OLS assumptions in finite sample

OLS estimation

In this short note, we use financial data to see if the assumptions of the finite sample framework hold. Our dependent variable is called 'HFI' and denotes a weighted index of all the Hedge fund returns of the credit Suisse database. We shall decompose the general HF strategy into 3 potential risks that are the financial market (approximated by the excess returns of the S&P 500), the SMB (small minus big anomaly) and the default risk (DEF).

In this exercise, we shall do the following:

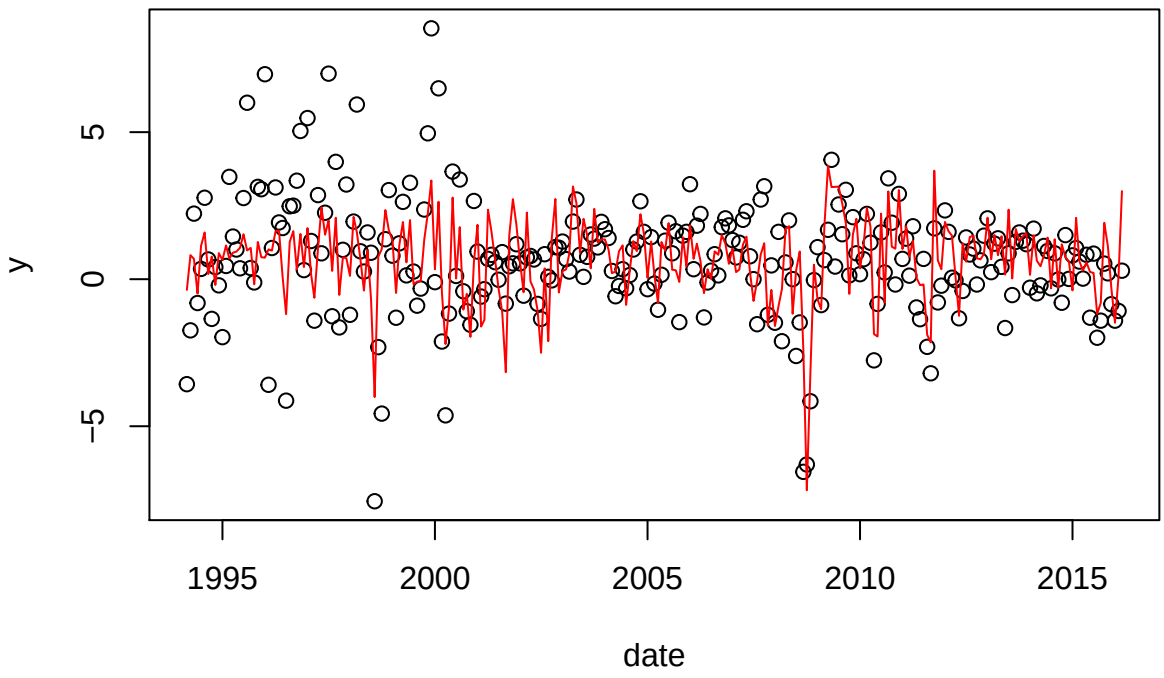
1. Compute the OLS estimates.
2. Compute the residuals.
3. See what kind of assumptions could be violated. To do so, we will plot the empirical density function of the residuals and compare it with a standard normal distribution. We shall also perform a statistical test to see if the Normality assumption holds.

First, we load the data.

```
url = "https://raw.githubusercontent.com/adufays/GDP_expectancy/main/DATA_HF.csv"
R_DATA <- read.csv(url)
X = cbind(R_DATA$Intercept,R_DATA$Mkt,R_DATA$SMB,R_DATA$DEF)
y = R_DATA$HF.Index
date = as.Date(R_DATA$date,'%d-%m-%Y')
```

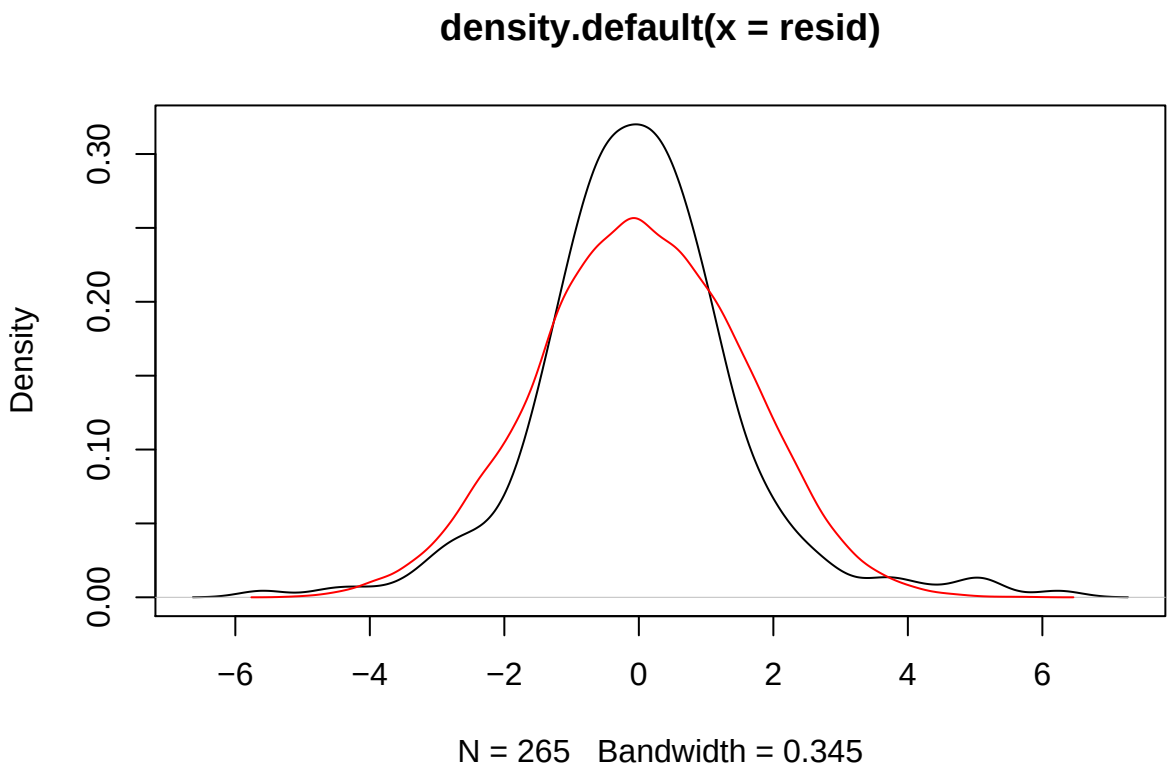
Now that we have the dependent and the explanatory variables, we apply the OLS formula to compute the OLS estimates.

```
inv_X_X = solve(t(X)%*%X)
beta_ols = inv_X_X%*%t(X)%*%y
plot(date,y)
yhat = X%*%beta_ols ## Fitted values
lines(date,yhat,col='red')
```



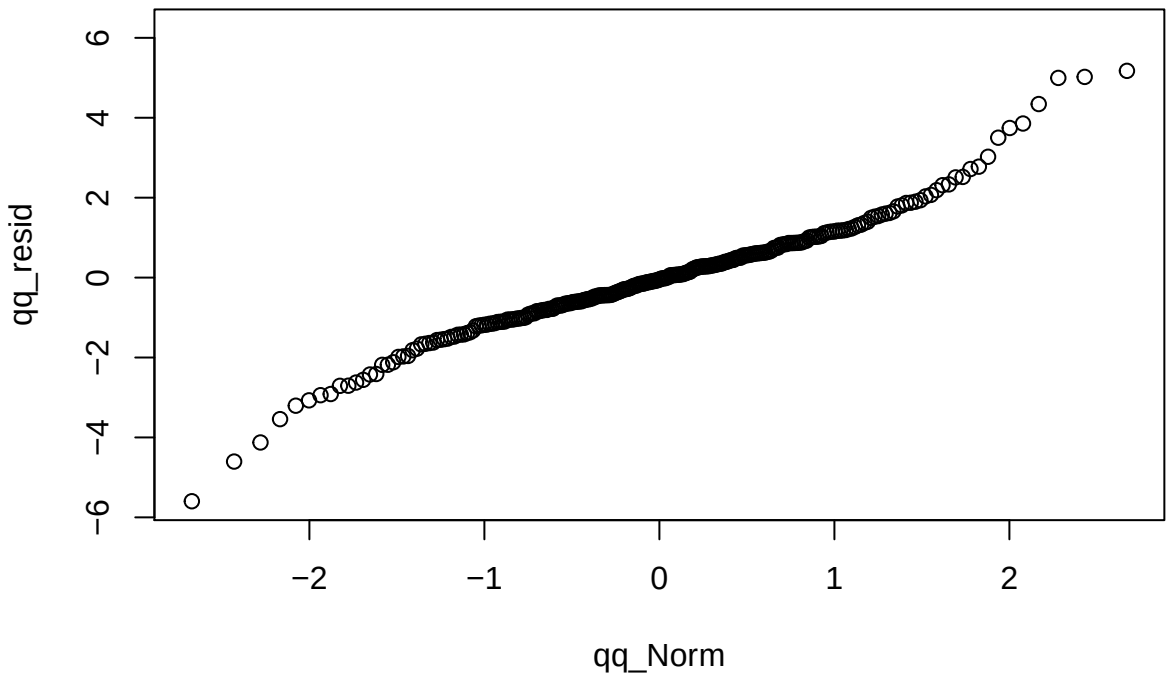
Let us compute the residuals and let us see if they look like draws from a normal distribution.

```
resid = y - X%*%beta_ols
T = length(resid)
plot(density(resid))
mean = mean(resid)
vari = var(resid)
draws_from_Normal = mean + sqrt(vari[1])*rnorm(10000)
lines(density(draws_from_Normal),col = 'red')
```



We can now have a look at the kurtosis since the density plot suggests that the residuals exhibit fat tails. We first do a QQ-plot to see if the quantiles of the two distributions coincide. Then, we perform a bootstrap test on the kurtosis using monte carlo simulations.

```
## QQ-plot
quantile = seq(1/T,1,1/T)
qq_Norm = qnorm(quantile)
qq_resid = sort(resid)
plot(qq_Norm,qq_resid)
```



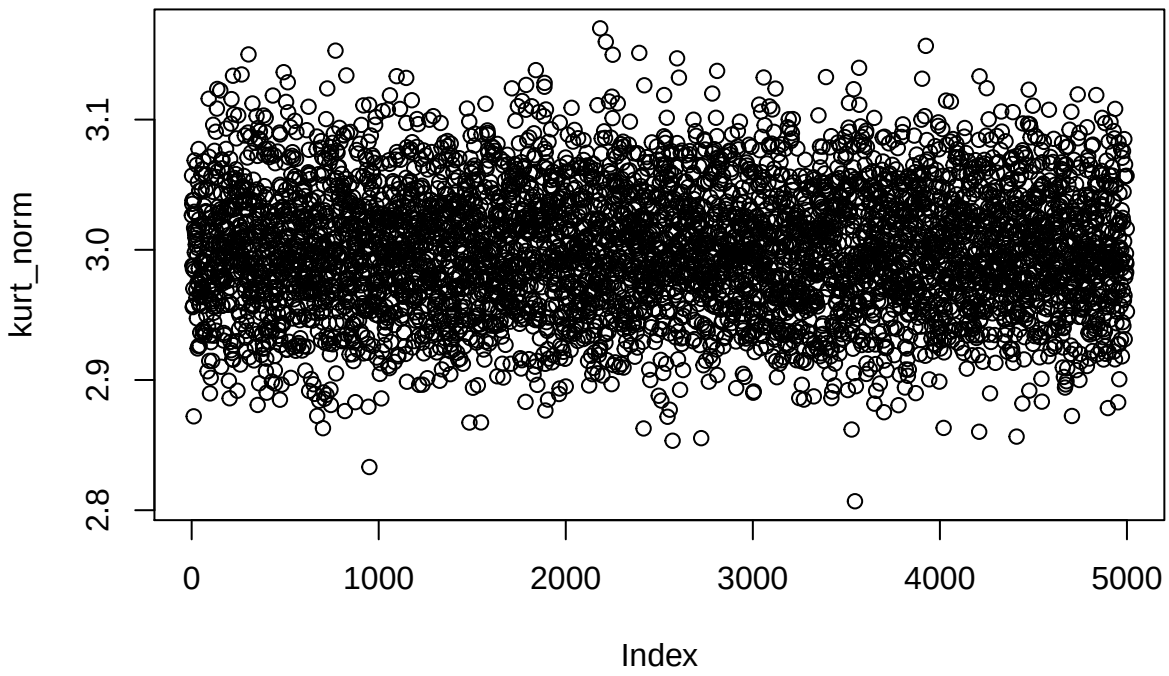

```
## Simple test on the kurtosis by monte carlo
kurtosis = function(x){
  mean_x = mean(x)
  kurt = mean((x-mean_x)^4)/(mean((x-mean_x)^2)^2)
  return(kurt)
}
nb_replic = 5000
mean = mean(resid)
vari = var(resid)
kurt_norm = numeric(nb_replic)
for(i in 1:nb_replic){
  kurt_norm[i] = kurtosis(mean + sqrt(vari[1])*rnorm(10000))
}
kurt_resid = kurtosis(resid)
p_val = 2*(1-sum(kurt_norm<kurt_resid)/nb_replic)
(p_val)
```

```
## [1] 0
```

```
(kurt_resid)
```

```
## [1] 5.684287
```

```
plot(kurt_norm)
```


kurt_norm
2.8
2.9
3.0
3.1
0
1000
2000
3000
4000
5000
Index