install.packages("moments")

library(quantmod)

library(tseries)

library( GGally)

library(lmtest)

library(moments)

#descarga serie de tiempo: Define el horizonte temporal

getSymbols.yahoo('AMZN',env=globalenv(),return.class="xts",

from='2020-08-30',to='2021-08-30',

periodicity='daily')

getSymbols.yahoo('^GSPC',env=globalenv(),return.class="xts",

from='2020-08-30',to='2021-08-30',

periodicity='daily')

#Especifica el campo de la serie de datos. En este caso fueron los precios ajustados.

AMZN <- AMZN[,"AMZN.Adjusted"]

GSPC <- GSPC[,"GSPC.Adjusted"]

#Calcula los retornos de ambas varaibles

AMZNrt<- diff(log(AMZN))

DGSPC <- diff(log(GSPC))

#Grafica ambas variables en su forma original y en sus retornos

chart\_Series(AMZNrt)

chart\_Series(DGSPC)

#define que estas trabajando con una serie de tiempo ts

AMZNrt<- ts(AMZNrt,frequency=365)

DGSPC<- ts(DGSPC,frequency=365)

#Estima la regresión

regresion <- lm(AMZNrt ~ DGSPC)

summary(regresion)

#Normalidad en los residuos

media <- mean(sresid)

varianza <- var(sresid)

desv\_estandar <- sd(sresid)

asimetria <- skewness(sresid)

curtosis <- kurtosis(sresid)

# Mostrar los resultados

cat("Media:", media, "\n")

cat("Varianza:", varianza, "\n")

cat("Desviación estándar:", desv\_estandar, "\n")

cat("Asimetría:", asimetria, "\n")

cat("Curtosis:", curtosis, "\n")

jarque.bera.test(residuals(regresion))

sresid <- resid(regresion)

hist(sresid, freq = FALSE, breaks = 30, col = "lightblue", border = "black",

main = "Distribución de los Residuos Studentizados",

xlab = "Residuos Studentizados", ylab = "Densidad")

lines(density(sresid), col = "red", lwd = 2)

xfit <- seq(min(sresid), max(sresid), length = 100)

yfit <- dnorm(xfit, mean = mean(sresid), sd = sd(sresid))

lines(xfit, yfit, col = "blue", lwd = 2)

legend("topright", legend = c("Densidad Kernel", "Distribución Normal"),

col = c("red", "blue"), lwd = 2, bty = "n")

# Prueba de heterocedasticidad breusch pagan test

library(lmtest)

bptest(regresion)

# Independencia de los errores durbin watson test

dwtest(regresion)