Aplicación 2.1. Valoración de activos en el mercado de valores

J. Ramajo

2020

En esta aplicación se aplica el modelo CAPM (Capital Asset Pricing Model) a la empresa americana Mobil.

## 1. Lectura de datos y análisis exploratorio básico

En primer lugar, se leen los datos y se les da formato de series temporales:

library(readr)  
CAPM\_MOBIL <- read\_csv("CAPM\_MOBIL.csv")

## Parsed with column specification:  
## cols(  
## ER\_MOBIL = col\_double(),  
## ER\_M = col\_double()  
## )

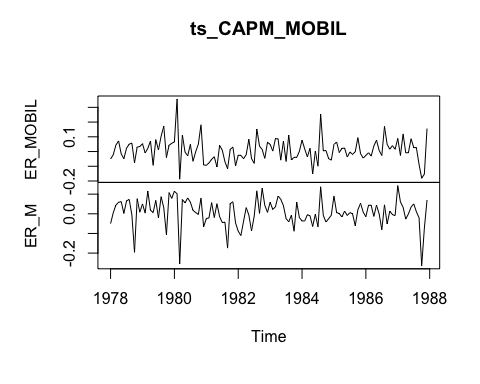
dim(CAPM\_MOBIL)

## [1] 120 2

head(CAPM\_MOBIL, n=10)

## # A tibble: 10 x 2  
## ER\_MOBIL ER\_M  
## <dbl> <dbl>  
## 1 -0.0509 -0.0499   
## 2 -0.0219 0.00506  
## 3 0.0437 0.0447   
## 4 0.0721 0.0581   
## 5 -0.0161 0.0619   
## 6 -0.0483 0.00173  
## 7 0.0227 0.0657   
## 8 0.0499 0.0729   
## 9 0.0576 -0.00445  
## 10 -0.0758 -0.196

ts\_CAPM\_MOBIL <- ts(CAPM\_MOBIL, start=c(1978,1), end = c(1987,12), frequency = 12)  
plot(ts\_CAPM\_MOBIL)



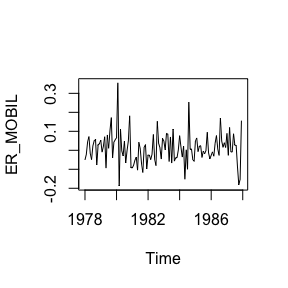
Estadísticos descriptivos de las variables en el fichero CAPM\_MOBIL:

summary(ts\_CAPM\_MOBIL)

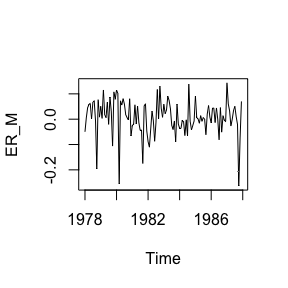
## ER\_MOBIL ER\_M   
## Min. :-0.187810 Min. :-0.263580   
## 1st Qu.:-0.039905 1st Qu.:-0.020220   
## Median : 0.005910 Median : 0.006210   
## Mean : 0.009353 Mean : 0.007153   
## 3rd Qu.: 0.049960 3rd Qu.: 0.054125   
## Max. : 0.355270 Max. : 0.143460

Gráficas e histogramas de la variables:

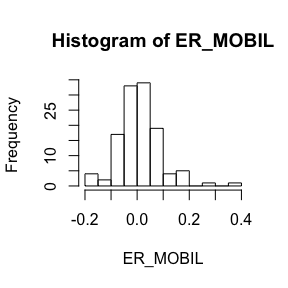
ER\_MOBIL <- ts\_CAPM\_MOBIL[,"ER\_MOBIL"]  
ER\_M <- ts\_CAPM\_MOBIL[,"ER\_M"]  
ts.plot(ER\_MOBIL)



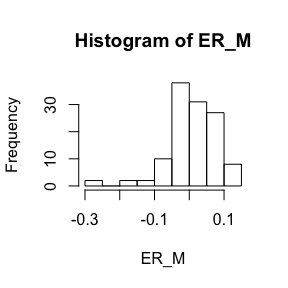
ts.plot(ER\_M)



with(CAPM\_MOBIL, hist(ER\_MOBIL))



with(CAPM\_MOBIL, hist(ER\_M))



## 2. Análisis econométrico básico

Ajuste de un modelo de regresión lineal ( funciónlm() ):

CAPM <- lm (ER\_MOBIL~ER\_M)  
summary(CAPM)

##   
## Call:  
## lm(formula = ER\_MOBIL ~ ER\_M)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.145562 -0.044644 0.000819 0.036722 0.278652   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.004241 0.005881 0.721 0.472   
## ER\_M 0.714695 0.085615 8.348 1.52e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06407 on 118 degrees of freedom  
## Multiple R-squared: 0.3713, Adjusted R-squared: 0.366   
## F-statistic: 69.69 on 1 and 118 DF, p-value: 1.524e-13

NOTA: Las funciones brief() y S(), ambas en el paquete **car** , proporcionan alternativas a las salida estándar de la función summary del paquete básico.

library(car)

## Loading required package: carData

brief(CAPM)

## (Intercept) ER\_M  
## Estimate 0.00424 0.7147  
## Std. Error 0.00588 0.0856  
##   
## Residual SD = 0.0641 on 118 df, R-squared = 0.371

# \*\*\*\*\*\*\*\*\*\*\*\*

S(CAPM)

## Call: lm(formula = ER\_MOBIL ~ ER\_M)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.004241 0.005881 0.721 0.472   
## ER\_M 0.714695 0.085615 8.348 1.52e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard deviation: 0.06407 on 118 degrees of freedom  
## Multiple R-squared: 0.3713  
## F-statistic: 69.69 on 1 and 118 DF, p-value: 1.524e-13   
## AIC BIC   
## -314.92 -306.56

Contrastes de hipótesis e intervalos de confianza:

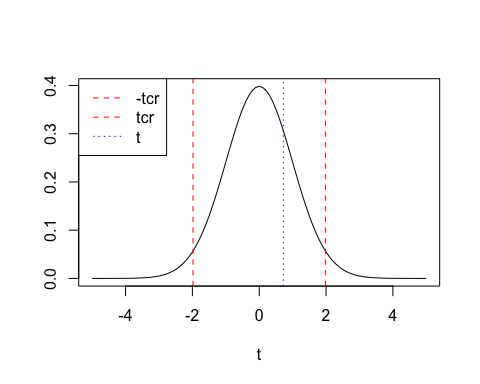
#  
# Contrastes de hipótesis  
#  
# beta1=0 versus beta1≠0  
#  
# Estadístico t  
#  
# Método del valor crítico  
#  
alpha <- 0.05 # nivel de significación  
b1 <- coef(CAPM)[[1]] # estimación del parámetro beta1  
seb1 <- sqrt(vcov(CAPM)[1,1]) # estimación de la desviación típica de beta1  
c <- 0  
df <- df.residual(CAPM) # grados de libertad  
t <- (b1-c)/seb1 # estadístico t  
t

## [1] 0.7210872

tcr <- qt(1-alpha/2, df) # valor crítico  
tcr

## [1] 1.980272

#  
# Gráfico de la función de densidad t de Student, valor crítico y estadístico t:  
#  
curve(dt(x, df), -5, 5, ylab=" ", xlab="t")  
abline(v=c(-tcr, tcr, t), col=c("red", "red", "blue"), lty=c(2,2,3))  
legend("topleft", legend=c("-tcr", "tcr", "t"), col=c("red", "red", "blue"), lty=c(2, 2, 3))



#  
# Método del P-valor  
#  
p <- 2\*(1-pt(abs(t), df))  
p

## [1] 0.4722821

#  
# Estadístico F  
#  
H\_0 <- "(Intercept)=0"  
linearHypothesis(CAPM,H\_0,test="F")

## Linear hypothesis test  
##   
## Hypothesis:  
## (Intercept) = 0  
##   
## Model 1: restricted model  
## Model 2: ER\_MOBIL ~ ER\_M  
##   
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 119 0.48659   
## 2 118 0.48445 1 0.0021347 0.52 0.4723

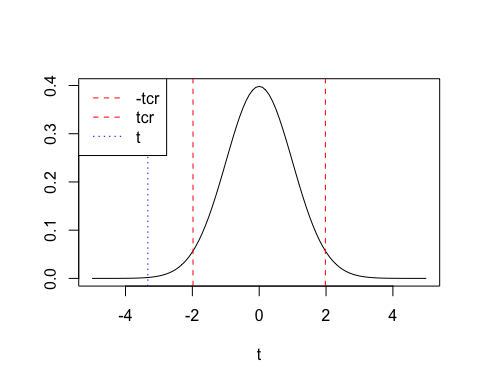
#  
# beta2=1 versus beta1≠1  
#  
# Estadístico t  
#  
# Método del valor crítico  
alpha <- 0.05 # nivel de significación  
b2 <- coef(CAPM)[[2]] # estimación del parámetro beta1  
seb2 <- sqrt(vcov(CAPM)[2,2]) # estimación de la desviación típica de beta1  
c <- 1  
df <- df.residual(CAPM) # grados de libertada  
t <- (b2-c)/seb2 # estadístico t  
t

## [1] -3.332411

tcr <- qt(1-alpha/2, df) # valor crítico  
tcr

## [1] 1.980272

#  
# Gráfico de la función de densidad t de Student, valor crítico y estadístico t:  
#   
curve(dt(x, df), -5, 5, ylab=" ", xlab="t")  
abline(v=c(-tcr, tcr, t), col=c("red", "red", "blue"), lty=c(2,2,3))  
legend("topleft", legend=c("-tcr", "tcr", "t"), col=c("red", "red", "blue"), lty=c(2, 2, 3))



#  
# Método del P-valor  
#  
p <- 2\*(1-pt(abs(t), df))  
p

## [1] 0.001150302

#  
# Estadístico F  
#  
H\_0 <- "ER\_M=1"  
linearHypothesis(CAPM,H\_0,test="F")

## Linear hypothesis test  
##   
## Hypothesis:  
## ER\_M = 1  
##   
## Model 1: restricted model  
## Model 2: ER\_MOBIL ~ ER\_M  
##   
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 119 0.53004   
## 2 118 0.48445 1 0.045592 11.105 0.00115 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#  
# Contraste conjunto: beta1=0, beta2=1  
#  
H\_0 <- c("(Intercept) = 0", "ER\_M = 1")  
linearHypothesis(CAPM,H\_0,test="F")

## Linear hypothesis test  
##   
## Hypothesis:  
## (Intercept) = 0  
## ER\_M = 1  
##   
## Model 1: restricted model  
## Model 2: ER\_MOBIL ~ ER\_M  
##   
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 120 0.53062   
## 2 118 0.48445 2 0.046172 5.6232 0.004649 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

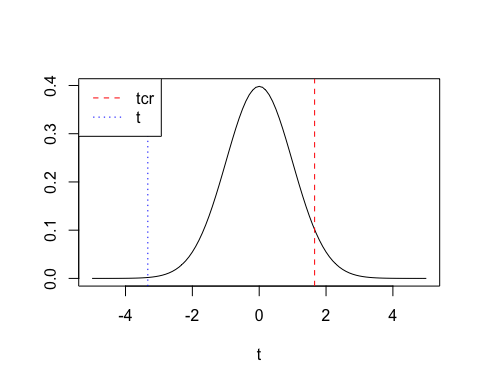
#  
# Contrastes unilaterales  
#   
# beta2≤1 versus beta2>1  
#  
c <- 1  
alpha <- 0.05  
t <- (b2-c)/seb2  
t

## [1] -3.332411

tcr <- qt(1-alpha, df) # alpha no se divide por 2  
tcr

## [1] 1.65787

curve(dt(x, df), -5, 5, ylab=" ", xlab="t")  
abline(v=c(tcr, t), col=c("red", "blue"), lty=c(2, 3))  
legend("topleft", legend=c("tcr", "t"),  
 col=c("red", "blue"), lty=c(2, 3))



#  
p <- 1-pt(t, df)  
p

## [1] 0.9994248

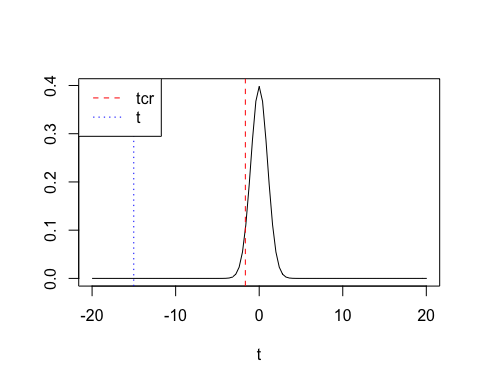
#   
# beta2≥2 versus beta2<2  
#  
c <- 2  
alpha <- 0.05  
t <- (b2-c)/seb2  
t

## [1] -15.01258

tcr <- qt(alpha, df) # alpha no se divide por 2  
tcr

## [1] -1.65787

curve(dt(x, df), -20, 20, ylab=" ", xlab="t")  
abline(v=c(tcr, t), col=c("red", "blue"), lty=c(2, 3))  
legend("topleft", legend=c("tcr", "t"),  
 col=c("red", "blue"), lty=c(2, 3))



#  
p <- pt(t, df)  
p

## [1] 1.923507e-29

#  
# Intervalos de confianza (automáticos)  
#  
IntConf <- confint(CAPM)  
print(IntConf)

## 2.5 % 97.5 %  
## (Intercept) -0.007405435 0.01588706  
## ER\_M 0.545153680 0.88423640

#  
# Intervalos de confianza (manualmente)  
#  
alpha <- 0.05  
tc <- qt(1-alpha/2, df)  
#  
inf\_b1 <- b1-tc\*seb1 # cota inferior  
sup\_b1 <- b1+tc\*seb1 # cota superior  
inf\_b1 ; sup\_b1

## [1] -0.007405435

## [1] 0.01588706

#  
inf\_b2 <- b2-tc\*seb2 # cota inferior  
sup\_b2 <- b2+tc\*seb2 # cota superior  
inf\_b2 ; sup\_b2

## [1] 0.5451537

## [1] 0.8842364

#  
# Ajuste del modelo (R^2) y ANOVA  
#  
s\_CAPM <- summary(CAPM)  
print(s\_CAPM)

##   
## Call:  
## lm(formula = ER\_MOBIL ~ ER\_M)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.145562 -0.044644 0.000819 0.036722 0.278652   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.004241 0.005881 0.721 0.472   
## ER\_M 0.714695 0.085615 8.348 1.52e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06407 on 118 degrees of freedom  
## Multiple R-squared: 0.3713, Adjusted R-squared: 0.366   
## F-statistic: 69.69 on 1 and 118 DF, p-value: 1.524e-13

names(s\_CAPM)

## [1] "call" "terms" "residuals" "coefficients"   
## [5] "aliased" "sigma" "df" "r.squared"   
## [9] "adj.r.squared" "fstatistic" "cov.unscaled"

R2 <- s\_CAPM$r.squared  
R2

## [1] 0.3712874

#   
T <- nobs(CAPM)  
K <- T-df.residual(CAPM)  
T ; K

## [1] 120

## [1] 2

F\_0 <- (R2/(K-1))/((1-R2)/(T-K))  
F\_0

## [1] 69.68511

anova(CAPM)

## Analysis of Variance Table  
##   
## Response: ER\_MOBIL  
## Df Sum Sq Mean Sq F value Pr(>F)   
## ER\_M 1 0.28609 0.286094 69.685 1.524e-13 \*\*\*  
## Residuals 118 0.48445 0.004106   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#  
# Intervalo confianza para sigma  
#  
s2 <- s\_CAPM$sigma^2  
s2

## [1] 0.004105522

#  
alpha <- 0.05  
chisqcr1 <- qchisq(alpha/2, df)   
chisqcr2 <- qchisq(1-alpha/2, df)  
chisqcr1 ; chisqcr2

## [1] 89.82707

## [1] 149.9569

inf\_s2 <- (T-K)\*s2/chisqcr2 # cota inferior  
sup\_s2 <- (T-K)\*s2/chisqcr1 # cota superior  
inf\_s2 ; sup\_s2

## [1] 0.003230605

## [1] 0.005393158

#  
s <- sqrt(s2)  
inf\_s <- sqrt(inf\_s2)  
sup\_s <- sqrt(sup\_s2)  
inf\_s ; s ; sup\_s

## [1] 0.05683841

## [1] 0.06407434

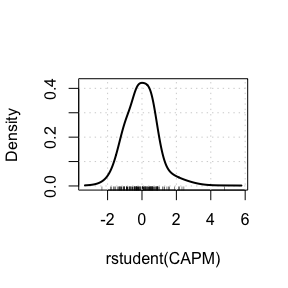
## [1] 0.07343812

## 3. Diagnósticos de la regresión

rstudent() : residuos estudentizados

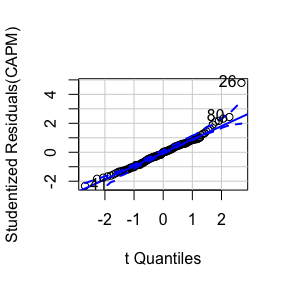
densityPlot() : estimación de la distribución de los residuos (adaptive kernel density estimator)

densityPlot(rstudent(CAPM))



qqPlot() : comparación de los residuos estudentizados con una distribution t de Student

qqPlot(CAPM, id=list(n=3))



## [1] 26 41 80

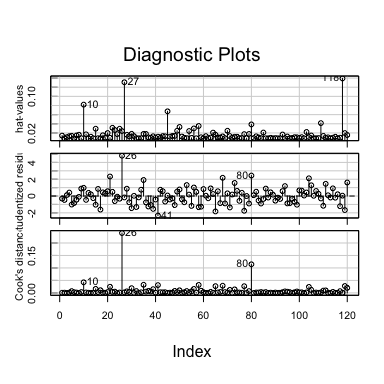
qqPlot() : búsqueda de **outliers** en la regresión

outlierTest(CAPM)

## rstudent unadjusted p-value Bonferroni p  
## 26 4.795108 4.8293e-06 0.00057951

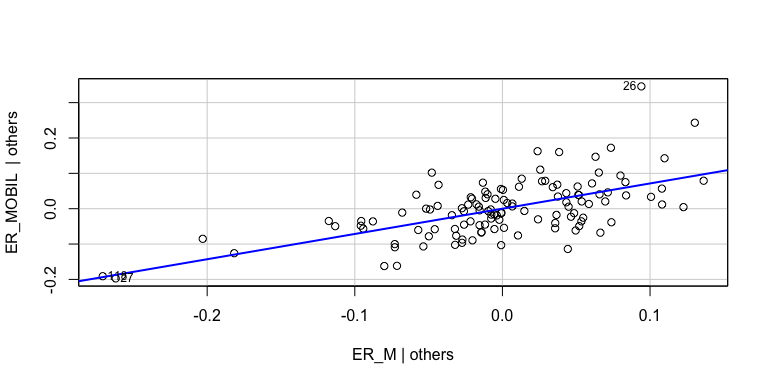
influenceIndexPlot : medidas de influencia

influenceIndexPlot(CAPM, vars=c("hat", "Studentized", "Cook"),   
 id=list(n=3))



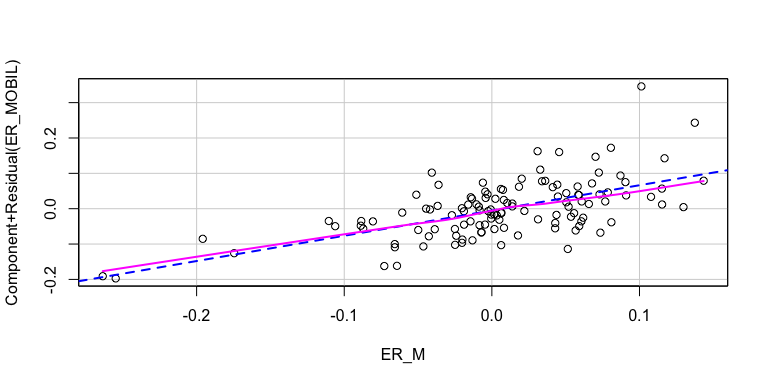
crPlots [Added-variable plots] : búsqueda de observaciones influyentes (para cada variable explicativa)

avPlots(CAPM,   
 id=list(cex=0.75, n=3, method="mahal"))



crPlots [Component-plus-residual plots] : chequeo de nonlinealidad (para cada variable explicativa)

crPlots(CAPM, smooth=list(span=0.7))



ncvTest : contraste de varianza no constante (heteroscedasticidad)

ncvTest(CAPM)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 10.58849, Df = 1, p = 0.0011379

ncvTest(CAPM, var.formula= ~ ER\_M) # con sólo una variable explicativa coincide con el anterior

## Non-constant Variance Score Test   
## Variance formula: ~ ER\_M   
## Chisquare = 10.58849, Df = 1, p = 0.0011379