

Tipologia i cicle de dades · PRACTICA 2 · 2017-2018

EEES · Màster de data Science

Relació i patrons entre moviments de índexs de borsa i cryptomoneda

Nom i Cognoms: Albert Costas Gutiérrez

UOC practica 2. Tipologia i cicle de vida.

Descripció

Datasets per la comparació de moviments i patrons entre els principals índexs borsatils espanyols i les crypto-monedes

Links	Fitxers
Repositori github: https://github.com/acostasg/scraping Repositori kaggle Open data: <ul style="list-style-type: none"> https://www.kaggle.com/acostasg/stock-index https://www.kaggle.com/acostasg/crypto-currencies https://www.kaggle.com/acostasg/cryptocurrenciesvsstockindex 	<ul style="list-style-type: none"> Document PDF amb les respostes de les preguntes i els noms dels components del grup. Fitxer amb el codi Python per obtenir les dades Fitxer R amb correlació d'atributs Carpeta CSV amb les dades

Estructura

```

scraping
├── pdf
│   ├── acostasg-PRACTICA_1.pdf # Document pdf de la practica 1, components grup
│   └── acostasg-PRACTICA_2.pdf # Document pdf de la practica 2, components grup
├── csv # datasets
│   ├── crypto_currencies
│   │   └── ... # fitxers csv
│   ├── stock_index
│   │   └── ... # directoris per data amb els csv
│   └── dataset
│       └── dataset.csv # dataset preparat per al script R unifica els anteriors
├── projects
│   ├── scraping_crypto_currencies.py # scraping url criptomoneda
│   ├── scraping_stock_indexes.py # scraping url el economista
│   └── cleanAndTransform.py # script para limpiar i unificar en un dataset
├── R
│   └── script.r # script R i amb correlació d'atributs i model
├── README.md
├── cleanData.py # fitxer python per netejar, unificar i transformar les dades
├── scraping.py # fitxer python inicial
└── setup.py

```

Autors

Albert Costas Gutierrez - acostasg@uoc.edu

Llicència

Database released under Open Database License, individual contents under Database Contents License.

Fonts de dades

- <http://www.eleconomista.es>
- <https://coinmarketcap.com>

Les dades de borsa i crypto-moneda estan en última instància sota llicència de les webs respectivament.

Respostes a les preguntes

1. Descripció del dataset

Perquè és important i quina pregunta/problema pretén respondre?

En aquest hi ha 2 datasets (els qual netejarem i transformarem en un únic dataset), però l'objectiu és poder **comparar** en el mateix període de temps si hi ha **relació o es podrien patrons** comuns entre els **moviments borsatils** dels principals índexs espanyols i els **moviments de les crypto-monedes**.

En aquest cas el context és detectar o preveure els **diferents moviments que es produeixen per una serie factors**, tant de moviment interns (compra-venda), com externs (moviments polítics, econòmics, etc...), en els principals índexs borsatils espanyols i de les crypto-monedes.

Hem seleccionat diferents fonts de dades per generar fitxers «csv», **guardar diferents valors en el mateix període de temps**. És important destacar que ens interessa més les tendències alcistes o baixes, que podem calcular o recuperar en aquests períodes de temps, en el nou dataset un cop netejades les dades pasarem **totes les monedes a Euros, ja que en la cryptomonedes està en dolors, ho farem el script de neteja de python**.

En aquest cas **el «trading» en cryptomoneda** és relativament nou, força popular per la seva formulació com a mitja digital d'intercanvi, **utilitzant un protocol que garanteix la seguretat, integritat i equilibri** del seu estat de compte per mitjà d'un entramat d'agents.

La comunitat podrà respondre, entre altres preguntes, a:

Està afectant o hi ha **patrons comuns** en les cotitzacions de cryptomonedes i el mercat de valors principals del país d'Espanya?

Els efectes o agents **externs afecten per igual a les accions o cryptomonedes**

Hi ha **relacions cause efecte** entre les accions i cryptomonedes?

2. Neteja de les dades

2.1. Selecció de les dades d'interès a analitzar. Quins són els camps més rellevants per tal de respondre al problema?

En aquest cas el contingut està format per diferents csv, especialment tenim els fitxers de moviments de **cryptomoneda**, els quals s'ha generat **un fitxer per dia del període de temps estudiat**.

Pel que fa als moviments del principals **índexs borsatils s'ha generat una carpeta per dia del període, en cada directori un fitxer amb cadascun del noms dels índexs**. Degut això s'han comprimit aquests últims abans de publicar-los en el directori de «open data» kaggle.com.

Pel que fa als camps, ens **interessà detectar els moviments alcistes i baixistes**, o almenys aquelles que tenen un patró similar en les cryptomonedes i els índexs. Els camps especialment son els camps comuns els qual netejarem i crearem el dataset parat per l'«script» d'R on generarem la correlació d'atributs i el model:

Camps comuns o nous (transformació i discretització):

- **Data:** Data de la observació
- **Nom:** Nom empresa o cryptomoneda, per identificar de quina moneda o index estem representant.
- **Símbol:** Símbol de la moneda o del index borsatil, per realitzar gràfic posteriorment d'una forma mes senzilla que el nom.
- **Preu:** Valor en euros d'una acció o una cryptomoneda (transformarem la moneda a euros en el cas de estigui en dòlars amb l'última cotització (un dollar a 0,8501 euro)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	0.00	0.09	49.96	2.18	82001.24

Clarament tenim valors extrem legítims degut al bitcoin.

- **Volum:** En euros/volum 24 hores,acumulat de les transaccions diàries en milions d'euros.
- **Tipus_cotitzacio:** Valor nou que agregarem per discretitzar entre la cotització: baix (0 i 1), normal (1 i 100), alt (100 i 1000), molt_alt (>1000)

alta	baixa	molt_alta	normal
625	32827	231	13176

```
def get_type_value(value):
    if value <= 1:
        return 'baixa'
    if 1 < value <= 100:
```

```

        return 'normal'
    if 100 < value <= 1000:
        return 'alta'
    else:
        return 'molt_alta'

return 'molt_alta'

```

- **Tipus:** Generarem aquest per agrupar i identificar cryptomonedes de índex de borsa, tindrà 2 valors: cryptomonedes/borsa.

```

crypto_moneda :36753
index_borsatil:10106

```

Crypto-currencies (no els utilitzarem):

- **Simbol:** Símbol o acrònim de la moneda
- **Cap de mercat:** Valor total de totes les monedes en el moment actual
- **Oferta circulant:** Valor en oportunitat de negoci
- **% 1h, % 2h i %7d,** tant per cent del valor la moneda en 1h, 2h o 7d sobre la resta de cryptomonedes.

Stock Index (no els utilitzarem):

- **Estat:** Estat final en tancament en alta o baixa del dia.
- **Var. Per cent:** Variació en el moment del tancament amb tant per cent respecte el dia anterior
- **Var. En euros:** Variació en el moment del tancament amb euros respecte el dia anterior.
- **Capitalització:** Valor de l'empresa respecte les seves accions.
- **PER:** La ràtio preu-benefici
- **Rent./Div:** Rendibilitat de l'acció respecte el valor inicial de la acció.

2.2. Les dades contenen zeros o elements buits? I valors extrems? Com gestionaries cadascun d'aquests casos?

Com es pot veure en script de python per netejar les dades: **clearData.py** aquest netejar, transforma i unifica les dades procedents de les diferents fonts en el fitxer dataset.csv.

El dataset.csv resultat amb les dades preparades per anàlisi tenen els camps:

```
HEADERS = ['Data', 'Tipus', 'Nom', 'Simbol', 'Preu (Euros)', 'Tipus_cotitzacio']
```

Disposem de 46859 observacions amb 6 atributs:

	Data	Tipus	Nom	Simbol	Preu..Euros.	Tipus_cotitzacio
1	2017-11-19	crypto_moneda	BTCBitcoin	BTC	6.832132e+03	molt_alta
2	2017-11-19	crypto_moneda	ETHEthereum	ETH	2.985163e+02	alta
3	2017-11-19	crypto_moneda	BCHBitcoin Cash	BCH	1.003034e+03	molt_alta
4	2017-11-19	crypto_moneda	XRPRipple	XRP	1.952729e-01	baixa
5	2017-11-19	crypto_moneda	LTCLitecoin	LTC	6.010265e+01	normal
6	2017-11-19	crypto_moneda	DASHDash	DASH	3.720960e+02	alta

Data
dataset 46859 obs. of 6 variables

Pel que fa les estratègies que hem seguir per al zeros i elements buits són:

- Les monedes o index borsatils que puntualment no hi havia preu, estava en blanc s'ha agregar zeros, ja que son poques dades que estan buides i es un factor molt important per al nostre objectiu.

```
if value:
    return float(value) * EURO_VALUE_FROM_DOLAR
else:
    return 0
```

- Pel que fa les index borsatils, que no hi havia símbol no es disposava, s'ha agregat els tres primers caràcters del nom del valor, per poder realitzar gràfics posteriorment amb una nom curt.

```
format(get_value_euros(row[4][1:]))
```

Finalment sobre les monedes s'ha passat a euros, ja que les cryptomonedes el preu era amb dollars, i els index borsatils en euros, amb una precisió de decimals de 9 dígit, què és recomana per monedes.

A mes s'ha unificat la data dels valors amb un format estandard YYYY-m-d.

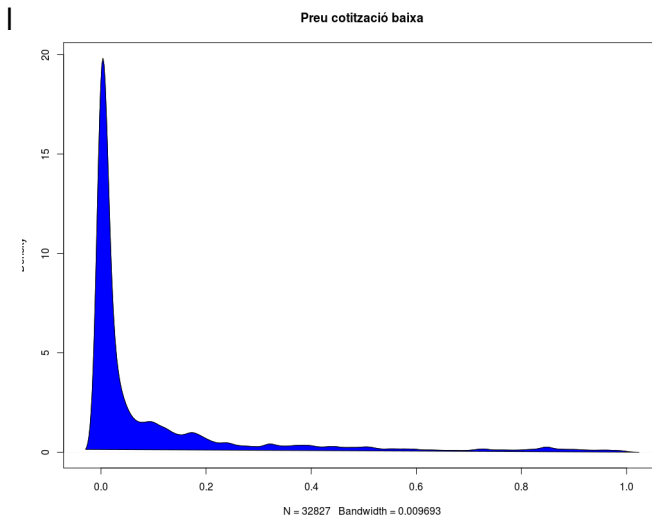
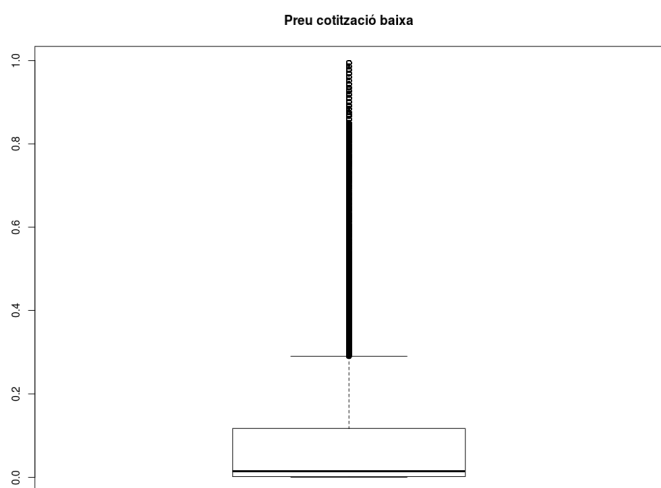
```
FORMAT_DATA = '%d_%m_%Y'
```

Pel que fa els valors extrems em d'observar primer la freqüències, ho farem pel tipus de cotització que hem discretitzat, tindrem una primera aproximació a la similitud de tendència:

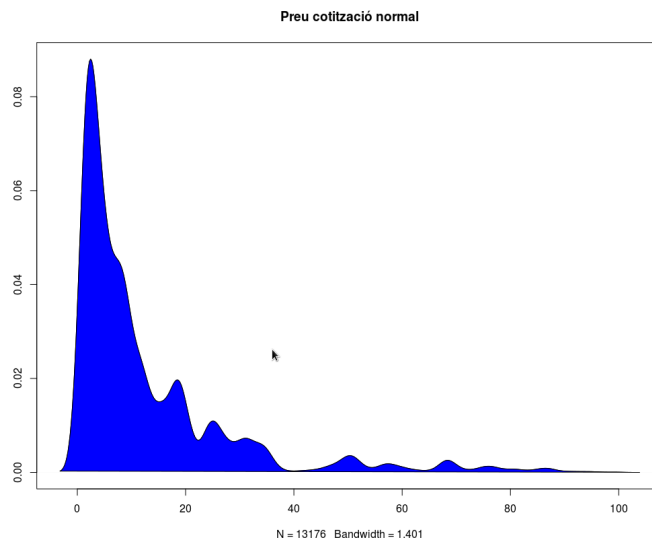
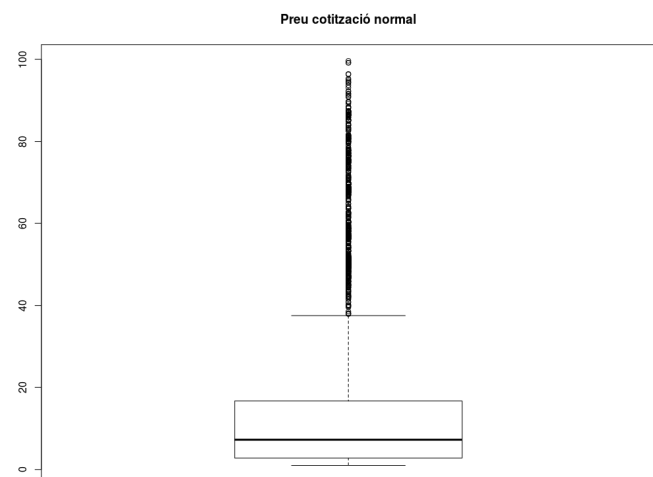
La diferencia entre cryptomonedes es molt alta, especialment amb el màxim o valor extrem del bitcoin, amb la resta. Per veure la densitat i al cert valors extrems legítims utilitzarem el tipus de cotització (baixa, normal, alta i molt alta) per veure la freqüència.

En definitiva els **valors extrems en aquest es tracta de valors atípics legítims**, no son degut a cap error humà o de mostreig, i en aporta una major font d'informació ja que possiblement si hi ha una tendència enter els stock index i les cryptomendes serà precisament pel bitcoin, com valor extrem de la resta de cryptomonedes.

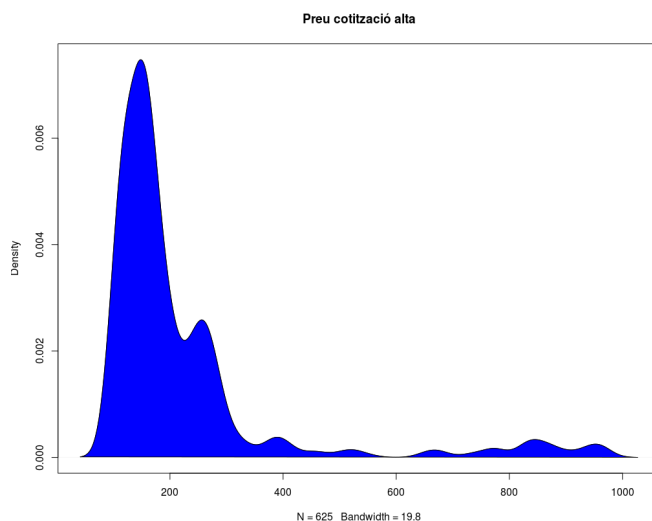
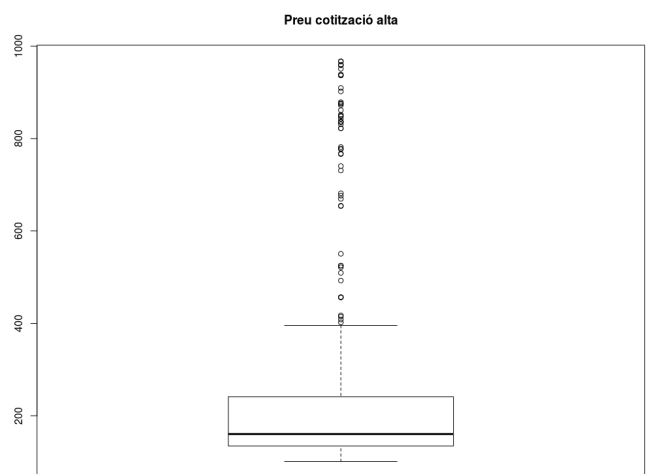
Cotització baixa:



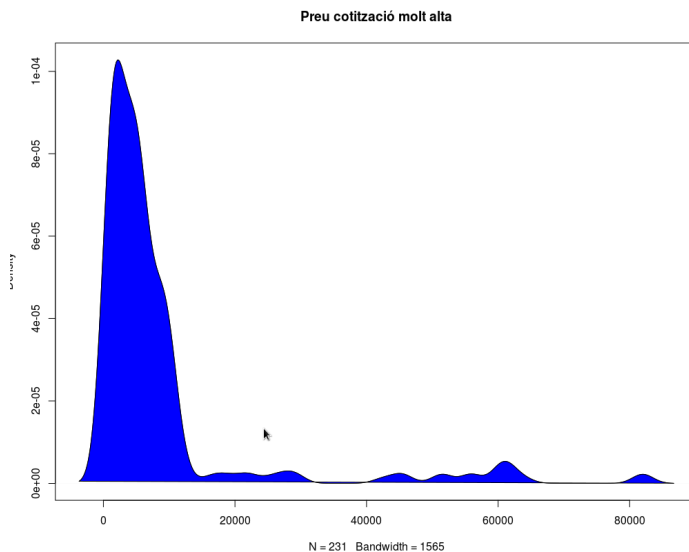
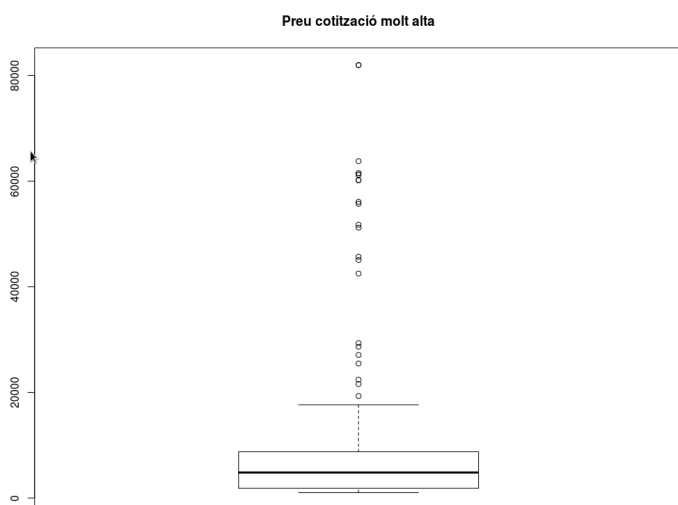
Cotització normal:



Cotització alta:

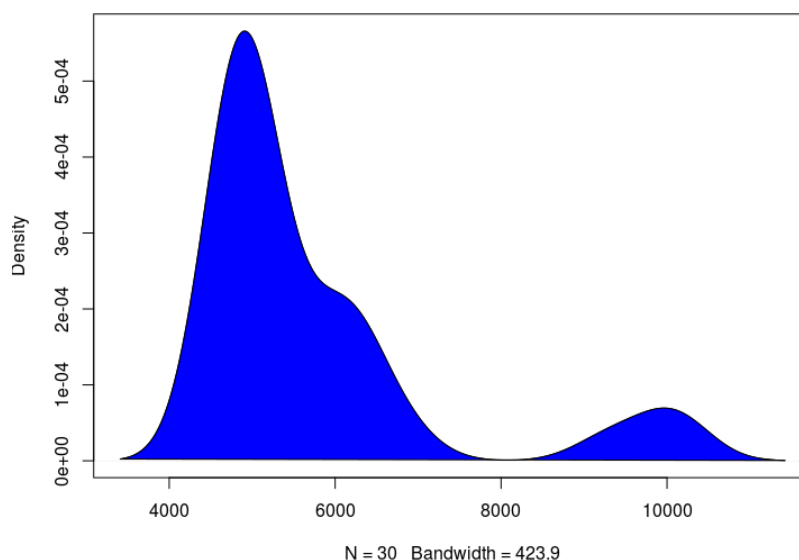


Cotització molt alta:



Com podem veure en cada grup del tipus de cotització la distribució es comporta de la mateixa forma, la gran concentració de valors està en el rang més baix de preu de cada discretització, per tant tenim representat valors extrems legítims que ens indiquen que sempre hi ha cotització molt més altes que la majoria es on hi ha més volum d'euros en les accions o cryptomonedes.

Una observació curiosa és un valor extrem el **bitcoin com a** cryptomonedra que està en un moviment alcista molt clar. Podem fer una distribució del bitcoin per veure aquesta tendència:



3. Anàlisis de les dades:

3.1. Selecció dels grups de dades que es volen analitzar/comparar.

En aquest cas el grup de dades que volem comprar son el preu de les cotitzacions, i **discretitzar per grup de baixa, normal, alta i molt alta**, per veure relacions del valor qualitatiu de tipus, si es cryptomoneda o index borsatil:

- **Data:** Data de la observació
- **Nom:** Nom empresa o cryptomoneda, per identificar de quina moneda o index estem representant.
- **Símbol:** Símbol de la moneda o del index borsatil, per realitzar gràfic posteriorment d'una forma mes senzilla que el nom.
- **Preu:** Valor en euros d'una acció o una cryptomoneda (transformarem la moneda a euros en el cas de estigui en dòlars amb l'última cotització (un dollar a 0,8501 euro))
- **Tipus_cotitzacio:** Valor nou que agregarem per discretitzar entre la cotització: baix (0 i 1), normal (1 i 100), alt (100 i 1000), molt_alt (>1000)

Tenim 2 grups cyptomonedas i «stock index», a més de 4 grups per tipus de preu en la cotització, posteriorment utilitzarem algorismes de agrupació per veure similitud amb el primer grup.

3.2. Comprovació de la normalitat i homogeneïtat de la variància. Si és necessari (i possible), aplicar transformacions que normalitzin les dades.

Com hem vist en la pregunta 2, no hi hagut mes remei que discretitzar per tipus

de preu en la transacció, ja que el valors extrems de algunes cryptomonedes eren molt alt (bitcoin), igual es ha passat en les stock indexs.

Amb el test de Levene amb els grups de cryptomonedes i stock index, p-valor es menor de 0.05 per tant es rechaza la hipotesi null:

```
Rcmdr> with(dataset, tapply(Preu..Euros., Tipus, var, na.rm=TRUE))
crypto_moneda index_borsatil
1758612.334      501.601

Rcmdr> leveneTest(Preu..Euros. ~ Tipus, data=dataset, center="median")
Levene's Test for Homogeneity of Variance (center = "median")
      Df F value Pr(>F)
group  1 13.562 0.000231 ***
      47183
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Per tant hem aplicat transformació per obtenir grup homogenis i poder tenir un gràfic adequat amb la freqüència dels valors. Tot i això aplicarem un algorisme de agrupació en la pregunta 5, i ara realitzarem el test de Levene per veure la homogeneïtat dels grups per tipus de valor:

```
Rcmdr> with(dataset, tapply(Preu..Euros., list(Tipus, Tipus_cotitzacio), var, na.rm=TRUE))
      alta      baixa molt_alta  normal
crypto_moneda 38146.8202 0.03796463 203807965 236.7990
index_borsatil  506.1773 0.06039860      NA 238.1323

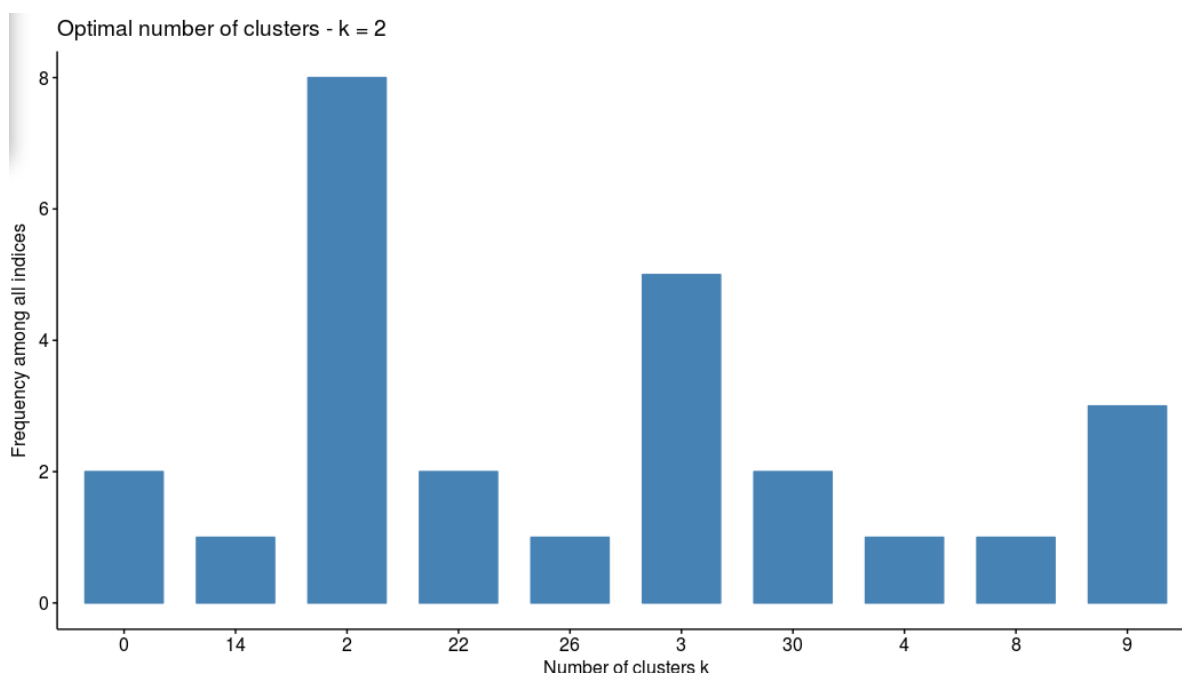
Rcmdr> leveneTest(Preu..Euros. ~ Tipus*Tipus_cotitzacio, data=dataset, center="median")
Levene's Test for Homogeneity of Variance (center = "median")
      Df F value Pr(>F)
group   6 1864.7 < 2.2e-16 ***
      47178
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tampoc tenim normalitat ni grups homogenis, degut al valors extrems que no em descartat segurament, en tot cas, el que farem al punt 3.3 es usar algorisme de kmeans per trobar la agrupació mes optima i poder veure realment si hi ha una relació entre als valors de la cryptomonedes i els index borsatil.

3.3. Aplicació de proves estadístiques (tantes com sigui possible) per comparar els grups de dades.

En aquest cas utilitzarem un model d'agregació que ens permetrà fer perdicions

d'atributs observant els veïns més pròxims, per majoria o mitjana. El que farem es normalitzar els valors i utilitzarem l'algorisme kmeans amb R per crear cluster o grups de dades òptims, a més del NbClust un paquet que iterare amb algorisme kmeans per proposar el numero de clusters o grups mes òptims:



Conclusion

=====

* According to the majority rule, the best number of clusters is 2 .

Podem veure que el cluster mes optim està formar per 2 grups:

```
clusters_2 <- kmeans(price_norm,2, 15)
print(clusters_2)

K-means clustering with 2 clusters of sizes 47168, 17

Cluster means:
Preu..Euros.
1  -0.01698065
2  47.11429870

Within cluster sum of squares by cluster:
[1] 6930.837 2503.691
(between_SS / total_SS = 80.0 %)
```

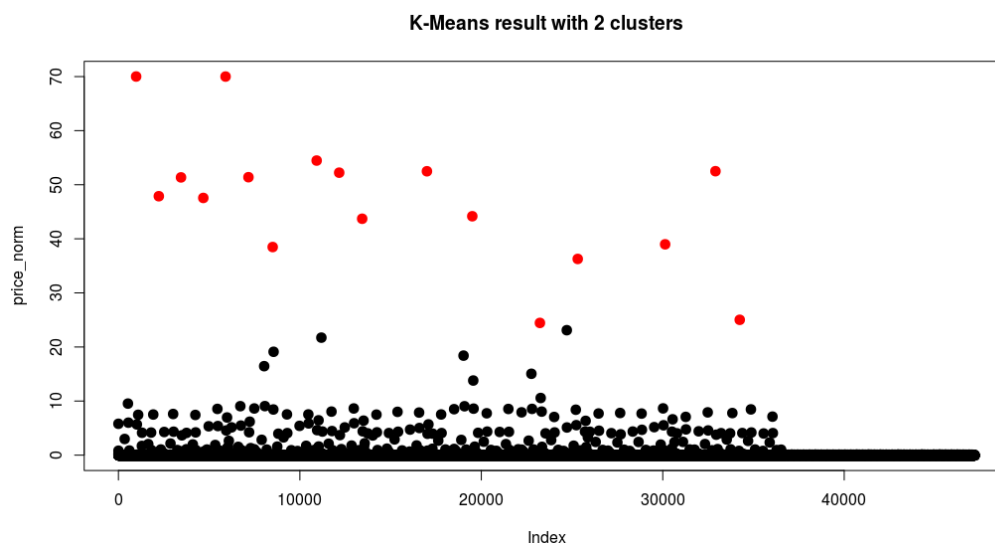
Un 80% es una agrupació de molt bona qualitat, anem a representar-ho.

4. Representació dels resultats a partir de taules i gràfiques.

Durant tot la practica ja em anat representat els resultat, tant les freqüències per veure els valors extrems, al tenir sols un valor quantitatiu, el preu, no em pogut fer una regressió lineal, d'altra banda els hem vist amb facilitat.

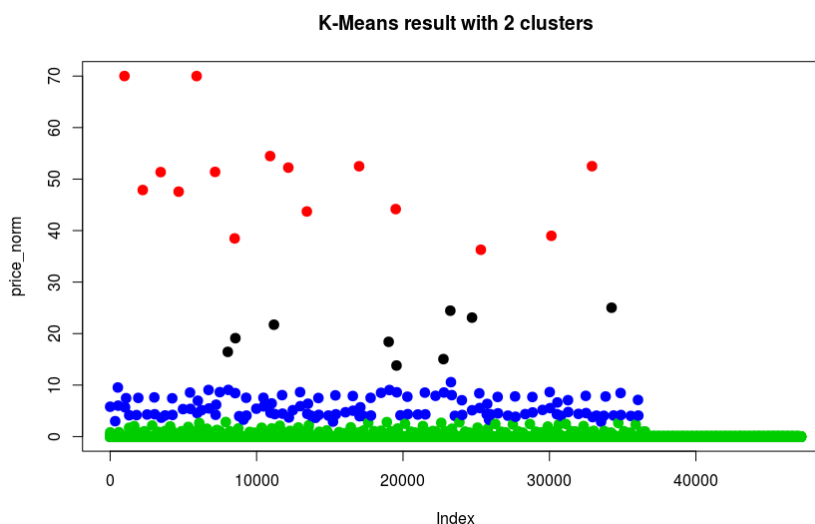
Finament, em pogut agrupar mitjançant algoritme de kmeans amb 2 grups, hem normalitzat les dades perquè no ens interessa donar mes pes al valor mes alt, el nostres objectiu es veure tendències. Un cop em normalitzat les dades que podem veure gràficament:

```
plot(price_norm, col =(clusters_2$cluster) , main="K-Means result with 2 clusters", pch=20, cex=2)
```



Per alta banda, si realitzem una agrupació per 4 grups, podem veure es molt similar a la discretització per tipus de preu (baix, normal, alt i molt alt):

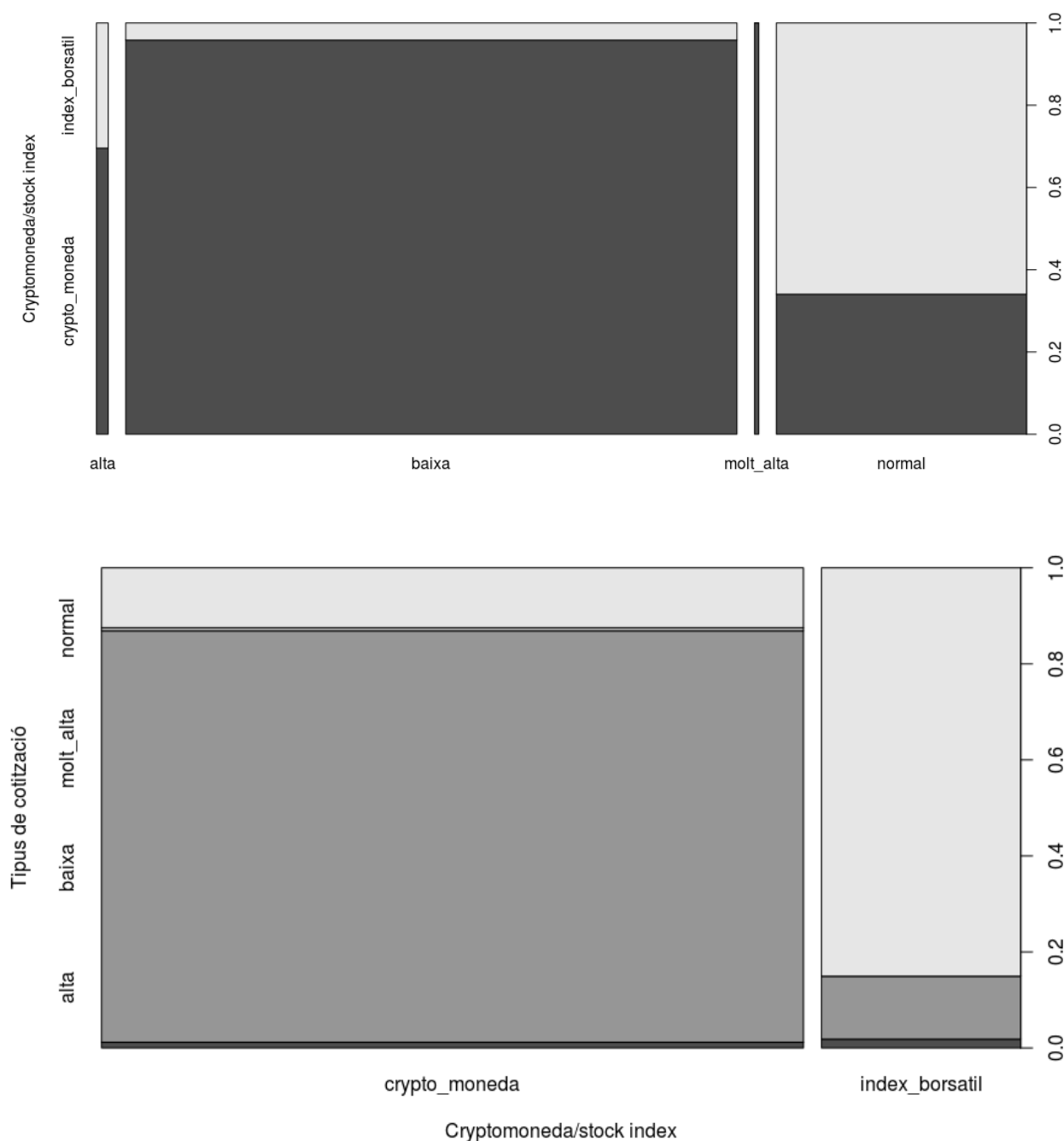
```
plot(price_norm, col =(clusters_4$cluster) , main="K-Means result with 4 clusters", pch=20, cex=2)
```



A és podem veure la distribució de les discretitzacions o grups:

```
#comparació tipus de preus o rangs amb cryptomonedes/stock índex
plot(dataset$Tipus_cotitzacio, dataset$Tipus, xlab = "Tipus de cotització", ylab = "Cryptomoneda/stock index")

#inversa de la anterior comparació de cryptomonedes i sotck index amb el tipus de preus
plot(dataset$Tipus, dataset$Tipus_cotitzacio, xlab = "Cryptomoneda/stock index", ylab = "Tipus de cotització")
```



5. Resolució del problema.

A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?

6. Codi:

Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python.

El codi amb python i R està en <https://github.com/acostasg/scraping>, exactament hem de destacar:

- El fitxer de codi de python per la neteja i transformació de dades
 - <https://github.com/acostasg/scraping/blob/master/clearData.py>
- I el fitxer de codi R amb l'anàlisi i els gràfics del conjunt de dades netejades i transformades (s'ha discretitzat un atribut):
 - <https://github.com/acostasg/scraping/blob/master/R/script.r>

Referencies

- <https://stackoverflow.com/questions/6159900/correct-way-to-write-line-to-file-in-python>
- Llibre manual: Richard Lawson. Web Scraping with Python. Packt Publishing Ltd, October 2015. 174 p. ISBN 9781782164371
- Regressió lineal simple amb R.
https://www.uam.es/personal_pdi/ciencias/joser/paginaR/regresion.html
- The R Graph Gallery. <https://www.r-graph-gallery.com/>
- Exemple de com fer un gràfic de Kmeans usant ACP. <http://apuntes-r.blogspot.com.es/2014/10/ejemplo-graficar-kmeans-usando-acp.html>
- <https://docs.python.org/2/tutorial/inputoutput.html>