> require(ISLR)

Loading required package: ISLR

> names(Smarket)

[1] "Year" "Lag1" "Lag2" "Lag3" "Lag4" "Lag5"

[7] "Volume" "Today" "Direction"

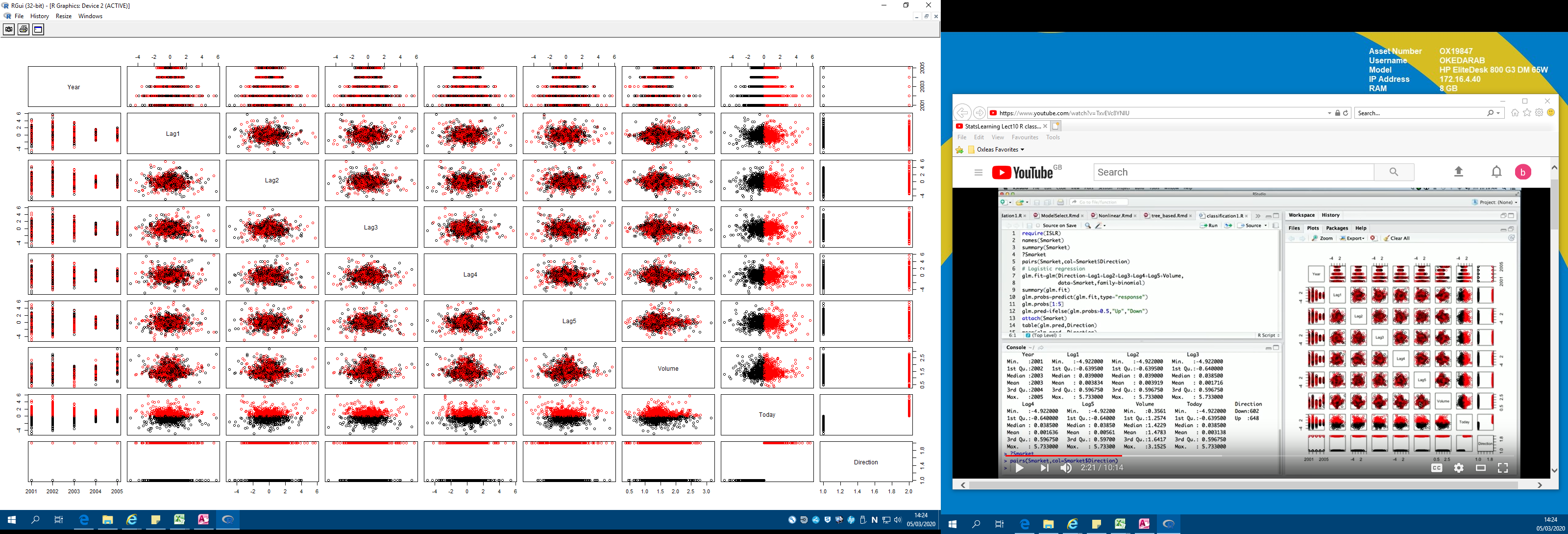
> summary(Smarket)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Year** | **Lag1** | **Lag2** | **Lag3** | **Lag4** | **Lag5** | **Volume** | **Today** |
| **Min** | 2001 | 4.922000 | -4.922000 | -4.922000 | -4.922000 | -4.92200 | 0.3561 | -4.922000 |
| **1st qua** | 2002 | -0.639500 | -0.639500 | -0.640000 | -0.640000 | -0.64000 | 1.2574 | -0.639500 |
| **Median** | 2002 | 0.039000 | 0.039000 | 0.038500 | 0.038500 | 0.03850 | 1.4229 | 0.038500 |
| **Mean** | 2003 | 0.003834 | 0.003919 | 0.001716 | 0.001636 | 0.00561 | 1.4783 | 0.003138 |
| **3rd quartile** | 2004 | 0.596750 | 0.596750 | 0.596750 | 0.596750 | 0.59700 | 1.6417 | 0.596750 |
| **Max** | 2005 | 5.733000 | 5.733000 | 5.733000 | 5.733000 | 5.73300 | 3.1525 | 5.733000 |

Direction

* Down: 602
* Up: 648

# Plot the variables in ‘smarket’ and use colour as a binary response indicator

> pairs(Smarket,col=Smarket$Direction)

# Logistic regression (aka Logit model)

# The response is ‘Direction’ - whether the stock market goes up or down

The predictors are the ‘LagN’ and ‘volume’ variables.

‘Family = binomial’ tells the Generalised linear model (glm) to fit a logistic regression model

> glm.fit-glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial)

Error in glm.fit - glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + :

non-numeric argument to binary operator

> glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial)

> summary(glm.fit)

Call: glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +

Volume, family = binomial, data = Smarket)

Deviance Residuals:

# Measure of model fit- shows the distribution of the deviance residuals for individual cases used in the model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

# Note that none of the Coefficients are significant. It is possible that this is due to some of the variables being correlated.

# For every one unit change in Lag5, the log odds of ‘up direction’ (versus down direction) increases by 0.010

(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 (# Deviance for the model with all the predictors in)

AIC: 1741.6

Number of Fisher Scoring iterations: 3

# Making predictions based on this model

# This will make predictions on the training data that we used to fit the model and give a vector of fitted probabilities

> glm.probs=predict(glm.fit,type="response")

> glm.probs[1:5]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 0.5070841 | 0.4814679 | 0.4811388 | 0.5152224 | 0.5107812 |

# Turn probabilities into classifications using ‘ifelse’ command. Tell R that if probability is less than 0.5, label it as ‘down’. If it is greater than 0.5, label it as ‘up’.

> glm.pred=ifelse(glm.probs>0.5,"Up","Down")

> attach(Smarket) #Ensures variables available by name

> table(glm.pred,Direction)

|  |  |  |
| --- | --- | --- |
|  | Direction: Down | Direction: Up |
| glm.pred: Down | 145 | 141 |
| glm.pred: Up | 457 | 507 |

\*\*\*\*# Note that the diagonals show correct classification, off-diagonals - errors

> mean(glm.pred==Direction) # Mean classification performance – cases where glm.pred is equal to the true direction

[1] 0.5216

# Note the model is only performing slightly better than chance, it is possible the model is overfitted

> # Make training and test set

> train = Year<2005 # For every observation that the year is before 2005, its marked as ‘True’, those after 2005 marked as ‘False’

>glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial,subset=train) # Model will only include observations before 2005

> glm.probs=predict(glm.fit,newdata=Smarket[!train,],type="response") # Predicting observations not in ‘train’ set – observations of > 2005.

> glm.pred=ifelse(glm.probs >0.5,"Up","Down")

> Direction.2005=Smarket$Direction[!train]

> 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

> # This model is predicting worse than null rate which is 0.50/50%, so we may have overfit the data. So will fit a smaller model

> # Fitting smaller model – using only Lag 1 and 2 variables

> glm.fit=glm(Direction~Lag1+Lag2,data=Smarket,family=binomial,subset=train)

> glm.probs=predict(glm.fit,newdata=Smarket[!train,],type="response")

> glm.pred=ifelse(glm.probs >0.5,"Up","Down")

Error: unexpected '>' in ">"

> glm.pred=ifelse(glm.probs >0.5,"Up","Down")

> 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

> summary(glm.fit)

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

> # Neither of the predictors became significant but we did get a better prediction of 56% with this smaller model.