

# Regresión Logística

Cleber Perez

2024-11-05

## Lectura Datos

```
library(ISLR)
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse
2.0.0 —
## ✓ dplyr      1.1.4    ✓ readr      2.1.5
## ✓ forcats    1.0.0    ✓ stringr    1.5.1
## ✓ ggplot2    3.5.1    ✓ tibble     3.2.1
## ✓ lubridate  1.9.3    ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts —
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

## Análisis de los datos

```
head(Weekly)
```

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
## 1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
## 2	1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
## 3	1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up
## 4	1990	3.514	-2.576	-0.270	0.816	1.572	0.1616300	0.712	Up
## 5	1990	0.712	3.514	-2.576	-0.270	0.816	0.1537280	1.178	Up
## 6	1990	1.178	0.712	3.514	-2.576	-0.270	0.1544440	-1.372	Down

```
glimpse(Weekly)
```

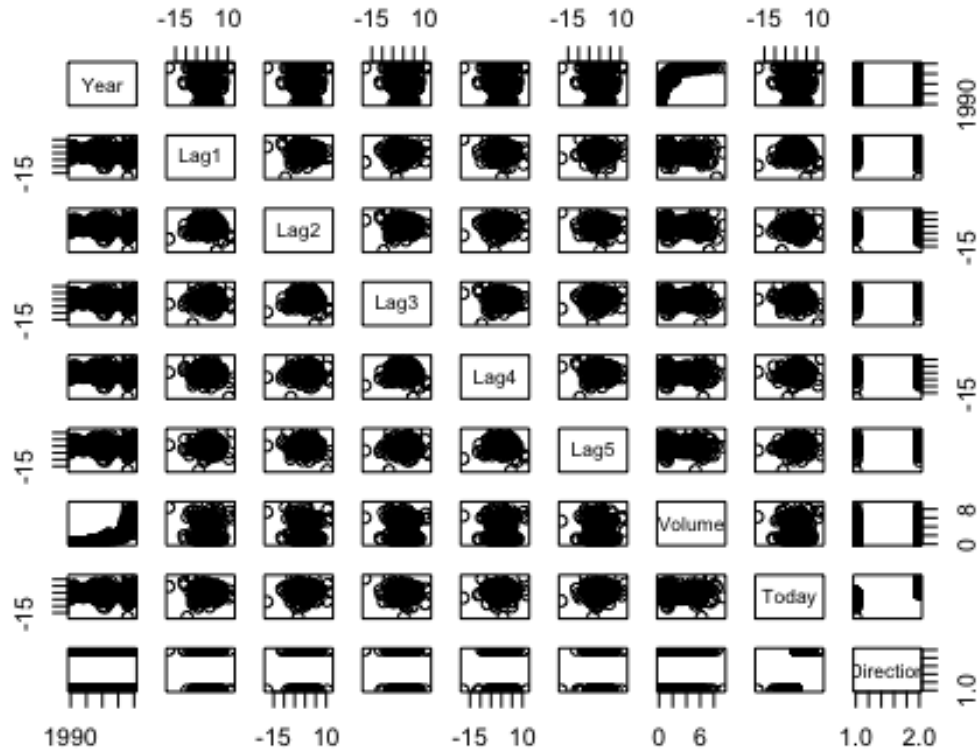
```
## Rows: 1,089
## Columns: 9
## $ Year      <dbl> 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990,
1990, ...
## $ Lag1      <dbl> 0.816, -0.270, -2.576, 3.514, 0.712, 1.178, -1.372,
0.807, 0...
## $ Lag2      <dbl> 1.572, 0.816, -0.270, -2.576, 3.514, 0.712, 1.178, -
1.372, 0...
## $ Lag3      <dbl> -3.936, 1.572, 0.816, -0.270, -2.576, 3.514, 0.712,
1.178, -...
```

```
## $ Lag4      <dbl> -0.229, -3.936, 1.572, 0.816, -0.270, -2.576, 3.514,
0.712, ...
## $ Lag5      <dbl> -3.484, -0.229, -3.936, 1.572, 0.816, -0.270, -2.576,
3.514,...
## $ Volume    <dbl> 0.1549760, 0.1485740, 0.1598375, 0.1616300, 0.1537280,
0.154...
## $ Today     <dbl> -0.270, -2.576, 3.514, 0.712, 1.178, -1.372, 0.807,
0.041, 1...
## $ Direction <fct> Down, Down, Up, Up, Up, Down, Up, Up, Up, Down, Down,
Up, Up...
```

**summary**(Weekly)

```
##      Year      Lag1      Lag2      Lag3
## Min.   :1990   Min.   :-18.1950 Min.   :-18.1950 Min.   :-18.1950
## 1st Qu.:1995   1st Qu.: -1.1540  1st Qu.: -1.1540  1st Qu.: -1.1580
## Median :2000   Median :  0.2410  Median :  0.2410  Median :  0.2410
## Mean   :2000   Mean    :  0.1506  Mean    :  0.1511  Mean    :  0.1472
## 3rd Qu.:2005   3rd Qu.:  1.4050  3rd Qu.:  1.4090  3rd Qu.:  1.4090
## Max.    :2010   Max.    : 12.0260  Max.    : 12.0260  Max.    : 12.0260
##      Lag4      Lag5      Volume      Today
## Min.   :-18.1950 Min.   :-18.1950 Min.   :0.08747 Min.   :-18.1950
## 1st Qu.: -1.1580 1st Qu.: -1.1660 1st Qu.:0.33202 1st Qu.: -1.1540
## Median :  0.2380 Median :  0.2340 Median :1.00268 Median :  0.2410
## Mean    :  0.1458 Mean    :  0.1399 Mean    :1.57462 Mean    :  0.1499
## 3rd Qu.:  1.4090 3rd Qu.:  1.4050 3rd Qu.:2.05373 3rd Qu.:  1.4050
## Max.    : 12.0260 Max.    : 12.0260 Max.    :9.32821 Max.    : 12.0260
## Direction
## Down:484
## Up  :605
##
##
##
##
```

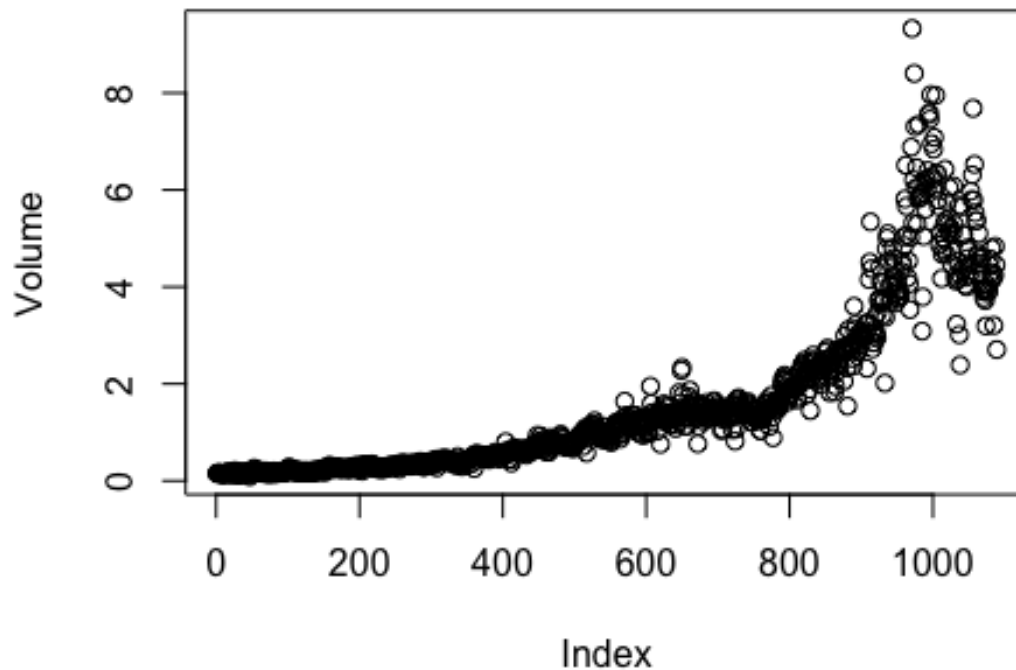
**pairs**(Weekly)



```
cor(Weekly[, -9])
```

```
##           Year           Lag1           Lag2           Lag3           Lag4
## Year      1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1     -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2     -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3     -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4     -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5     -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume    0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today    -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##           Lag5           Volume           Today
## Year     -0.030519101  0.84194162 -0.032459894
## Lag1     -0.008183096 -0.06495131 -0.075031842
## Lag2     -0.072499482 -0.08551314  0.059166717
## Lag3      0.060657175 -0.06928771 -0.071243639
## Lag4     -0.075675027 -0.06107462 -0.007825873
## Lag5      1.000000000 -0.05851741  0.011012698
## Volume   -0.058517414  1.00000000 -0.033077783
## Today     0.011012698 -0.03307778  1.000000000
```

```
attach(Weekly)
plot(Volume)
```



## Cálculo del modelo logístico

Modelo con todos los predictores, excluyendo "Today"

```
modelo.log.m <- glm(Direction ~ . - Today, data
= Weekly, family = binomial)
summary(modelo.log.m)
```

```
##
## Call:
## glm(formula = Direction ~ . - Today, family = binomial, data = Weekly)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 17.225822  37.890522   0.455   0.6494
## Year        -0.008500   0.018991  -0.448   0.6545
## Lag1        -0.040688   0.026447  -1.538   0.1239
## Lag2         0.059449   0.026970   2.204   0.0275 *
## Lag3        -0.015478   0.026703  -0.580   0.5622
## Lag4        -0.027316   0.026485  -1.031   0.3024
## Lag5        -0.014022   0.026409  -0.531   0.5955
## Volume       0.003256   0.068836   0.047   0.9623
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.2  on 1081  degrees of freedom
## AIC: 1502.2
##
## Number of Fisher Scoring iterations: 4

contrasts(Direction)

##      Up
## Down  0
## Up    1

confint(object = modelo.log.m, level = 0.95)

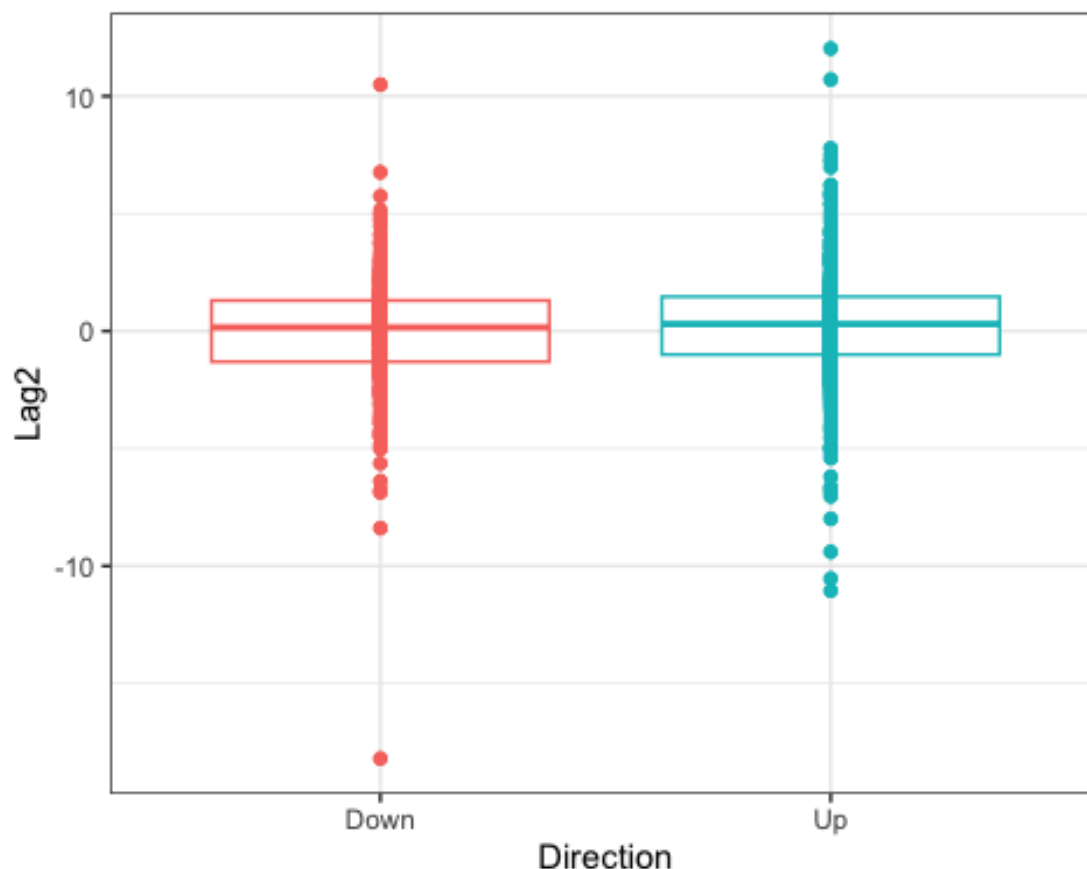
## Waiting for profiling to be done...

##              2.5 %      97.5 %
## (Intercept) -56.985558236 91.66680901
## Year        -0.045809580  0.02869546
## Lag1        -0.092972584  0.01093101
## Lag2         0.007001418  0.11291264
## Lag3        -0.068140141  0.03671410
## Lag4        -0.079519582  0.02453326
## Lag5        -0.066090145  0.03762099
## Volume      -0.131576309  0.13884038
```

## Modelo logístico con variables significativas

### Gráfico de las variables significativas (boxplot), ejemplo: Lag2):

```
ggplot(data = Weekly, mapping = aes(x = Direction, y = Lag2)) +
  geom_boxplot(aes(color = Direction)) +
  geom_point(aes(color = Direction)) +
  theme_bw() +
  theme(legend.position = "null")
```



### Training: observaciones desde 1990 hasta 2008

```
datos.entrenamiento <- (Year < 2009)
```

### Test: observaciones de 2009 y 2010

```
datos.test <- Weekly[!datos.entrenamiento, ]
```

### Verifica

```
nrow(datos.entrenamiento) + nrow(datos.test)
```

```
## integer(0)
```

### Ajuste del modelo logístico con variables significativas

```
modelo.log.s <- glm(Direction ~ Lag2, data = Weekly,
family = binomial, subset = datos.entrenamiento)
summary(modelo.log.s)
```

```
##
```

```
## Call:
```

```
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
##      subset = datos.entrenamiento)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

## Representación gráfica del modelo

El modelo devuelve las predicciones del logaritmo de Odds. La predicción se debe convertir en probabilidad. Eso se logra con el comando 'predict' y el 'type="response"'.

## Vector con nuevos valores interpolados en el rango del predictor Lag2:

```
nuevos_puntos <- seq(from = min(Weekly$Lag2), to = max(Weekly$Lag2), by =
0.5)
```

## Predicción de los nuevos puntos según el modelo con el comando predict() se calcula la probabilidad de que la variable respuesta pertenezca al nivel de referencia (en este caso "Up")

```
predicciones <- predict(modelo.log.s, newdata = data.frame(Lag2 =
nuevos_puntos), se.fit = TRUE, type = "response")
```

## Límites del intervalo de confianza (95%) de las predicciones

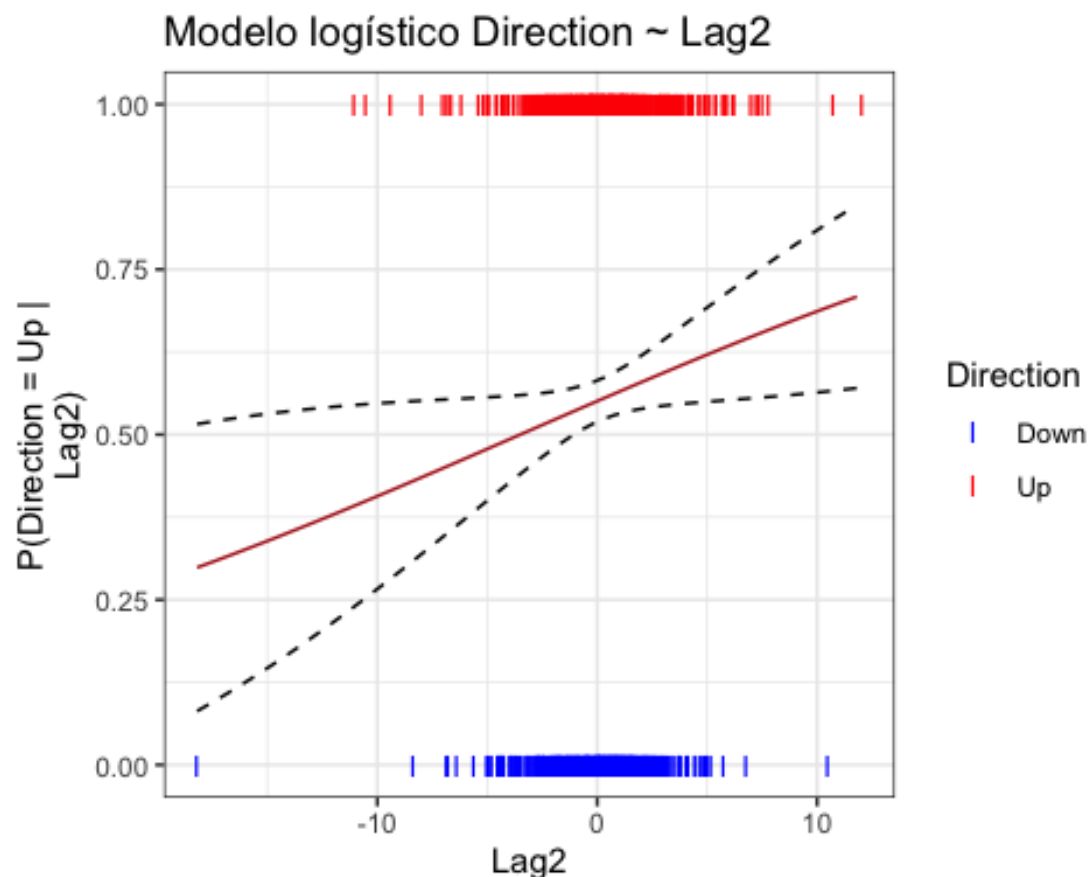
```
CI_inferior <- predicciones$fit - 1.96 * predicciones$se.fit
CI_superior <- predicciones$fit + 1.96 * predicciones$se.fit
```

## Matriz de datos con los nuevos puntos y sus predicciones

```
datos_curva <- data.frame(Lag2 = nuevos_puntos, probabilidad =
predicciones$fit, CI.inferior = CI_inferior, CI.superior = CI_superior)
```

## Codificación 0,1 de la variable respuesta Direction

```
Weekly$Direction <- ifelse(Weekly$Direction == "Down", yes = 0, no = 1)
ggplot(Weekly, aes(x = Lag2, y = Direction)) +
  geom_point(aes(color = as.factor(Direction)), shape = "I", size = 3) +
  geom_line(data = datos_curva, aes(y = probabilidad), color = "firebrick") +
  geom_line(data = datos_curva, aes(y = CI.superior), linetype = "dashed") +
  geom_line(data = datos_curva, aes(y = CI.inferior), linetype = "dashed") +
  labs(title = "Modelo logístico Direction ~ Lag2", y = "P(Direction = Up |
Lag2)", x = "Lag2") +
  scale_color_manual(labels = c("Down", "Up"), values = c("blue", "red")) +
  guides(color=guide_legend("Direction")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_bw()
```



## Evaluación del modelo

**Chi cuadrada:** Se evalúa la significancia del modelo con predictores con respecto al modelo nulo ("Residual deviance" vs "Null deviance"). Si valor p es menor que alfa será significativo

```
anova(modelo.log.s, test = 'Chisq')
```



```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Direction
##
## Terms added sequentially (first to last)
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                984      1354.7
## Lag2  1    4.1666      983      1350.5 0.04123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cálculo de las predicciones correctas así como de los falsos negativos y positivos. Normalmente se usa un límite de 0.5.

## Cálculo de la probabilidad predicha por el modelo con los datos de test

```
prob.modelo <- predict(modelo.log.s, newdata = datos.test, type = "response")
```

## Vector de elementos “Down”

```
pred.modelo <- rep("Down", length(prob.modelo))
```

## Sustitución de “Down” por “Up” si la $p > 0.5$

```
pred.modelo[prob.modelo > 0.5] <- "Up"
Direction.0910 = Direction[!datos.entrenamiento]
```

## Matriz de confusión

```
matriz.confusion <- table(pred.modelo, Direction.0910)
matriz.confusion
```

```
##           Direction.0910
## pred.modelo Down Up
##           Down    9  5
##           Up    34 56
```

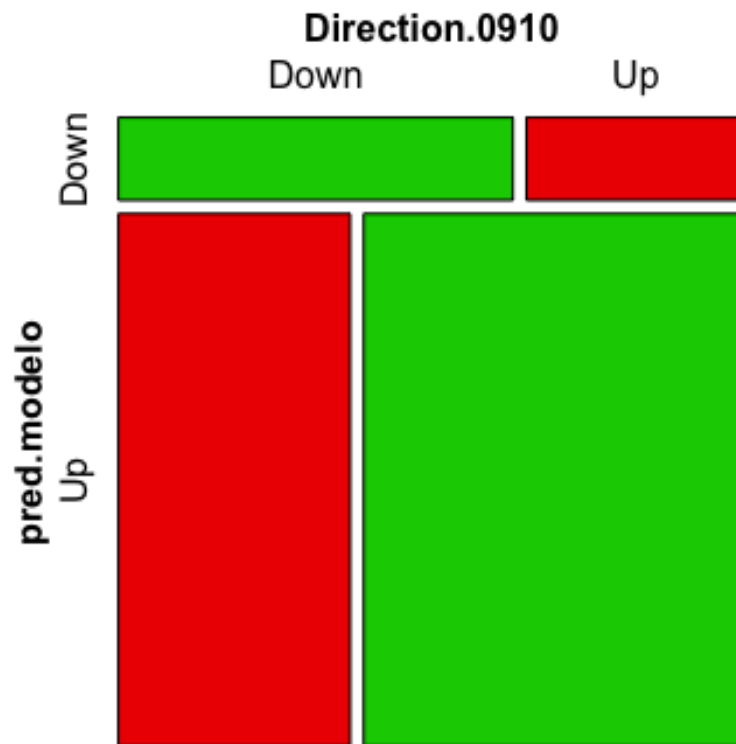
```
library(vcd)
```

```
## Loading required package: grid
```

```
##
## Attaching package: 'vcd'
```

```
## The following object is masked from 'package:ISLR':
##
##      Hitters

mosaic(matriz.confusion, shade = T, colorize = T,
gp = gpar(fill = matrix(c("green3", "red2", "red2", "green3"), 2, 2)))
```



```
mean(pred.modelo == Direction.0910)

## [1] 0.625
```

## Conclusion

Como pudimos observar desde el analisis inicial de las variables, la unica variable que tenia un nivel de significancia mayor a 0.05 era Log2 indicando que era la unica variable con significancia, sobre ella pasamos algunas pruebas mas y empezamos a definir el modelo, dividimos el set de entrenamiento y test, ajustamos el modelo y finalizamos con la representacion grafica de nuestro modelo donde creamos nuevos puntos sobre los que hacemos una prediccion segun nuestro modelo y lo graficamos, despues hacemos una evaluacion con chi cuadrada y creamos por ultimo nuestro matriz de confusion, donde podemos observar que nuestro bloque de up - up es grande, por lo tanto muchas predicciones son acertadas por nuestro modelo.