

M2.951 - Tipologia i cicle de vida de les dades

Pràctica 2 - Neteja i anàlisi de les dades

Autors: Roger Peris Serrano i Albert Cámara Viñals

Desembre, 2021

Contents

1 Details de l'activitat	2
1.1 Presentació	2
1.2 Competències	2
1.3 Objectius	2
1.4 Descripció de la Pràctica a realitzar	2
1.5 Recursos	3
1.6 Criteris de valoració	3
1.7 Format i data de lliurament	3
2 Resolució	5
2.1 Descripció del dataset	5
2.2 Importància i objectius de l'anàlisi	6
2.3 Neteja de dades	6
2.3.1 Lectura de dades	6
2.3.2 Previsualització de les dades d'interès	7
2.3.3 Zeros i atributs buits	23
2.3.4 Valors extrems (Outliers)	24
2.3.5 Discretització de variables	24
2.3.6 Transformació d'atributs	25
2.3.7 Creació de nous indicadors	27
2.3.8 Exportació de les dades preprocesades	31
2.4 Anàlisi de les dades	32
2.4.1 Selecció dels grups de dades a analitzar	32
2.4.2 Comprobació de la normalitat	32
2.5 Test estadístics	50
2.5.1 ¿Quines variables quantitatives influeixen més a les valoracions?	50
2.5.2 ¿Els cotxes elèctric tenen un preu més elevat que els cotxes de benzina?	51
2.5.3 ¿La proporció de furgonetes és més petit que la d'utilitaris?	53
2.5.4 ¿El preu diari del lloguer del vehicle és diferent en funció del tipus de vehicle?	54
2.5.5 ¿El preu diari del lloguer del vehicle és diferent en funció del tipus de combustible?	54
2.6 Model de regressió lineal múltiple per preveure el preu diari d'un vehicle	55
2.7 Model d'arbre de regressió per preveure el preu diari d'un vehicle	61
2.8 Model d'arbre de classificació per preveure si un vehicle serà llogat	65
2.9 Conclusions	74
3 Bibliografia	75

1 Details de l'activitat

1.1 Presentació

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. Per fer aquesta pràctica haureu de treballar en grups de 2 persones. Haureu de lliurar un sol fitxer amb l'enllaç Github (<https://github.com>) on es trobin les solucions incloent els noms dels components de l'equip. Podeu utilitzar la Wiki de Github per descriure el vostre equip i els diferents arxius que corresponen a la vostra entrega. Cada membre de l'equip haurà de contribuir amb el seu usuari Github. Malgrat que no es tracta del mateix enunciat, els següents exemples d'edicions anteriors us poden servir com a guia:

- Exemple: <https://github.com/Bengis/nba-gap-cleaning>
- Exemple complex (fitxer adjunt).

1.2 Competències

En aquesta pràctica es desenvolupen les següents competències del Màster de Data Science:

- 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.

1.3 Objectius

Els objectius concrets d'aquesta pràctica són:

- 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.

1.4 Descripció de la Pràctica a realitzar

L'objectiu d'aquesta activitat serà el tractament d'un dataset, que pot ser el creat a la pràctica 1 o bé qualsevol dataset lliure disponible a Kaggle (<https://www.kaggle.com>). Alguns exemples de dataset amb els que podeu treballar són:

- Red Wine Quality (<https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009>).
- Titanic: Machine Learning from Disaster (<https://www.kaggle.com/c/titanic>).

L'últim exemple correspon a una competició activa a Kaggle de manera que, opcionalment, podeu aprofitar el treball realitzat durant la pràctica per entrar en aquesta competició.

Seguint les principals etapes d'un projecte analític, les diferents tasques a realitzar (i justificar) són les següents:

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?
2. Integració i selecció de les dades d'interès a analitzar.

3. Neteja de les dades.
 - Les dades contenen zeros o elements buits? Com gestionaries aquests casos?
 - Identificació i tractament de valors extrems.
4. Anàlisi de les dades.
 - Selecció dels grups de dades que es volen analitzar/comparar (planificació delsanàlisis a aplicar).
 - Comprovació de la normalitat i homogeneïtat de la variància.
 - Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents.
5. Representació dels resultats a partir de taules i gràfiques.
6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?
7. 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.

1.5 Recursos

Els següents recursos són d'utilitat per la realització de la pràctica:

- Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Megan Squire (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Wes McKinney (2012). Python for Data Analysis. O'Reiley Media, Inc.
- Tutorial de Github <https://guides.github.com/activities/hello-world>.

1.6 Criteris de valoració

Tots els apartat són obligatoris. La ponderació dels exercicis és la següent:

- Els apartats 1, 2 i 6 valen 0,5 punts.
- Els apartats 3, 5 i 7 valen 2 punts.
- L'apartat 4 val 2,5 punts.

Es valorarà la idoneïtat de les respostes, que han de ser clares i completes. Les diferents etapes han d'estar ben justificades i accompanyades del codi corresponent. També es valorarà la síntesi i claredat, a través de l'ús de comentaris, del codi resultant, així com la qualitat de les dades finals analitzades.

1.7 Format i data de lliurament

Durant la setmana del 21 al 25 de desembre el grup podrà lliurar al professor una entrega parcial opcional. Aquesta entrega parcial és molt recomanable per tal de rebre assessorament sobre la pràctica i verificar que la direcció presa és la correcta. Es lliuraran comentaris als estudiants que hagin efectuat l'entrega parcial però no comptarà per la nota de la pràctica. En l'entrega parcial els estudiants hauran de lliurar per correu electrònic, al professor encarregat de l'aula, l'enllaç al repositori Github amb el que hagin avançat.

Pel que fa a l'entrega final, cal lliurar un únic fitxer que contingui l'enllaç a Github, el qual no es podrà modificar posteriorment a la data d'entrega, on hi hagi:

1. Una Wiki on hi hagi els noms dels components del grup i una descripció dels fitxers.

2. Un document Word, Open Office o PDF amb les respostes a les preguntes i els noms dels components del grup. A més, al final de document, haurà d'aparèixer la següent taula de contribucions al treball, la qual ha de signar cada integrant del grup amb les seves inicials. Les inicials representen la confirmació de que l'integrant ha participat en aquell apartat. Tots els integrants han de participar en cadascun dels apartats, de manera que, idealment, els apartats hauran d'estar signats per tots els integrants.
3. Una carpeta amb el codi generat per analitzar les dades.
4. El fitxer CSV amb les dades originals.
5. El fitxer CSV amb les dades finals analitzades.

Aquest document de l'entrega final de la Pràctica 2 s'ha de lliurar a l'espai de Lliurament i Registre d'AC de l'aula abans de les 23:59 del dia 5 de gener. No s'acceptaran lliuraments fora de termini.

2 Resolució

Contribucions	Firma
Investigació prèvia	RPS, ACV
Redacció de les respostes	RPS, ACV
Desenvolupament codi	RPS, ACV

```
# Carreguem les llibreries que farem servir
library(tidyverse)
library(ggplot2)
library(dplyr)
library(GGally)
library(arules)
library(reshape2)
library(regclass)
library(car)
library(nortest)
library(gridExtra)
library(ResourceSelection)
library(DescTools)
library(vcd)
library(ggpubr)
library(corrplot)
library(rpart)
library(rpart.plot)
library(gmodels)
library(rattle)
library(Metrics)
library(caret)
library(DMwR)
library(ROSE)
library(grid)
library(gridExtra)

old <- theme_set(theme_minimal())
```

2.1 Descripció del dataset

El conjunt de dades que analitzarem serà: Cornell Car Rental Dataset, s'ha obtingut del repositori de dades Kaggle (<https://www.kaggle.com/kushleshkumar/cornell-car-rental-dataset?select=CarRentalData.csv>). Aquest joc de dades és un recull de registres de diferents portals de lloguer de vehicles a les principals ciutats d'Estats Units, i s'ha generat mitjançant *webscraping*, amb una extracció relitzada al Julol del 2020. Està format per 16 característiques (columnes o atributs) que presentan 5851 registres (files o observacions). Cada una de les obseravacions es correspon a les característiques d'un vehicle de lloguer.

Les característiques, atributs o variables per cada observació (es manté el nom original en anglès) són:

- *fuelType*: Tipus de combustible utilitzat pel vehicle.
- *rating*: Qualificació acumulativa del cotxe per part dels clients
- *renterTripsTaken*: Nombre de viatges realitzats per aquest vehicle (durada desconeguda)
- *reviewCount*: Nombre de valoracions
- *location.city*: Ciutat en què es troba el vehicle
- *location.country*: País en què es troba el vehicle

- *location.latitude*: Coordenada geogràfica (latitud) en què es troba el vehicle
- *location.longitude*: Coordenada geogràfica (longitud) en què es troba el vehicle
- *location.state*: Estat en què es troba el vehicle
- *owner.id*: Identificador (ID) del propietari del vehicle
- *rate.daily*: Tarifa diària en dòlars
- *vehicle.make*: Marca del vehicle
- *vehicle.model*: Model del vehicle
- *vehicle.type*: Tipus de vehicle
- *vehicle.year*: Any del vehicle (matriculació)
- *airportcity*: Ciutat de l'aeroport més proper a la localització en què es troba el vehicle (normalment es des d'on es realitza el lloguer del vehicle).

2.2 Importància i objectius de l'anàlisi

Amb l'entrada en vigor de les zones de zero emissions a les grans ciutats, la classificació de vehicles segons el seu grau de contaminació i de l'objectiu de zero emissions al 2050, promogut per la Comissió Europea han generat un gran debat envers la compra o lloguer de vehicles. De fet hi ha diversos estudis que mostren que en els últims anys hi ha hagut un increment en el mercat de lloguer de vehicles o *renting*. Per tant a partir del joc de dades escollit “Cornell Car Rental Dataset” ens proposem:

- Realitzar una anàlisi detallada dels atributs propis del sector del lloguer de vehicles per tal d'extreuren nou coneixement i que aquest pugui aportar valor.
- Dur a terme diferents contrastos d'hipòtesis que permetin identificar propietats interessants subjacents en les mostres que puguin ser inferides respecte a la població.
- Generar un model que permeti preveure el preu per dia d'un automòbil de lloguer.
- Generar un model que permeti preveure, donades les característiques d'un vehicle de lloguer, si aquest serà llogat o no.
- Generar un model que pugui preveure els ingressos anuals que aportarà un automòbil de lloguer com a aproximació als beneficis que aportaria.

Aquests anàlisis són de gran rellevància en gairebé qualsevol sector relacionat amb el lloguer de vehicles. Per una banda pot resultar interessant per les agències de lloguer, ja que aquestes podran conèixer amb més detall els seus clients, i per tant generar ofertes personalitzades o decidir quin és el preu més adient o quin model de vehicle és més adequat per renovar la flota. D'altra banda també pot resultar interessant per a l'usuari final, ja que a partir de les seves necessitats podrà determinar si li convé llogar un vehicle o no o si el seu preu és corresponent amb el del mercat actual.

2.3 Neteja de dades

2.3.1 Lectura de dades

En primer lloc, obrim el fitxer de dades i examinem el tipus de dades amb els que R ha interpretat cada variable. A més examinem també els valors resum de cada tipus de variable.

Comencem carregant el joc de dades en un dataframe d'R.

```
# Carreguem el joc de dades
ds <- read.csv("../data/CarRentalDataV1.csv", stringsAsFactors = FALSE,
              header = TRUE, sep = ",", strip.white = TRUE)
```

Tot seguit examinem l'estructura del joc de dades, per validar que s'han interpretat correctament.

```
# Verifiquem les dimensions del joc de dades
dim(ds)
```

```
## [1] 5851 16
```

```
# Verifiquem l'estructura del joc de dades
str(ds)

## 'data.frame':      5851 obs. of  16 variables:
##   $ fuelType        : chr  "ELECTRIC" "ELECTRIC" "HYBRID" "GASOLINE" ...
##   $ rating          : num  5 5 4.92 5 5 5 4.42 4.9 5 4.76 ...
##   $ renterTripsTaken : num  13 2 28 21 3 13 13 12 1 22 ...
##   $ reviewCount     : num  12 1 24 20 1 12 12 10 1 17 ...
##   $ location.city    : chr  "Seattle" "Tijeras" "Albuquerque" "Albuquerque" ...
##   $ location.country : chr  "US" "US" "US" "US" ...
##   $ location.latitude: num  47.4 35.1 35.1 35.1 35.2 ...
##   $ location.longitude: num  -122 -106 -107 -107 -107 ...
##   $ location.state   : chr  "WA" "NM" "NM" "NM" ...
##   $ owner.id         : num  12847615 15621242 10199256 9365496 3553565 ...
##   $ rate.daily       : num  135 190 35 75 47 58 42 117 102 49 ...
##   $ vehicle.make     : chr  "Tesla" "Tesla" "Toyota" "Ford" ...
##   $ vehicle.model    : chr  "Model X" "Model X" "Prius" "Mustang" ...
##   $ vehicle.type     : chr  "suv" "suv" "car" "car" ...
##   $ vehicle.year     : num  2019 2018 2012 2018 2010 ...
##   $ airportcity      : chr  "Albuquerque" "Albuquerque" "Albuquerque" "Albuquerque" ...
```

A partir dels detalls anteriors podem veure que disposem d'un joc de dades amb 5851 observacions, amb 16 atributs o variables per observació i que el tipus de dades s'ha interpretat correctament.

2.3.2 Previsualització de les dades d'interès

A continuació realitzem una previsualització, anàlisi visual de les dades, per intentar identificar les possibles anomalies, distribucions i característiques de les diferents variables.

Per això el primer que farem serà distingir les variables categòriques de les variables numèriques. la qual cosa ens facilitarà el tractament futur.

```
categorical_features_names = c("fuelType", "location.city", "location.country",
  "location.state", "owner.id", "vehicle.make", "vehicle.model",
  "vehicle.type", "airportcity")

numeric_features_names = c("rating", "renterTripsTaken", "reviewCount",
  "location.latitude", "location.longitude", "rate.daily",
  "vehicle.year")
```

Anàlisi univariant

Comencem visualitzant els principals descriptors estadístics de cadascuna de les variables numèriques.

```
# Mostrem un resum dels principals estadístics de cada
# variable
summary(ds[, numeric_features_names])
```

```
##      rating    renterTripsTaken  reviewCount    location.latitude
##  Min.   :1.00   Min.   : 0.00   Min.   : 0.00   Min.   :21.27
##  1st Qu.:4.90   1st Qu.: 5.00   1st Qu.: 4.00   1st Qu.:30.45
##  Median :5.00   Median :18.00   Median :16.00   Median :35.55
##  Mean   :4.92   Mean   :33.48   Mean   :28.45   Mean   :35.58
##  3rd Qu.:5.00   3rd Qu.:46.00   3rd Qu.:39.00   3rd Qu.:40.00
##  Max.   :5.00   Max.   :395.00  Max.   :321.00  Max.   :64.89
##  NA's    :501
##      location.longitude    rate.daily      vehicle.year
```

```

##  Min.   : -158.17   Min.   : 20.00   Min.   :1955
##  1st Qu.: -117.16  1st Qu.: 45.00   1st Qu.:2014
##  Median : -95.67   Median : 69.00   Median :2016
##  Mean   : -99.63   Mean   : 93.69   Mean   :2015
##  3rd Qu.: -81.54   3rd Qu.:110.00   3rd Qu.:2018
##  Max.   : -68.82   Max.   :1500.00  Max.   :2020
##

```

2.3.2.1 Anàlisi dels atributs numèrics

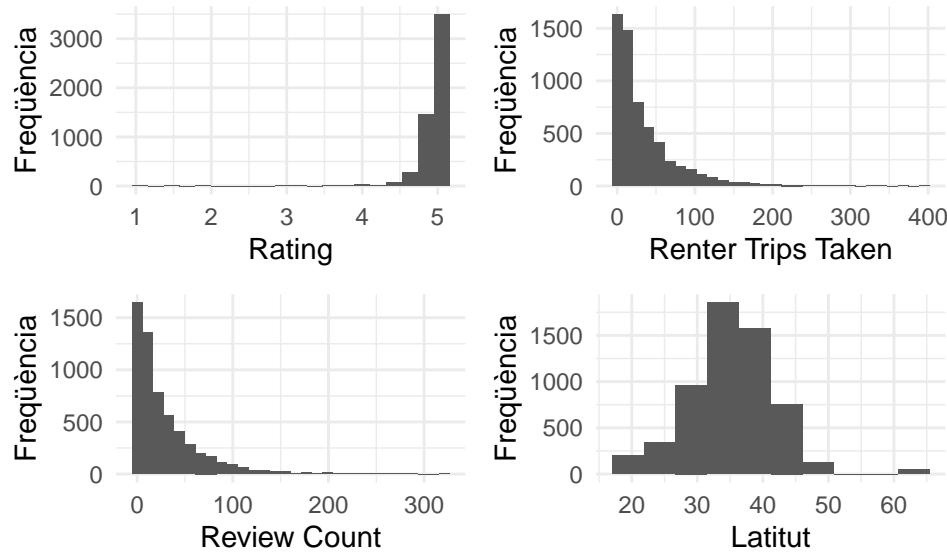
Tot seguit visualitzem les distribucions dels valors de les variables numèriques.

```

# Calculem histogrames de les variables numèriques
hist01 <- ggplot(data = ds, aes(x = rating)) + geom_histogram(bins = 20) +
  xlab("Rating") + ylab("Freqüència")
hist02 <- ggplot(data = ds, aes(x = renterTripsTaken)) + geom_histogram(bins = 30) +
  xlab("Renter Trips Taken") + ylab("Freqüència")
hist03 <- ggplot(data = ds, aes(x = reviewCount)) + geom_histogram(bins = 30) +
  xlab("Review Count") + ylab("Freqüència")
hist04 <- ggplot(data = ds, aes(x = location.latitude)) + geom_histogram(bins = 10) +
  xlab("Latitud") + ylab("Freqüència")
hist05 <- ggplot(data = ds, aes(x = location.longitude)) + geom_histogram(bins = 10) +
  xlab("Longitud") + ylab("Freqüència")
hist06 <- ggplot(data = ds, aes(x = rate.daily)) + geom_histogram(bins = 30) +
  xlab("Daily rate") + ylab("Freqüència")
hist07 <- ggplot(data = ds, aes(x = vehicle.year)) + geom_histogram(bins = 13) +
  xlab("Vehicle Year") + ylab("Freqüència")

grid.arrange(hist01, hist02, hist03, hist04, nrow = 2, ncol = 2)

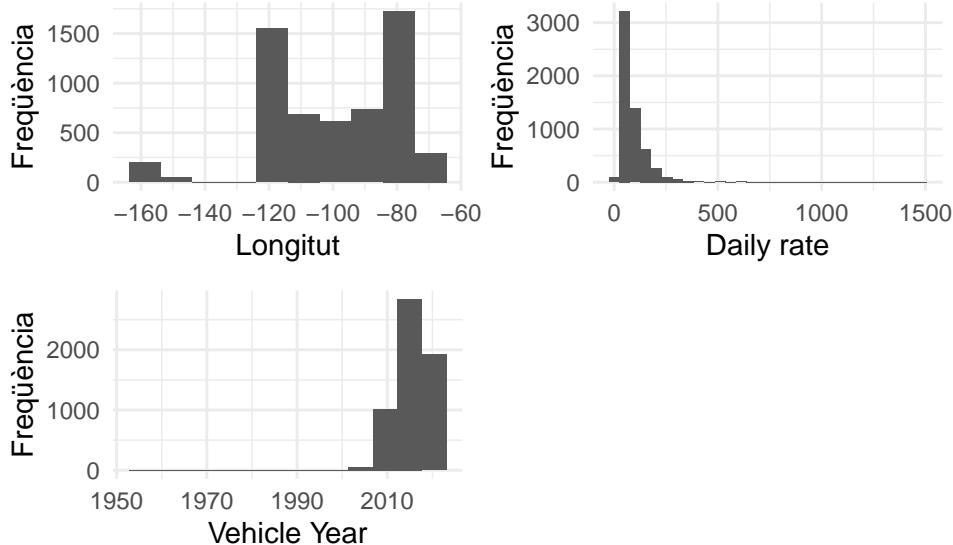
```



```

grid.arrange(hist05, hist06, hist07, nrow = 2, ncol = 2)

```



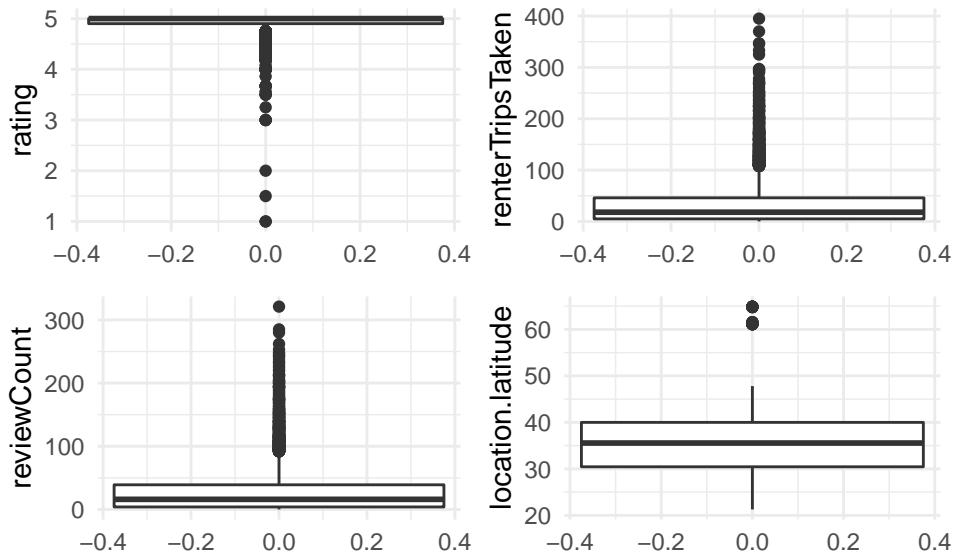
A partir de les visualitzacions anteriors podem fer les següents apreciacions:

- **rating:** Histograma unimodal amb cua a la dreta on podem observar que la gran majoria de vehicles reben valoracions elevades, en concret la puntuació màxima (5). A més s'observa que la presència de vehicles amb baixa puntuació és insignificant.
- **renterTripsTaken:** Histograma unimodal amb cua a l'esquerra, la qual cosa ens indica que els vehicles se solen llogar poques vegades, per sota de 100 vegades. I amb una mitjana de 33.48 lloguers.
- **reviewCount:** Histograma unimodal amb cua a l'esquerra, la qual cosa ens indica que els vehicles solen rebre poques valoracions, per sota de 100 vegades. I amb una mitjana de 28.45 valoracions.
- **location.latitude:** Histograma amb un distribució bastant normal. La qual cosa ens fa pensar que gran part dels vehicles es localitzen en una mateixa latitud, coordenada geogràfica (latitud) en què es troba el vehicle.
- **location.longitude:** Histograma bimodal, amb un pic prop de -120 i un altre pic prop de -80, de nou podem dir que la zona on s'estudia el lloguer de vehicles també queda bastant acotada horitzontalment, coordenada geogràfica (longitud) en què es troba el vehicle.
- **rate.daily:** Histograma unimodal amb cua a l'esquerra, la qual cosa ens indica que els vehicles se solen llogar per menys de 500 dòlars/dia. Amb un lloguer mitjà de 93.69 dòlars/dia, un lloguer mínim de 20 dòlars/dia i un lloguer màxim de 1500 dòlars/dia.
- **vehicle.year:** Histograma unimodal amb cua a l'esquerra, la qual cosa ens indica que els vehicles de lloguer no solen ser gaire antics. Sent la mitja de 5 anys, el vehicle més antic del 1955 i el vehicle més nou del 2020.

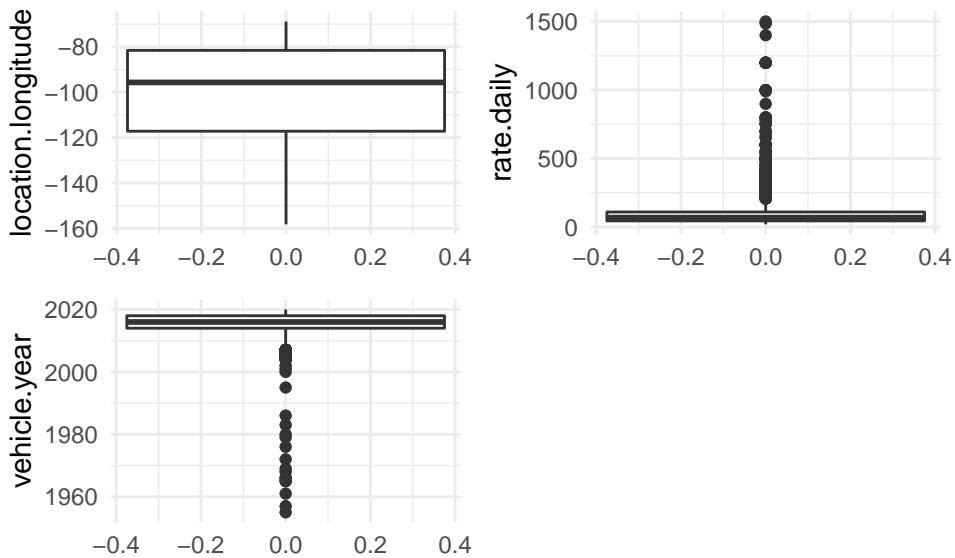
Un altre de les visualitzacions que sol ser de força utilitat per analitzar els jocs de dades són els diagrames de caixes, amb aquests podem observar la dispersió de les dades. A continuació visualitzarem els diagrames de caixes dels atributs numèrics.

```
# Calculem diagrames de caixa de les variables numèriques
box01 <- ggplot(ds, aes(y = rating)) + geom_boxplot()
box02 <- ggplot(ds, aes(y = renterTripsTaken)) + geom_boxplot()
box03 <- ggplot(ds, aes(y = reviewCount)) + geom_boxplot()
box04 <- ggplot(ds, aes(y = location.latitude)) + geom_boxplot()
box05 <- ggplot(ds, aes(y = location.longitude)) + geom_boxplot()
box06 <- ggplot(ds, aes(y = rate.daily)) + geom_boxplot()
box07 <- ggplot(ds, aes(y = vehicle.year)) + geom_boxplot()

grid.arrange(box01, box02, box03, box04, nrow = 2, ncol = 2)
```



```
grid.arrange(box05, box06, box07, nrow = 2, ncol = 2)
```



A continuació podem veure un detall dels possibles valors extrems.

```
n <- length(ds$rating)
n = boxplot.stats(ds$rating)$n
## [1] 501
n = boxplot.stats(ds$renterTripsTaken)$n
## [1] 0
n = boxplot.stats(ds$reviewCount)$n
## [1] 0
```

```
n = boxplot.stats(ds$location.latitude)$n  
## [1] 0  
n = boxplot.stats(ds$location.longitude)$n  
## [1] 0  
n = boxplot.stats(ds$rate.daily)$n  
## [1] 0  
n = boxplot.stats(ds$vehicle.year)$n  
## [1] 0
```

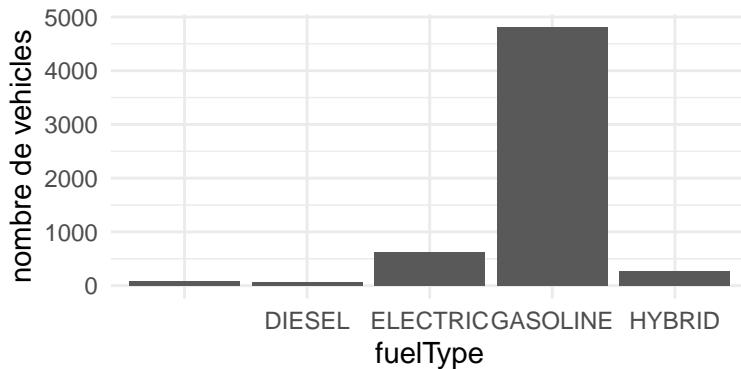
A partir del resultats anteriors podem dir que la única variable que presenta valors extrems és les valoracions 'rating'. En concret tenim 501 observacions que s'alluyen molt dels valors esperats per aquesta variable.

2.3.2.2 Anàlisi dels atributs categòrics

Continuem visualitzant les distribucions dels valors de les variables categòriques

Revisem la variable fuelType

```
ggplot(data = ds, aes(x = fuelType)) + geom_bar() + ylab("nombre de vehicles")
```



Podem observar que per la majoria de vehicles el combustible és **GASOLINE**, seguits de lluny pels **ELECTRIC** i després pels **HYBRID**. Finalment els **DIESEL** son residuals i queda un subgrup sense etiquetar que el tractarem tot seguit.

Fem un petit tractament de valors Nuls. Aquest el realitzem calculant quin és el valor més freqüent de combustible i substituint els valors nuls per aquest.

```
# Obtenim el valor més freqüent
most_freq_fuelType <- names(which.max(table(ds$fuelType)))
```

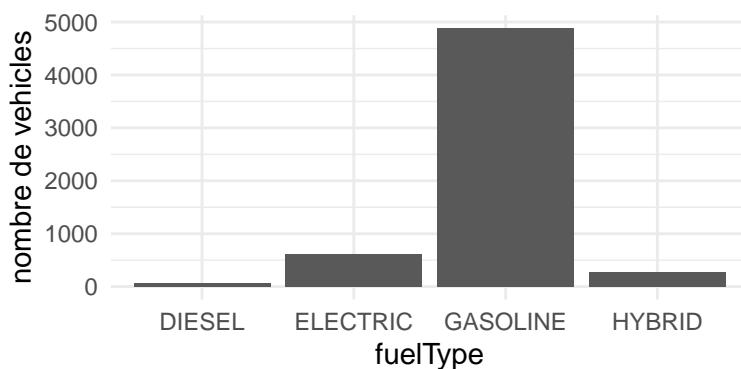
```
most_freq_fuelType
```

```
## [1] "GASOLINE"
```

```
# Prenem el valor més freqüent per als valors buits
ds$fuelType[ds$fuelType == ""] = most_freq_fuelType
```

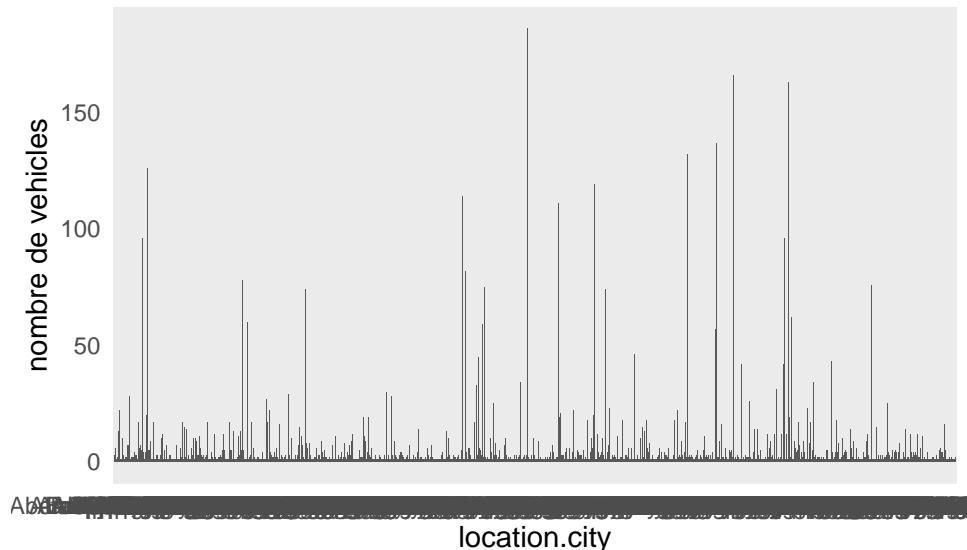
Grafiquem de nou fuelType

```
ggplot(data = ds, aes(x = fuelType)) + geom_bar() + ylab("nombre de vehicles")
```



Revisem la variable location.city

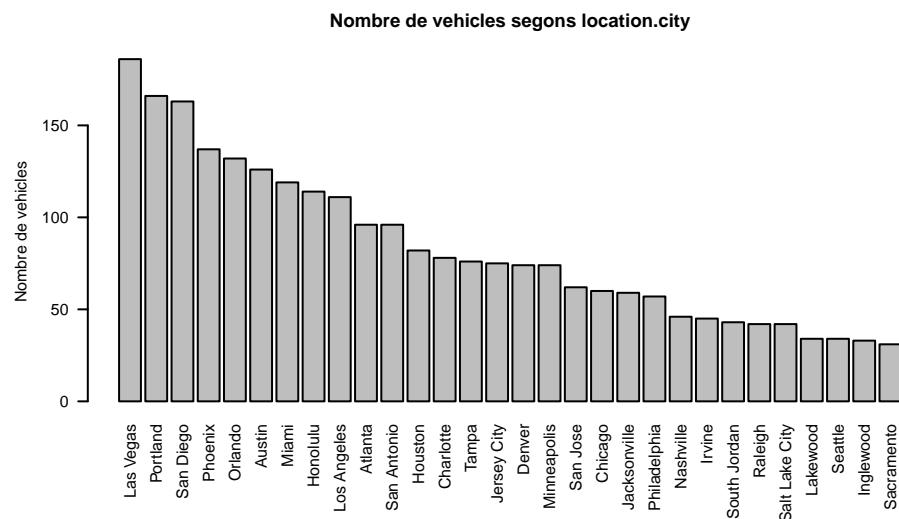
```
ggplot(data = ds, aes(x = location.city)) + geom_bar() + ylab("nombre de vehicles")
```



Veiem que hi ha tantes categories que és impossible revisar acuradament la gràfica. Llavors optem per graficar només les 30 categories més comuns.

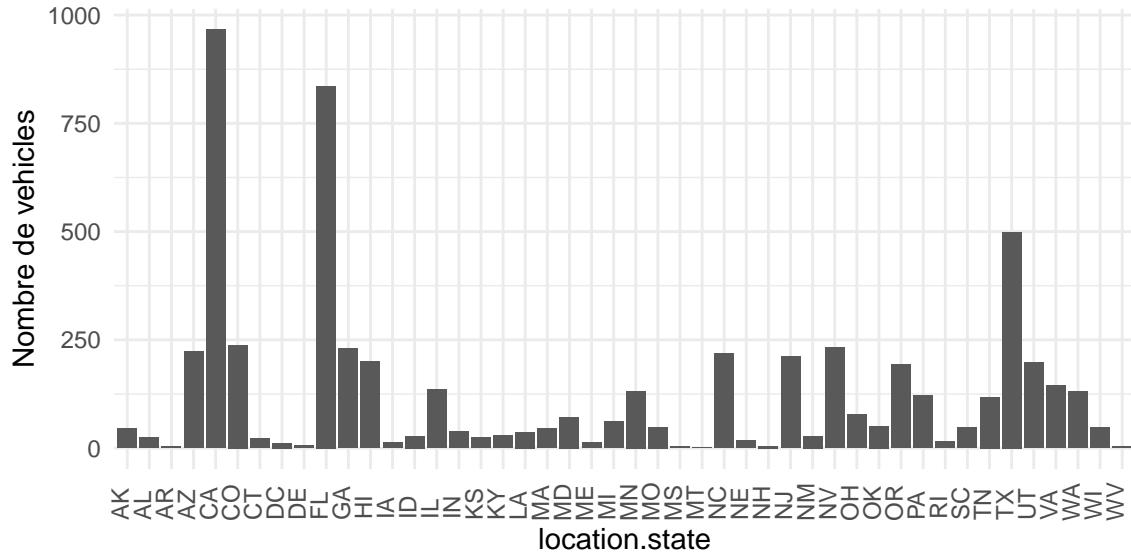
```
# Seleccionem les dades
location.city_ordered <- sort(table(ds$location.city), decreasing = TRUE)

# Creem la visualització
par(las = 2, cex = 0.5, mar = c(8, 4, 4, 2))
barplot(location.city_ordered[0:30], main = "Nombre de vehicles segons location.city",
       ylab = "Nombre de vehicles")
```



Revisem la variable location.state

```
ggplot(data = ds, aes(x = location.state)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
    vjust = 0.1, hjust = 0.1)) + ylab("Nombre de vehicles")
```



Veiem que entre les categories destaquen **CA**, **FL** i **TX**.

Ens proposem afegir un atribut continu, al qual mitjançant la població dels estats, es pugui obviar l'abreviatura categorica dels estats. Ens descarreguem les dades del cens dels estats unit des de: <https://www.census.gov/data/datasets/time-series/demo/popest/2010s-state-total.html> i simplifiquem la taula descarregada en un csv que carreguem a continuació.

```
# Carreguem la taula de població per estat
sp <- read.csv("../data/state-population.csv", stringsAsFactors = FALSE,
    header = TRUE, sep = ",", strip.white = TRUE)

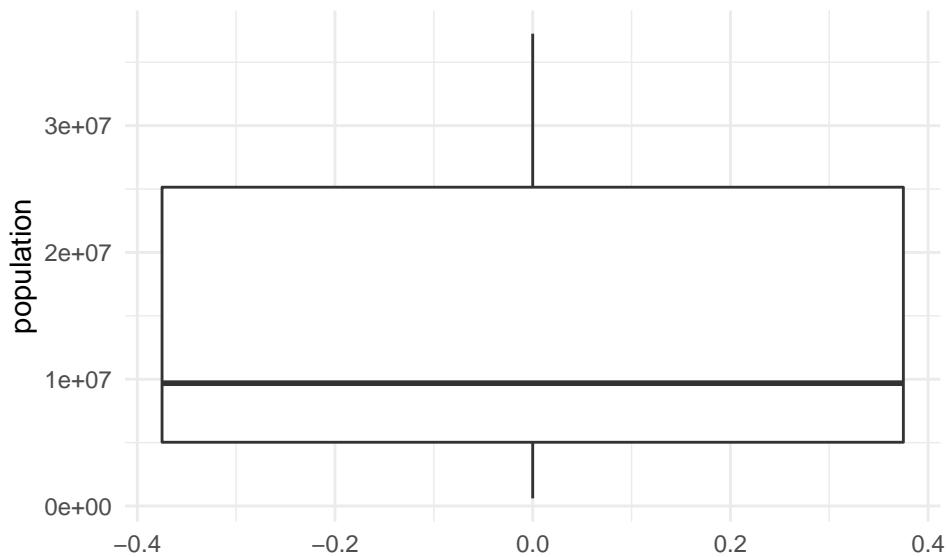
# Creem una nova variable on hi introduim la població de
# l'estat on es troba el vehicle
ds$population <- ds$location.state

# Assignem la població segons l'estat en el que es troba
for (i in sp$state) {
    ds$population[ds$population == i] = sp$Census[sp$state ==
        i]
}

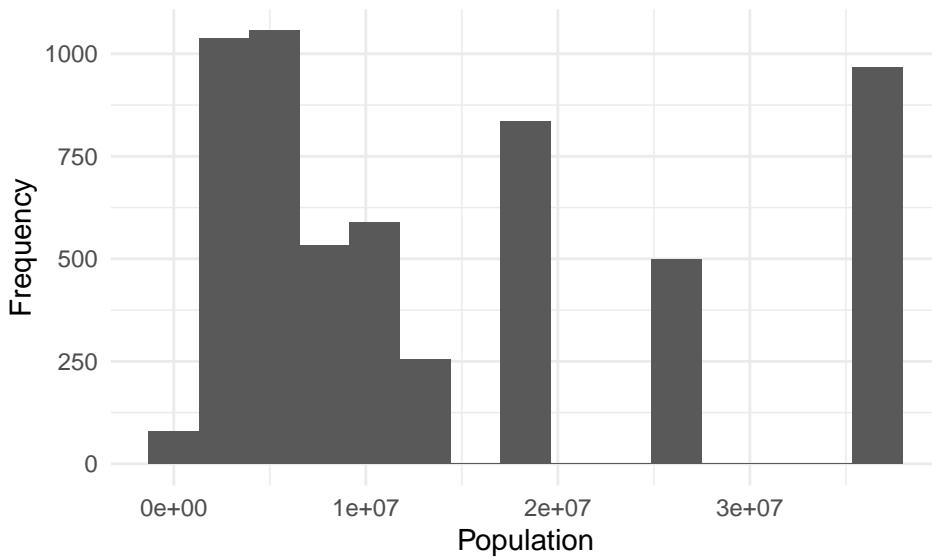
# convertim la nova variable en numèrica
ds$population <- as.numeric(ds$population)
```

Podemaprofitar per a calcular els diagrames d'aquesta nova variable numèrica.

```
# Calculem diagrames de caixa i de punts de la nova variable
# numèrica
ggplot(ds, aes(y = population)) + geom_boxplot()
```

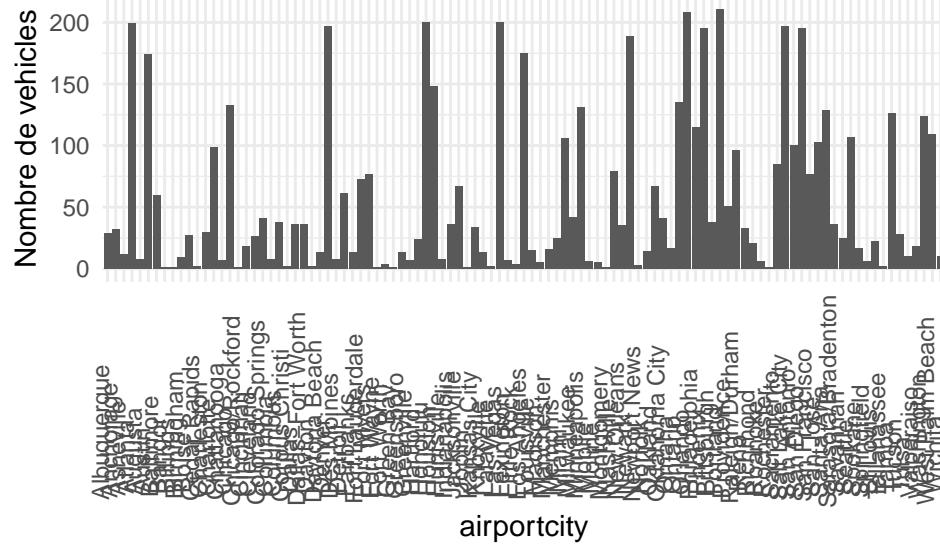


```
ggplot(data = ds, aes(x = population)) + geom_histogram(bins = 15) +  
  xlab("Population") + ylab("Frequency")
```



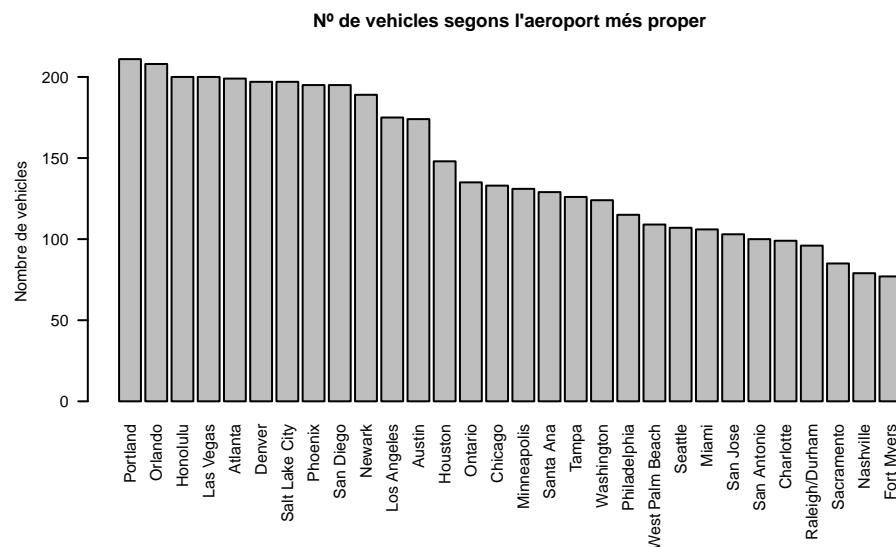
Revisem la variable airportcity

```
ggplot(data = ds, aes(x = airportcity)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.1, hjust = 0.1)) + ylab("Nombre de vehicles")
```



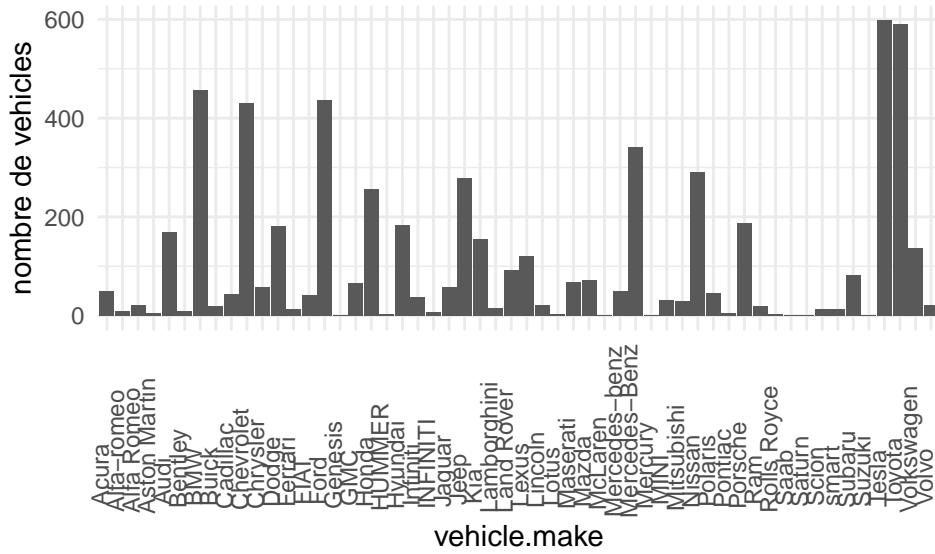
Tornem a trobar-nos que hi ha massa categories per a fer una visualització cómode. Llavors optem per graficar només les 30 categories més comuns.

```
# Seleccionem les dades
airportcity_ordered <- sort(table(ds$airportcity), decreasing = TRUE)
# Creem la visualització
par(las = 2, cex = 0.5, mar = c(8, 4, 4, 2))
barplot(airportcity_ordered[0:30], main = "Nº de vehicles segons l'aeroport més proper",
       ylab = "Nombre de vehicles")
```



Revisem la variable vehicle.make

```
ggplot(data = ds, aes(x = vehicle.make)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.1, hjust = 0.1)) + ylab("nombre de vehicles")
```



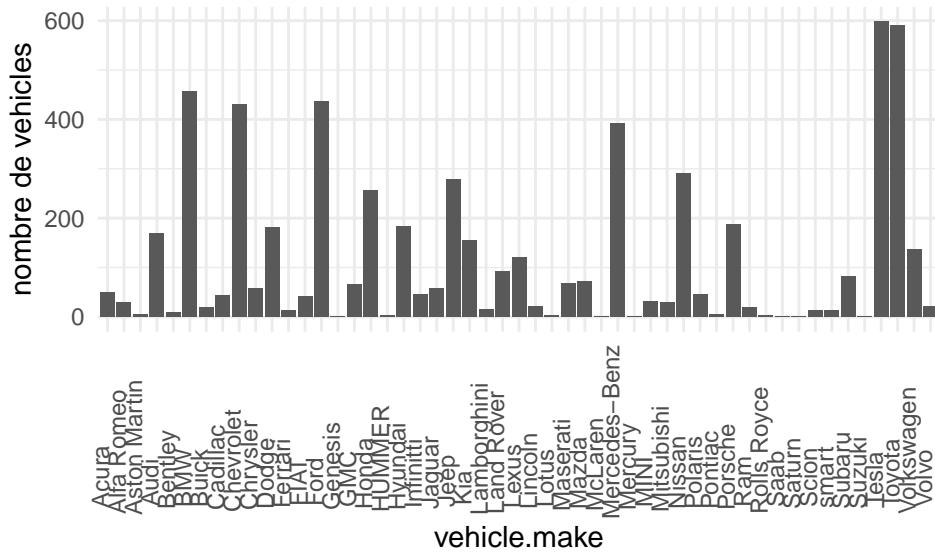
Aquí ens trobem al límit del que seria una visualització raonable, i veiem que hi ha categories repetides ja que es troben en algun cas mal escrites (per falta de lletres majúscules principalment).

Per tant realitzem una primera correcció de les categories per a continuació reproduir de nou la visualització.

```
# Correcció de les marques de vehicles, diferències
# tipogràfiques
ds$vehicle.make[ds$vehicle.make == "Alfa-romeo"] = "Alfa Romeo"
ds$vehicle.make[ds$vehicle.make == "Mercedes-benz"] = "Mercedes-Benz"
ds$vehicle.make[ds$vehicle.make == "Infiniti"] = "Infinitti"
ds$vehicle.make[ds$vehicle.make == "INFINITI"] = "Infinitti"

ds$vehicle.make <- droplevels.factor(ds$vehicle.make)

ggplot(data = ds, aes(x = vehicle.make)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.1, hjust = 0.1)) + ylab("nombre de vehicles")
```



A la vista de les dades es pot observar que *TESLA* i *TOYOTA* son les marques més freqüents, seguides de *BMW*, *CHEVROLET*, *FORD* i *Mercedes-Benz*.

Ens trobem que igualment hi ha un nombre molt elevat de categories per aquesta variable (51 concretament) quan de fet, tot el joc de dades amb prou feines arriba a les 6000 observacions. Llavors ens plantegem crear una nova variable **rang** agrupant les marques en 3 grups segons la gamma (“average,” “high” o “luxury”):

- average: ‘FIAT,’ ‘Ford,’ ‘Honda,’ ‘Hyundai,’ ‘Jeep,’ ‘Kia,’ ‘Land Rover,’ ‘Mazda,’ ‘Mitsubishi,’ ‘Nissan,’ ‘Polaris,’ ‘Pontiac,’ ‘Saturn,’ ‘Scion,’ ‘Suzuki,’ ‘Toyota,’ ‘Volkswagen,’ ‘Volvo.’
- high: ‘Alfa Romeo,’ ‘Audi,’ ‘BMW,’ ‘Chevrolet,’ ‘Chrysler,’ ‘Dodge,’ ‘GMC,’ ‘HUMMER,’ ‘Mercedes-Benz,’ ‘Tesla,’ ‘MINI,’ ‘Ram,’ ‘Saab,’ ‘smart,’ ‘Subaru.’
- luxury: ‘Acura,’ ‘Aston Martin,’ ‘Bentley,’ ‘Buick,’ ‘Cadillac,’ ‘Ferrari,’ ‘Genesis,’ ‘Infiniti,’ ‘Jaguar,’ ‘Lamborghini,’ ‘Lexus,’ ‘Lincoln,’ ‘Lotus,’ ‘Maserati,’ ‘McLaren,’ ‘Mercury,’ ‘Porsche,’ ‘Rolls Royce.’

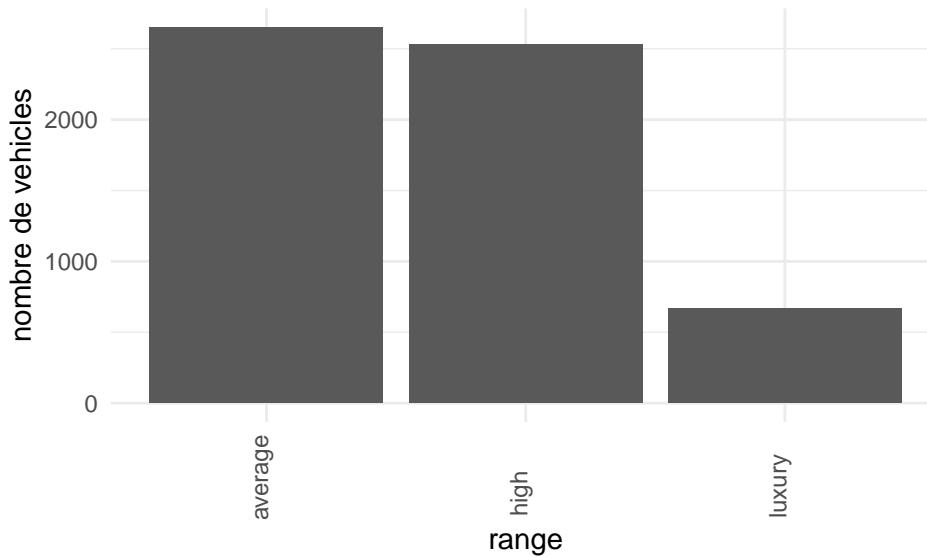
```
# Creem vectors per cada categoria de la nova variables
average = c("FIAT", "Ford", "Honda", "Hyundai", "Jeep", "Kia",
           "Land Rover", "Mazda", "Mitsubishi", "Nissan", "Polaris",
           "Pontiac", "Saturn", "Scion", "Suzuki", "Toyota", "Volkswagen",
           "Volvo")
high = c("Alfa Romeo", "Audi", "BMW", "Chevrolet", "Chrysler",
        "Dodge", "GMC", "HUMMER", "Mercedes-Benz", "Tesla", "MINI",
        "Ram", "Saab", "smart", "Subaru")
luxury = c("Acura", "Aston Martin", "Bentley", "Buick", "Cadillac",
           "Ferrari", "Genesis", "Infiniti", "Jaguar", "Lamborghini",
           "Lexus", "Lincoln", "Lotus", "Maserati", "McLaren", "Mercury",
           "Porsche", "Rolls Royce")

# Creem i assignem la nova variable;
ds$range[is.element(ds$vehicle.make, average)] = "average"
ds$range[is.element(ds$vehicle.make, high)] = "high"
ds$range[is.element(ds$vehicle.make, luxury)] = "luxury"

# Indiquem a R que es tracta de una variable categòrica
ds$range <- droplevels.factor(ds$range)
```

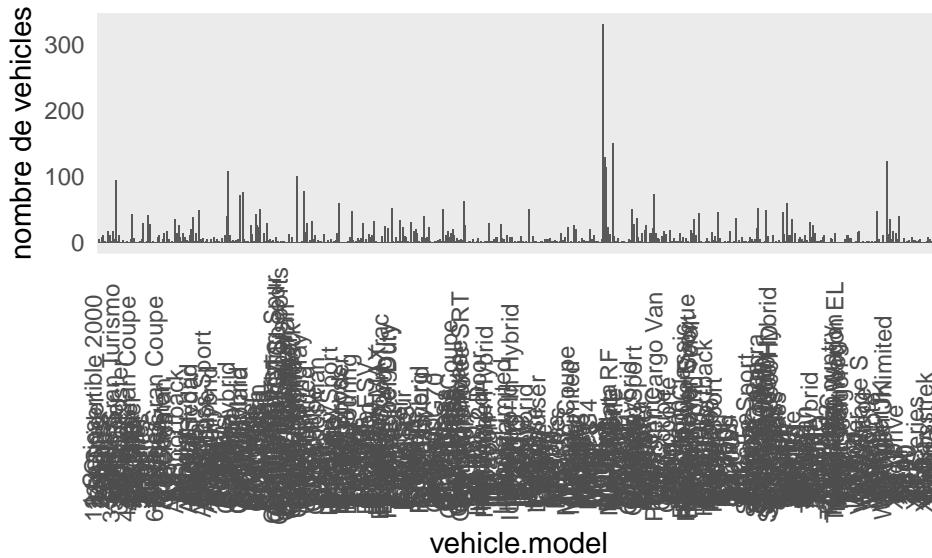
Visualitzem la nova variable categòrica

```
ggplot(data = ds, aes(x = range)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
   vjust = 0.1, hjust = 0.1)) + ylab("nombre de vehicles")
```



Revisem la variable vehicle.model

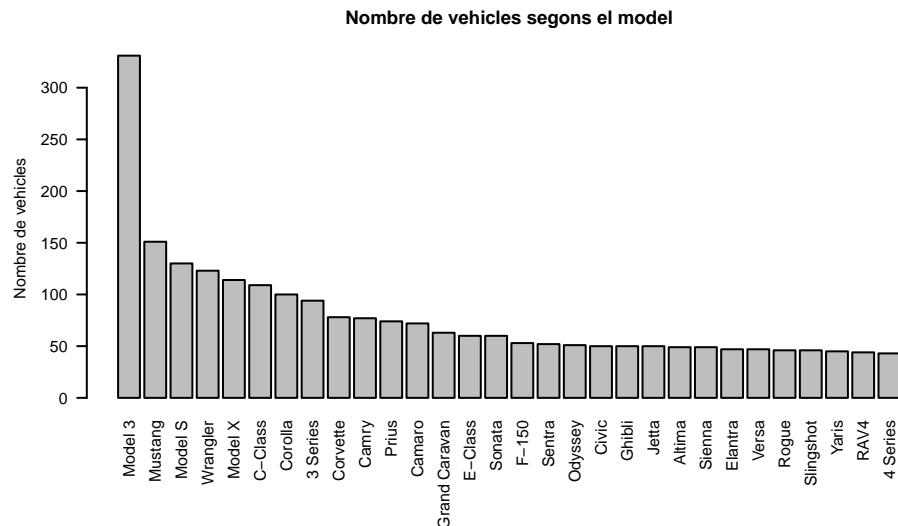
```
ggplot(data = ds, aes(x = vehicle.model)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.1, hjust = 0.1)) + ylab("nombre de vehicles")
```



Tornem a trobar-nos que hi ha massa categories per a fer una visualització cómode. Llavors optem per graficar només les 30 categories més comuns.

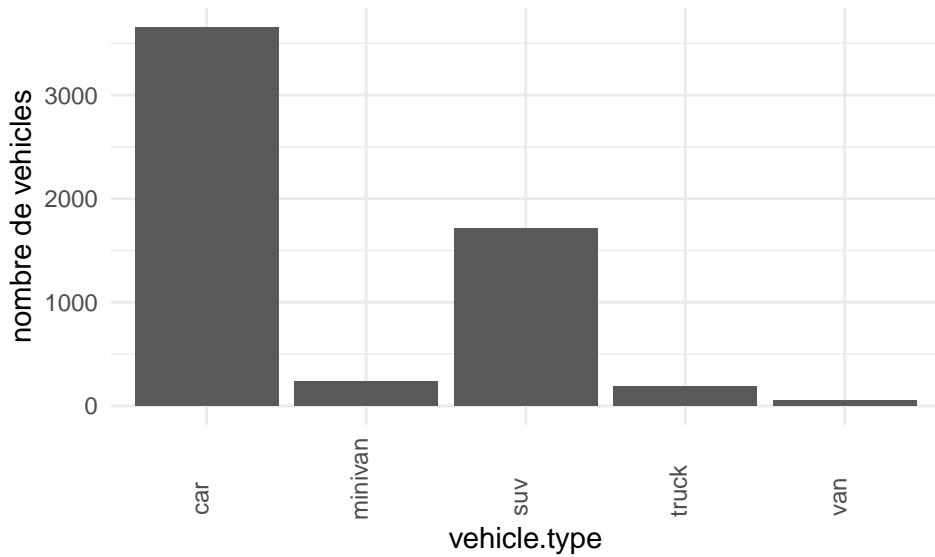
```
# Seleccionem les dades
vehicle.model_ordered <- sort(table(ds$vehicle.model), decreasing = TRUE)

# Creem la visualització
par(las = 2, cex = 0.5, mar = c(8, 4, 4, 2))
barplot(vehicle.model_ordered[0:30], main = "Nombre de vehicles segons el model",
       ylab = "Nombre de vehicles")
```



Revisem la variable `vehicle.type`

```
ggplot(data = ds, aes(x = vehicle.type)) + geom_bar() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.1, hjust = 0.1)) + ylab("nombre de vehicles")
```

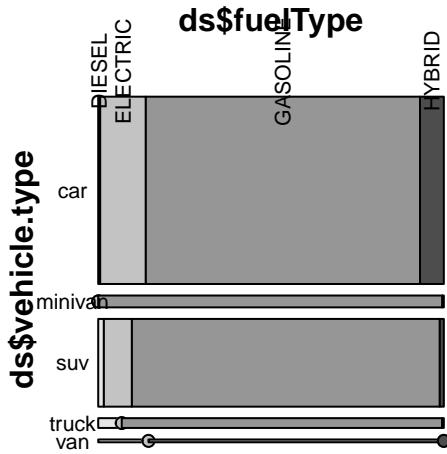


S'observa que la categoria **car** seguida de lluny per **SUV** són les més comuns, sent la resta (**minivan**, **truck** i **van**) gairebé residuals.

Visualització multivariant de les dades categòriques

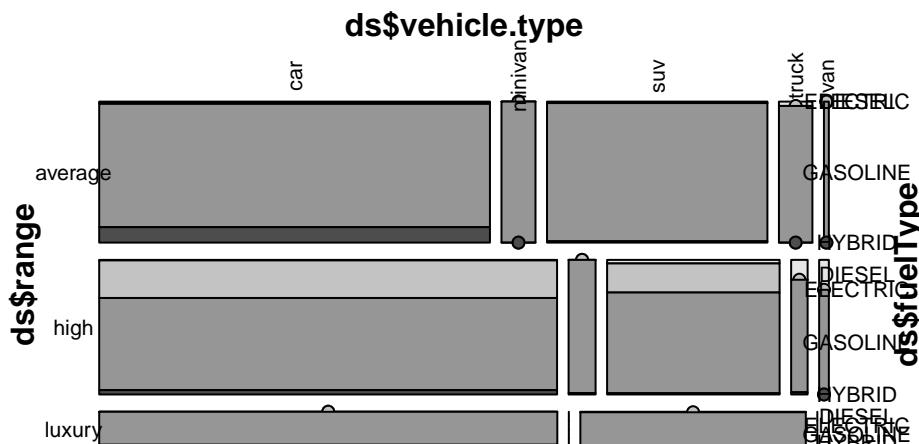
Usem el gràfic de mosaic per visualitzar la combinació entre **fuelType** i **vehicle.type**

```
mosaic(ds$fuelType ~ ds$vehicle.type, gp_labels = gpar(fontsize = 8),
       las = 2, cex.axis = 2, rot_labels = c(90, 90, 0, 0), shade = TRUE)
```



Aprofitem per viisualitzar la combinació de les variables anteriors i **range**

```
mosaic(ds$fuelType ~ ds$range + ds$vehicle.type, gp_labels = gpar(fontsize = 8),
       las = 2, cex.axis = 2, rot_labels = c(90, 0, 0, 0), shade = TRUE)
```

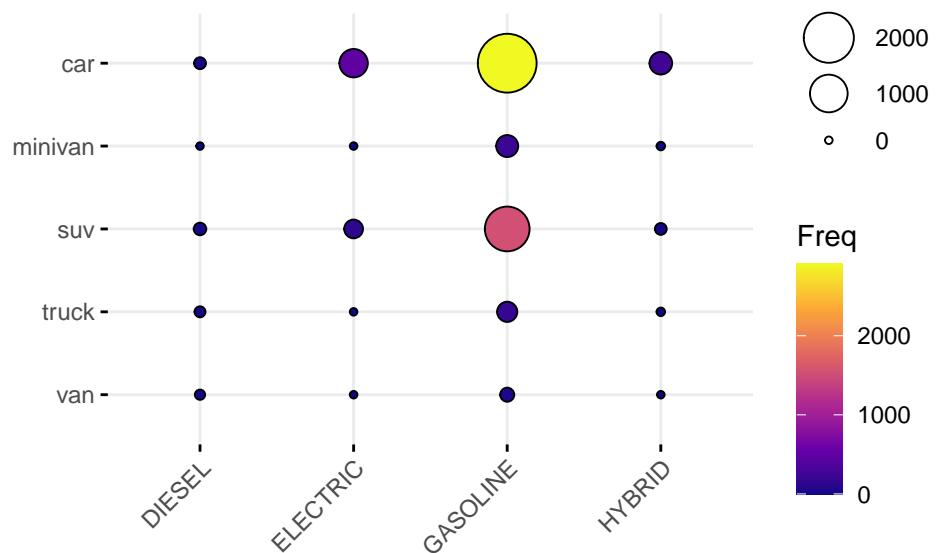


Gràcies al gràfic de mosaic es pot observar que només hi ha la categoria **HYBRID** en vehicles tipus **car** (en la resta és residual). Per altre banda DESTACA que no hi ha vehicles **ELECTRIC** per **van** ni **minivan** les quals son gairebé totalment **GASOLINE**.

Alternativament visualitzem les mateixes dades amb un balloonplot

```
# Preparam les dades
taula = as.data.frame(table(ds$fuelType, ds$vehicle.type))
```

```
# Visualitza el Ballonplot  
ggballoonplot(taula, fill = "value") + scale_fill_viridis_c(option = "C")
```



2.3.3 Zeros i atributs buits

Ara farem el tractament dels valors buits i convertirem les variables discretes a factors.

```
# Estadístiques de valors buits, validem si hi ha valors
# buits
colSums(is.na(ds))

##          fuelType            rating   renterTripsTaken      reviewCount
##                0                  501                   0                      0
## location.city location.country location.latitude location.longitude
##                0                      0                   0                      0
## location.state       owner.id      rate.daily     vehicle.make
##                0                      0                   0                      0
## vehicle.model    vehicle.type   vehicle.year    airportcity
##                0                      0                   0                      0
##      population        range
##                0                      0

# Estadístiques de valors buits, validem si hi ha valors
# buits
colSums(ds == "")

##          fuelType            rating   renterTripsTaken      reviewCount
##                0                  NA                   0                      0
## location.city location.country location.latitude location.longitude
##                0                      0                   0                      0
## location.state       owner.id      rate.daily     vehicle.make
##                0                      0                   0                      0
## vehicle.model    vehicle.type   vehicle.year    airportcity
##                0                      0                   0                      0
##      population        range
##                0                      0
```

Es pot veure que queda 1 variable amb valors buits (rating). Anem a calcular doncs quin és el seu valor més freqüent

```
most_freq_rating <- names(which.max(table(ds$rating)))

most_freq_rating

## [1] "5"
```

Assignem el valor més freqüent als valors buits i comprovem que després del tractament no hi hagi valors buits

```
# Prenem el valor més freqüent per als valors buits
ds$rating[is.na(ds$rating)] = as.numeric(most_freq_rating)

# Visualitzem de nou si hi ha valors buits
colSums(is.na(ds))

##          fuelType            rating   renterTripsTaken      reviewCount
##                0                  0                   0                      0
## location.city location.country location.latitude location.longitude
##                0                      0                   0                      0
## location.state       owner.id      rate.daily     vehicle.make
##                0                      0                   0                      0
## vehicle.model    vehicle.type   vehicle.year    airportcity
```

```

##          0          0          0          0
##      population      range
##          0          0

colSums(ds == "")

```

```

##      fuelType      rating renterTripsTaken      reviewCount
##          0          0          0          0
##  location.city location.country location.latitude location.longitude
##          0          0          0          0
##  location.state      owner.id      rate.daily      vehicle.make
##          0          0          0          0
##  vehicle.model      vehicle.type      vehicle.year      airportcity
##          0          0          0          0
##      population      range
##          0          0

```

2.3.4 Valors extrems (Outliers)

Després del tractament dels valors buits examinem de nou els possibles valors extrems.

```

n <- length(ds$rating)

n - boxplot.stats(ds$rating)$n

```

```

## [1] 0

```

Podem veure com el joc de dades ja no presenta valors extrems. I que els valors extrems de la variable ‘rating’ es corresponen amb dades buides.

2.3.5 Discretització de variables

Ara examinarem per quines variables tindria sentit realitzar una discretització.

```

# Per a quines variables tindria sentit un procés de
# discretització?
apply(ds, 2, function(x) length(unique(x)))

```

```

##      fuelType      rating renterTripsTaken      reviewCount
##          4          80          238          203
##  location.city location.country location.latitude location.longitude
##         964          1          5725          5716
##  location.state      owner.id      rate.daily      vehicle.make
##          46          3093          294          51
##  vehicle.model      vehicle.type      vehicle.year      airportcity
##         526          5          34          103
##      population      range
##          46          3

```

Per aquelles variables amb pocs valors possibles podem realitzar una discretització. Per això convertim les variables discretes a factors d’R.

```

# Convertim les variables discretes a factors
ds[, categorical_features_names] <- lapply(ds[, categorical_features_names],
    factor)

# Mostrem el resultat
str(ds)

```

```

## 'data.frame': 5851 obs. of 18 variables:
## $ fuelType      : Factor w/ 4 levels "DIESEL","ELECTRIC",...: 2 2 4 3 3 3 3 3 3 ...
## $ rating       : num  5 5 4.92 5 5 5 4.42 4.9 5 4.76 ...
## $ renterTripsTaken : num  13 2 28 21 3 13 13 12 1 22 ...
## $ reviewCount   : num  12 1 24 20 1 12 12 10 1 17 ...
## $ location.city : Factor w/ 964 levels "Aberdeen Township",...: 801 877 7 7 7 7 7 7 7 7 ...
## $ location.country : Factor w/ 1 level "US": 1 1 1 1 1 1 1 1 1 ...
## $ location.latitude : num  47.4 35.1 35.1 35.1 35.2 ...
## $ location.longitude: num  -122 -106 -107 -107 -107 ...
## $ location.state   : Factor w/ 46 levels "AK","AL","AR",...: 44 32 32 32 32 32 32 32 32 ...
## $ owner.id        : Factor w/ 3093 levels "5105","12107",...: 2602 3026 2192 2007 1002 1726 918 20...
## $ rate.daily      : num  135 190 35 75 47 58 42 117 102 49 ...
## $ vehicle.make    : Factor w/ 51 levels "Acura","Alfa Romeo",...: 48 48 49 14 10 32 16 14 14 14 ...
## $ vehicle.model   : Factor w/ 526 levels "1 Series","124 Convertible 2000",...: 318 318 348 322 41...
## $ vehicle.type    : Factor w/ 5 levels "car","minivan",...: 3 3 1 1 1 3 3 3 1 3 ...
## $ vehicle.year    : num  2019 2018 2012 2018 2010 ...
## $ airportcity     : Factor w/ 103 levels "Albuquerque",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ population      : num  6724540 2059179 2059179 2059179 2059179 ...
## $ range           : Factor w/ 3 levels "average","high",...: 2 2 1 1 2 2 2 1 1 1 ...

```

2.3.6 Transformació d'atributs

A continuació realitzarem algunes transformacions sobre alguns atributs, amb la finalitat de generar diferents punts de vista de les dades.

```

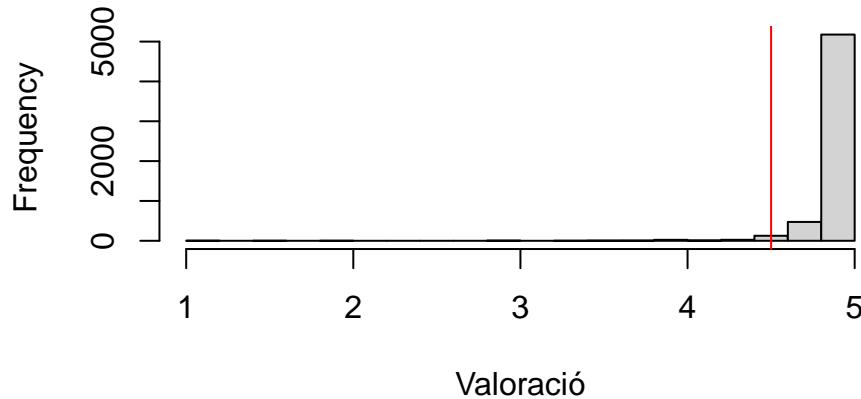
# Discretització amb intervals prefixats, de les valoracions
table(discretize(ds$rating, method = "fixed", c(0, 4.5, Inf),
                 labels = c("Bad", "Good")))

##
## Bad Good
## 97 5754

hist(ds$rating, breaks = 20, main = "Discretització amb intervals prefixats",
     xlab = "Valoració")
cuts_rating <- discretize(ds$rating, method = "fixed", c(0, 4.5,
               Inf), onlycuts = TRUE)
abline(v = cuts_rating, col = "red")

```

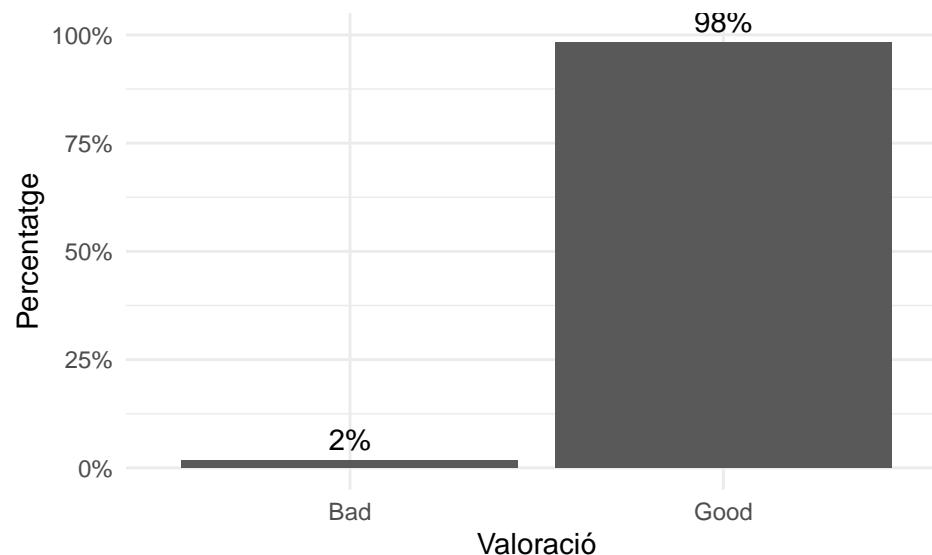
Discretització amb intervals prefixats



```
# Discretització amb intervals prefixats, de les valoracions
ds["rating.discret"] <- discretize(ds$rating, method = "fixed",
  breaks = c(0, 4.5, Inf), labels = c("Bad", "Good"))

ds$rating.discret = as.factor(ds$rating.discret)

# Calculem gràfic de barres
ggplot(data = ds, aes(x = rating.discret)) + geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(y = ((..count..)/sum(..count..)), label = scales::percent(..count..)/sum(..count..)),
    stat = "count", vjust = -0.5) + scale_y_continuous(labels = scales::percent,
    limits = c(0, 1)) + xlab("Valoració") + ylab("Percentatge")
```



La variable **population** per exemple, podria ser tallada amb els 4 quartils que té, per crear 4 categories diferenciant els vehicles ubicats en estats amb més o menys població. Així creem **population.discr**

```
summary(ds$population)
```

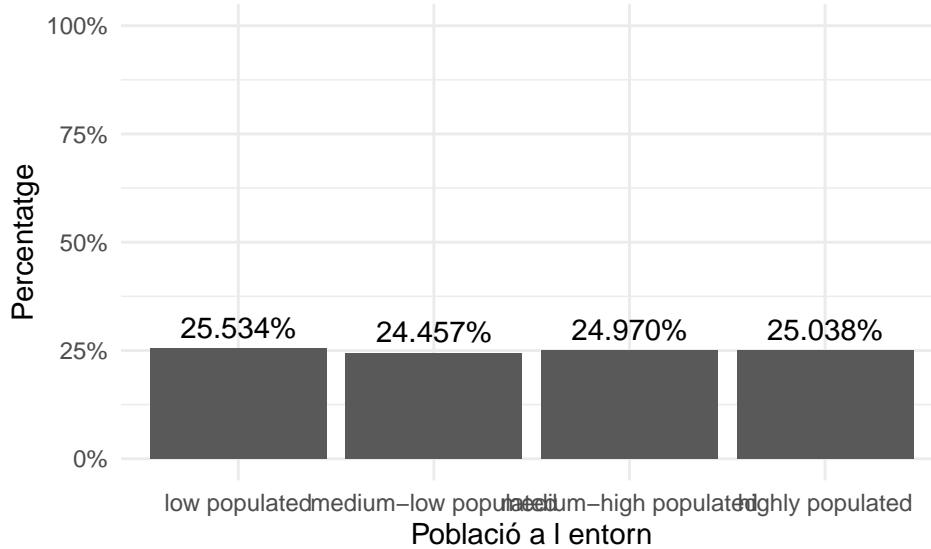
```
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
```

```

##   601723 5029196 9687653 14746411 25145561 37253956
ds["population.discr"] <- discretize(ds$population, method = "fixed",
  c(601722, 5029197, 9687653, 25145561, 37253957), labels = c("low populated",
  "medium-low populated", "medium-high populated", "highly populated"))

# Calculem gràfic de barres
ggplot(data = ds, aes(x = population.discr)) + geom_bar(aes(y = ..count../sum(..count..))) +
  geom_text(aes(y = ((..count..)/sum(..count..))), label = scales::percent(..count../sum(..count..))),
  stat = "count", vjust = -0.5) + scale_y_continuous(labels = scales::percent,
  limits = c(0, 1)) + xlab("Població a l entorn") + ylab("Percentatge")

```



Veiem que les 4 categories tenen pràcticament assignat un tamany igual de mostra.

```
# Factoritzem la nova variable finalment
```

```
ds$population.discr = as.factor(ds$population.discr)
```

2.3.7 Creació de nous indicadors

Un altre dels passos interessant en l'anàlisi d'un joc de dades és la generació de nous atributs a partir dels existents.

Indicador: age

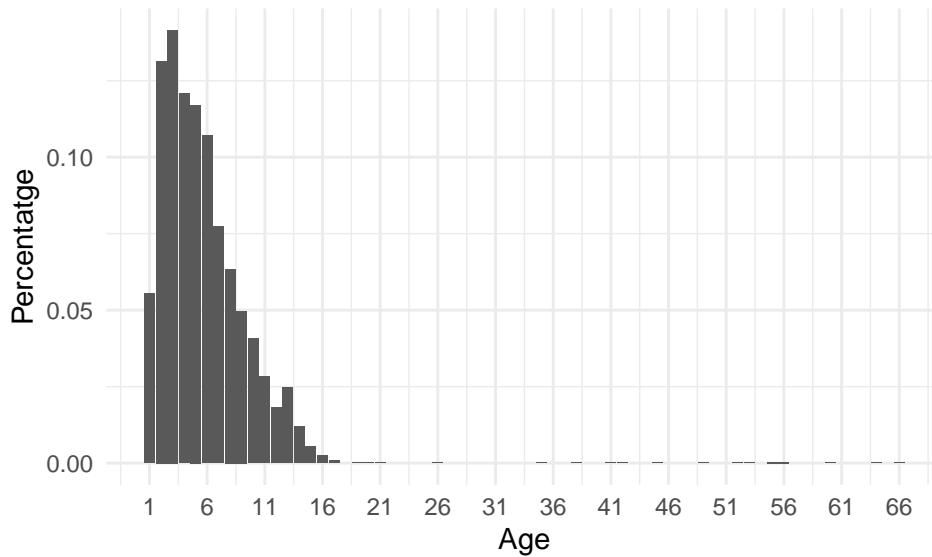
Creem un nou indicador o atribut on emmagatzemem l'antiguitat del vehicle.

```

# Antiguitat del vehicle
ds["age"] <- as.integer(format(Sys.Date(), "%Y")) - ds$vehicle.year

# Calculem gràfic de barres de la nova variable
ggplot(data = ds, aes(x = age)) + geom_bar(aes(y = ..count../sum(..count..))) +
  scale_x_continuous(breaks = round(seq(min(ds$age), max(ds$age),
  by = 5), 1)) + xlab("Age") + ylab("Percentatge")

```

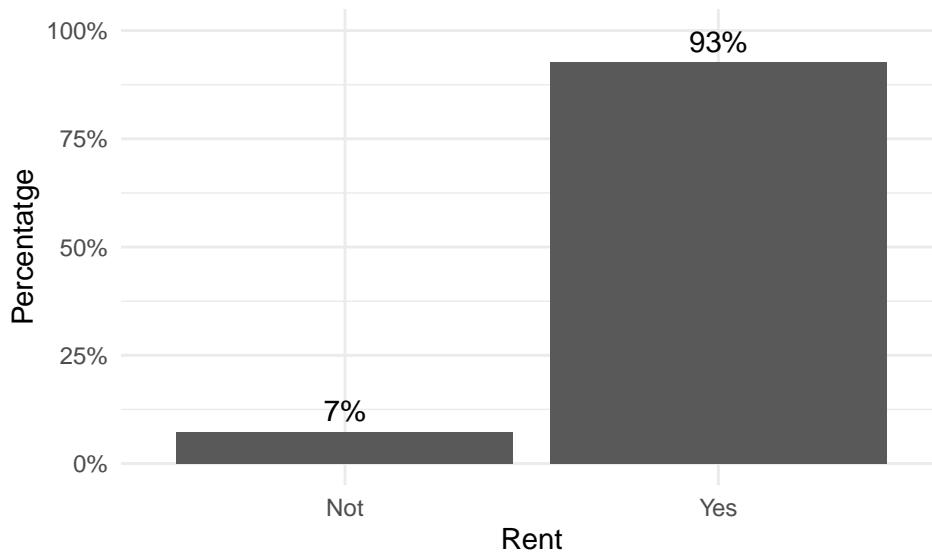


Indicador: rent

Creem un altre indicador o atribut nou on emmagatzemem si el vehicle va ser llogat o no.

```
# Vehicle llogat/no llogat
ds["rent"] <- ifelse(ds["renterTripsTaken"] > 0, 1, 0)
ds$rent <- as.factor(ds$rent)
levels(ds$rent) <- c("Not", "Yes")

# Calculem gràfic de barres de la variable objectiu
ggplot(data = ds, aes(x = rent)) + geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(y = ((..count..)/sum(..count..))), label = scales::percent(..count../sum(..count..))),
            stat = "count", vjust = -0.5) + scale_y_continuous(labels = scales::percent,
  limits = c(0, 1)) + xlab("Rent") + ylab("Percentatge")
```



Indicador: income

Creem un altre indicador o atribut nou on emmagatzemem els ingressos anuals que va aportar el vehicle.

Per crear aquesta variable **income** (ingressos anuals mitjans), s'assumeix el següent:

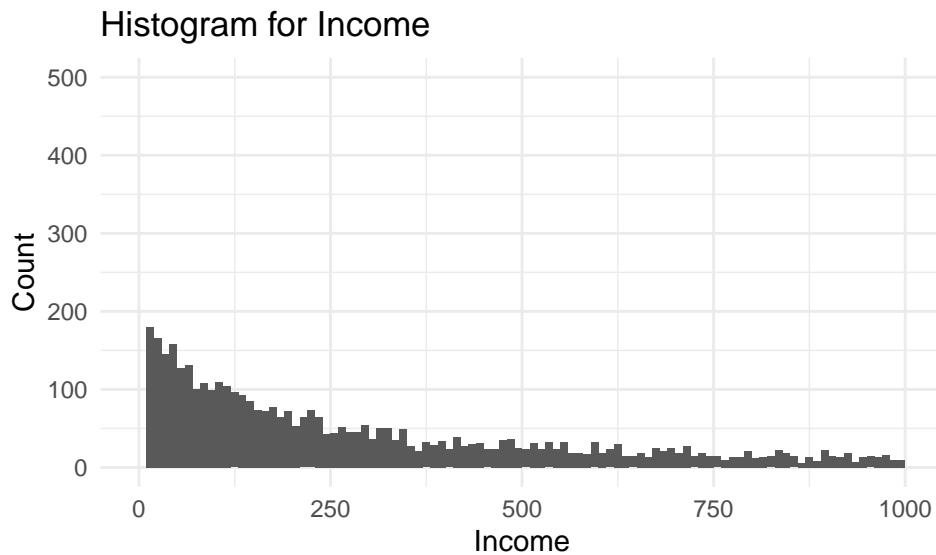
- **renterTripsTaken** és equivalent al nombre de dies que el cotxe ha estat llogat i que per tant es poden aproximar els ingressos totals aportats pel vehicle com a resultat del producte entre **renterTripsTaken** i **rate.daily**.
- s'assumeix que **age** representa tots els anys de vida del cotxe, per tant si es vol calcular els ingressos anuals cal repartir-los entre age

```
# Ingressos anuals vehicle

# En primer lloc establim que la edat mínima amb la que
# treballem és d'un any per tal d'evitar divisors de '0'
edat = ds$age
edat[edat == 0] = 1

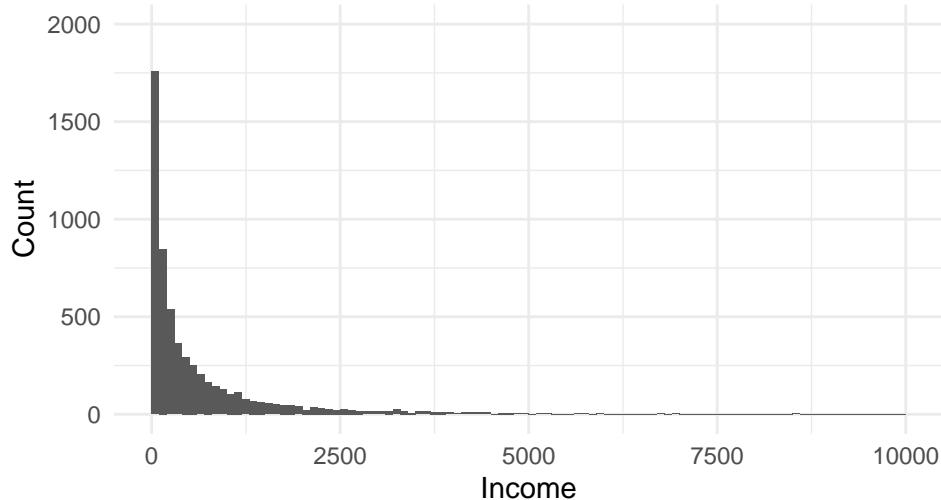
ds["income"] <- ds$rate.daily * ds$renterTripsTaken/edat

# Calculem gràfic de barres de la nova variable
ggplot(data = ds, aes(x = ds$income)) + geom_histogram(breaks = seq(0,
  1000, by = 10), alpha = 1) + labs(title = "Histogram for Income",
  x = "Income", y = "Count") + xlim(c(0, 1000)) + ylim(c(0,
  500))
```



```
ggplot(data = ds, aes(x = ds$income)) + geom_histogram(breaks = seq(0,
  10000, by = 100), alpha = 1) + labs(title = "Histogram for Income",
  x = "Income", y = "Count") + xlim(c(0, 10000)) + ylim(c(0,
  2000))
```

Histogram for Income



Indicador: frequency

Creem un altre indicador o atribut nou on emmagatzemem la freqüència amb la que es va llogar el vehicle (**frequency**).

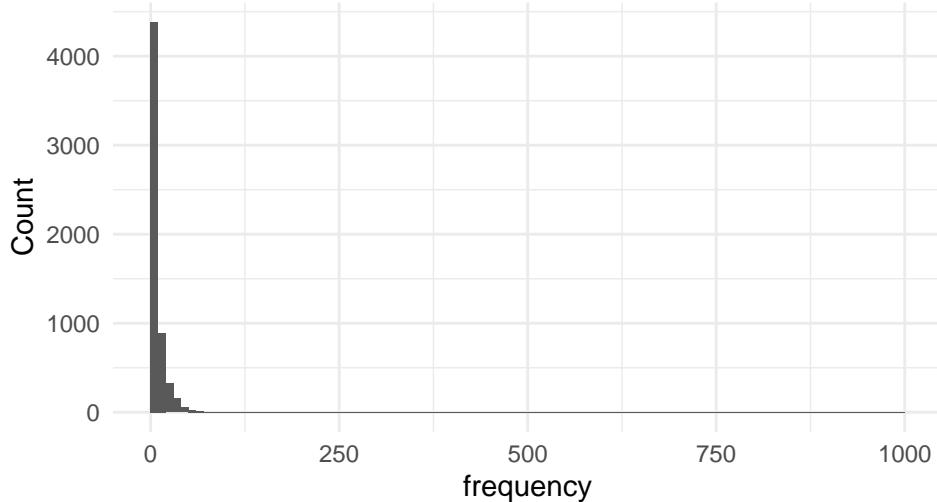
```
# Freqüència d'us del vehicle (vegades a l'any)

# En primer lloc establim que la edat mínima amb la que
# treballem és d'un any per tal d'evitar divisors de '0'
edat = ds$age
edat[edat == 0] = 1

ds["frequency"] <- ds$renterTripsTaken/edat

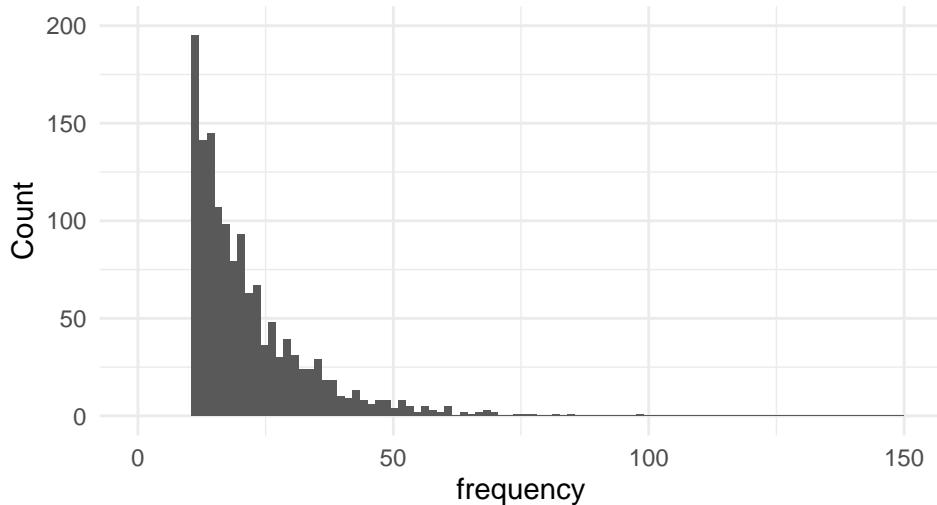
# Calculem gràfic de barres de la nova variable
ggplot(data = ds, aes(x = ds$frequency)) + geom_histogram(breaks = seq(0,
  1000, by = 10), alpha = 1) + labs(title = "Histogram for frequency",
  x = "frequency", y = "Count")
```

Histogram for frequency



```
ggplot(data = ds, aes(x = ds$frequency)) + geom_histogram(breaks = seq(0, 150, by = 1.5), alpha = 1) + labs(title = "Histogram for frequency", x = "frequency", y = "Count") + xlim(c(0, 150)) + ylim(c(0, 200))
```

Histogram for frequency



2.3.8 Exportació de les dades preprocessades

Una vegada que hem realitzat sobre el conjunt de dades inicial els procediments de preprocessament integració, validació i neteja, procedim a guardar aquest nou joc de dades en un fitxer anomenat CarRentalDataV1_Clean.csv.

```
# Exportació de les dades preprocessades a un fitxer .CSV  
write.csv(ds, "../data/CarRentalDataV1_Clean.csv")
```

2.4 Anàlisi de les dades

La gran majoria dels atributs presents en el conjunt de dades es corresponen amb característiques dels diversos vehicles de lloguer o de la seva ubicació, per tant serà convenient tenir-los en consideració durant la realització de les analisis. No obstant això, podem prescindir de tres atributs, l'estat (location.country; que sempre és *US*) i les dues coordenades geogràfiques (location.latitude i location.longitude) ja que la informació que ens aporten aquestes variables ja es troba implícita en la resta d'atributs de localització i, per tant, són menys rellevants a l' hora de resoldre el nostre problema.

2.4.1 Selecció dels grups de dades a analitzar

```
# Eliminem location.country i les coordenades geogràfiques
ds <- subset(ds, select = -c(location.latitude, location.longitude,
    location.country))

categorical_features_names = c("fuelType", "location.city", "location.state",
    "owner.id", "vehicle.make", "vehicle.model", "vehicle.type",
    "airportcity", "range", "population.dscr")

numeric_features_names = c("rating", "renterTripsTaken", "reviewCount",
    "rate.daily", "age", "income", "frequency", "population")
```

2.4.2 Comprobació de la normalitat

Per a la comprovació que els valors que prenen les nostres variables quantitatives provenen d'una població distribuïda normalment, utilitzarem la prova de normalitat d'Anderson-Darling. Per això, es comprova que per cada prova s'obté un *p*-valor superior al nivell de significació prefixat = 0.05 (Nivell de confiança del 95%). Si això es compleix, llavors es considera que la variable en qüestió segueix una distribució normal.

Test de normalitat

```
alpha = 0.05
col.names = colnames(ds)

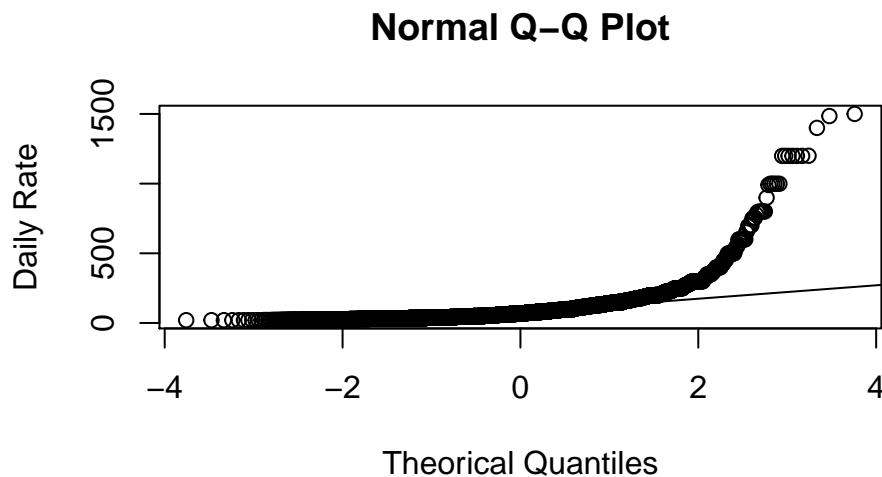
for (i in 1:ncol(ds)) {
  if (i == 1)
    cat("Variables que no presenten una distribució normal:\n")
  if (is.integer(ds[, i]) | is.numeric(ds[, i])) {
    p_val = ad.test(ds[, i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(ds) - 1)
        cat(", ")
      if (i%%3 == 0)
        cat("\n")
    }
  }
}

## Variables que no presenten una distribució normal:
## rating, renterTripsTaken,
## reviewCount, rate.daily, vehicle.year,
## population, age,
## incomefrequency
```

A partir del resultat anterior podem veure com que cap dels atributs quantitatius presenta una distribució normal. Tot i així ens centrem ara un moment el la variable dependent o variable objectiu en el nostre estudi, el preu diari del vehicle.

Si observem els valors mínim, mitjà i màxim veiem clarament que no és una variable amb distribució normal. Però en la majoria de models numèrics d'aprenentatge automàtic, necessitarem que les variables segueixin la distribució normal. Per això podem emprar visualitzacions de les dades per analitzar la normalitat. Concretament, el gràfic Q-Q, on la Q denota quantil, és un tipus de visualització que s'utilitza per a diagnosticar la desviació de les dades de la mostra en relació amb una població normal.

```
qqnorm(ds$rate.daily, ylab = "Daily Rate", xlab = "Theoretical Quantiles",
       main = "Normal Q-Q Plot")
qqline(ds$rate.daily)
```



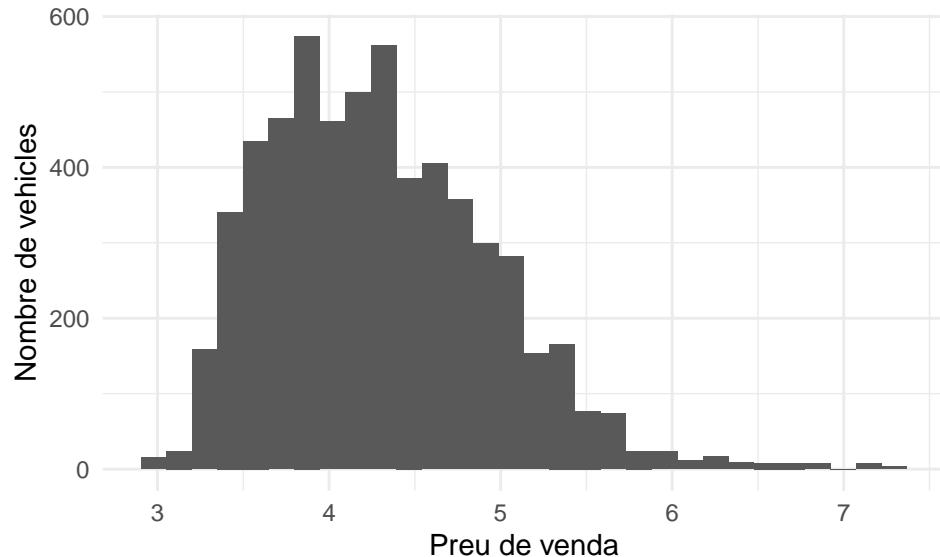
Quan tenim una variable que no és normal, una de les transformacions clàssiques que pot funcionar és aplicar el logaritme a la variable.

Alternativament podem usar la funció *BoxCox* la qual ens ajuda a seleccionar la transformació óptima de la variable per a que s'acabi distribuint de forma “normal.”

Normalització de Daily Rate (Preu)

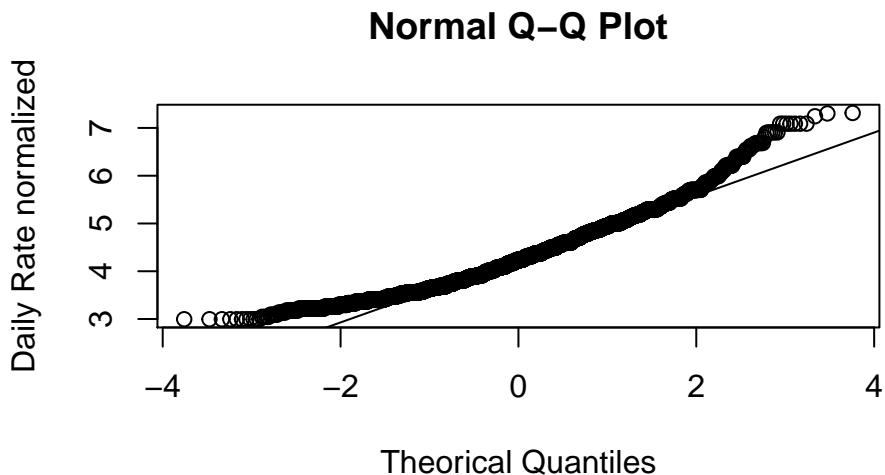
```
ds$rate.daily <- log(ds$rate.daily)

ggplot(ds, aes(x = rate.daily)) + geom_histogram() + ylab("Nombre de vehicles") +
  xlab("Preu de venda")
```



Veiem com ara sí que té forma normal. Ho comprovem amb el Q-Q plot per confirmar-ho.

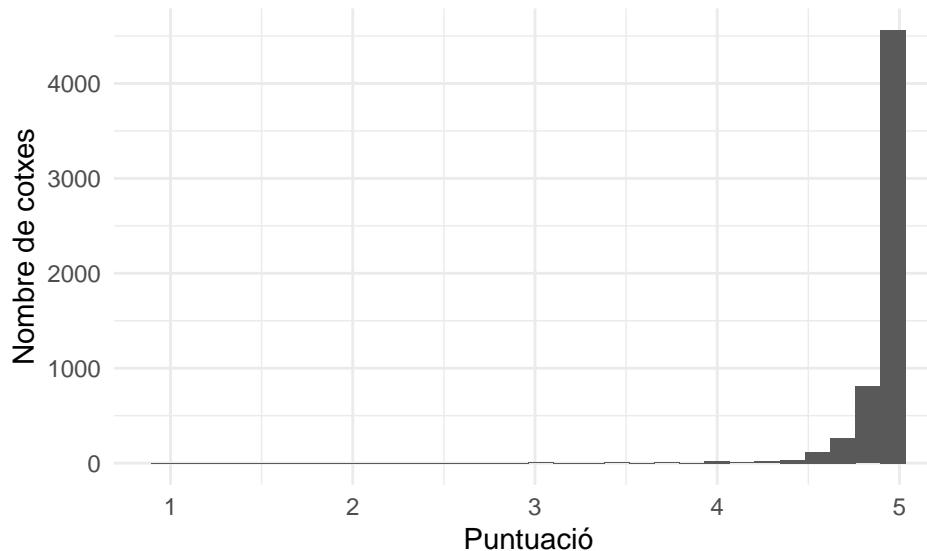
```
qqnorm(ds$rate.daily, ylab = "Daily Rate normalized", xlab = "Theoretical Quantiles",
       main = "Normal Q-Q Plot")
qqline(ds$rate.daily)
```



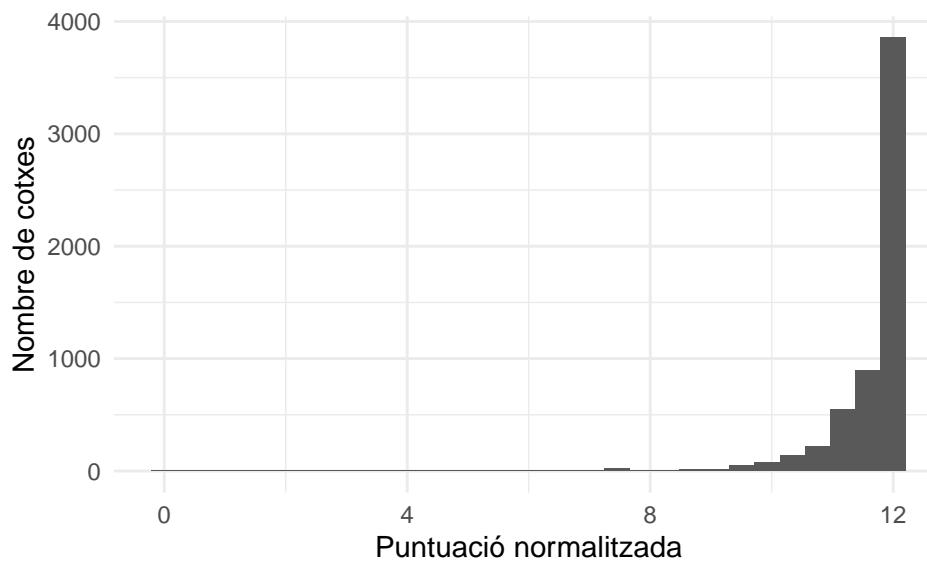
Normalització de Rating (Puntuació)

```
lambda_optima <- BoxCoxLambda(ds$rating)
ds$rating_norm <- BoxCox(ds$rating, lambda = lambda_optima)

ggplot(ds, aes(x = rating)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Puntuació")
```



```
ggplot(ds, aes(x = rating_norm)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Puntuació normalitzada")
```

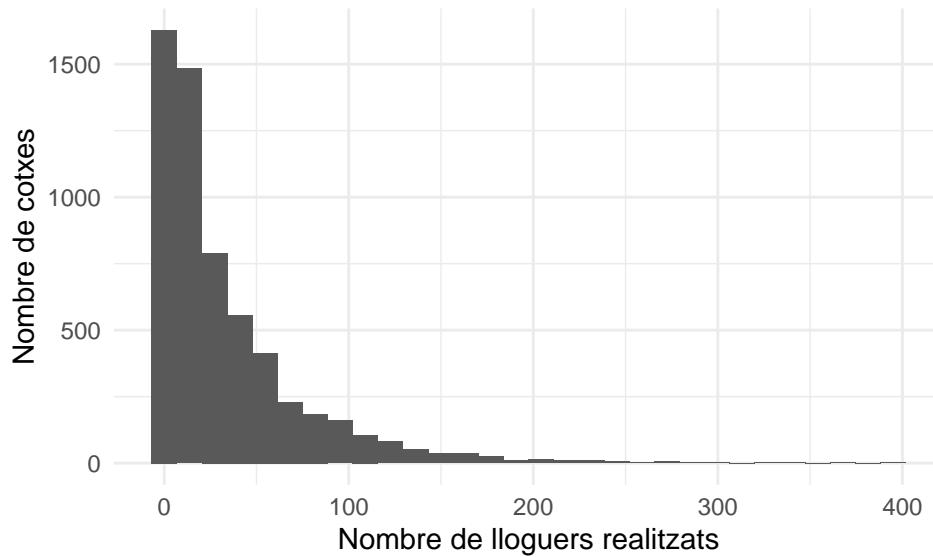


Veiem de forma evident que, tot i amb la conversió, no s'obté forma de distribució normal.

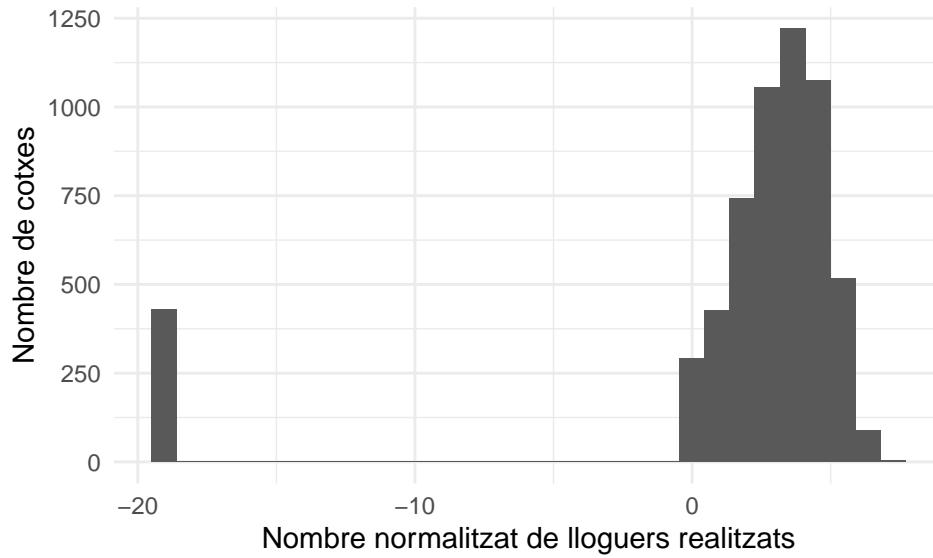
Normalització de Renter Trips Taken (Nombre de lloguers realitzats)

```
lambda_optima <- BoxCoxLambda(ds$renterTripsTaken)
ds$renterTripsTaken_norm <- BoxCox(ds$renterTripsTaken, lambda = lambda_optima)

ggplot(ds, aes(x = renterTripsTaken)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Nombre de lloguers realitzats")
```



```
ggplot(ds, aes(x = renterTripsTaken_norm)) + geom_histogram() +
  ylab("Nombre de cotxes") + xlab("Nombre normalitzat de lloguers realitzats")
```

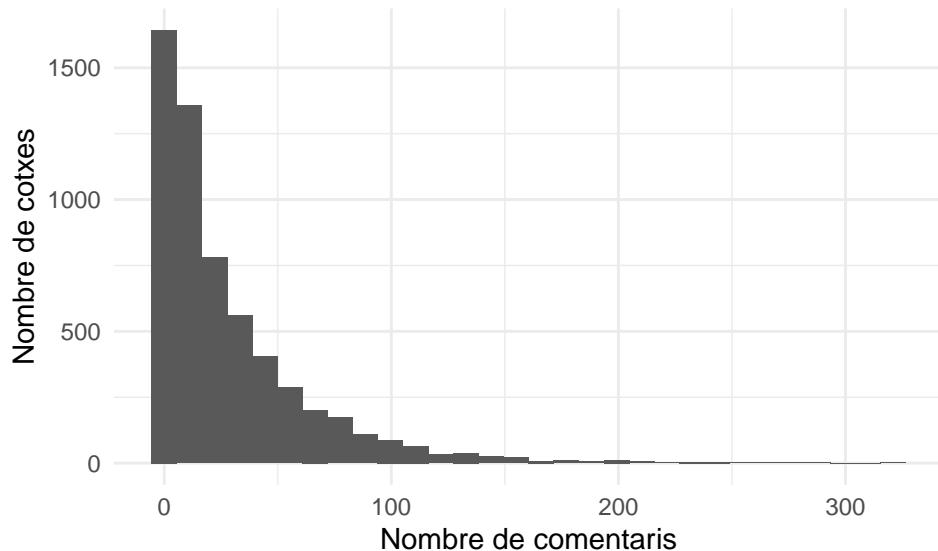


Veiem que, tot i amb la conversió, no s'obté forma de distribució normal ja que queden un grup significatiu de valors aïllats.

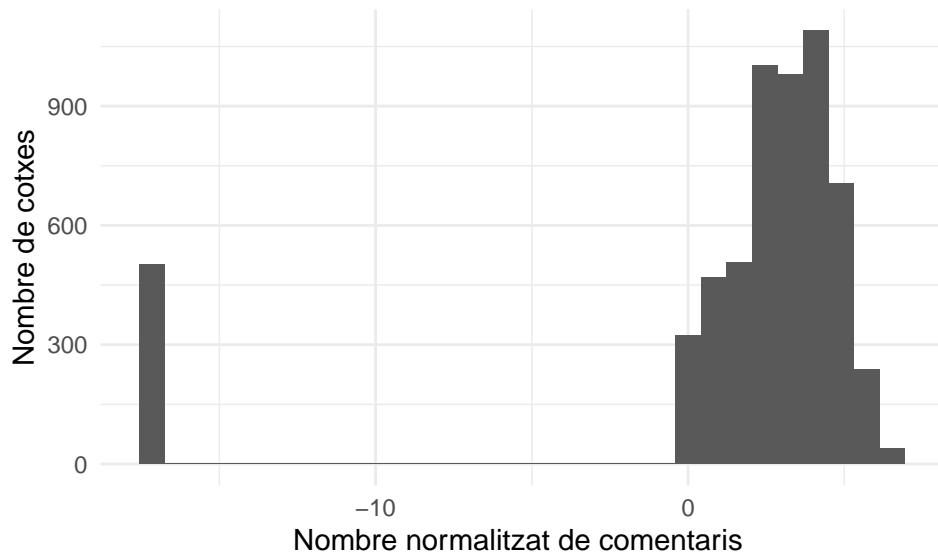
Normalització de reviewCount (Nombre de comentaris)

```
lambda_optima <- BoxCoxLambda(ds$reviewCount)
ds$reviewCount_norm <- BoxCox(ds$reviewCount, lambda = lambda_optima)

ggplot(ds, aes(x = reviewCount)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Nombre de comentaris")
```



```
ggplot(ds, aes(x = reviewCount_norm)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Nombre normalitzat de comentaris ")
```

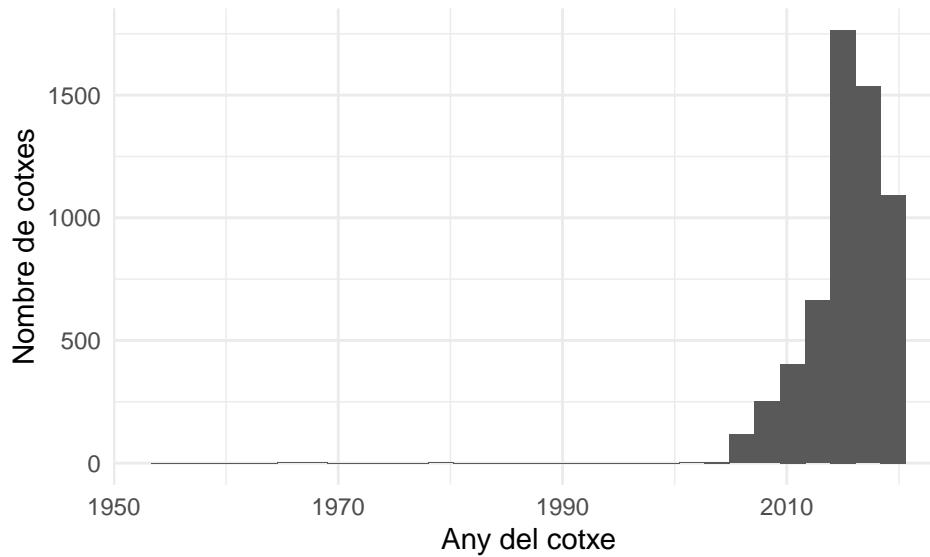


Veiem que, tot i amb la conversió, no s'obté forma de distribució normal ja que queden un grup significatiu de valors aïllats.

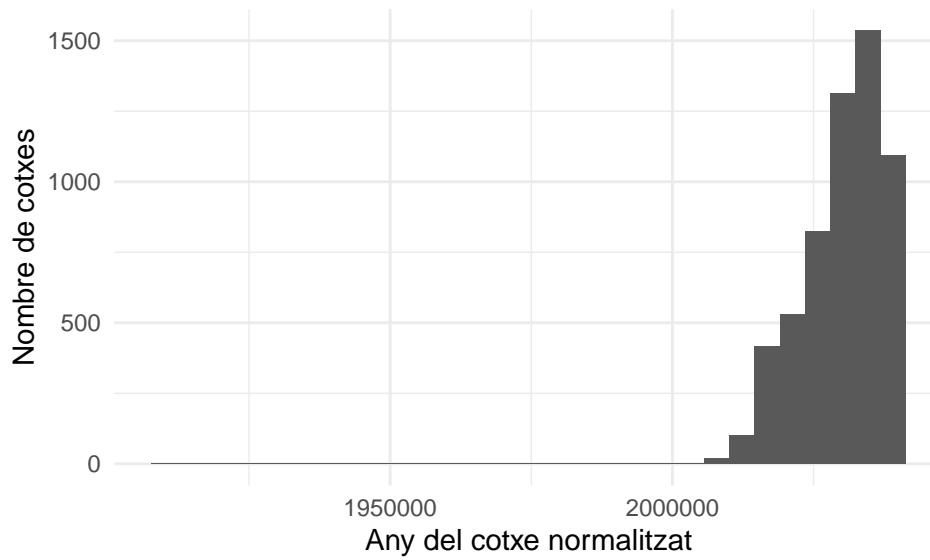
Normalització de vehicle.year (Any del cotxe)

```
lambda_optima <- BoxCoxLambda(ds$vehicle.year)
ds$vehicle.year_norm <- BoxCox(ds$vehicle.year, lambda = lambda_optima)

ggplot(ds, aes(x = vehicle.year)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Any del cotxe")
```



```
ggplot(ds, aes(x = vehicle.year_norm)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Any del cotxe normalitzat")
```

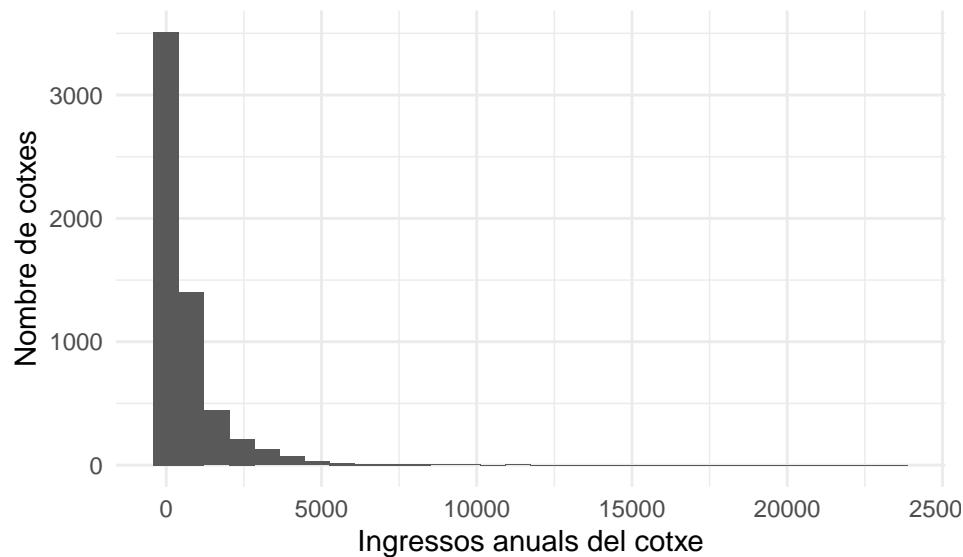


Veiem que, tot i amb la conversió, no s'obté forma de distribució normal.

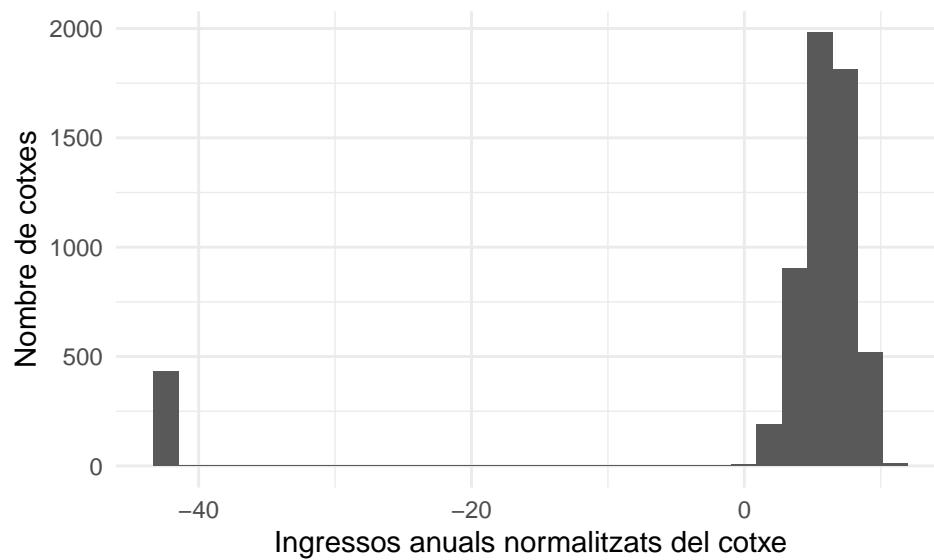
Normalització de Income (Ingressos anuals del cotxe)

```
lambda_optima <- BoxCoxLambda(ds$income)
ds$income_norm <- BoxCox(ds$income, lambda = lambda_optima)

ggplot(ds, aes(x = income)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Ingressos anuals del cotxe")
```



```
ggplot(ds, aes(x = income_norm)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Ingressos anuals normalitzats del cotxe")
```

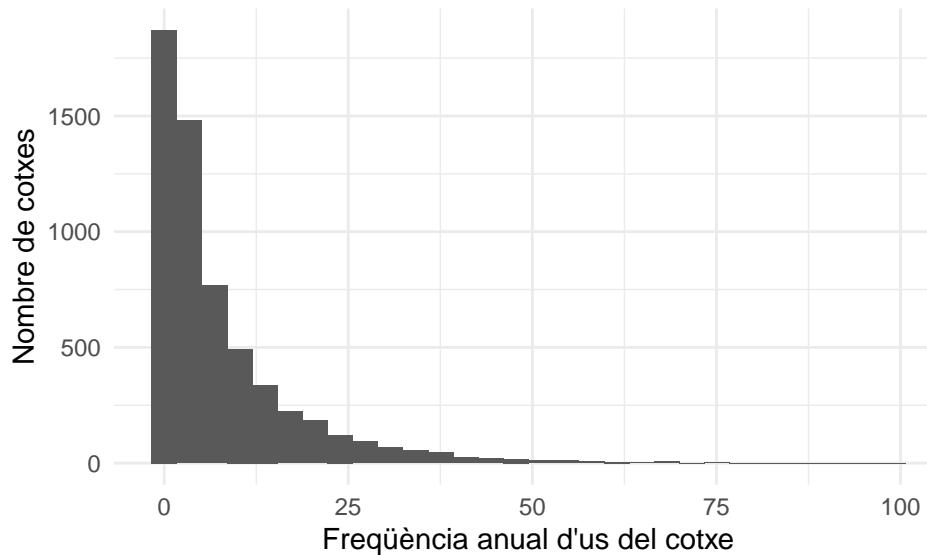


Veiem que, tot i amb la conversió, no s'obté forma de distribució normal ja que queden un grup significatiu de valors aïllats.

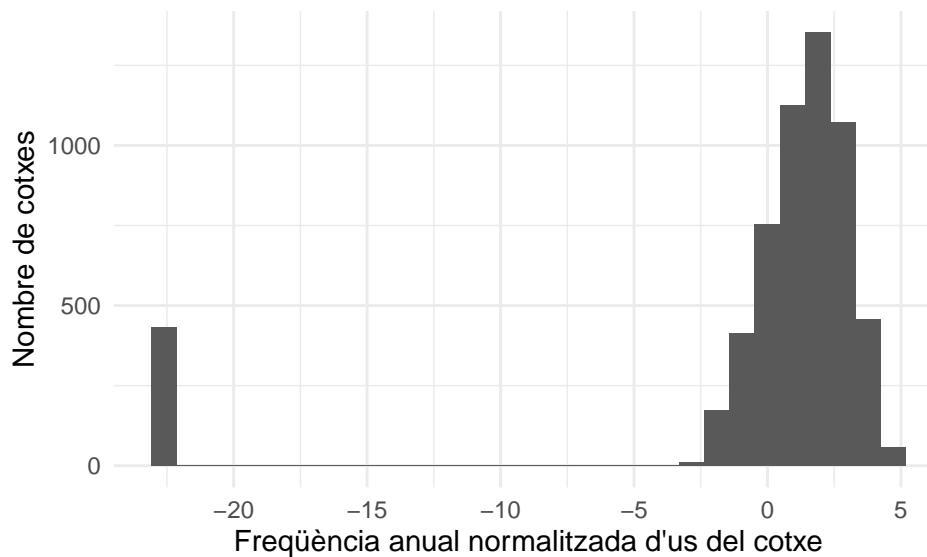
Normalització de Frecuency (freqüència anual d'us del cotxe)

```
lambda_optima <- BoxCoxLambda(ds$frequency)
ds$frequency_norm <- BoxCox(ds$frequency, lambda = lambda_optima)

ggplot(ds, aes(x = frequency)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Freqüència anual d'us del cotxe")
```



```
ggplot(ds, aes(x = frequency_norm)) + geom_histogram() + ylab("Nombre de cotxes") +
  xlab("Freqüència anual normalitzada d'us del cotxe")
```



Veiem que, tot i amb la conversió, no s'obté forma de distribució normal ja que queden un grup significatiu de valors aïllats.

Neteja de noves variables creades que no han tingut èxit en la normalització

```
# Eliminem location.country i les coordenades geogràfiques
ds <- subset(ds, select = -c(rating_norm, renterTripsTaken_norm,
  reviewCount_norm, vehicle.year_norm, income_norm, frequency_norm))

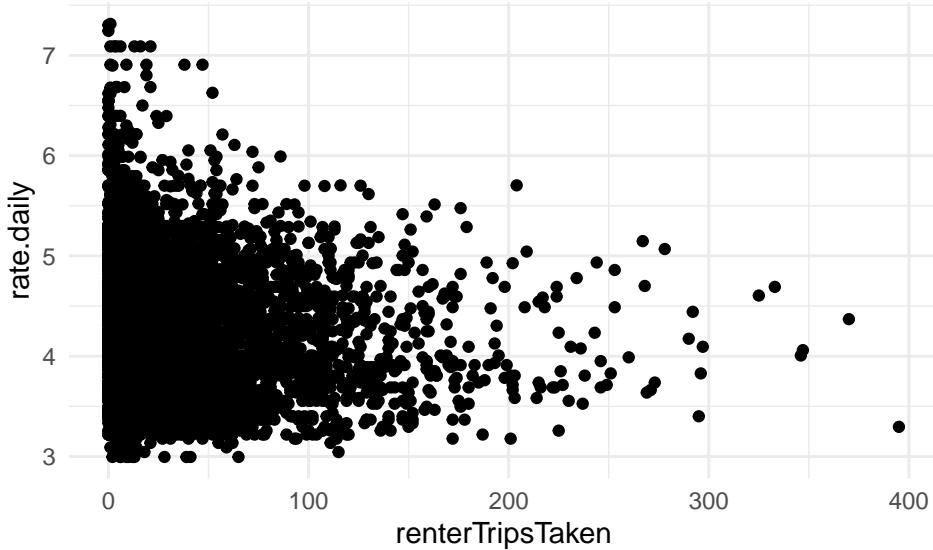
numeric_features_names = c("rating", "renterTripsTaken", "reviewCount",
  "rate.daily", "age", "income", "frequency", "population")
```

Anàlisi multivariant

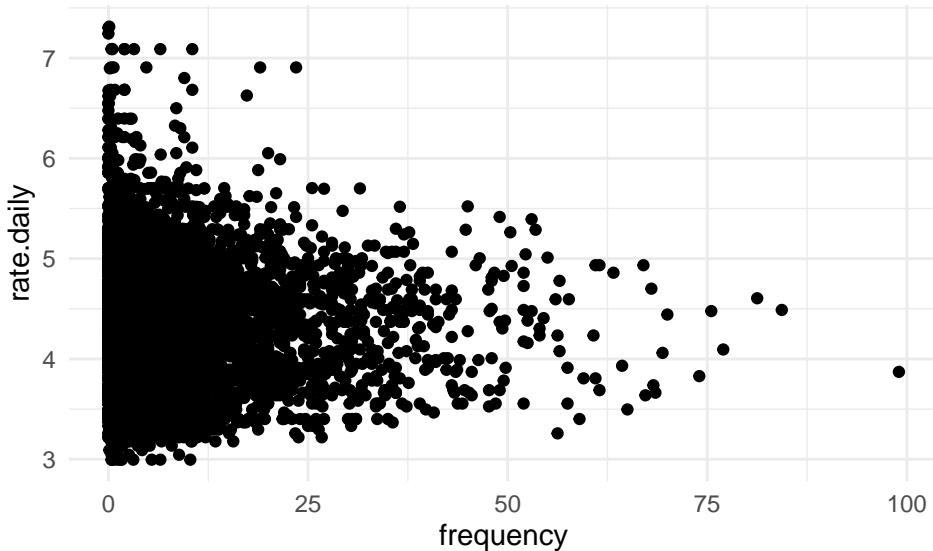
Per analitzar un conjunt de dades necessitem tenir en compte més d'una variable alhora. L'anàlisi bivariant permet identificar les relacions entre dues variables, i fins quina manera una pot predir l'altra.

En aquest cas podem veure quina és la relació entre el preu diari del lloguer i el nombre de vegades que s'ha llogat el vehicle amb un scatter plot o diagrama de punts.

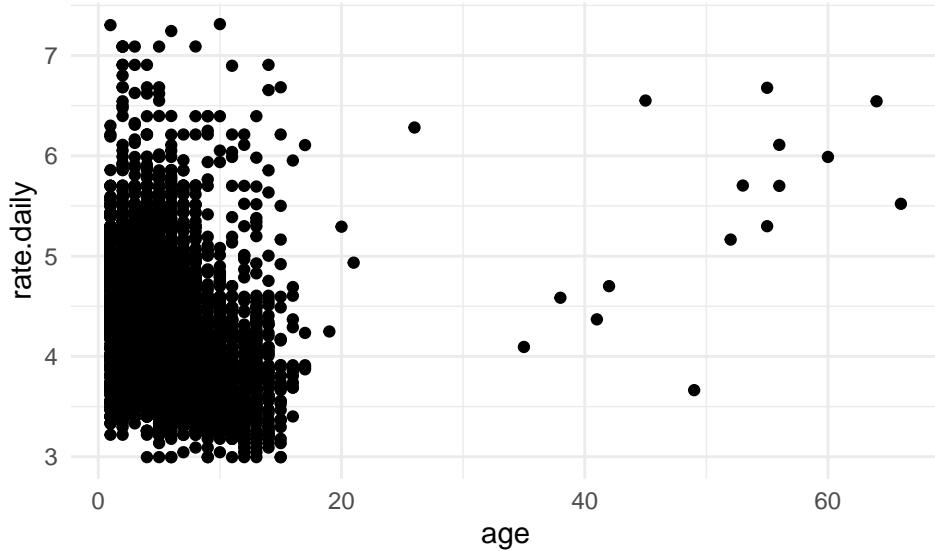
```
# Relació entre el preu diari del lloguer i el nombre de  
# vegades que s'ha llogat el vehicle  
ggplot(ds, aes(x = renterTripsTaken, y = rate.daily)) + geom_point()
```



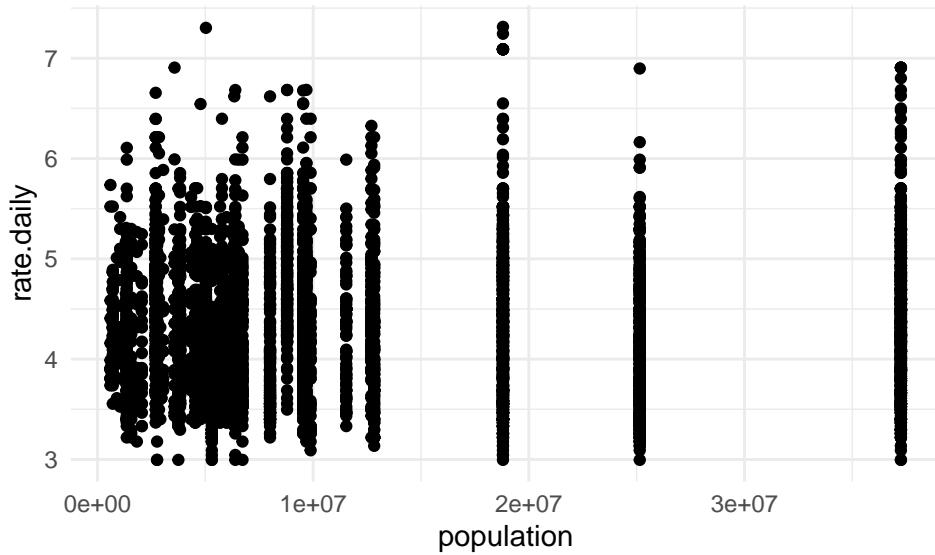
```
# Relació entre el preu diari del lloguer i la freqüència  
# amb que s'ha llogat un vehicle  
ggplot(ds, aes(x = frequency, y = rate.daily)) + geom_point()
```



```
# Relació entre e l'edat i el nombre de vegades que s'ha  
# llogat el vehicle  
ggplot(ds, aes(x = age, y = rate.daily)) + geom_point()
```

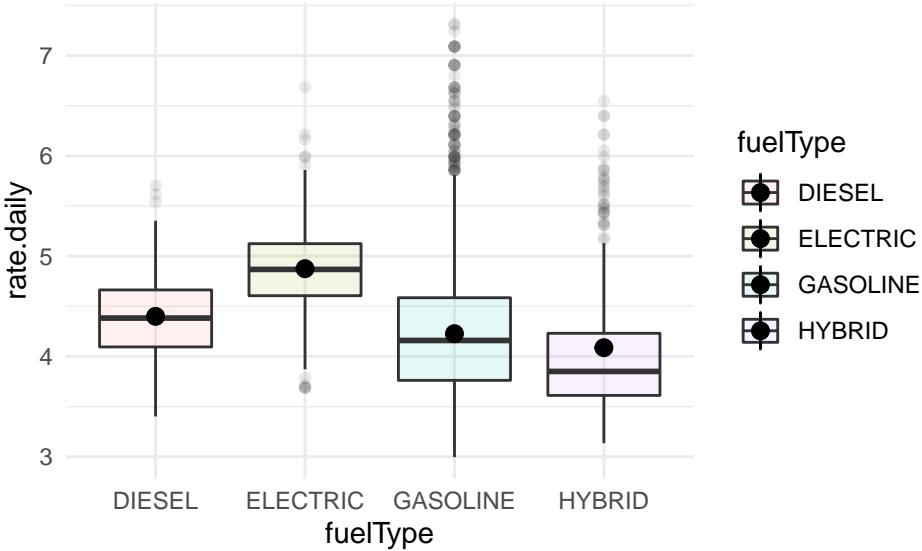


```
# Relació entre la població de l'estat i el nombre de vegades
# que s'ha llogat el vehicle
ggplot(ds, aes(x = population, y = rate.daily)) + geom_point()
```



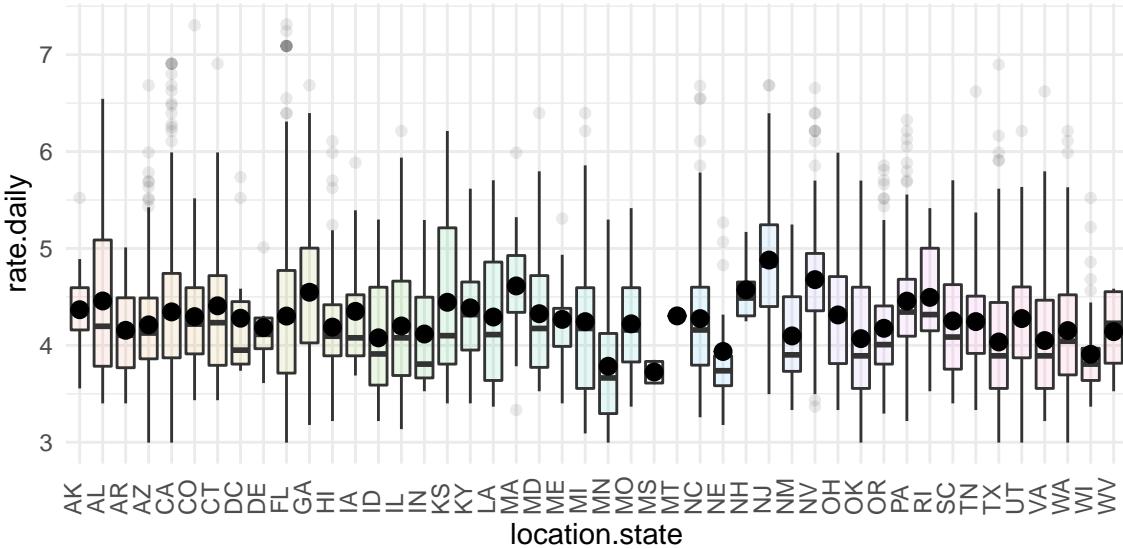
En el cas d'una variable categòrica com el tipus de combustible o la marca del vehicle, podem visualitzar la relació amb un boxplot o diagrama de caixes.

```
# Relació entre el preu diari del lloguer i el tipus de
# combustible
ggplot(ds, aes(x = fuelType, y = rate.daily, fill = fuelType)) +
  geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean)
```



A partir del gràfic anterior podem observar com el tipus de combustible sembla que influeix al preu diari de lloguer. Essent el lloguer més car pels vehicles elèctrics i el més econòmic el dels vehicles híbrids. Mentre que el preu dels vehicles diesel i gasolina és similar.

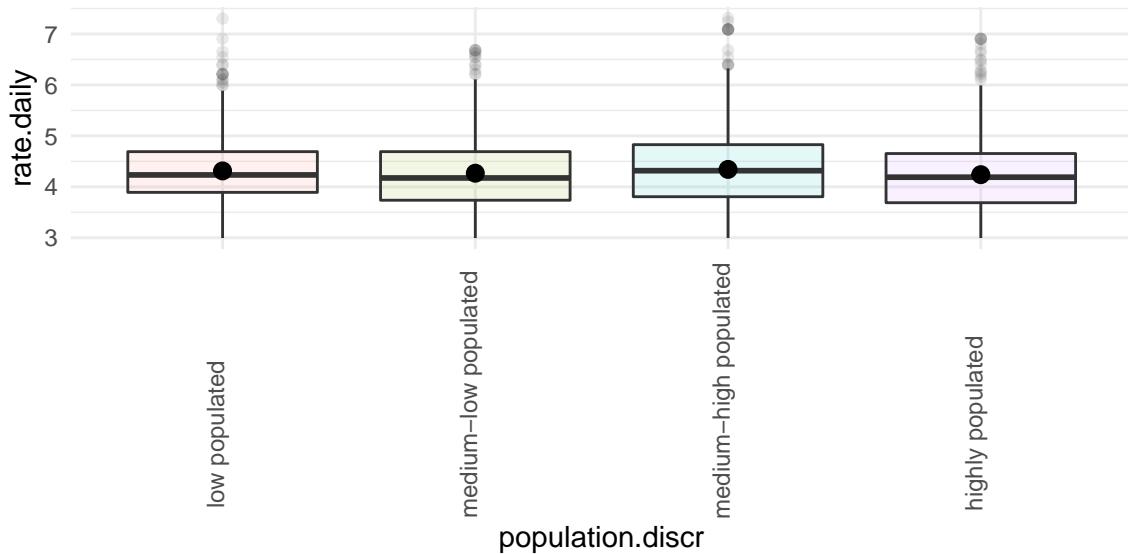
```
# Relació entre el preu diari del lloguer i l'estat on es localitza el vehicle
# ggplot(ds, aes(x = location.state, y = rate.daily, fill = location.state)) +
#   geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean) +
#   theme(axis.text.x = element_text(angle = 90, vjust = 0.1,
#   hjust = 0.1)) + theme(legend.position = "none")
```



A partir del gràfic anterior podem observar com l'estat on es localitza el vehicle sembla que influeix al preu diari de lloguer. Essent el lloguer més car pels vehicles de l'estat d'NM.

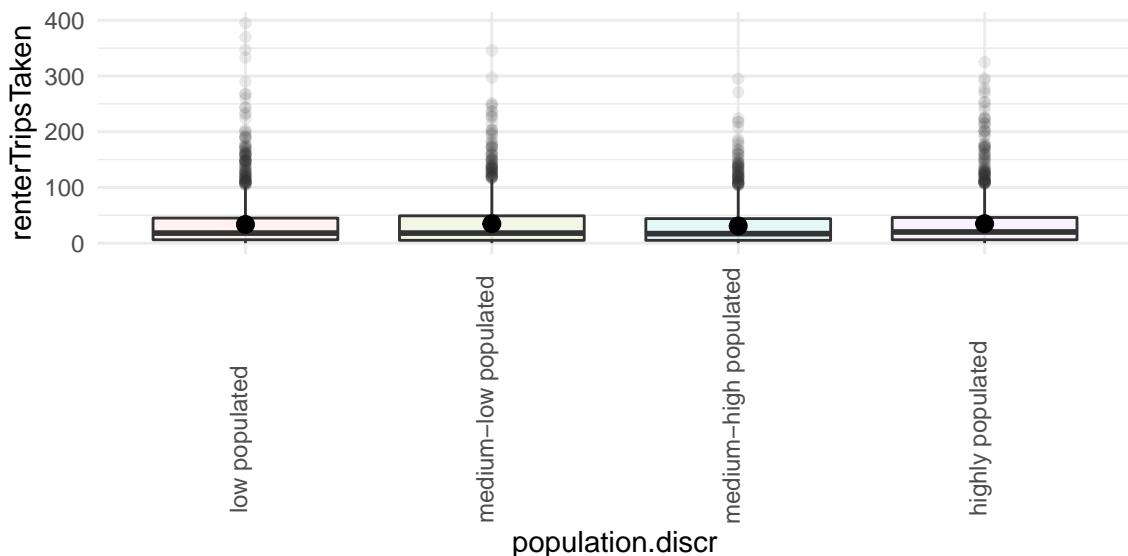
```
# Relació entre el preu diari del lloguer i la població que té l'estat on es troba el cotxe (discret)
# ggplot(ds, aes(x = population.discr, y = rate.daily, fill = population.discr)) +
#   geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean) +
```

```
theme(axis.text.x = element_text(angle = 90, vjust = 0.1,
                                  hjust = 0.1)) + theme(legend.position = "none")
```



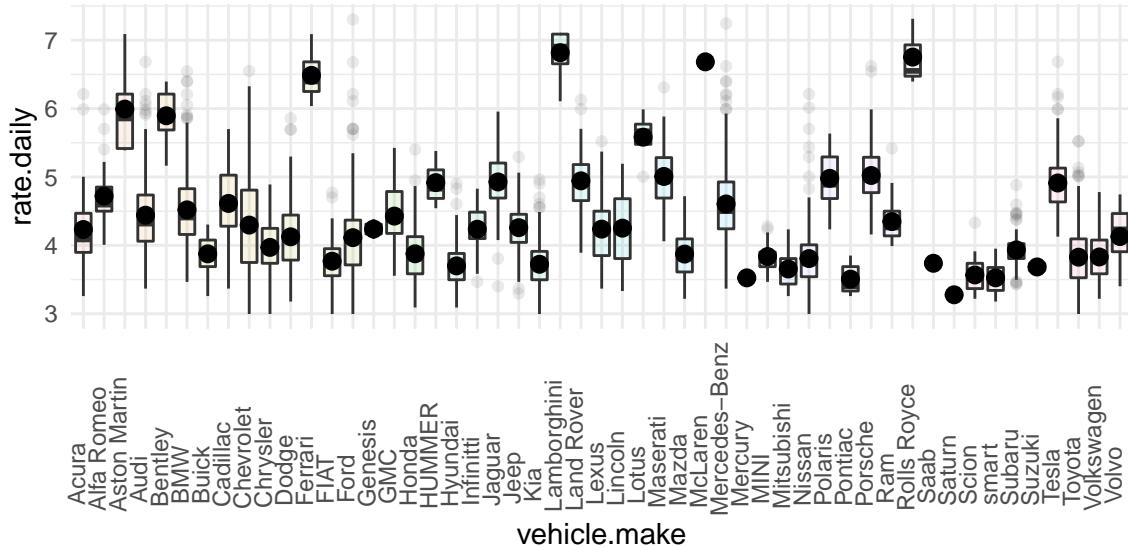
A partir del gràfic anterior podem observar com l'estat on es localitza el vehicle sembla no influeix en el preu diari del lloguer.

```
# Relació entre el renterTripstaken i la població que té
# l'estat on es troba el cotxe (discret)
ggplot(ds, aes(x = population.discr, y = renterTripsTaken, fill = population.discr)) +
  geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.1,
                                    hjust = 0.1)) + theme(legend.position = "none")
```



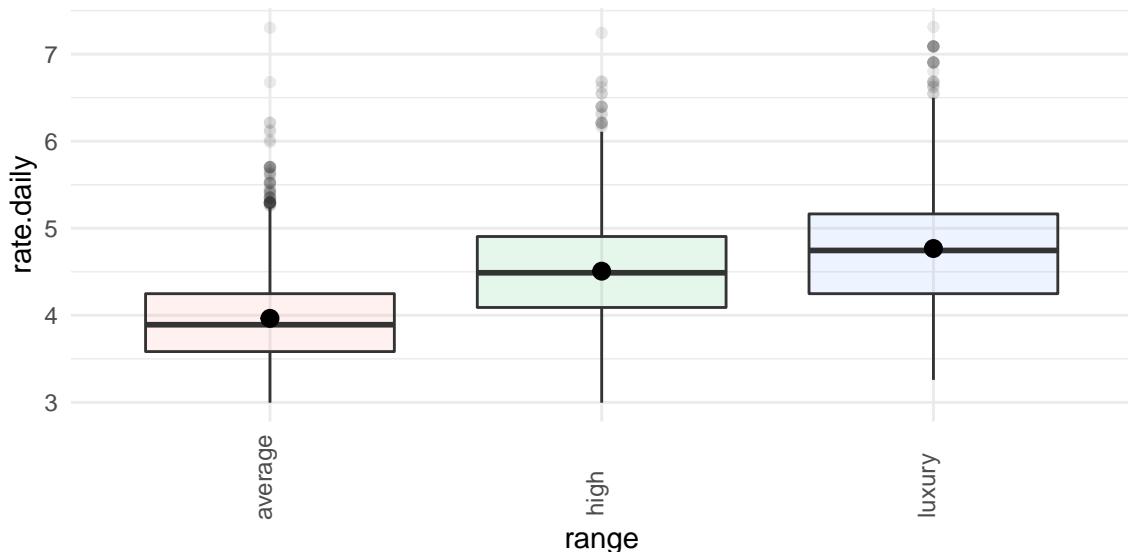
```
# Relació entre el preu diari del lloguer i la marca del
# vehicle
ggplot(ds, aes(x = vehicle.make, y = rate.daily, fill = vehicle.make)) +
  geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean) +
```

```
theme(axis.text.x = element_text(angle = 90, vjust = 0.1,
                                  hjust = 0.1)) + theme(legend.position = "none")
```



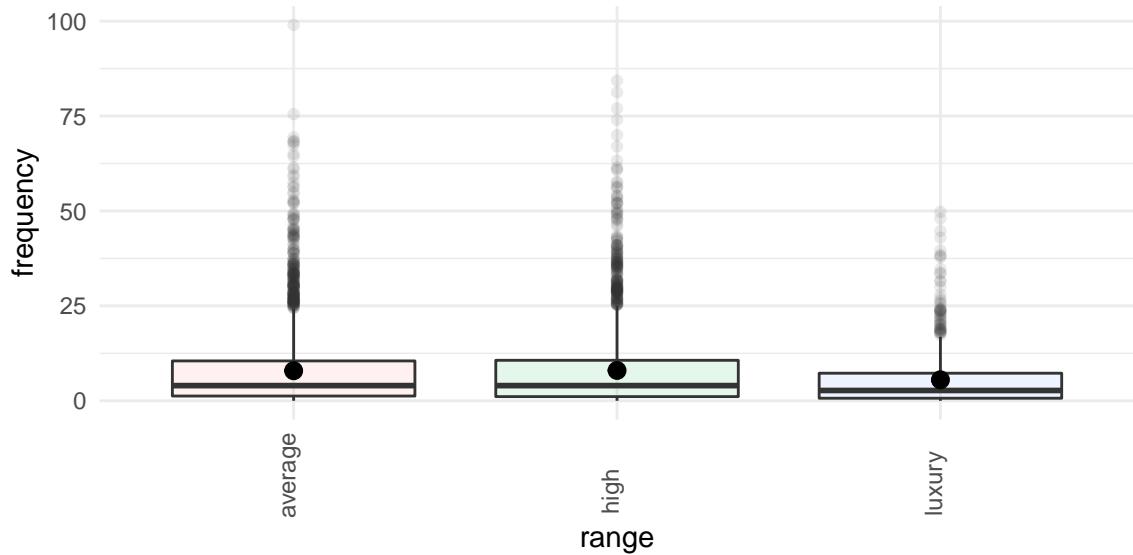
A partir del gràfic anterior podem observar com la marca del vehicle sembla que influeix al preu diari de lloguer. Essent el lloguer més car pels vehicles d'alta gamma, com ja podíem intuir, on destaca *Rolls Royce*, *Lamborgini*, *Aston Martin* i *Ferrari*.

```
# Relació entre el preu diari del lloguer i la gamma del
# vehicle
ggplot(ds, aes(x = range, y = rate.daily, fill = range)) + geom_boxplot(alpha = 0.1) +
  stat_summary(fun.y = mean) + theme(axis.text.x = element_text(angle = 90,
  vjust = 0.1, hjust = 0.1)) + theme(legend.position = "none")
```

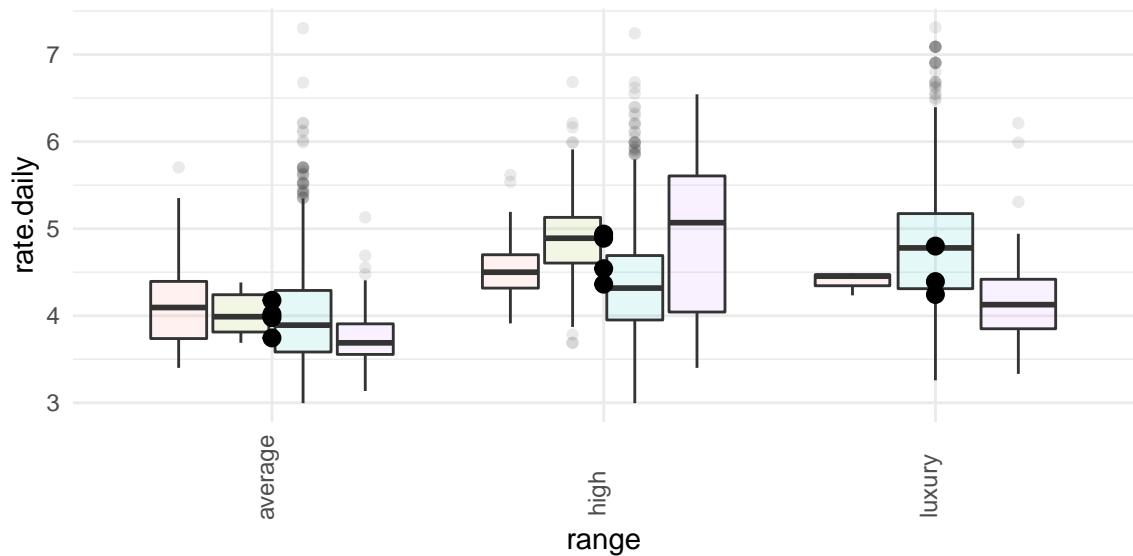


```
# Relació entre la freqüència del lloguer i la gamma del
# vehicle
ggplot(ds, aes(x = range, y = frequency, fill = range)) + geom_boxplot(alpha = 0.1) +
  stat_summary(fun.y = mean) + theme(axis.text.x = element_text(angle = 90,
```

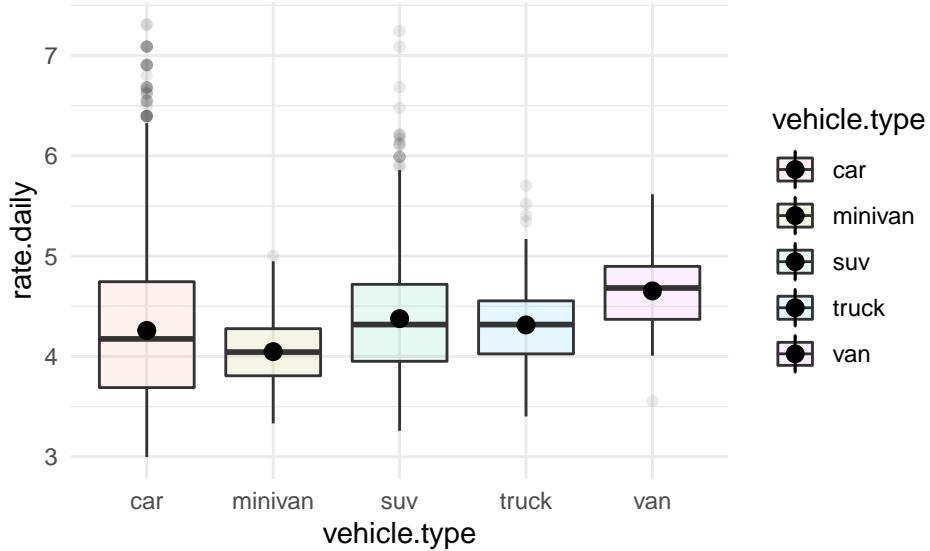
```
vjust = 0.1, hjust = 0.1)) + theme(legend.position = "none")
```



```
# Relació entre el preu diari del lloguer, la gamma del
# vehicle i el lcombustible
ggplot(ds, aes(x = range, y = rate.daily, fill = fuelType)) +
  geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.1,
                                    hjust = 0.1)) + theme(legend.position = "none")
```



```
# Relació entre el preu diari del lloguer i el tipus de
# vehicle
ggplot(ds, aes(x = vehicle.type, y = rate.daily, fill = vehicle.type)) +
  geom_boxplot(alpha = 0.1) + stat_summary(fun.y = mean)
```

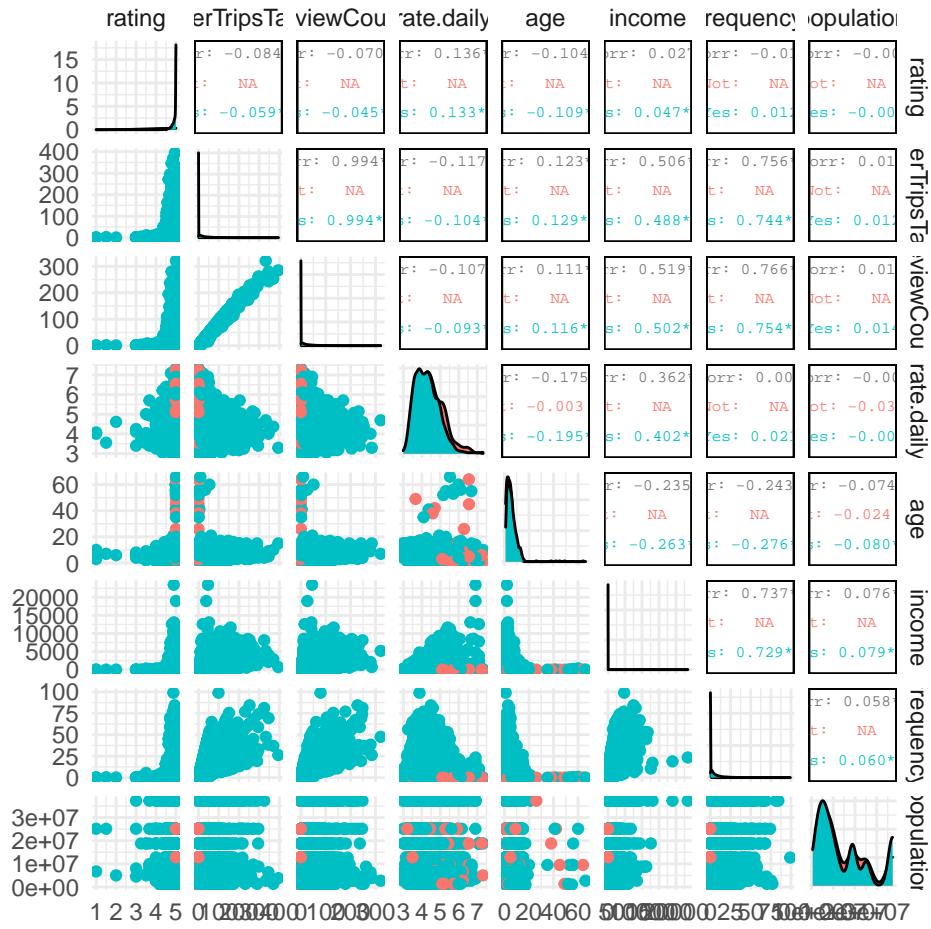


A partir del gràfic anterior podem observar com el tipus del vehicle sembla que influeix al preu diari de lloguer. Essent el lloguer més car per les furgonetes (*van*), i el més econòmic per les mini furgonetes (*minivan*). Mentre que per la resta de tipus de vehicles *car*, *suv* i *truck* no s'observa una gran diferència en el preu.

Si volem visualitzar alhora les relacions creuades entre diverses variables, podem fer un pairplot.

Examinem els atributs numèrics envers si el vehicle va se llogat o no.

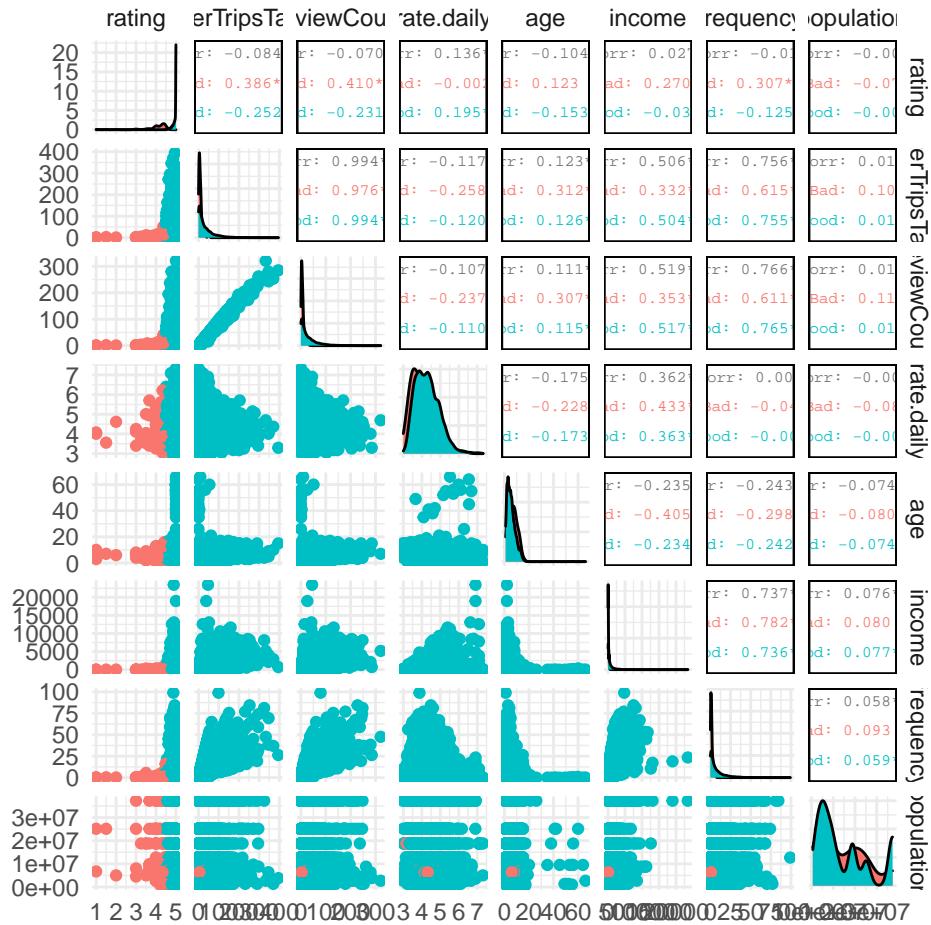
```
ggpairs(ds, columns = numeric_features_names, mapping = aes(color = rent),
       upper = list(continuous = wrap("cor", size = 2)))
```



A partir del pairplot anterior veiem que hi ha un solapament entre totes les funcions de densitat, en funció de si va ser llogat o no el vehicle, amb la qual cosa podem determinar que no hi ha cap atribut numèric que ens permeti identificar si el vehicle va ser llogat o no directament. D'altre banda podem destacar la relació lineal que s'observa entre l'atribut *renderTripsTaken* i *reviewCount*, on tal i com podriem pensar quans més cops es lloga un vehicle més valoracions rep.

Examinem els atributs numèrics envers la qualificació que va obtenir el vehicle.

```
ggpairs(ds, columns = numeric_features_names, mapping = aes(color = rating.discret),
       upper = list(continuous = wrap("cor", size = 2)))
```

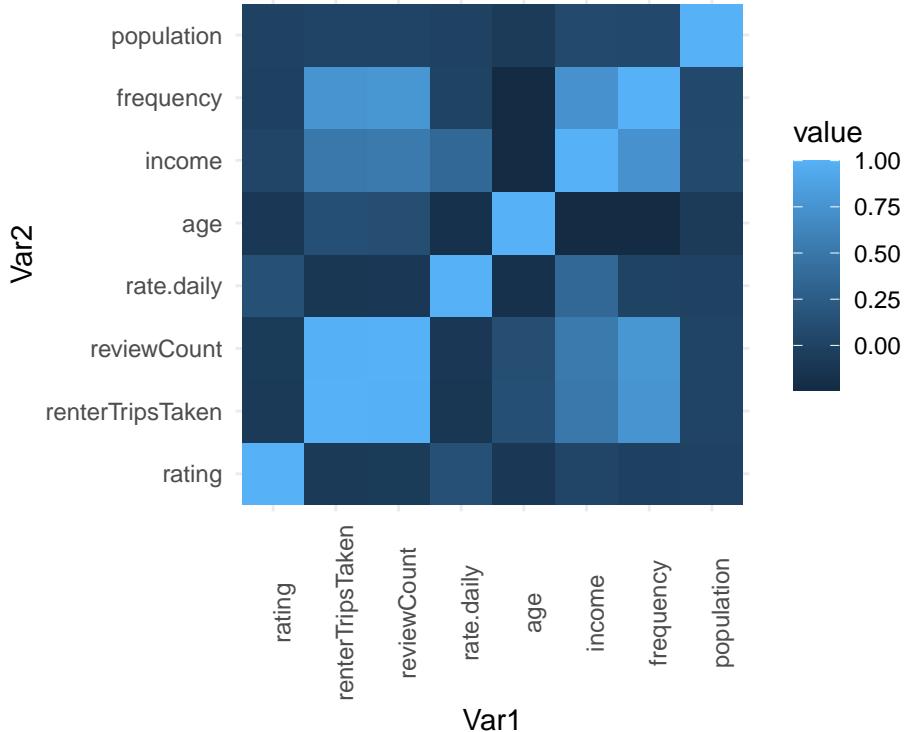


A partir del pairplot anterior veiem que hi ha un solapament entre totes les funcions de densitat, en funció de la qualificació obtinguda pel vehicle de lloguer, amb la qual cosa podem determinar que no hi ha cap atribut numèric que ens permeti identificar la puntuació del vehicle directament. D'altre banda també s'observa la relació lineal que hi ha entre l'atribut *renderTripsTaken* i *reviewCount*, on tal i com podriem pensar quans més cops es lloga un vehicle més valoracions rep.

Finalment, ens interessarà veure quins conjunts de variables estan relacionats entre si. Per això, farem servir tècniques estadístiques d'anàlisi multivariant.

Una de les eines més útils és calcular la matriu de correlació entre les variables. Amb la funció `qplot` i la correlació de variables, calculada amb la funció `cor`, podem visualitzar de manera fàcil aquelles variables més correlacionades, que corresponen a una intensitat major de color.

```
heat <- ds[, numeric_features_names]
qplot(x = Var1, y = Var2, data = melt(cor(heat, use = "p")),
      fill = value, geom = "tile") + theme(axis.text.x = element_text(angle = 90)) +
      coord_fixed()
```



2.5 Test estadístics

2.5.1 ¿Quines variables quantitatives influeixen més a les valoracions?

En primer lloc, procedim a realitzar una anàlisi de correlació entre les diferents variables per determinar quines d'elles exerceixen una major influència sobre el preu diari del lloguer del vehicle. Per a això, s'utilitzarà el coeficient de correlació de Spearman, ja que hem vist que tenim dades que no segueixen una distribució normal.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calcular el coeficiente de correlación para cada variable
# cuantitativa con respecto al campo 'precio'
for (i in 1:(ncol(ds) - 1)) {
  if (is.integer(ds[, i]) | is.numeric(ds[, i])) {
    spearman_test = cor.test(ds[, i], ds[, "rate.daily"],
      method = "spearman")
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value

    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(ds)[i]
  }
}
```

```
print(corr_matrix)
```

```
##                  estimate      p-value
## rating           0.23150081 4.931486e-72
## renterTripsTaken -0.13480526 3.905934e-25
## reviewCount      -0.12556188 5.388263e-22
## rate.daily        1.00000000 0.000000e+00
## vehicle.year      0.30145076 3.439955e-123
## population       -0.01551285 2.354549e-01
## age              -0.30145076 3.439955e-123
## income            0.32421223 2.832769e-143
```

A partir del resultat obtingut podem identificar quines són les variables més correlacionades amb el preu diari de lloguer en funció de la seva proximitat amb els valors -1 i +1. Tenint en compte això, queda palès com la variable més rellevant en la fixació del preu és l'antiguitat del vehicle (age) seguida de les valoracions. La variable **income** obté obviament una elevada correlació no obstant o entrem en valorar-la ja què està creada amb un calcul a partir de la mateixa **rate.daily**

Nota. Per a cada coeficient de correlació es mostra també el seu *p*-*valor* associat, ja que aquest pot donar informació sobre el pes estadístic de la correlació obtinguda.

2.5.2 ¿Els cotxes elèctric tenen un preu més elevat que els cotxes de benzina?

Per a avaluar si els cotxes elèctrics tenen un preu de lloguer diari més elevat que els cotxes de benzina, podem aplicar un test d'hipòtesis de dues mostres. Tal i com veurem a continuació.

Hipòtesi nul · la i alternativa

Comencem amb la definició de la hipòtesi nul · la i de la hipòtesi alternativa.

- Hipòtesi nul · la:

$$H_0 : \mu_1 = \mu_2$$

- Hipòtesi alternativa:

$$H_1 : \mu_1 > \mu_2$$

A continuació revisem si es compleix l'assumpció de normalitat per a la variable 'rate.daily' i a partir d'això, podrem explicar quin test podem aplicar per al test d'hipòtesis de dues mostres.

Test de normalitat

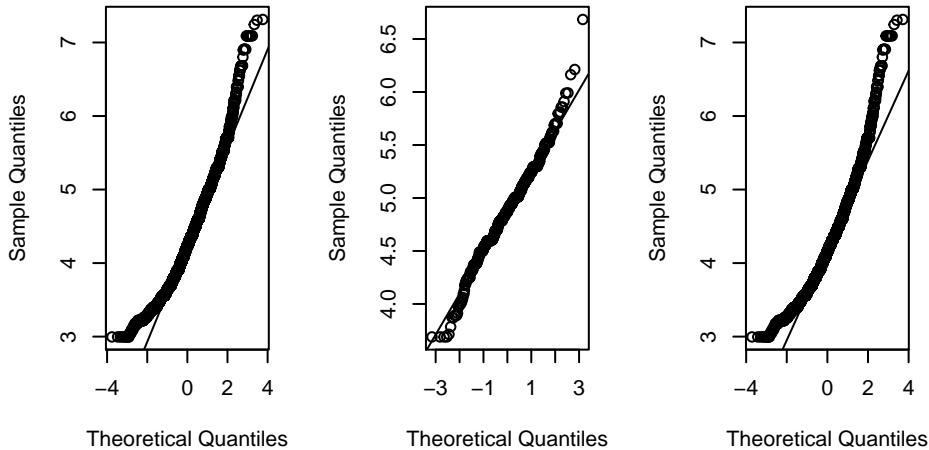
En primer lloc avaluem l'assumpció de normalitat per a la variable 'rate.daily'

```
par(mfrow = c(1, 3))
x <- ds$rate.daily
qq1 <- qqnorm(x, main = "Normal Q-Q Plot 'Daily rate'")
qqline(x)

x <- ds$rate.daily[ds$fuelType == "ELECTRIC"]
qq2 <- qqnorm(x, main = "Normal Q-Q Plot 'DR - ELECTRIC'")
qqline(x)

x <- ds$rate.daily[ds$fuelType == "GASOLINE"]
qq3 <- qqnorm(x, main = "Normal Q-Q Plot 'DR - GASOLINE'")
qqline(x)
```

Normal Q-Q Plot 'Daily rormal Q-Q Plot 'DR – ELEtrmal Q-Q Plot 'DR – GAS'



En aquest cas, els punts estan pràcticament sobre la línia, i, per tant, es pot assumir normalitat,

Per tant podem dir que totes dues poblacions es distribueixen normalment, però ens faltaria saber, donat que la variància poblacional és desconeguda, si aquestes poblacions presenten variàncies iguals o variàncies diferents.

Test d'igualtat de variàncies

Per a aplicar l'estadístic adequat, cal comprovar si les variàncies de les dues poblacions són iguals. Per això, apliquem primer el test d'igualtat de variàncies.

Per això podem realitzar un altre test, tal que:

- Hipòtesi nul·la:

$$H_0 : \sigma_1^2 = \sigma_2^2$$

- Hipòtesi alternativa:

$$H_1 : \sigma_1^2 \neq \sigma_2^2$$

```
# Obtenim les dades
ELECTRIC <- ds$rate.daily[ds$fuelType == "ELECTRIC"]
GASOLINE <- ds$rate.daily[ds$fuelType == "GASOLINE"]
```

```
# Calculem el test d'igualtat de variàncies, amb la funció
# var.test d'R
var.test(ELECTRIC, GASOLINE)
```

```
##
##  F test to compare two variances
##
## data:  ELECTRIC and GASOLINE
## F = 0.39604, num df = 621, denom df = 4884, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3528624 0.4469429
## sample estimates:
## ratio of variances
## 0.39604
```

Donat que el valor *p-value* obtingut és menor que el nivell de significació (0.05) podem rebutjar la hipòtesi nul·la i per tant podem dir que la variància de ambdues poblacions és diferent amb un nivell de confiança del 95%.

A partir dels resultats anteriors podem dir que el test a aplicar serà un test d'hipòtesis de dues mostres de poblacions independents amb distribucions normals i variàncies desconegudes i diferents.

Test de la mitjana de dues mostres independents amb variància desconeguda i diferents

Arribats aquest punt, amb la funció *t.test* d'R que ens permet realitzar el contrast d'hipòtesis directament, realitzem el test de mitjanes.

```
# variàncies diferents
t.test(ds$rate.daily[ds$fuelType == "ELECTRIC"], ds$rate.daily[ds$fuelType ==
  "GASOLINE"], alternative = "greater", var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data: ds$rate.daily[ds$fuelType == "ELECTRIC"] and ds$rate.daily[ds$fuelType == "GASOLINE"]
## t = 35.363, df = 1070.4, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.6179538      Inf
## sample estimates:
## mean of x mean of y
## 4.874673 4.226547
```

El valor *p-value* obtingut és menor que el nivell de significació (0.05) i, per tant, podem rebutjar la hipòtesi nul·la d'igualtat de mitjanes de preu de lloguer diari entre els cotxes elèctrics i els cotxes de benzina. Per tant podem dir que els cotxes elèctrics tenen un lloguer més elevat que els de benzina, amb un nivell de confiança del 95%.

2.5.3 ¿La proporció de furgonetes és més petit que la d'utilitaris?

Ara ens preguntem si la proporció de furgonetes de lloguer és més petit que la d'utilitaris o vehicles convencionals.

Hipòtesi nul·la i alternativa

Per tal de donar resposta a la pregunta formulada, comencem amb la definició de la hipòtesi nul·la i de la hipòtesi alternativa.

- Hipòtesi nul·la:

$$H_0 : p = 0.5$$

- Hipòtesi alternativa:

$$H_1 : p_1 < 0.5$$

En aquest cas específicament, ens preguntem si la proporció de furgonetes és més petita que la d'utilitaris, o el que és el mateix, si la proporció de furgonetes és igual a 0.5.

Test unilateral d'una mostra sobre la proporció

A continuació podem realitzar els càlculs pertinents que ens permetin decidir si podem rebutjar la hipòtesi nul·la o no.

Podem fer servir, la funció *prop.test* pròpia d'R que ens permet realitzar directament contrastos d'hipòtesis sobre proporcions.

```

n <- length(ds$vehicle.type)

prop.test(x = sum(ds$vehicle.type == "van" | ds$vehicle.type ==
  "minivan"), n = n, p = 0.5, alternative = "less", correct = FALSE)

##
## 1-sample proportions test without continuity correction
##
## data: sum(ds$vehicle.type == "van" | ds$vehicle.type == "minivan") out of n, null probability 0.5
## X-squared = 4759.3, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
## 0.00000000 0.05390773
## sample estimates:
##          p
## 0.04905144

```

D'altra banda el *p*-valor és inferior al nivell de significació (*p*-value <), per tant podem rebutjar la hipòtesi nul·la. Això ens permet dir que la proporció de furgonetes és més petita que la d'utilitàries, amb un nivell de confiança del 95%.

2.5.4 ¿El preu diari del lloguer del vehicle és diferent en funció del tipus de vehicle?

Per tal de donar resposta a aquesta pregunta, realitzarem un contrast de la igualtat de varianza mitjançant un test de *Bartlett* en dues o més poblacions sense la necessitat de que la mida dels grups que comparem sigui la mateixa.

```

car <- ds[ds$vehicle.type == "car", "rate.daily"]
minivan <- ds[ds$vehicle.type == "minivan", "rate.daily"]
suv <- ds[ds$vehicle.type == "suv", "rate.daily"]
truck <- ds[ds$vehicle.type == "truck", "rate.daily"]
van <- ds[ds$vehicle.type == "van", "rate.daily"]

bartlett.test(list(car, minivan, suv, truck, van))

```

```

##
## Bartlett test of homogeneity of variances
##
## data: list(car, minivan, suv, truck, van)
## Bartlett's K-squared = 349.44, df = 4, p-value < 2.2e-16

```

El *p*-valor és inferior al nivell de significació (*p*-value <), per tant podem rebutjar la hipòtesi nul·la. Això ens permet dir que el preu diari del lloguer del vehicle depén del tipus de vehicle, amb un nivell de confiança del 95%.

Podem veure que tenim cinc grups diferents (5 categories de vehicles) que volem comparar. A partir del test realitzat hi ha evidències de que la varianza no és la mateixa en tots els grups. Fet que ja podrem intuir al diagrama de caixes o boxplot on mostravem l'atribut 'rate.daily' envers 'vehicle.type'.

2.5.5 ¿El preu diari del lloguer del vehicle és diferent en funció del tipus de combustible?

Per tal de donar resposta a aquesta pregunta, realitzarem un contrast de la igualtat de varianza mitjançant un test de *Bartlett* en dues o més poblacions sense la necessitat de que la mida dels grups que comparem sigui la mateixa.

```

diesel <- ds[ds$vehicle.type == "car", "rate.daily"]
electric <- ds[ds$vehicle.type == "minivan", "rate.daily"]

```

```

gasoline <- ds[ds$vehicle.type == "suv", "rate.daily"]
hybrid <- ds[ds$vehicle.type == "truck", "rate.daily"]

bartlett.test(list(diesel, electric, gasoline, hybrid))

```

```

##
##  Bartlett test of homogeneity of variances
##
## data:  list(diesel, electric, gasoline, hybrid)
## Bartlett's K-squared = 327.66, df = 3, p-value < 2.2e-16

```

El *p*-valor és inferior al nivell de significació (*p*-value <), per tant podem rebutjar la hipòtesi nul · la. Això ens permet dir que el preu diari del lloguer del vehicle depén del tipus de combustible, amb un nivell de confiança del 95%.

Podem veure que tenim quatre grups diferents (4 tipus de combustibles) que volem comparar. A partir del test realitzat hi ha evidències de que la varianza no és la mateixa en tots els grups. Fet que ja podrem intuir al diagrama de caixes o boxplot on mostrarem l'atribut ‘rate.daily’ envers ‘fuelType’.

2.6 Model de regressió lineal múltiple per preveure el preu diari d'un vehicle

A continuació ens proposem estimar per mètode dels mínims quadrats ordinaris un model lineal que expliqui la variable *rate.daily* en funció de *age* i *renterTripsTaken*. En aquest cas farem servir atributs numèrics.

Model de regressió lineal múltiple 1 (Preu ~ age + renterTripsTaken)

```

model.lm1 <- lm(formula = rate.daily ~ age + renterTripsTaken,
                  data = ds)

summary(model.lm1)

##
## Call:
## lm(formula = rate.daily ~ age + renterTripsTaken, data = ds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.3593 -0.4833 -0.0676  0.3867  3.7207 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.4880624  0.0151804 295.648 < 2e-16 ***
## age         -0.0260129  0.0020630 -12.609 < 2e-16 ***
## renterTripsTaken -0.0014895  0.0001995  -7.468 9.34e-14 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6343 on 5848 degrees of freedom
## Multiple R-squared:  0.03969,    Adjusted R-squared:  0.03936 
## F-statistic: 120.8 on 2 and 5848 DF,  p-value: < 2.2e-16

```

Així doncs l'equació de regressió és:

$$\hat{y} = -0.0260 * \text{age} - 0.0014 * \text{renterTripsTaken} + 4.46$$

Es pot observar que tant la variable *age* i *renterTripsTaken* són significatives perquè $\text{Pr}(>|t|) < 0.05$.

Finalment el coeficient de determinació ajustat per aquest model és: $R^2 = 0.03936$. Això ens diu que el model de regressió múltiple obtingut explica el 3.936% de la variabilitat del preu del lloguer diari de vehicles. Com que és molt proper al 0%, en principi és un model bastant dolent, i per tant tindrà poc poder predictiu, gairebé nul.

Anem a veure ara si amb la introducció de noves variables al model, aconseguim un altre model que presenti una millor capacitat predictora. En aquest cas utilitzarem només atributs categòrics.

Model de regressió lineal múltiple 2 (Preu ~ age + renterTripsTaken + fuelType + vehicle.make + vehicle.type)

```
model.lm2 <- lm(formula = rate.daily ~ fuelType + vehicle.make +
  vehicle.type, data = ds)

summary(model.lm2)

##
## Call:
## lm(formula = rate.daily ~ fuelType + vehicle.make + vehicle.type,
##      data = ds)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1.5042 -0.2904 -0.0467  0.2322  3.2925 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.07867   0.08557  47.664 < 2e-16 ***
## fuelTypeELECTRIC 0.12838   0.08955   1.434 0.151762  
## fuelTypeGASOLINE  0.06056   0.05655   1.071 0.284238  
## fuelTypeHYBRID   0.14354   0.06377   2.251 0.024418 *  
## vehicle.makeAlfa Romeo 0.51480   0.10340   4.979 6.59e-07 ***
## vehicle.makeAston Martin 1.85335   0.21011   8.821 < 2e-16 ***
## vehicle.makeAudi   0.24863   0.07218   3.445 0.000576 *** 
## vehicle.makeBentley 1.75567   0.16229   10.818 < 2e-16 ***
## vehicle.makeBMW    0.33873   0.06691   5.063 4.26e-07 *** 
## vehicle.makeBuick   -0.35930   0.12064  -2.978 0.002911 ** 
## vehicle.makeCadillac 0.37954   0.09253   4.102 4.16e-05 *** 
## vehicle.makeChevrolet 0.07425   0.06714   1.106 0.268772  
## vehicle.makeChrysler -0.26109   0.08851  -2.950 0.003193 ** 
## vehicle.makeDodge    -0.12003   0.07263  -1.653 0.098459 .  
## vehicle.makeFerrari  2.34662   0.13960   16.810 < 2e-16 ***
## vehicle.makeFIAT    -0.37656   0.09404  -4.004 6.30e-05 *** 
## vehicle.makeFord    -0.12854   0.06721  -1.912 0.055866 .  
## vehicle.makeGenesis  0.09896   0.32288   0.306 0.759258  
## vehicle.makeGMC     0.07144   0.08444   0.846 0.397539  
## vehicle.makeHonda   -0.34824   0.06964  -5.001 5.87e-07 *** 
## vehicle.makeHUMMER  0.61246   0.26616   2.301 0.021421 *  
## vehicle.makeHyundai -0.47102   0.07164  -6.575 5.29e-11 *** 
## vehicle.makeInfiniti 0.00138   0.09145   0.015 0.987963  
## vehicle.makeJaguar   0.76621   0.08662   8.846 < 2e-16 *** 
## vehicle.makeJeep    -0.05529   0.06906  -0.801 0.423386  
## vehicle.makeKia     -0.47581   0.07318  -6.502 8.58e-11 *** 
## vehicle.makeLamborghini 2.65606   0.13193  20.133 < 2e-16 *** 
## vehicle.makeLand Rover 0.64091   0.07884  8.130 5.23e-16 *** 
## vehicle.makeLexus   0.01460   0.07548   0.193 0.846606
```

```

## vehicle.makeLincoln      0.01360   0.11643   0.117  0.906992
## vehicle.makeLotus        1.44280   0.23273   6.199  6.06e-10 ***
## vehicle.makeMaserati     0.85099   0.08392  10.140  < 2e-16 ***
## vehicle.makeMazda        -0.31926   0.08277  -3.857  0.000116 ***
## vehicle.makeMcLaren      2.54413   0.45214   5.627  1.92e-08 ***
## vehicle.makeMercedes-Benz 0.40798   0.06740   6.054  1.51e-09 ***
## vehicle.makeMercury       -0.78009   0.32285  -2.416  0.015712 *
## vehicle.makeMINI          -0.30311   0.10266  -2.953  0.003165 **
## vehicle.makeMitsubishi    -0.55361   0.10451  -5.297  1.22e-07 ***
## vehicle.makeNissan         -0.40409   0.06863  -5.888  4.13e-09 ***
## vehicle.makePolaris        0.84107   0.09183   9.160  < 2e-16 ***
## vehicle.makePontiac       -0.63313   0.21011  -3.013  0.002596 **
## vehicle.makePorsche        0.83397   0.07138  11.684  < 2e-16 ***
## vehicle.makeRam            -0.11563   0.12409  -0.932  0.351469
## vehicle.makeRolls Royce   2.61396   0.26620   9.820  < 2e-16 ***
## vehicle.makeSaab          -0.40156   0.45214  -0.888  0.374508
## vehicle.makeSaturn         -0.86226   0.32288  -2.670  0.007595 **
## vehicle.makeScion          -0.57227   0.13960  -4.099  4.20e-05 ***
## vehicle.makesmart          -0.61887   0.13568  -4.561  5.19e-06 ***
## vehicle.makeSubaru         -0.32552   0.08018  -4.060  4.98e-05 ***
## vehicle.makeSuzuki         -0.53812   0.32278  -1.667  0.095543 .
## vehicle.makeTesla          0.66800   0.09531   7.009  2.67e-12 ***
## vehicle.makeToyota         -0.40607   0.06632  -6.123  9.80e-10 ***
## vehicle.makeVolkswagen     -0.33638   0.07459  -4.509  6.63e-06 ***
## vehicle.makeVolvo          -0.11213   0.11452  -0.979  0.327546
## vehicle.typeminivan        0.18395   0.03391   5.424  6.06e-08 ***
## vehicle.typesuv            0.16548   0.01498  11.050  < 2e-16 ***
## vehicle.typetruck          0.34785   0.03631   9.580  < 2e-16 ***
## vehicle.typevan            0.46723   0.06255  7.470  9.21e-14 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4476 on 5793 degrees of freedom
## Multiple R-squared:  0.5263, Adjusted R-squared:  0.5216
## F-statistic: 112.9 on 57 and 5793 DF,  p-value: < 2.2e-16

```

Si examinem ara el coeficient de determinació ajustat per aquest nou model és: $R^2 = 0.52$. Això ens diu que el model de regressió múltiple obtingut explica el 52.22% de la variabilitat del preu del lloguer diari de vehicles. Com que és molt proper al 50%, en principi no és un model gaire bo, i per tant tindrà poc poder predictiu. Tot i així podem veure com hi ha un increment considerable sobre la variabilitat explicada respecte al primer model. A més, usant la variable **vehicle.make** que té 51 categories, sobre una mostra de menys de 6000 observacions, podem estar sobreajustant el model. En aquest sentit, fem a continuació la prova de crear una nova versió del mateix model amb la variable **range** (en lloc de **vehicle.make**).

Model de regressió lineal múltiple x (Preu ~ age + renterTripsTaken + fuelType + range + vehicle.type)

```

model.lm2_2 <- lm(formula = rate.daily ~ fuelType + range + vehicle.type,
                    data = ds)

summary(model.lm2_2)

##
## Call:
## lm(formula = rate.daily ~ fuelType + range + vehicle.type, data = ds)
## 
```

```

## Residuals:
##      Min     1Q   Median     3Q    Max
## -1.4526 -0.3544 -0.0605  0.2826  3.4112
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.93378   0.06731  58.446 < 2e-16 ***
## fuelTypeELECTRIC      0.47559   0.06989   6.805 1.11e-11 ***
## fuelTypeGASOLINE      -0.04181   0.06632  -0.630   0.528
## fuelTypeHYBRID        -0.07101   0.07421  -0.957   0.339
## rangehigh              0.43367   0.01635  26.527 < 2e-16 ***
## rangeluxury            0.81873   0.02372  34.517 < 2e-16 ***
## vehicle.typeminivan   -0.03140   0.03714  -0.845   0.398
## vehicle.typesuv        0.17584   0.01614  10.897 < 2e-16 ***
## vehicle.typetruck      0.28060   0.04091   6.859 7.64e-12 ***
## vehicle.typevan         0.47093   0.07461   6.312 2.95e-10 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5434 on 5841 degrees of freedom
## Multiple R-squared:  0.2961, Adjusted R-squared:  0.295
## F-statistic: 273 on 9 and 5841 DF, p-value: < 2.2e-16

```

En aquest cas el coeficient de determinació és de 0.295. És inferior al model anterior però també és un mòdel molt més senzill i per tant robust. No obstant encara ens trobem lluny de poder obtenir una predicción prou “bona” de **rate.daily**.

Model de regressió lineal múltiple 3 (Preu ~ age + renterTripsTaken + fuelType + vehicle.make + vehicle.type)

Ara probarem a generar un nou model, però aquest cop combinarem atributs categòrics i atributs numèrics com a variables predictores.

```

model.lm3 <- lm(formula = rate.daily ~ age + renterTripsTaken +
  fuelType + vehicle.make + vehicle.type, data = ds)

summary(model.lm3)

```

```

##
## Call:
## lm(formula = rate.daily ~ age + renterTripsTaken + fuelType +
##     vehicle.make + vehicle.type, data = ds)
##
## Residuals:
##      Min     1Q   Median     3Q    Max
## -1.4679 -0.2846 -0.0451  0.2218  3.5253
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.2311780   0.0851934  49.666 < 2e-16 ***
## age                   -0.0180611   0.0015205 -11.879 < 2e-16 ***
## renterTripsTaken      -0.0005815   0.0001420  -4.096 4.26e-05 ***
## fuelTypeELECTRIC       0.0668885   0.0884100   0.757 0.449337
## fuelTypeGASOLINE        0.0365756   0.0558002   0.655 0.512189
## fuelTypeHYBRID         0.1166403   0.0629064   1.854 0.063763 .
## vehicle.makeAlfa Romeo 0.4753796   0.1020759   4.657 3.28e-06 ***

```

```

## vehicle.makeAston Martin    1.9065989  0.2072218   9.201 < 2e-16 ***
## vehicle.makeAudi           0.2444084  0.0711597   3.435 0.000597 ***
## vehicle.makeBentley        1.8216603  0.1601206  11.377 < 2e-16 ***
## vehicle.makeBMW            0.3425710  0.0659822   5.192 2.15e-07 ***
## vehicle.makeBuick          -0.3516723  0.1189331  -2.957 0.003120 **
## vehicle.makeCadillac       0.3618105  0.0912341   3.966 7.41e-05 ***
## vehicle.makeChevrolet     0.0760267  0.0662404   1.148 0.251124
## vehicle.makeChrysler      -0.2577568  0.0872590  -2.954 0.003150 **
## vehicle.makeDodge          -0.1382077  0.0716446  -1.929 0.053771 .
## vehicle.makeFerrari        2.3948375  0.1376934  17.393 < 2e-16 ***
## vehicle.makeFIAT          -0.3427637  0.0927563  -3.695 0.000222 ***
## vehicle.makeFord           -0.1207694  0.0662800  -1.822 0.068490 .
## vehicle.makeGenesis        0.0496364  0.3183348   0.156 0.876097
## vehicle.makeGMC           0.0801708  0.0832452   0.963 0.335554
## vehicle.makeHonda          -0.3292116  0.0687075  -4.791 1.70e-06 ***
## vehicle.makeHUMMER         0.7486350  0.2626373   2.850 0.004381 **
## vehicle.makeHyundai        -0.4941139  0.0706751  -6.991 3.03e-12 ***
## vehicle.makeInfiniti       -0.0044169  0.0901541  -0.049 0.960927
## vehicle.makeJaguar         0.7527778  0.0854145   8.813 < 2e-16 ***
## vehicle.makeJeep           -0.0630605  0.0681497  -0.925 0.354836
## vehicle.makeKia            -0.4912122  0.0721733  -6.806 1.10e-11 ***
## vehicle.makeLamborghini    2.6389033  0.1300650  20.289 < 2e-16 ***
## vehicle.makeLand Rover     0.6378077  0.0777226   8.206 2.79e-16 ***
## vehicle.makeLexus          0.0323093  0.0744290   0.434 0.664235
## vehicle.makeLincoln        0.0018687  0.1147901   0.016 0.987012
## vehicle.makeLotus          1.4909414  0.2294833   6.497 8.88e-11 ***
## vehicle.makeMaserati       0.8271645  0.0827573   9.995 < 2e-16 ***
## vehicle.makeMazda          -0.3129117  0.0816281  -3.833 0.000128 ***
## vehicle.makeMcLaren        2.5082389  0.4457433   5.627 1.92e-08 ***
## vehicle.makeMercedes-Benz  0.4044692  0.0664475   6.087 1.22e-09 ***
## vehicle.makeMercury        -0.6447902  0.3184609  -2.025 0.042943 *
## vehicle.makeMINI           -0.2900896  0.1012106  -2.866 0.004169 **
## vehicle.makeMitsubishi     -0.5716323  0.1030670  -5.546 3.05e-08 ***
## vehicle.makeNissan          -0.4106626  0.0677123  -6.065 1.40e-09 ***
## vehicle.makePolaris        0.7878368  0.0906208   8.694 < 2e-16 ***
## vehicle.makePontiac        -0.5039509  0.2073744  -2.430 0.015123 *
## vehicle.makePorsche        0.8701432  0.0704499  12.351 < 2e-16 ***
## vehicle.makeRam             -0.1206797  0.1223279  -0.987 0.323916
## vehicle.makeRolls Royce   2.8825731  0.2634893  10.940 < 2e-16 ***
## vehicle.makeSaab           -0.3140029  0.4457879  -0.704 0.481226
## vehicle.makeSaturn          -0.7231035  0.3185202  -2.270 0.023232 *
## vehicle.makeScion          -0.4896068  0.1378240  -3.552 0.000385 ***
## vehicle.makesmart          -0.5850727  0.1337858  -4.373 1.25e-05 ***
## vehicle.makeSubaru          -0.3392380  0.0790873  -4.289 1.82e-05 ***
## vehicle.makeSuzuki          -0.4024498  0.3185273  -1.263 0.206471
## vehicle.makeTesla          0.6538783  0.0940757   6.951 4.04e-12 ***
## vehicle.makeToyota          -0.3732902  0.0654849  -5.700 1.25e-08 ***
## vehicle.makeVolkswagen     -0.3320587  0.0735394  -4.515 6.45e-06 ***
## vehicle.makeVolvo           -0.1357022  0.1129107  -1.202 0.229469
## vehicle.typeminivan        0.1908045  0.0334779   5.699 1.26e-08 ***
## vehicle.typesuv            0.1507048  0.0148240  10.166 < 2e-16 ***
## vehicle.typetruck          0.3159778  0.0359131   8.798 < 2e-16 ***
## vehicle.typevan            0.4352635  0.0618136   7.042 2.12e-12 ***
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4413 on 5791 degrees of freedom
## Multiple R-squared:  0.5398, Adjusted R-squared:  0.5351
## F-statistic: 115.1 on 59 and 5791 DF,  p-value: < 2.2e-16

```

Si examinem ara el coeficient de determinació ajustat per aquest nou model és: $R^2 = 0.5356$. Això ens diu que el model de regressió múltiple obtingut explica el 53.56% de la variabilitat del preu del lloguer diari de vehicles. Com que és molt proper al 50%, en principi no és un model gaire bo, i per tant tindrà poc poder predictiu. Tot i així podem veure com hi ha un increment considerable sobre la variabilitat explicada respecte al primer model, i un increment poc significatiu respecte el segon model. La qual cosa ens indica que a priori els atributs que millor expliquen el preu en un model lineal són les característiques del vehicle: el combustible, la marca i el tipus de vehicle, però no el nombre de vegades que el vehicle ha estat usat.

Ara probarem de fer un nou el model usant, com abans la variable **range** enllloc de **vehicle.make** i afegint la variable **population.discr**.

```

model.lm4 <- lm(formula = rate.daily ~ age + renterTripsTaken +
  fuelType + range + vehicle.type + population.discr, data = ds)

summary(model.lm4)

```

```

##
## Call:
## lm(formula = rate.daily ~ age + renterTripsTaken + fuelType +
##     range + vehicle.type + population.discr, data = ds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3637 -0.3525 -0.0567  0.2710  3.5557
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               4.1040023  0.0689585 59.514 < 2e-16  
## age                     -0.0165468  0.0018039 -9.173 < 2e-16  
## renterTripsTaken        -0.0008044  0.0001702 -4.726 2.34e-06 
## fuelTypeELECTRIC         0.4135131  0.0693672  5.961 2.65e-09 
## fuelTypeGASOLINE          -0.0637852  0.0655379 -0.973  0.3305  
## fuelTypeHYBRID            -0.0710719  0.0732838 -0.970  0.3322  
## rangehigh                 0.4271999  0.0161673 26.424 < 2e-16  
## rangeluxury                0.8250907  0.0235031 35.106 < 2e-16  
## vehicle.typeminivan       -0.0339728  0.0367493 -0.924  0.3553  
## vehicle.typesuv             0.1544843  0.0161793  9.548 < 2e-16  
## vehicle.typetruck          0.2488555  0.0405717  6.134 9.15e-10 
## vehicle.typevan              0.4173757  0.0738684  5.650 1.68e-08 
## population.discrmedium-low populated -0.0067448  0.0199276 -0.338  0.7350  
## population.discrmedium-high populated  0.0399455  0.0199715  2.000  0.0455  
## population.discrhighly populated    -0.0920112  0.0199607 -4.610 4.12e-06 
##
## (Intercept) ***                                          
## age          ***
## renterTripsTaken ***
## fuelTypeELECTRIC ***
## fuelTypeGASOLINE ***
## fuelTypeHYBRID ***
## rangehigh    ***

```

```

## rangeluxury          ***
## vehicle.typeminivan ***
## vehicle.typesuv       ***
## vehicle.typetruck     ***
## vehicle.typevan       ***
## population.discrmedium-low populated
## population.discrmedium-high populated *
## population.discrhighly populated    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.536 on 5836 degrees of freedom
## Multiple R-squared:  0.3157, Adjusted R-squared:  0.314
## F-statistic: 192.3 on 14 and 5836 DF,  p-value: < 2.2e-16

```

Podem observar que la millora obtinguda al afegir **population.discr** i **vehicle.type** és irrisoria i que per tant aquestes dues variables no afecten el resultat, almenys de forma lineal.

2.7 Model d'arbre de regressió per preveure el preu diari d'un vehicle

Comencem triant el subconjunt d'entrenament i el de prova. Nosaltres crearem dos conjunts de dades directament amb un rang.

```

# Creem el conjunt d'entrenament i el de prova
set.seed(555)
indexes = sample(1:nrow(ds), size = floor((2/3) * nrow(ds)))
train <- ds[indexes, ]
test <- ds[-indexes, ]

```

Després d'una extracció aleatòria de casos cal realitzar una anàlisi de dades mínim per assegurar-nos de no obtenir valors esbiaixats pels valors que conté cada mostra.

```

# Verifiquem les dimensions del conjunt d'entrenament
dim(train)

```

```

## [1] 3900   21
# Verifiquem les dimensions del conjunt de prova
dim(test)

```

```

## [1] 1951   21
# Obtenim l'atribut de classe de la resta
trainY <- train[, c("rate.daily")]
testY <- test[, c("rate.daily")]

```

Model d'Arbre de regressió

A continuació ens proposem generar un model d'arbre de regressió que expliqui la variable *rate.daily* en funció de les característiques del vehicle.

```

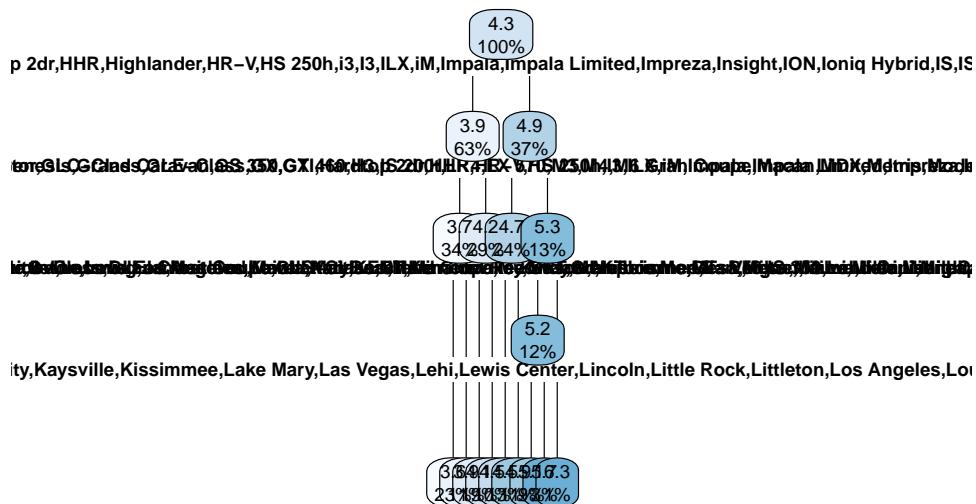
# Generem el model d'arbre
model_cart_rate <- rpart(rate.daily ~ fuelType + location.city +
  location.state + vehicle.make + vehicle.model + vehicle.type,
  method = "anova", data = train)

# Mostrem un resum de la generació de l'arbre
# summary(model_cart)

```

```
# printcp(model_cart_rate)

# Mostrem en un gràfic l'arbre obtingut
rpart.plot(model_cart_rate, cex = 0.6)
```



Un cop generat el model, podem comprovar la seva qualitat predint la classe per a les dades de prova que hem reservat al principi.

```
predicted_model_cart_rate <- predict(model_cart_rate, test, type = "vector")

# test RMSE
rmse_cart_rate <- rmse(predicted_model_cart_rate, testY)

# test MAE
mae_cart_rate <- mae(predicted_model_cart_rate, testY)

print(sprintf("RMSE: %.4f", rmse_cart_rate))

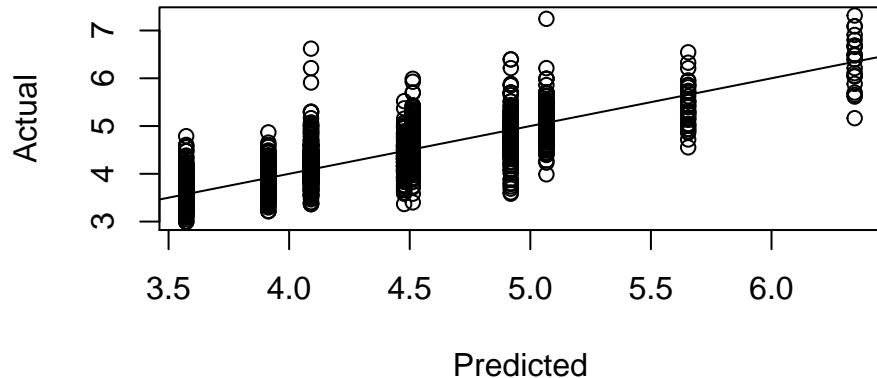
## [1] "RMSE: 0.3826"

print(sprintf("MAE: %.4f", rmse_cart_rate))

## [1] "MAE: 0.3826"
```

Examinem els valors predicts envers els valors reals.

```
plot(predicted_model_cart_rate, testY, xlab = "Predicted", ylab = "Actual")
abline(0, 1)
```



Examinem la importància de cada una dels atributs predictors en el model.

```
model_cart_rate$variable.importance
```

```
##   vehicle.model    vehicle.make   location.city      fuelType location.state
##     1211.96274      636.65430      545.53745      239.29902      127.07061
##   vehicle.type
##     49.25973
```

A partir del resultat obtingut podem veure que l'atribut que més influeix en el preu del lloguer dels vehicles és el model de cotxe.

Model d'Arbre de regressió amb variables més simples

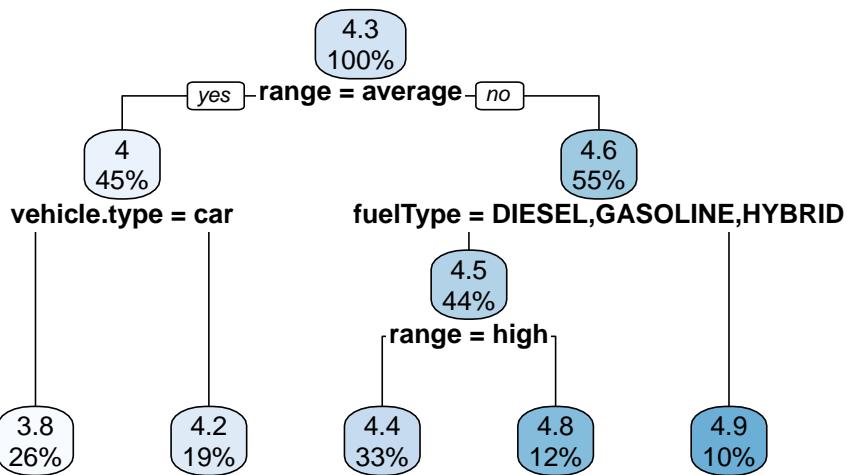
A continuació ens proposem generar un model d'arbre de regressió que expliqui la variable *rate.daily* en funció de les característiques del vehicle.

```
# Generem el model d'arbre
model_cart_rate_2 <- rpart(rate.daily ~ fuelType + range + population.discr +
  vehicle.type, method = "anova", data = train)

# Mostrem un resum de la generació de l'arbre
# summary(model_cart)

# printcp(model_cart_rate)

# Mostrem en un gràfic l'arbre obtingut
rpart.plot(model_cart_rate_2, cex = 0.8)
```



Un cop generat el model, podem comprovar la seva qualitat predint la classe per a les dades de prova que hem reservat al principi.

```

predicted_model_cart_rate_2 <- predict(model_cart_rate_2, test,
                                         type = "vector")

# test RMSE
rmse_cart_rate_2 <- rmse(predicted_model_cart_rate_2, testY)

# test MAE
mae_cart_rate_2 <- mae(predicted_model_cart_rate_2, testY)

print(sprintf("RMSE: %.4f", rmse_cart_rate_2))

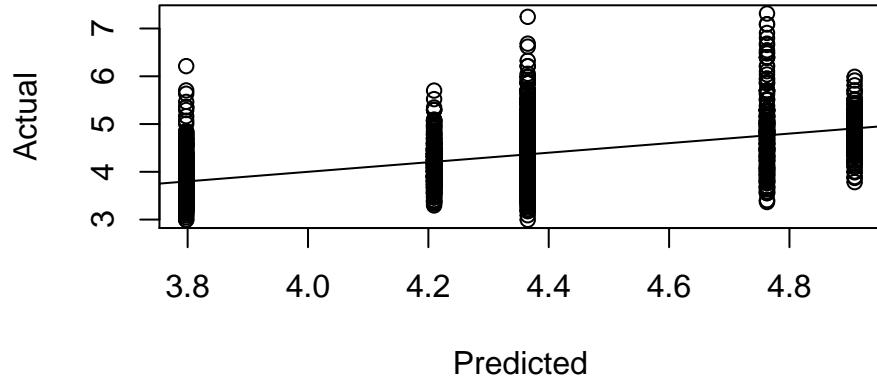
## [1] "RMSE: 0.5337"
print(sprintf("MAE: %.4f", rmse_cart_rate_2))

## [1] "MAE: 0.5337"
  
```

Examinem els valors predictius envers els valors reals.

```

plot(predicted_model_cart_rate_2, testY, xlab = "Predicted",
      ylab = "Actual")
abline(0, 1)
  
```



Examinem la importància de cada una dels atributs predictors en el model.

```
model_cart_rate_2$variable.importance
```

```
##          range      vehicle.type      fuelType population.discr
## 377.980443     84.061997    72.969652      5.756592
```

A partir del resultat obtingut podem veure que l'atribut que més influeix en el preu del lloguer dels vehicles és la gamma del cotxe **range**.

2.8 Model d'arbre de classificació per preveure si un vehicle serà llogat

A continuació ens proposem generar un model d'arbre de classificació que expliqui la variable *rent* en funció de les característiques del vehicle. És a dir que ens permeti donades les característiques d'un vehicle predir si aquest serà llogat o no.

Model d'Arbre de classificació

Obtenim la classe objectiu

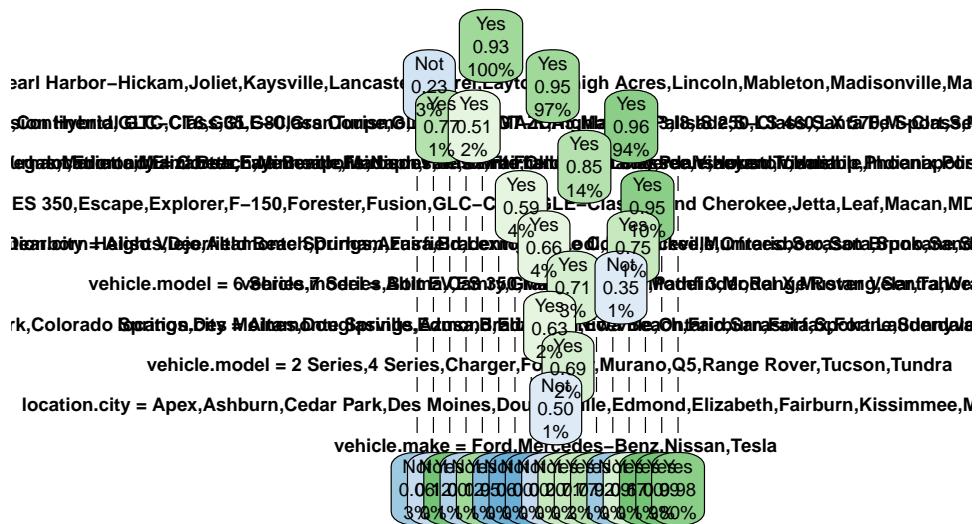
```
# Obtenim l'atribut de classe de la resta
trainY <- train[, c("rent")]
testY <- test[, c("rent")]
```

```
# Generem el model d'arbre
model_cart_rent <- rpart(rent ~ fuelType + location.city + location.state +
  vehicle.make + vehicle.model + vehicle.type, method = "class",
  data = train)

# Mostrem un resum de la generació de l'arbre
# summary(model_cart)

# printcp(model_cart_rent)

# Mostrem en un gràfic l'arbre obtingut
rpart.plot(model_cart_rent, cex = 0.6)
```



Una cop generat el model, podem comprovar la seva qualitat predint la classe per a les dades de prova que hem reservat al principi.

```
predicted_model_cart_rent <- predict(model_cart_rent, test, type = "class")

print(sprintf("La precisió de l'arbre és: %.4f %", 100 * sum(predicted_model_cart_rent == testY)/length(predicted_model_cart_rent)))

## [1] "La precisió de l'arbre és: 89.9539 %"
```

Quan hi ha poques classes, la qualitat de la predicció també es pot analitzar mitjançant una matriu de confusió que identifica els tipus d'errors commesos.

```
mat_conf_cart_rent <- table(testY, Predicted = predicted_model_cart_rent)

mat_conf_cart_rent
```

```
##      Predicted
## testY  Not    Yes
##    Not     6   134
##    Yes    62  1749
```

Una altra manera de calcular el percentatge de registres correctament classificats és utilitzant la matriu de confusió:

```
porcentaje_correct <- 100 * sum(diag(mat_conf_cart_rent))/sum(mat_conf_cart_rent)
print(sprintf("El % de registres correctament classificats és: %.4f %",
            porcentaje_correct))
```

```
## [1] "El % de registres correctament classificats és: 89.9539 %"
```

A més, tenim a la nostra disposició el paquet gmodels per a obtenir informació més completa

```
CrossTable(testY, predicted_model_cart_rent, prop.chisq = FALSE,
           prop.c = FALSE, prop.r = FALSE, dnn = c("Reality", "Prediction"))
```

```
##
##      Cell Contents
## |-----|
```

```

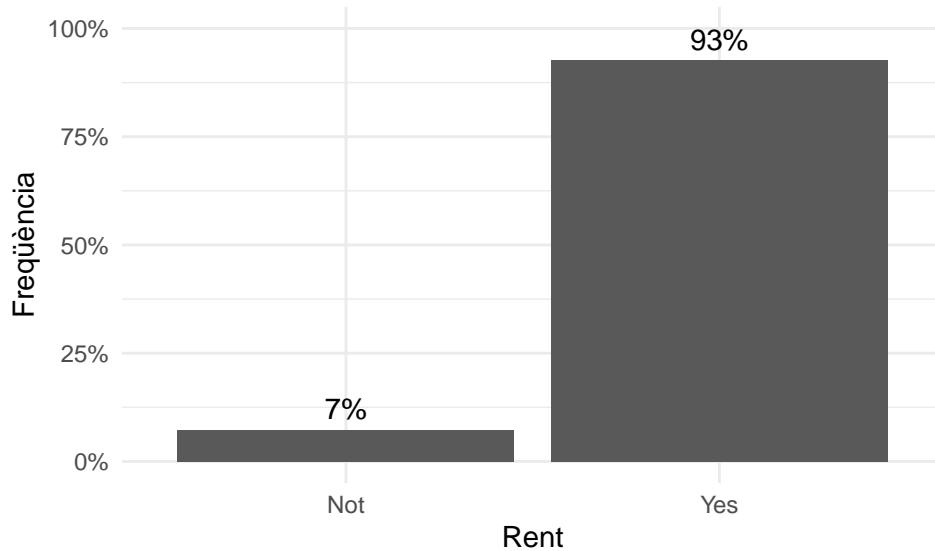
## | N |
## |   N / Table Total |
## |-----|
## 
## 
## Total Observations in Table: 1951
## 
## 
##           | Prediction
## Reality |      Not |      Yes | Row Total |
## -----|-----|-----|-----|
##       Not |      6 |    134 |     140 |
##       | 0.003 | 0.069 |      |
## -----|-----|-----|-----|
##       Yes |     62 | 1749 | 1811 |
##       | 0.032 | 0.896 |      |
## -----|-----|-----|-----|
## Column Total |     68 | 1883 | 1951 |
## -----|-----|-----|-----|
## 
## 
```

Durant tot l'estudi previ em deixat de banda un fet important que es va veure durant la fase d'anàlisis del joc de dades. Si recordem el nombre d'observacions pertanyents a una classe respecte a l'altre és força diferent per a l'atribut objectiu 'rent'.

```

# Calculem histogrammes de les variables de classe
ggplot(data = ds, aes(x = rent)) + geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(y = ((..count..)/sum(..count..))), label = scales::percent(..count..)/sum(..count..)),
            stat = "count", vjust = -0.5) + scale_y_continuous(labels = scales::percent,
            limits = c(0, 1)) + xlab("Rent") + ylab("Freqüència")

```



Les classificacions desequilibrades suposen un desafiament per a la generació de models de predicción, ja que la majoria dels algorismes utilitzats per a la classificació van ser dissenyats entorn l'assumpció d'un nombre igual d'observacions per a cada classe. Això produeix que els models obtinguts presentin un rendiment predictiu deficient, en concret per a la classe minoritària. A més això sol ser un problema perquè normalment, la classe minoritària és la més important i per tant el model és més sensible als errors de classificació de la

classe minoritària que la classe majoritària.

Per tal de reduir els efectes causats per aquest fet proposem aplicar diferents mètodes de submostreig abans de generar els models.

```
set.seed(2)

# Escollim els atributs sobre els quals treballarem
train.data <- train

train.data[, "Class"] <- train["rent"]

test.data <- test

test.data[, "Class"] <- test["rent"]

# Construïm diferents mostres a partir del conjunt de dades
# original

# Down sampling
down_train <- downSample(x = train.data[, -ncol(train.data)],
                           y = train.data$Class)

# Up Sampling
up_train <- upSample(x = train.data[, -ncol(train.data)], y = train.data$Class)

# Mix up/down sampling SMOTE
smote_train <- SMOTE(Class ~ ., data = train.data)

# Mostrem el nombre d'observacions pertanyents a cada una de
# les classes després de la generació de mostres Conjunt
# original
table(train.data$Class)

## 
## Not Yes
## 291 3609

# Down sampling
table(down_train$Class)

## 
## Not Yes
## 291 291

# Up sampling
table(up_train$Class)

## 
## Not Yes
## 3609 3609

# Smote sampling
table(smote_train$Class)

## 
## Not Yes
## 873 1164
```

Un cop obtingudes diferents mostres del conjunt de dades originals amb submostreig anem a realitzar l'entrenament d'un model d'arbre de decisió (CART) per cada un dels subconjunts.

```
# Model training - CART
set.seed(4)
down_outside <- rpart(rent ~ fuelType + location.city + location.state +
  vehicle.make + vehicle.model + vehicle.type, method = "class",
  data = down_train)

set.seed(5)
up_outside <- rpart(rent ~ fuelType + location.city + location.state +
  vehicle.make + vehicle.model + vehicle.type, method = "class",
  data = up_train)

set.seed(6)
smote_outside <- rpart(rent ~ fuelType + location.city + location.state +
  vehicle.make + vehicle.model + vehicle.type, method = "class",
  data = smote_train)
```

I per acabar realitzem les corresponents prediccions en el conjunt de test

```
predicted_model_cart_rent <- predict(down_outside, test.data,
  type = "class")

divisor = length(predicted_model_cart_rent)
print(sprintf("La precisió de l'arbre amb downsampling és: %.4f %%",
  100 * sum(predicted_model_cart_rent == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb downsampling és: 60.4305 %"

predicted_model_cart_rent <- predict(up_outside, test.data, type = "class")

divisor = length(predicted_model_cart_rent)
print(sprintf("La precisió de l'arbre amb upsampling és: %.4f %%",
  100 * sum(predicted_model_cart_rent == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb upsampling és: 73.4495 %"

predicted_model_cart_rent <- predict(smote_outside, test.data,
  type = "class")

divisor = length(predicted_model_cart_rent)
print(sprintf("La precisió de l'arbre amb smote sampling és: %.4f %%",
  100 * sum(predicted_model_cart_rent == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb smote sampling és: 80.4716 %"
```

Tot i que podem veure que la precisió ha disminuit, pels models obtinguts després de realitzar un submostreig per tractar el conjunt desequilibrat, podem veure com aquests nous models ofereixen una millor resposta a la classe minoritària, la que recull el no llogar.

Model d'Arbre de classificació simplificat

Obtenim la classe objectiu

```
# Obtenim l'atribut de classe de la resta
trainY <- train[, c("rent")]
testY <- test[, c("rent")]
```

```

# Generem el model d'arbre
model_cart_rent_2 <- rpart(rent ~ fuelType + population.discr +
    range + vehicle.type, method = "class", data = train)

# Mostrem un resum de la generació de l'arbre
# summary(model_cart)

# printcp(model_cart_rent)

# Mostrem en un gràfic l'arbre obtingut
rpart.plot(model_cart_rent_2)

```

Yes
0.93
100%

Un cop generat el model, podem comprovar la seva qualitat predint la classe per a les dades de prova que hem reservat al principi.

```

predicted_model_cart_rent_2 <- predict(model_cart_rent_2, test,
    type = "class")

print(sprintf("La precisió de l'arbre és: %.4f %%", 100 * sum(predicted_model_cart_rent_2 ==
    testY)/length(predicted_model_cart_rent_2)))

```

[1] "La precisió de l'arbre és: 92.8242 %"

Quan hi ha poques classes, la qualitat de la predicció també es pot analitzar mitjançant una matriu de confusió que identifica els tipus d'errors comesos.

```

mat_conf_cart_rent_2 <- table(testY, Predicted = predicted_model_cart_rent_2)

mat_conf_cart_rent_2

```

```

##      Predicted
## testY  Not  Yes
##    Not      0 140
##    Yes      0 1811

```

Una altra manera de calcular el percentatge de registres correctament classificats és utilitzant la matriu de confusió:

```

porcentaje_correct_2 <- 100 * sum(diag(mat_conf_cart_rent_2))/sum(mat_conf_cart_rent_2)
print(sprintf("El % de registres correctament classificats és: %.4f %",
    porcentaje_correct_2))

```

```
## [1] "El % de registres correctament classificats és: 92.8242 %"
```

A més, tenim a la nostra disposició el paquet gmodels per a obtenir informació més completa

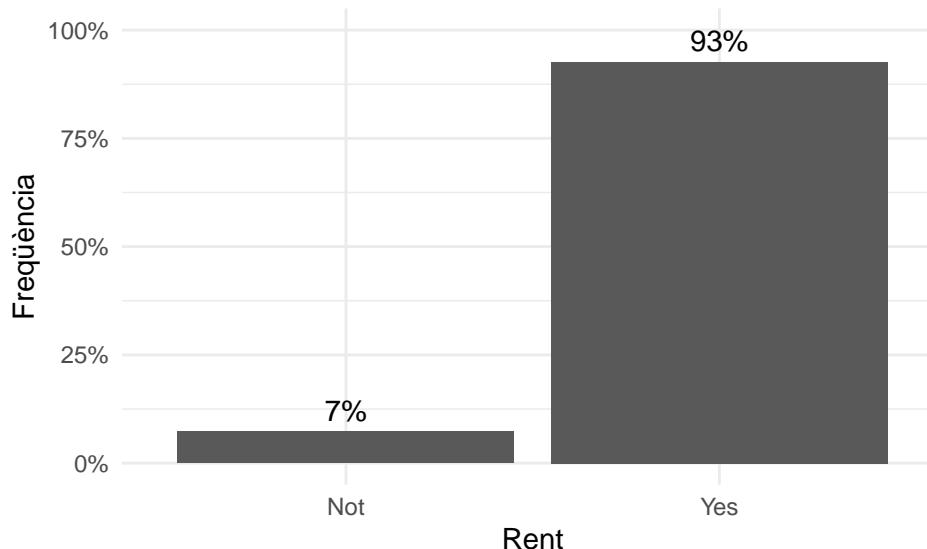
```
CrossTable(testY, predicted_model_cart_rent_2, prop.chisq = FALSE,
    prop.c = FALSE, prop.r = FALSE, dnn = c("Reality", "Prediction"))
```

```

##
##
##      Cell Contents
## |-----|
## |                   N |
## |           N / Table Total |
## |-----|
##
##
## Total Observations in Table:  1951
##
##
##          | predicted_model_cart_rent_2
## testY |       Yes | Row Total |
## -----|-----|-----|
##     Not |     140 |     140 |
##         |     0.072 |     |
## -----|-----|-----|
##     Yes |   1811 |   1811 |
##         |     0.928 |     |
## -----|-----|-----|
## Column Total | 1951 | 1951 |
## -----|-----|-----|
##
##
```

Durant tot l'estudi previ em deixat de banda un fet important que es va veure durant la fase d'anàlisis del joc de dades. Si recordem el nombre d'observacions pertanyents a una classe respecte a l'altre és força diferent per a l'atribut objectiu 'rent'.

```
# Calculem histogrammes de les variables de classe
ggplot(data = ds, aes(x = rent)) + geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(y = ((..count..)/sum(..count..))), label = scales::percent(..count..)/sum(..count..)),
  stat = "count", vjust = -0.5) + scale_y_continuous(labels = scales::percent,
  limits = c(0, 1)) + xlab("Rent") + ylab("Freqüència")
```



Les classificacions desequilibrades suposen un desafiatament per a la generació de models de predicció, ja que la majoria dels algorismes utilitzats per a la classificació van ser dissenyats entorn l'assumpció d'un nombre igual d'observacions per a cada classe. Això produeix que els models obtinguts presentin un rendiment predictiu deficient, en concret per a la classe minoritària. A més això sol ser un problema perquè normalment, la classe minoritària és la més important i per tant el model és més sensible als errors de classificació de la classe minoritària que la classe majoritària.

Per tal de reduir els efectes causats per aquest fet proposem aplicar diferents mètodes de submostreig abans de generar els models.

```
set.seed(2)

# Escollim els atributs sobre els quals treballarem
train.data <- train

train.data[, "Class"] <- train["rent"]

test.data <- test

test.data[, "Class"] <- test["rent"]

# Construïm diferents mostres a partir del conjunt de dades
# original

# Down sampling
down_train <- downSample(x = train.data[, -ncol(train.data)],
                           y = train.data$Class)

# Up Sampling
up_train <- upSample(x = train.data[, -ncol(train.data)], y = train.data$Class)

# Mix up/down sampling SMOTE
smote_train <- SMOTE(Class ~ ., data = train.data)

# Mostrem el nombre d'observacions pertanyents a cada una de
# les classes després de la generació de mostres Conjunt
```

```

# original
table(train.data$Class)

##
## Not Yes
## 291 3609

# Down sampling
table(down_train$Class)

##
## Not Yes
## 291 291

# Up sampling
table(up_train$Class)

##
## Not Yes
## 3609 3609

# Smote sampling
table(smote_train$Class)

##
## Not Yes
## 873 1164

```

Un cop obtingudes diferents mostres del conjunt de dades originals amb submostreig anem a realitzar l'entrenament d'un model d'arbre de decisió (CART) per cada un dels subconjunts.

```

# Model training - CART
set.seed(4)
down_outside <- rpart(rent ~ fuelType + population.dscr + range +
    vehicle.type, method = "class", data = down_train)

set.seed(5)
up_outside <- rpart(rent ~ fuelType + population.dscr + range +
    vehicle.type, method = "class", data = up_train)

set.seed(6)
smote_outside <- rpart(rent ~ fuelType + population.dscr + range +
    vehicle.type, method = "class", data = smote_train)

```

I per acabar realitzem les corresponents prediccions en el conjunt de test

```

predicted_model_cart_rent_2 <- predict(down_outside, test.data,
    type = "class")

divisor = length(predicted_model_cart_rent_2)
print(sprintf("La precisió de l'arbre amb downsampling és: %.4f %%",
    100 * sum(predicted_model_cart_rent_2 == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb downsampling és: 54.5361 %"

predicted_model_cart_rent_2 <- predict(up_outside, test.data,
    type = "class")

divisor = length(predicted_model_cart_rent_2)

```

```

print(sprintf("La precisió de l'arbre amb upsampling és: %.4f %%",
             100 * sum(predicted_model_cart_rent_2 == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb upsampling és: 68.7340 %"

predicted_model_cart_rent_2 <- predict(smote_outside, test.data,
                                         type = "class")

divisor = length(predicted_model_cart_rent_2)
print(sprintf("La precisió de l'arbre amb smote sampling és: %.4f %%",
             100 * sum(predicted_model_cart_rent_2 == test.data[, "Class"])/divisor))

## [1] "La precisió de l'arbre amb smote sampling és: 77.2424 %"

```

Tot i que podem veure que la precisió ha disminuit, pels models obtinguts després de realitzar un submostreig per tractar el conjunt desequilibrat, podem veure com aquests nous models ofereixen una millor resposta a la classe minoritària, la que recull el no llogar.

2.9 Conclusions

A l'inici d'aquest estudi ens proposavem realitzar una anàlisi detallada dels atributs propis del sector del lloguer de vehicles per tal d'extreuren nou coneixement i que aquest pugui aportar valor.

Per això hem començat el nostre estudi amb les tasques típiques d'un procés de mineria o anàlica de dades, incloent la neteja de dades, la imputació de valors nuls, el tractament de valors extrems, la transformació d'alguns atributs i la creació d'atributs nous. També hem realitzat una primera inspecció visual de les dades on ja s'han pogut determinar certes relacions entre variables. Com per exemple que el preu diari de lloguer depén de la marca i el tipus de vehicle, el combustible que utilitza o la seva ubicació.

Tot seguit hem dut a terme diferents contrastos d'hipòtesis que ens han permés identificar propietats interessants subjacents en les mostres que puguin ser inferides respecte a la població. Però primerament hem avaluat la normalitat dels principals atributs. A partir dels diferents resultats obtinguts podem conoure que:

- Que els cotxes elèctrics tenen un lloguer més elevat que els de benzina, amb un nivell de confiança del 95%.
- Que la proporció de furgonetes de lloguer és més petita que la d'utilitàries, amb un nivell de confiança del 95%.
- Que el preu diari del lloguer del vehicle depén del tipus de vehicle, amb un nivell de confiança del 95%.
- Que el preu diari del lloguer del vehicle depén del tipus de combustible, amb un nivell de confiança del 95%.

A continuació hem generat un model per intentar preveure el preu per dia d'un automòbil de lloguer, a partir de les seves característiques. Tant mitjançant un model basat en regressió lineal múltiple com amb un arbre de regressió. Usant aquest dos tipus de models no s'ha pogut obtenir una predicción que poguem calificar com a bona. No obstant s'ha pogut determinar els atributs més influents en el preu diari del lloguer dels vehicles. Aquests són el model, la marca del vehicle i la seva localització.

Finalment hem generat també un model que permet preveure, donades les característiques d'un vehicle de lloguer, si aquest serà llogat o no mitjançant un arbre de decisió per a classificació.

3 Bibliografia

(D. Liviano Solís, n.d.; Edwin de Jonge 2013; Bàguena, n.d.; Cichosz 2015)

Bàguena, J. Gibergans. n.d. “Regressió lineal múltiple.”

Cichosz, Paweł. 2015. “Data Mining Algorithms: Explained Using R.”

D. Liviano Solís, M. Pujol Jover. n.d. “Models de regressió i anàlisi multivariant amb R-Commander.”

Edwin de Jonge, Mark van der Loo. 2013. “An introduction to data cleaning with R.” https://cran.r-project.org/doc/contrib/de_Jonge+van_der_Loo-Introduction_to_data_cleaning_with_R.pdf.