Pràctica 2: Neteja i validació de les dades

José Antonio Forcada Sans

1 de gener de 2022

Índex

1. Detalls de l'activitat 3

1.1 Descripció 3

1.2. Objectius 3

1.3. Competències 3

2. Resolució 4

2.1. Descripció del dataset 4

2.2. Importància i objectius dels anàlisis 4

2.3. Neteja de dades 5

2.3.1. Integració i selecció de les dades d'interès a analitzar 5

2.3.2. Zeros i elements buits 6

2.3.3. Valors extrems 7

2.4. Anàlisi de les dades 10

2.4.1. Selecció dels grups de dades que es volen analitzar/comparar 11

2.4.2. Comprovació de la normalitat i homogeneïtat de la variància 12

2.4.3. Estimació del preu segons els diferents atributs 13

3. Conclusions 15

1. Detalls de l'activitat

1.1 Descripció

En aquesta pràctica s'elabora un cas pràctic orientat a aprendre a identificar les dades
rellevants per un projecte analític i usar les eines d'integració, neteja, validació i anàlisi de les mateixes

1.2. Objectius

Els objectius que es volen treballar en aquesta pràctica són el següents:

- Aprendre a aplicar els coneixements adquirits i la seva capacitat de resolució de problemes en entorns nous o poc coneguts dintre de contextos més amplis o multidisciplinaris.
- Saber identificar les dades rellevants i els tractaments necessaris (integració, neteja i validació) per dur a terme un projecte analític.
- Aprendre a analitzar les dades adequadament per abordar la informació continguda en les dades.
- Identificar la millor representació dels resultats per tal d'aportar conclusions sobre el problema plantejat en el procés analític.
- Actuar amb els principis ètics i legals relacionats amb la manipulació de dades en funció de l'àmbit d'aplicació.
- Desenvolupar les habilitats d'aprenentatge que els permetin continuar estudiant d'una manera que haurà de ser en gran manera autodirigida o autònoma.
- Desenvolupar la capacitat de cerca, gestió i ús d'informació i recursos en l'àmbit de la ciència de dades.

1.3. Competències

Les competències del Máster en Data Science que es portaran a terme són:

- Capacitat d'analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i resoldre'l.
- Capacitat per aplicar les tècniques específiques de tractament de dades (integració, transformació, neteja i validació) per al seu posterior anàlisi.

2. Resolució

2.1. Descripció del dataset

Aquest conjunt de dades s'ha obtingut mitjançant la implementació de la primera pràctica. A fi de poder entendre quina informació ens dona el nostre Dataset ens proposem a descriure'l:

Nom del atribut	Descripció	Tipus de variable
location	Localització. Barri/Districte	string
surface	Superfície del immoble	integer
rooms	Nombre d'habitacions	integer
bathrooms	Nombre de banys	integer
surface_price_rate	Ràtio preu/superfície	double
premium?	És un anunci premium	boolean
days_from_last_update	Dies des de la última actualitació per part de l'anunciant	integer
price	Preu	double

2.2. Importància i objectius dels anàlisis

A partir d'aquest Dataset es pot resoldre el repte de determinar quins dels atributs descrits anteriorment tenen una major influència en el preu dels diferents immobles descrits. A fi d'aconseguir-ho aquest Dataset ens permetrà la creació de models que ens permetran realitzar aquest anàlisis.

Una de les aplicacions més importants que poden tenir els possibles anàlisis fets sobre aquestes dades són verificacions dels diferents reglaments sobre els habitatges en lloguer a Barcelona. Un sistema automàtic de detecció de violacions dels reglaments podria evitar l'ús fraudulent de portals d'habitatge com és el cas de Habitaclia.es. També ens permetrien detectar estratègies poc ètiques per part de les immobiliàries que busquen confondre els possibles llogaters.

2.3. Neteja de dades

Abans de començar amb els nostres anàlisis caldrà realitzar diferents accions prèvies a fi de tenir unes dades vàlides i correctes que evitin errors ens els anàlisis.
Primer de tot ens caldrà carregar les nostres dades:

```
rent_data<- read.csv("barcelona_rents.csv", header= T, sep=",")
head(rent_data)
```

	X <int>	location <chr>	surface <int>	rooms <int>	bathrooms <int>	surface_price_rate <chr>	premium. <chr>	days_from_last_update <int>	price <int>
1	0	Barcelona - La Marina-MontjuÃ¯c	76	3	2	12,50	True	23	950
2	1	Barcelona - Dreta de lÂ´Eixample	79	2	2	15,06	True	0	1190
3	2	Barcelona - Sant Gervasi - Galvany	100	3	2	14,06	True	2	1406
4	3	Barcelona - Sagrada FamÃlia	59	2	1	16,22	True	22	957
5	4	Barcelona - Raval	90	3	1	16,56	True	2	1490
6	5	Barcelona - Sagrada FamÃlia	105	4	2	9,86	True	3	1035

6 rows

Podem executar la següent línia de codi per a poder analitzar el tipus de cada atribut llegit:

```
sapply(rent_data, class)
```

```
X              location               surface                 rooms
bathrooms    surface_price_rate              premium.
         "integer"           "character"               "integer"
"integer"              "integer"           "character"
"character"
days_from_last_update                 price
            "integer"             "integer"
```

Podem veure com excepte pel cas dels atributs “Premium” i “Surface_prince_rate” el tipus coincideix amb lo esperat.

2.3.1. Integració i selecció de les dades d’interès a analitzar

Podem veure com totes les variables extretes són potencialment útils per a el nostre anàlisis. Ara bé, la variable “X” que només indica l’orde no es serà necessària per tant eliminarem la primera columna

```
rent_data <- rent_data [,(-1),drop = FALSE]
sapply(rent_data, class)
```

```
location                surface                    rooms
bathrooms     surface_price_rate               premium.
            "integer"            "character"              "integer"
"integer"              "integer"            "character"
"character"
days_from_last_update                 price
            "integer"              "integer"
```

Observem com hem eliminat la columna 'X'

2.3.2. Zeros i elements buits

Primer de tot haurem de convertir l'atribut "Surface_price_rate" a "float":

```
rent_data$surface_price_rate <- sub(".","",rent_data$surface_price_rate,fixed = TRUE)
rent_data$surface_price_rate <- sub(",",".",rent_data$surface_price_rate)
rent_data$surface_price_rate = as.double(rent_data$surface_price_rate)
```

A fi de poder realitzar els nostres anàlisis caldrà identificar i tractar de manera adient aquells valors que siguin 0, nuls o buits.
Per a aconseguir-ho executarem les següents línies de codi:

```
summary(rent_data)
```

```
  location            surface           rooms          bathrooms      surface_price_rate   premium.
Length:5657        Min.   :  1,00   Min.   : 1,000   Min.   : 0,000   Min.   :    0,98   Length:5657
Class :character   1st Qu.: 56,00   1st Qu.: 2,000   1st Qu.: 1,000   1st Qu.:   13,01   Class :character
Mode  :character   Median : 72,00   Median : 2,000   Median : 1,000   Median :   16,27   Mode  :character
                   Mean   : 89,82   Mean   : 2,526   Mean   : 2,146   Mean   :   28,38
                   3rd Qu.: 96,00   3rd Qu.: 3,000   3rd Qu.: 2,000   3rd Qu.:   22,50
                   Max.   :981,00   Max.   :59,000   Max.   :96,000   Max.   :14666,67
                   NA's   :3        NA's   :9        NA's   :9        NA's   :36
days_from_last_update     price
Min.   : 0,00         Min.   :    220
1st Qu.: 0,00         1st Qu.:    900
Median : 2,00         Median :   1200
Mean   : 3,61         Mean   :   2580
3rd Qu.: 5,00         3rd Qu.:   1812
Max.   :47,00         Max.   :2200000
                      NA's   :2
```

Observem com les columnes "Surface", "rooms", "bathrooms", "Surface_price_rate" i "price" tenen elements NaN. A fi d'eliminar-los farem:

```
rent_data <- na.omit(rent_data)
summary(rent_data)
```

```
  location            surface           rooms          bathrooms      surface_price_rate   premium.
Length:5621        Min.   :  1,00   Min.   : 1,000   Min.   : 1,000   Min.   :    0,98   Length:5621
Class :character   1st Qu.: 56,00   1st Qu.: 2,000   1st Qu.: 1,000   1st Qu.:   13,01   Class :character
Mode  :character   Median : 72,00   Median : 2,000   Median : 1,000   Median :   16,27   Mode  :character
                   Mean   : 89,76   Mean   : 2,457   Mean   : 1,988   Mean   :   28,38
                   3rd Qu.: 96,00   3rd Qu.: 3,000   3rd Qu.: 2,000   3rd Qu.:   22,50
                   Max.   :981,00   Max.   :25,000   Max.   :47,000   Max.   :14666,67
 days_from_last_update     price
 Min.   : 0,000        Min.   :    220
 1st Qu.: 0,000        1st Qu.:    900
 Median : 2,000        Median :   1200
 Mean   : 3,615        Mean   :   2583
 3rd Qu.: 5,000        3rd Qu.:   1815
 Max.   :47,000        Max.   :2200000
```

Podem veure com s'han eliminat aquelles línies amb valors NaN, és a dir amb dades inconsistents. En aquest cas s'ha preferit eliminar les dades anòmales abans que intentar corregir-les. Per haver-ho fet caldria que les diferents línies tinguessin alguna relació amb les veïnes (per a poder extrapolar). A més el nombre total de línies eliminades ha sigut de 44 lo que representa només un 0.78% del total de ítems en el Dataset.

Ara procedirem a eliminar els valors = 0. Si ens fixem en la última taula, només l'atribut "days_from_last_update" conté valors 0. Si tenim en compte la informació que proporcionen podem veure que en aquest cas és un valor correcte, ja que si un anunci fou publicat el mateix dia que l'extracció el nombre de dies transcorreguts des de l'última actualització serà igual 0, per tant té sentit.

Ara anem a comprovar si per als atribut del tipus "character" tenim valors buits ("") i en cas afirmatiu els eliminarem:

```
nrow(rent_data)
rent_data = rent_data[!(is.na(rent_data$location) |
rent_data$location==""), ]
rent_data = rent_data[!(is.na(rent_data$premium) |
rent_data$premium==""), ]
nrow(rent_data)
```

```
[1] 5621
[1] 5621
```

Observem doncs que no s'ha eliminat cap ítem doncs per cap de les dues columnes teníem valors buits.

2.3.3. Valors extrems

Un cop identificats i eliminats aquells items amb valors buits o nuls, ens caldrà trobar i eliminar aquells valors que no són realistes i que per tant no són correctes.

Per a fer-ho tornarem a executar la comanda summary per veure quins són els valors màxims i mínims.

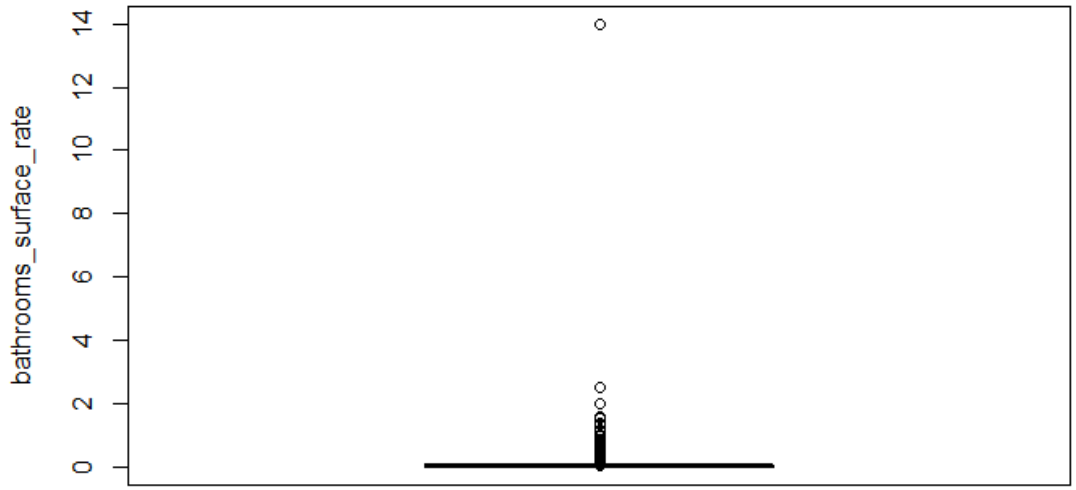
```
summary(rent_data)
```

```
   location            surface          rooms          bathrooms      surface_price_rate   premium.
 Length:5621        Min.   :  1,00   Min.   : 1,000   Min.   : 1,000   Min.   :    0,98   Length:5621
 Class :character   1st Qu.: 56,00   1st Qu.: 2,000   1st Qu.: 1,000   1st Qu.:   13,01   Class :character
 Mode  :character   Median : 72,00   Median : 2,000   Median : 1,000   Median :   16,27   Mode  :character
                    Mean   : 89,76   Mean   : 2,457   Mean   : 1,988   Mean   :   28,38
                    3rd Qu.: 96,00   3rd Qu.: 3,000   3rd Qu.: 2,000   3rd Qu.:   22,50
                    Max.   :981,00   Max.   :25,000   Max.   :47,000   Max.   :14666,67
 days_from_last_update     price
 Min.   : 0,000        Min.   :    220
 1st Qu.: 0,000        1st Qu.:    900
 Median : 2,000        Median :   1200
 Mean   : 3,615        Mean   :   2583
 3rd Qu.: 5,000        3rd Qu.:   1815
 Max.   :47,000        Max.   :2200000
```

Observem com tenim valors poc realistes a 'surface', 'surface_price_rate', 'bathroms' i 'price'.

A fi de poder comptabilitzar, identificar i eliminar aquests outliers anirem fixant-nos un per un.
Comencem per l'atribut 'bathrooms':

```
#Primer de tot tractem l'atribut 'bathrooms'
#Creem un nou atribut bathrooms surface a fi de detectar els outliers.
rent_data$bathrooms_surface_rate = rent_data$bathrooms/rent_data$surface
```

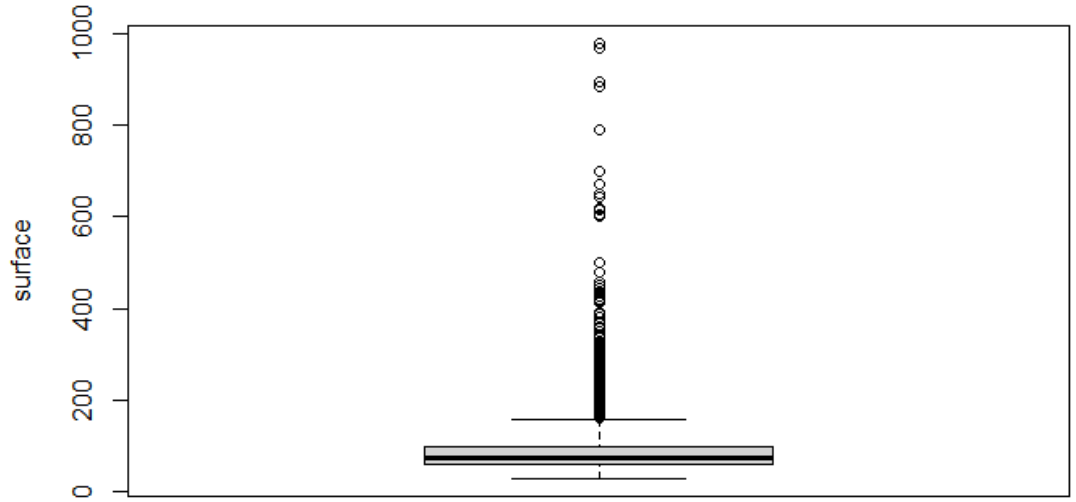



```
#Observem com tenim outliers clars. Procedim a eliminar-los.
nrow(rent_data)
rent_data = rent_data[-which(rent_data$bathrooms_surface_rate %in%
boxplot.stats(rent_data$bathrooms_surface_rate)$out),]
rent_data$bathrooms_surface_rate <- NULL
nrow(rent_data)
```

```
[1] 5621
[1] 5342
```

Un cop eliminades les 279 línies amb un valor de bathrooms fora de rang tractarem els outliers per a l'atribut 'surface'.

```
#Ara ens toca repetir el procés per a l'atribut 'surface'
boxplot(rent_data$surface,ylab = "surface")
```

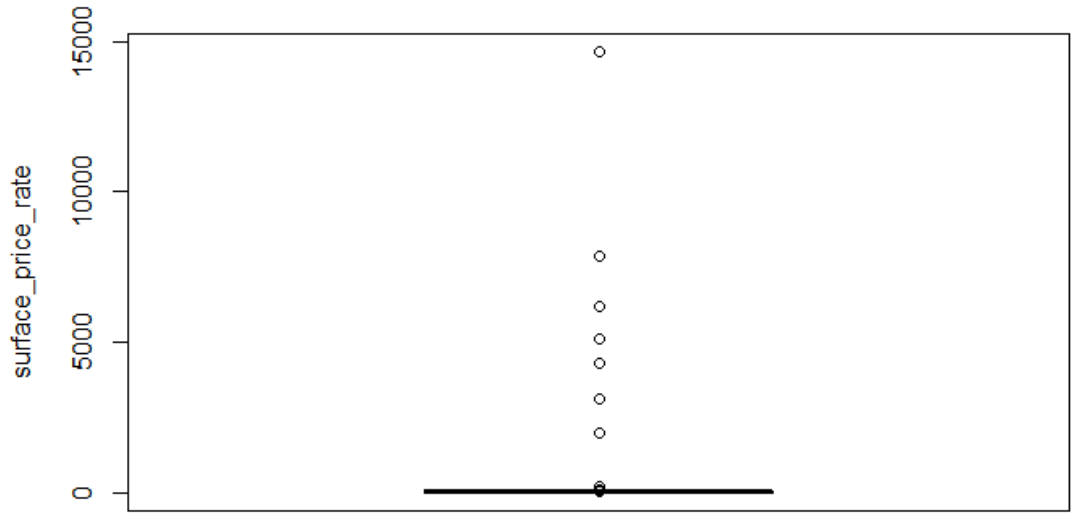



```
#Observem com tenim outliers clars. Procedim a eliminar-los.
nrow(rent_data)
rent_data = rent_data[-which(rent_data$surface %in%
boxplot.stats(rent_data$surface)$out),]
nrow(rent_data)
```

```
[1] 5342
[1] 4829
```

Continuem ara per l'atribut 'surface_price_rate'.

```
#Ara ens toca repetir el procés per a l'atribut 'surface_price_rate'
boxplot(rent_data$surface_price_rate,ylab = "surface_price_rate")
```

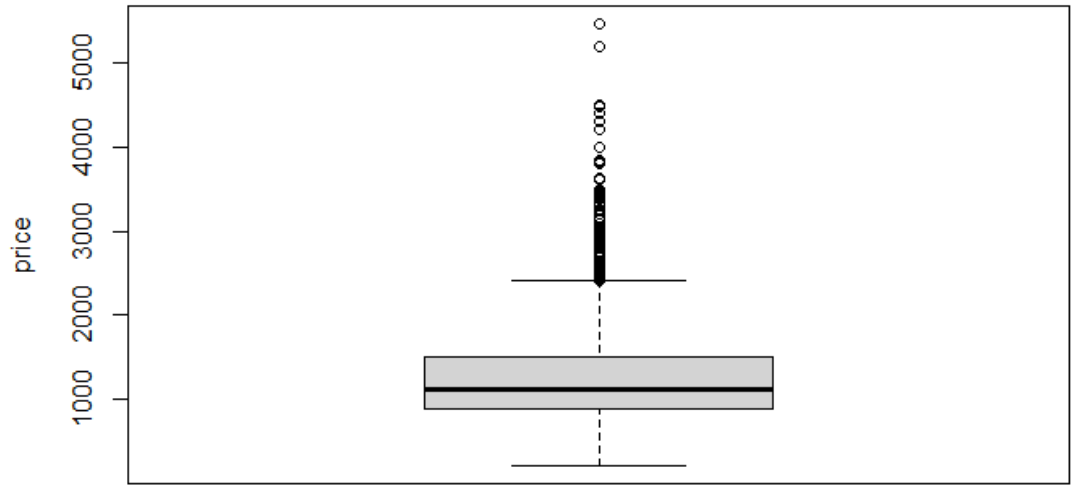



```
#Observem com tenim outliers clars. Procedim a eliminar-los.
nrow(rent_data)
rent_data = rent_data[-which(rent_data$surface_price_rate %in%
boxplot.stats(rent_data$surface_price_rate)$out),]
nrow(rent_data)
```

```
[1] 4829
[1] 4514
```

Per últim procedim igual per a l'atribut 'price'.

```
#Ara ens toca repetir el procés per a l'atribut 'price'
boxplot(rent_data$price,ylab = "price")
```




```
#Observem com tenim outliers clars. Procedim a eliminar-los.
nrow(rent_data)
rent_data = rent_data[-which(rent_data$price %in%
boxplot.stats(rent_data$price)$out),]
nrow(rent_data)
```

```
[1] 4514
[1] 4261
```

2.4. Anàlisi de les dades

Un cop amb les dades netes de valors zeros, nuls, buits i extrems podem procedir a l'anàlisi. En aquest apartat en plantegem tres punts a resoldre

Per a procedir a l'anàlisi ens caldrà convertir els atributs de tipus 'string' a numèric.

```
#Primer de tot ho passarem ambdos atributs a majúscules
rent_data$location <- toupper(rent_data$location)
rent_data$premium <- toupper(rent_data$premium.)
rent_data$premium.<- NULL

unique(rent_data$location)
unique(rent_data$premium)

#Primer de tot ho passarem ambdos atributs a majúscules
rent_data$location <- toupper(rent_data$location)
rent_data$premium <- toupper(rent_data$premium.)
rent_data$premium.<- NULL

unique(rent_data$location)
unique(rent_data$premium)

#Posterioment convertim els valors valor a numérics
newValue <- 0
for (value in unique(rent_data$premium))
{
  rent_data$premium[rent_data$premium==value]<- toString(newValue)

  newValue <- newValue + 1
}

newValue <- 0
for (value in unique(rent_data$location))
{
  rent_data$location[rent_data$location==value]<- toString(newValue)

  newValue <- newValue + 1
}

#Finalment convertim ambdos columnes a tipus integer
rent_data$location <- as.numeric(rent_data$location)
rent_data$premium <- as.numeric(rent_data$premium)
unique(rent_data$location)
unique(rent_data$premium)
```

```
[1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31 32 33 34
[36] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
58 59 60 61 62 63 64 65 66 67 68 69
[71] 70 71 72 73 74

[1] 0 1
```

Observem com s'han assignat valor numèrics als diferents valors de les columnes 'location' i 'premium'.

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

En aquest apartat se'ns demana que escollim aquells grups de dades sobre les quals volem procedir a l'anàlisi.
Per a esbrinar els atributs que influeixen més en el preu haurem de crear una matriu de correlacions de tots els atributs respecte el target 'price':

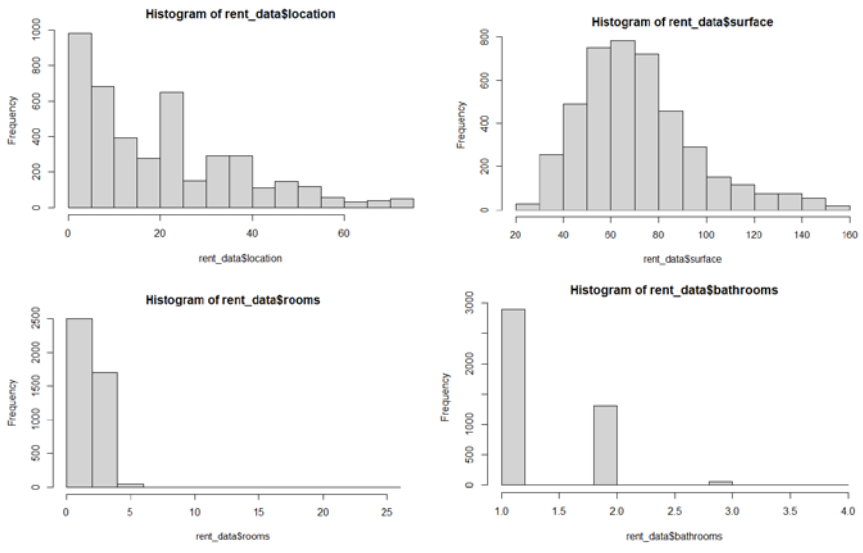
```r
```{r, echo=FALSE}
  corr_matrix <- data.frame(attribute = character(), correlation =
numeric())

  for (col.name in colnames(rent_data)){
    corr_matrix[nrow(corr_matrix) + 1,] =
list(col.name,cor(rent_data[c('price')], rent_data[c(col.name)],
use="complete.obs"))
  }
```
```

	attribute	correlation
1	location	-0.182260299
2	surface	0.525813356
3	rooms	0.159039245
4	bathrooms	0.473660482
5	surface_price_rate	0.476941803
6	days_from_last_update	-0.006922744
7	price	1.000000000
8	premium	0.005264104

Observem com l'atribut que més està correlacionada al preu és la superfície. Podem trobar també que els atributs 'premium', 'days_from_last_update' no són significatius i per tant els podem descartar per al nostre anàlisi. A més l'atribut 'surface_price_rate' no ens servirà ja que ja tenim un altre per 'surface' i el preu és precisament el que volem investigar.

2.4.2. Comprovació de la normalitat i homogeneïtat de la variància



Podem veure com només l'atribut 'surface' té distribucions normals.

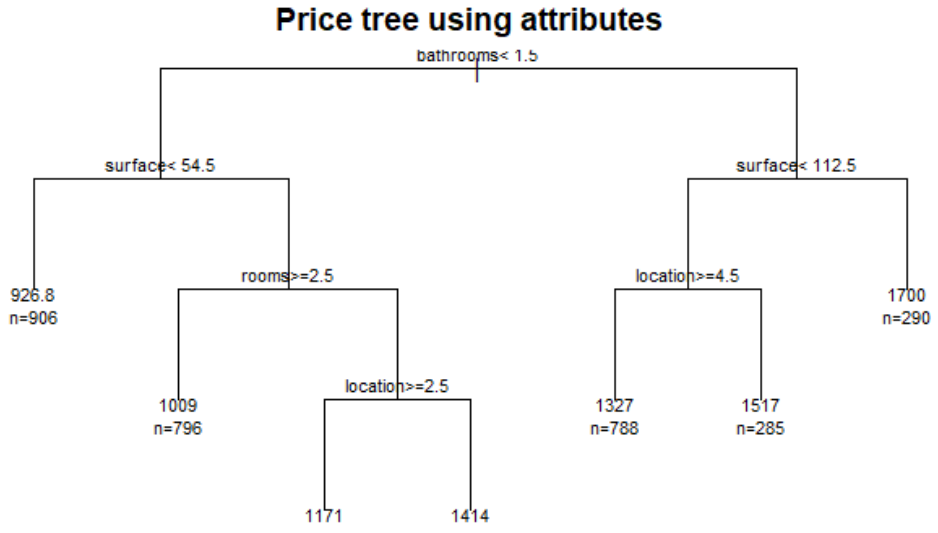
2.4.3. Estimació del preu segons els diferents atributs

Per a fer-ho generarem un model que inclogui aquells atributs amb una major correlació respecte el preu.

```
#primer creem dos subest de test i train

sample <- sample.int(n = nrow(rent_data), size =
floor(.70*nrow(rent_data)), replace = F)
train <- rent_data[sample, ]
test  <- rent_data[-sample, ]

#Procedmin a crear models de regressió per a estimar el preu i entrenem
fent servir les dades de train:
model <- rpart(price ~ surface + bathrooms + rooms+ location, method =
"anova", data = train)
summary(model)
plot(model, uniform = TRUE,
          main = "Price tree using attributes")
text(model, use.n = TRUE, cex = .7)
```



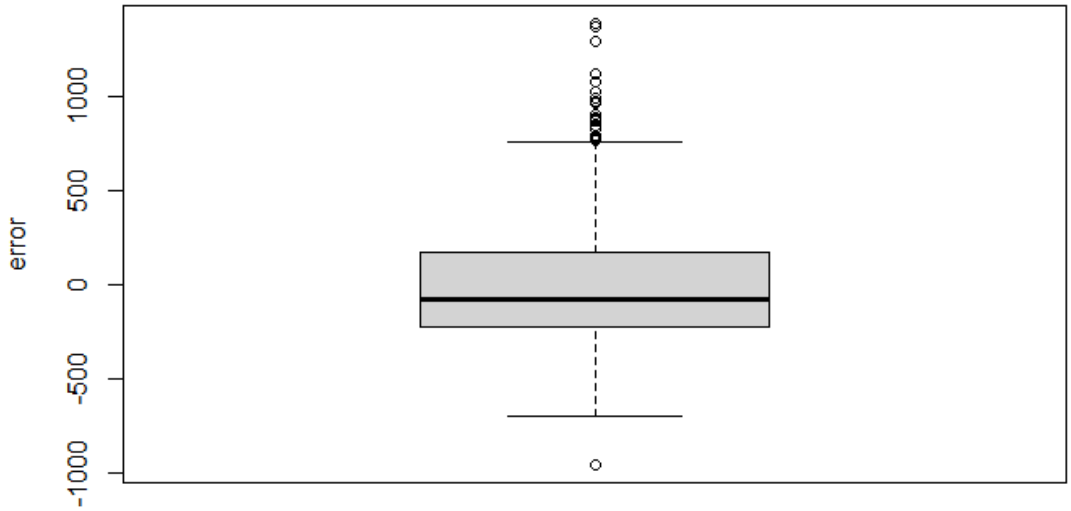
Un cop entrenat el model procedim a validar-lo amb les dades de test:

```
#Executem la validació
error <- data.frame(pred = predict(model,newdata=test), actual =
test$price)
mean((error$actual - error$pred)^2)
```

```
[1] 106721.6
```

Podem veure com s'ha obtingut un SEM força alt. A fi de veure com de vàlids són els resultats anem a veure com es distribueix l'error

```
#Mirem com queda distrbuït l'error:
boxplot(error$actual - error$pred,ylab = "error")
```



Observem com l'error en la predicció està molt centrar a 0 el que vol dir que la predicció és força bona malgrat tenim alguns resultats totalment erronis (outliers).

3. Conclusions

Podem concloure un cop fet aquesta pràctica que les característiques que més influeixen a l'hora d'establir un preu de lloguer d'un habitatge a la ciutat de Barcelona és la superfície, el nombre d'habitacions, el nombre de banys i la localització. Aquest estudi s'ha realitzat a partir de les dades extretes en la pràctica anterior tot eliminant aquells valors que s'han considerant erronis, o que no descrivien escenaris reals.
S'ha creat un model que s'ha entrant amb un conjunt de dades de train (creades com a subest del Dataset original) i posteriorment s'ha validat amb un altre subset de dades de test. Podem corroborar que excepte alguns error, el nostre model és capaç de predir amb un cert marge d'error el preu de l'habitatge. Els valors en que el model ha errat podrien ser casos on hi ha hagut error en la descripció o com s'ha dit abans en la introducció, casos on les característiques no siguin reals i per tant els anuncis no fossin correctes. Així doncs aquest model ens podria servir per validar els anuncis penjat al portal habitaclia.es i poder així detectar anuncis fraudulents.