lab8

shichenh

10/23/2017

```
library(ISLR)
library(FactoMineR)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

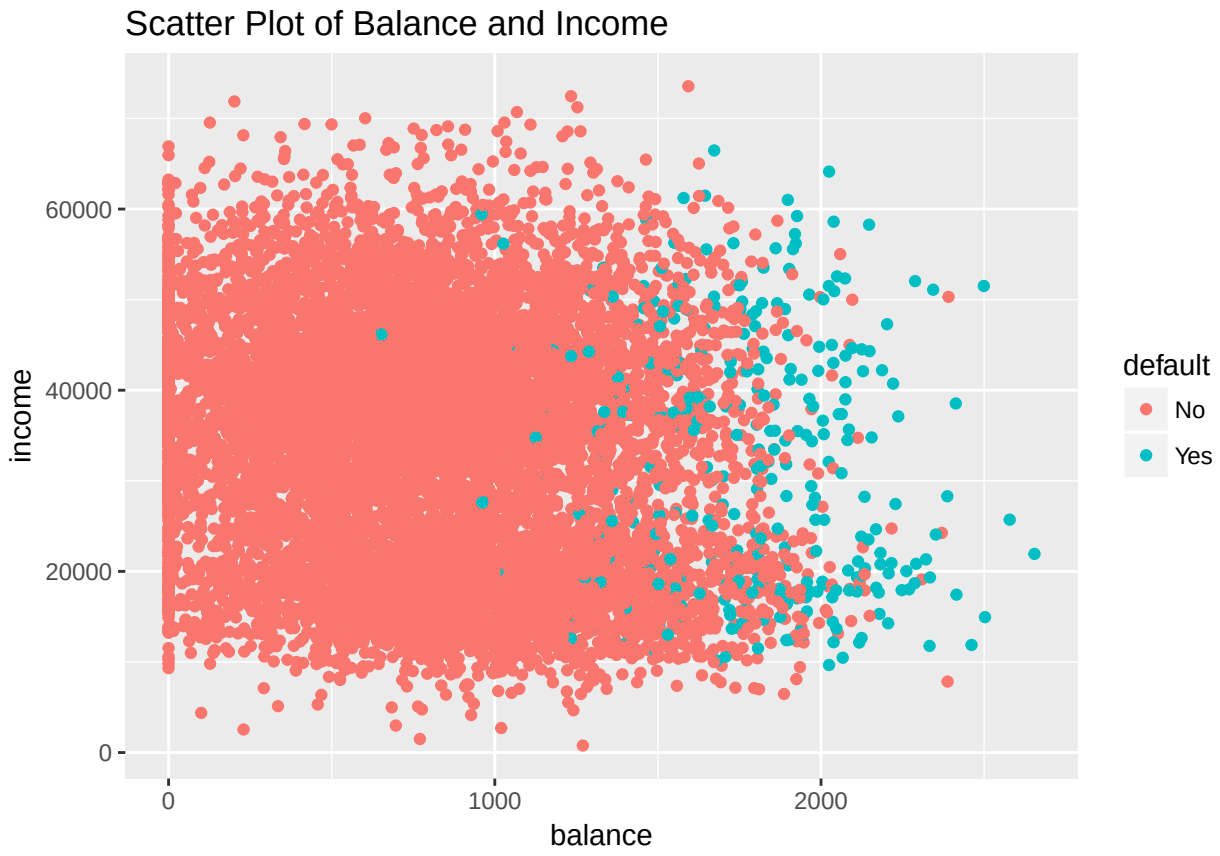
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

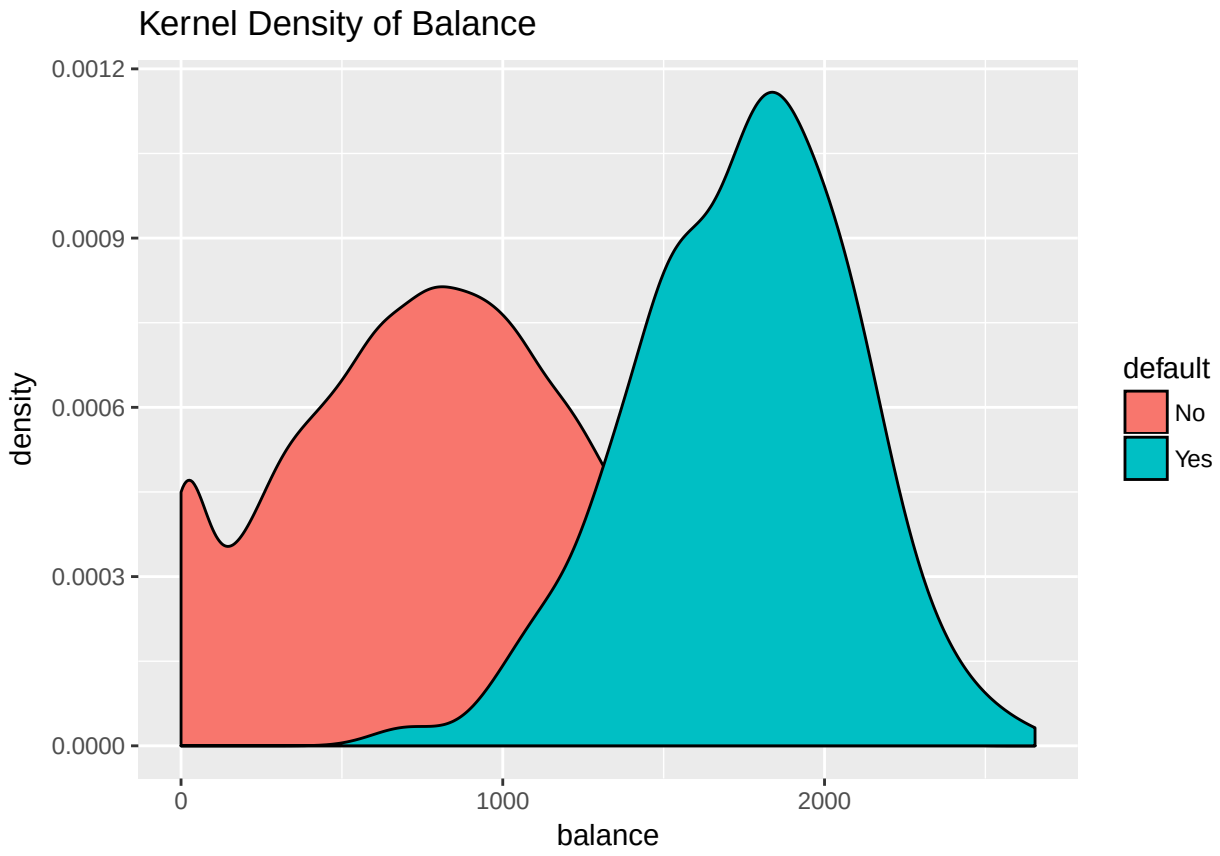
```
Default <- Default
```

EDA

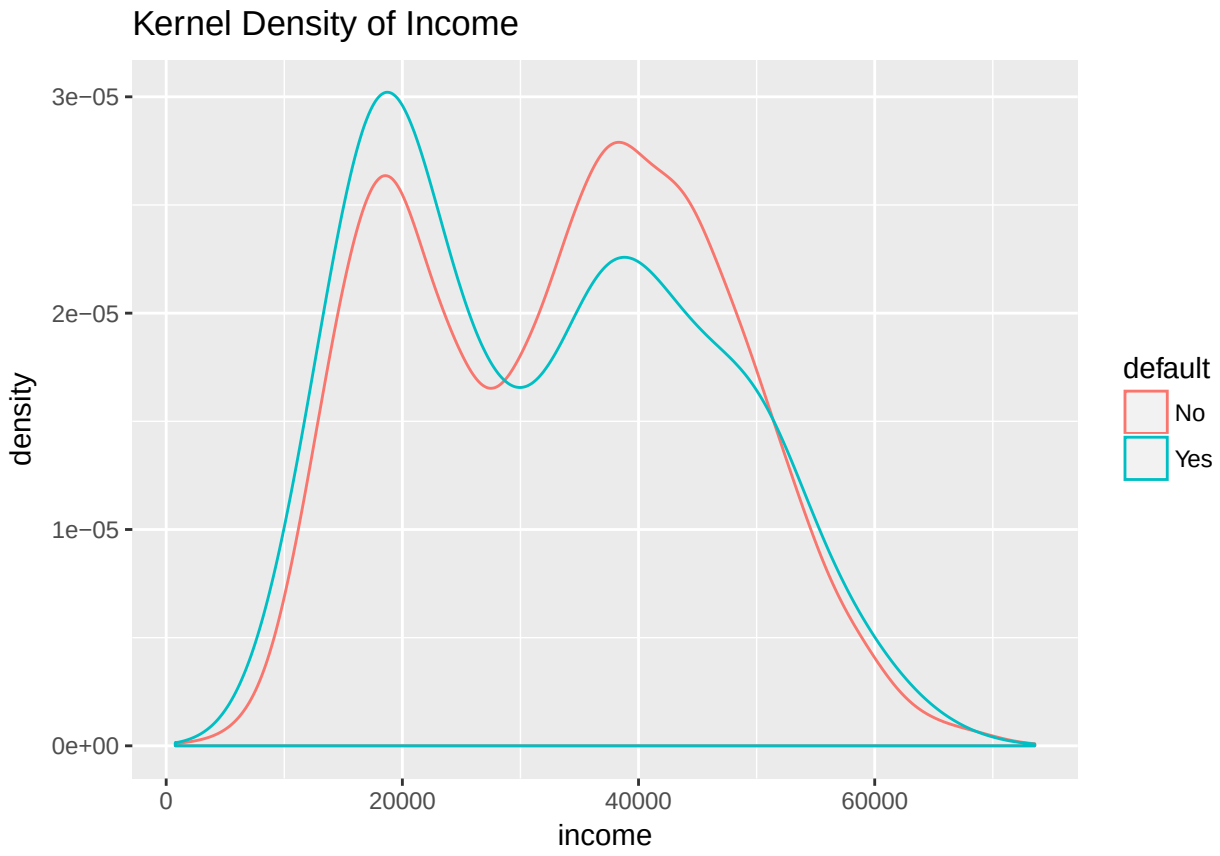
```
ggplot(Default) +
  geom_point(aes(x=balance, y=income, color=default)) +
  labs(title="Scatter Plot of Balance and Income")
```




```
ggplot(Default) +
  geom_density(aes(x=balance, fill=default)) +
  labs(title="Kernel Density of Balance")
```




```
ggplot(Default) +
  geom_density(aes(x=income, color=default)) +
  labs(title="Kernel Density of Income")
```



Default

OLS Regression

```
default_numeric <- rep(0, nrow(Default))
default_numeric[Default$default == 'Yes'] <- 1
Default$default_num <- default_numeric
ols_reg <- lm(default_num ~ balance, data = Default)
summary(ols_reg)
```

```
##
## Call:
## lm(formula = default_num ~ balance, data = Default)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23533 -0.06939 -0.02628  0.02004  0.99046
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.519e-02  3.354e-03  -22.42   <2e-16 ***
## balance      1.299e-04  3.475e-06   37.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

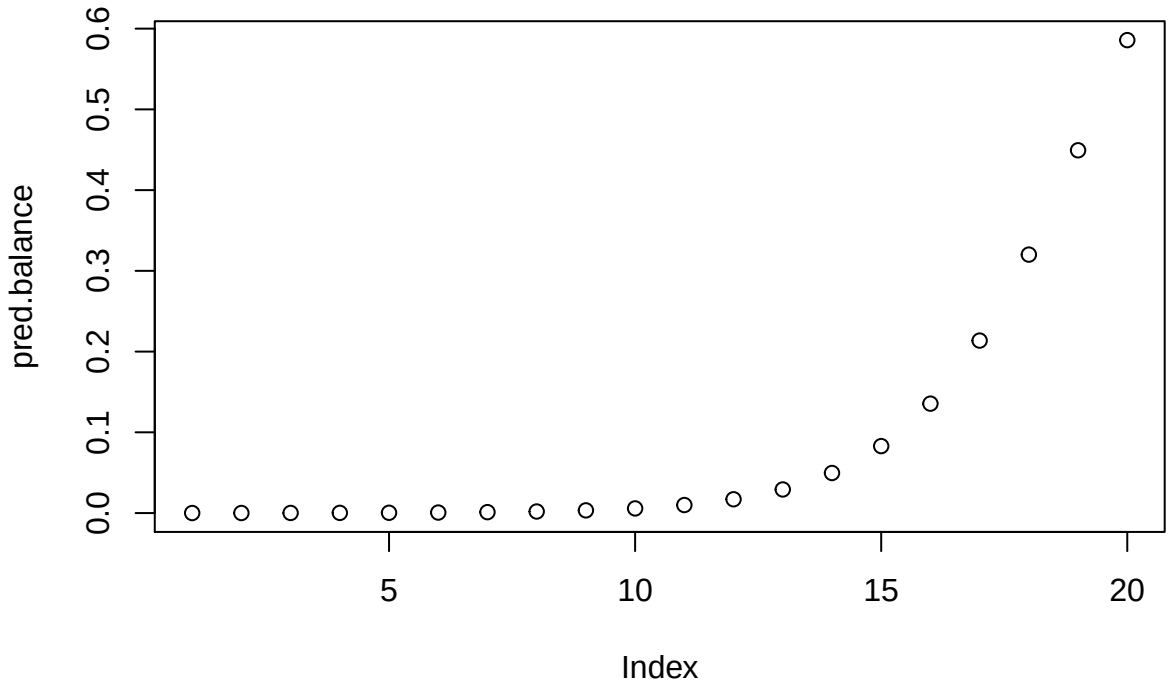
```
##
## Residual standard error: 0.1681 on 9998 degrees of freedom
## Multiple R-squared:  0.1226, Adjusted R-squared:  0.1225
## F-statistic:  1397 on 1 and 9998 DF,  p-value: < 2.2e-16
```

Logistic Regression

```
logreg_balance <- glm(default ~ balance, family = binomial, data = Default)
summary(logreg_balance)$coefficients
```

```
##                   Estimate   Std. Error   z value      Pr(>|z|)
## (Intercept) -10.651330614 0.3611573721 -29.49221 3.623124e-191
## balance       0.005498917 0.0002203702  24.95309 1.976602e-137
```

```
balance <- seq(100, 2000, by=100)
odds.balance <- exp(predict(logreg_balance, newdata = data.frame(balance)))
pred.balance <- odds.balance/(1+odds.balance)
plot(pred.balance)
```




```
logreg_student <- glm(default ~ student, family = binomial, data = Default)
summary(logreg_student)$coefficients
```

```
##                Estimate Std. Error    z value     Pr(>|z|)
## (Intercept) -3.5041278 0.07071301 -49.554219 0.0000000000
## studentYes   0.4048871 0.11501883   3.520181 0.0004312529
```

If the person is a student, then the log odds of it being default increases by 0.4048871.

```
logreg_all <- glm(default ~., family = binomial, data = Default)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
summary(logreg_all)
```

```
##
## Call:
## glm(formula = default ~ ., family = binomial, data = Default)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06   2.409e-06
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.657e+01  1.786e+04  -0.001    0.999
## studentYes   3.639e-13  1.201e+04   0.000    1.000
## balance     -3.185e-16  8.029e+00   0.000    1.000
## income       1.552e-17  4.065e-01   0.000    1.000
## default_num  5.313e+01  2.121e+04   0.003    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.9206e+03  on 9999  degrees of freedom
## Residual deviance: 5.8016e-08  on 9995  degrees of freedom
## AIC: 10
##
## Number of Fisher Scoring iterations: 25
```

Only *income* is not statistically significant.
The apparent contradiction between the opposite signs of the student coefficients may due to colinearity among student, balance and income.

Stock Market

```
Smarket <- Smarket
```

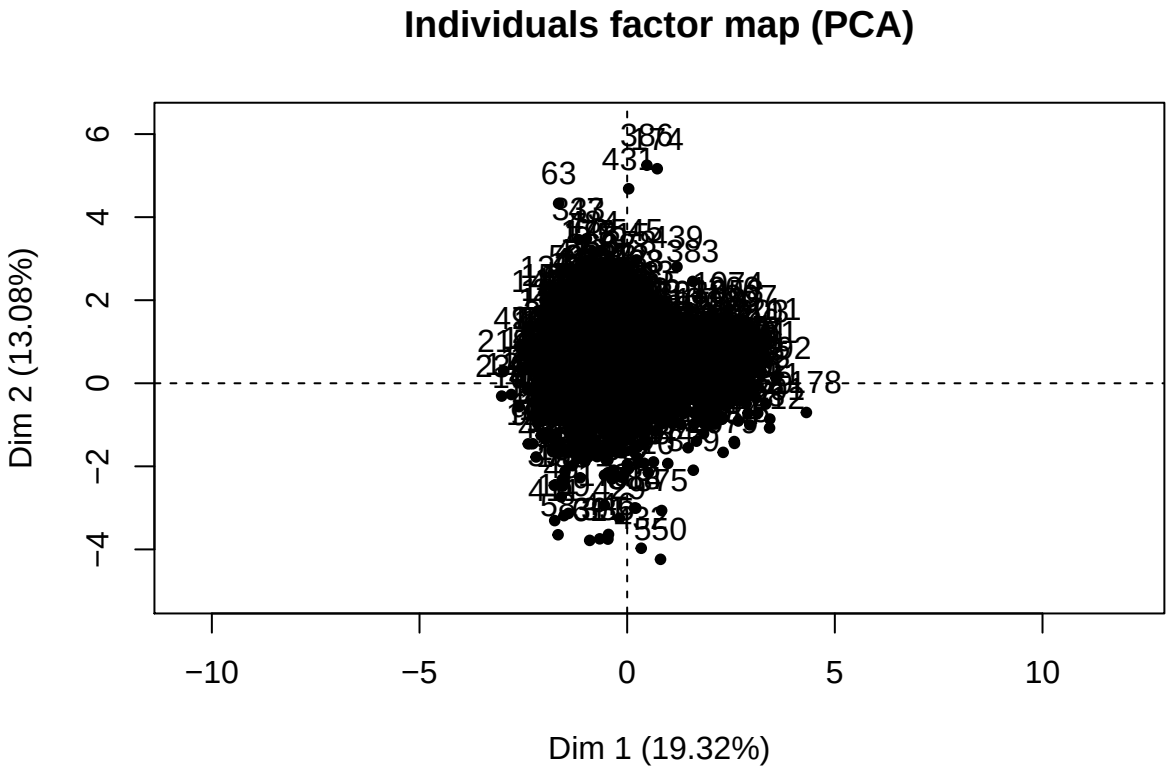
Variables

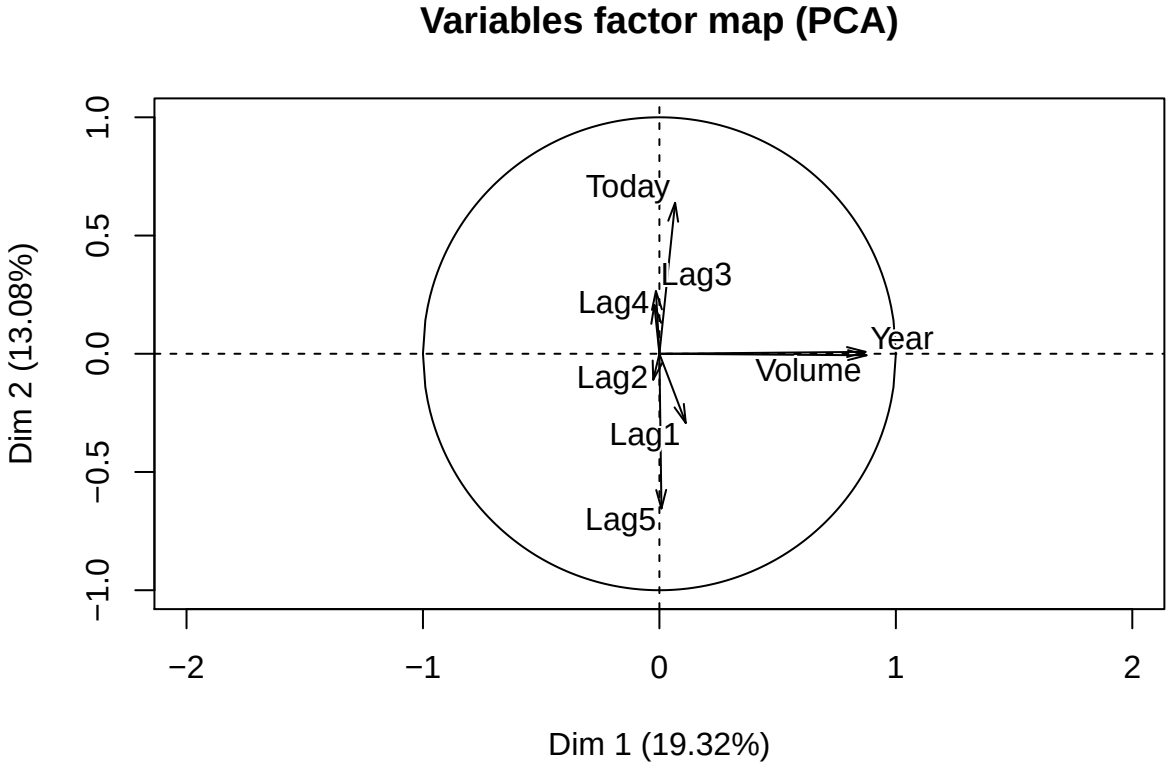
```
cor(Smarket %>% select(-Direction))
```

```
##              Year         Lag1         Lag2         Lag3         Lag4
## Year   1.00000000  0.029699649  0.030596422  0.033194581  0.035688718
## Lag1   0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911
## Lag2   0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533
## Lag3   0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036
## Lag4   0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000
## Lag5   0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641
## Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246
## Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527
##                Lag5      Volume        Today
## Year    0.029787995  0.53900647  0.030095229
## Lag1   -0.005674606  0.04090991 -0.026155045
## Lag2   -0.003557949 -0.04338321 -0.010250033
## Lag3   -0.018808338 -0.04182369 -0.002447647
## Lag4   -0.027083641 -0.04841425 -0.006899527
```

```
## Lag5    1.000000000 -0.02200231 -0.034860083
## Volume -0.022002315  1.00000000  0.014591823
## Today  -0.034860083  0.01459182  1.000000000
```

```
#pca
PCA(Smarket[ ,-9])
```



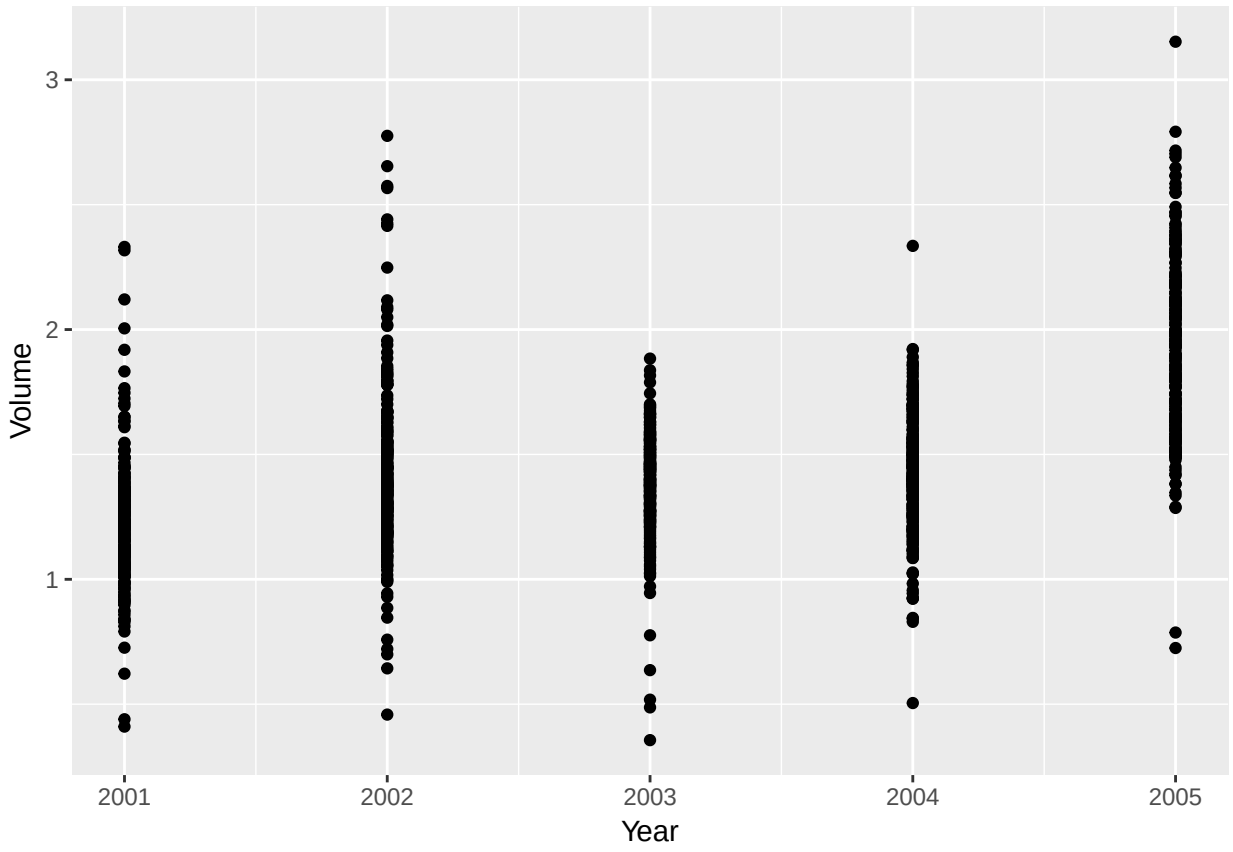


```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 1250 individuals, described by 8 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

```
#
```

The lag variables are not so correlated with today's return. So previous days return does not seem to correlate with today's return.

```
ggplot(Smarket) +
  geom_point(aes(x=Year, y=Volume))
```



Logistic Regression

```
logreg_smarket <- glm(Direction~., family = binomial, data = Smarket %>% select(-c(Year, Today)))
summary(logreg_smarket)
```

```
##
## Call:
## glm(formula = Direction ~ ., family = binomial, data = Smarket %>%
##     select(-c(Year, Today)))
##
## 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
```

None of the coefficients are significant. Lag1 coefficient is -0.073074. The negative sign means the Lag1 is inversely proportion to the log odds of Ups/Downs. If log1 increases, today's return is less likely to be up.

```
pred.smarket <- predict(logreg_smarket, newdata= Smarket %>% select(-c(Year, Today, Direction)), type="
pred.smarket[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
```

Parameters Estimation

Newton-Raphson

```
x <- model.matrix(Direction~., data = Smarket %>% select(-c(Year, Today)))

y <- as.matrix(Smarket %>%
  mutate(Direction = ifelse(as.numeric(Direction) == 2,1,0)) %>% select(Direction)
)
```

```r
#number of iterations
n.iter = 100
b0 <- as.matrix(rep(0, 7))
i <- 1

p <- exp(x %*% b0)/(1 + exp(x %*% b0))
w <- diag(p[1:length(p)]) * (1-p[1:length(p)])
z <- x %*% b0 + solve(w) %*% (y - p)
b <- solve(t(x) %*% w %*% x) %*% t(x) %*% (w) %*% z

while (sum(abs(b - b0)) >= 0.1^5 && i <= n.iter) {
  b0 <- b
  p <- exp(x %*% b0)/(1 + exp(x %*% b0))
  w <- diag(p[1:length(p)]) * (1-p[1:length(p)])
  z <- x %*% b0 + solve(w) %*% (y - p)
  b <- solve(t(x) %*% w %*% x) %*% t(x) %*% (w) %*% z
  i <- i + 1
}

print(sum(abs(b-b0)))
```

```
## [1] 3.00186e-07
```

```r
print(b)
```

```
##                      [,1]
## (Intercept) -0.126000259
## Lag1        -0.073073747
## Lag2        -0.042301345
## Lag3         0.011085108
## Lag4         0.009358938
## Lag5         0.010313069
## Volume       0.135440661
```

Simplified Algoirthm

```r
n.iter = 100
b0 <- as.matrix(rep(0, 7))
i <- 1
```

```r
p <- exp(x %*% b0)/(1 + exp(x %*% b0))
x.hat <- t(x) %*% (p*(1-p))
b <- b0 + solve(t(x) %*% x) %*% t(x) %*% (y-p)

while (sum(abs(b-b0)) >= 0.1^8 & i <= n.iter) {
  b0 <- b
  p <- exp(x %*% b0)/(1 + exp(x %*% b0))
  x.hat <- x
  for (j in 1:length(p)) {
    x.hat[i,] = x.hat[i,] * p[i] * (1 - p[i])
  }
  b <- b0 + solve(t(x) %*% x.hat) %*% t(x) %*% (y-p)
  i <- i + 1
}
```

```r
print(sum(abs(b-b0)))
```

```
## [1] 8.030544e-09
```

```r
print(b)
```

```
##                Direction
## (Intercept) -0.126000251
## Lag1        -0.073073743
## Lag2        -0.042301342
## Lag3         0.011085108
## Lag4         0.009358938
## Lag5         0.010313068
## Volume       0.135440653
```