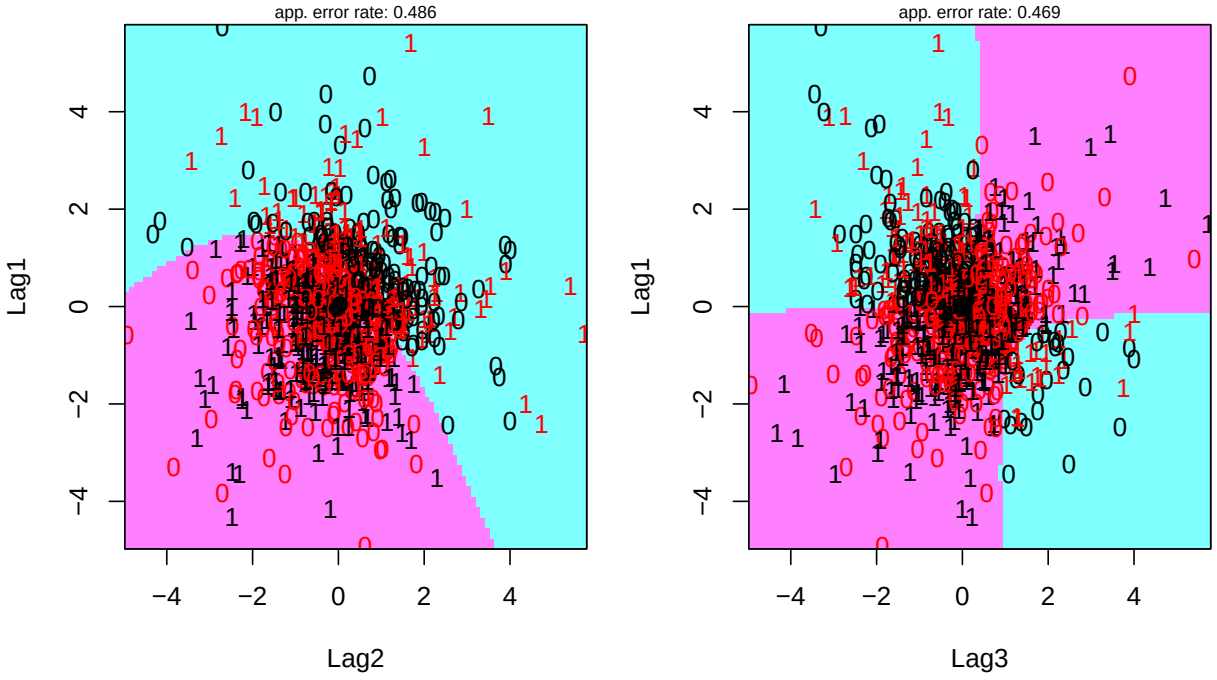
Creación del Modelo Cuadratico

Ya que el modelo lineal no fue tan bueno haremos un modelo cuadratico

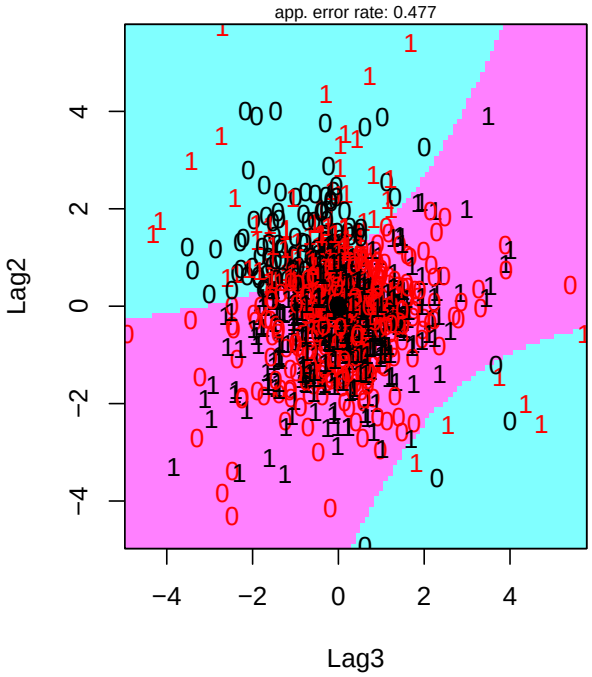
```
modelo_cua=qda(Direction ~ Lag1 + Lag2 + Lag3, data= datos_entrena)
modelo_cua
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2 + Lag3, data = datos_entrena)
##
## Prior probabilities of groups:
##        0        1
## 0.491984 0.508016
##
## Group means:
##          Lag1        Lag2         Lag3
## 0  0.04279022  0.03389409 -0.009806517
## 1 -0.03954635 -0.03132544  0.005834320
```

```
partimat(factor(Direction) ~ Lag1+Lag2+Lag3, data = datos_entrena, method= "qda", plot.matrix=FALSE)
```



Partition Plot



Hacemos la Predicción con el modelo cudratico usando la muestra de prueba

```
predicción_cua= predict(modelo_cua, datos_prueba)
predicción_cua$class[1:10]
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```

```
as.data.frame(predicción_cua)[1:10,]
```

```
##    class posterior.0 posterior.1
## 1      1   0.4869749   0.5130251
## 2      1   0.4754586   0.5245414
## 3      1   0.4586606   0.5413394
## 4      1   0.4628359   0.5371641
## 5      1   0.4991439   0.5008561
## 6      1   0.4914760   0.5085240
## 7      1   0.4889275   0.5110725
## 8      1   0.4831046   0.5168954
## 9      1   0.4863058   0.5136942
## 10     1   0.4729578   0.5270422
```

Haciendo la matriz de confusión

```
table(predicción_cua$class, datos_prueba$Direction)
```

```
##
##       0   1
##   0  26  21
##   1  85 120
```

```
mean(predicción_cua$class== datos_prueba$Direction)
```

```
## [1] 0.5793651
```

Conclusiones

```
mean(predicción$class== datos_prueba$Direction)
```

```
## [1] 0.5873016
```

```
mean(predicción_cua$class== datos_prueba$Direction)
```

```
## [1] 0.5793651
```

En conclusión podemos observar que ambos discriminadores tienen una precisión similar de casi el 60%, es importante mencionar que este valor puedo ser afectado ya que no existe normalidad en las variables, el cual es un supuesto al aplicar análisis de discriminante, los resultados obtenidos no son concluyentes sobre la predicción del valor de las acciones.