

# Estadística Descriptiva

## Análisis exploratorio de datos categóricos y binarios

Para estudiar numéricamente los datos categóricos, es suficiente con la proporción, porcentajes y frecuencias en la que ocurre cada una de las categorías. El siguiente ejemplo muestra el porcentaje de vuelos cancelados según la causa en el aeropuerto de Dallas/fort Worth desde 2010.

Cargamos en primer lugar algunas librerías que nos serán de utilidad:

```
#library(ascii)
library(stats)
library(corrplot)

## corrplot 0.92 loaded
library(descr)
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##   filter, lag
## The following objects are masked from 'package:base':
##   intersect, setdiff, setequal, union
library(ggplot2)
library(hexbin)
library(matrixStats)

##
## Attaching package: 'matrixStats'
## The following object is masked from 'package:dplyr':
##   count
library(tidyr)
library(vioplot)

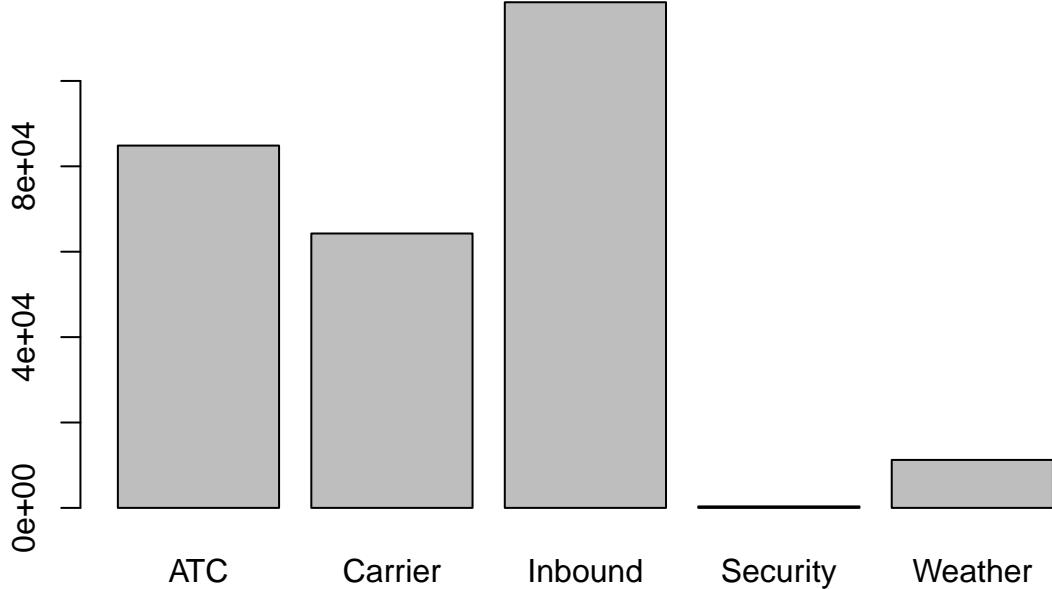
## Loading required package: sm
## Package 'sm', version 2.2-5.7: type help(sm) for summary information
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##   as.Date, as.Date.numeric
```

Veamos pues cómo cargar el conjunto de datos, ver la dimensión, el nombre de las variables:

```
dfw <- read.table(file="./dades/dfw_airline.txt", header=TRUE, fileEncoding = "WINDOWS-1252")
dfw
dim(dfw)
names(dfw)
```

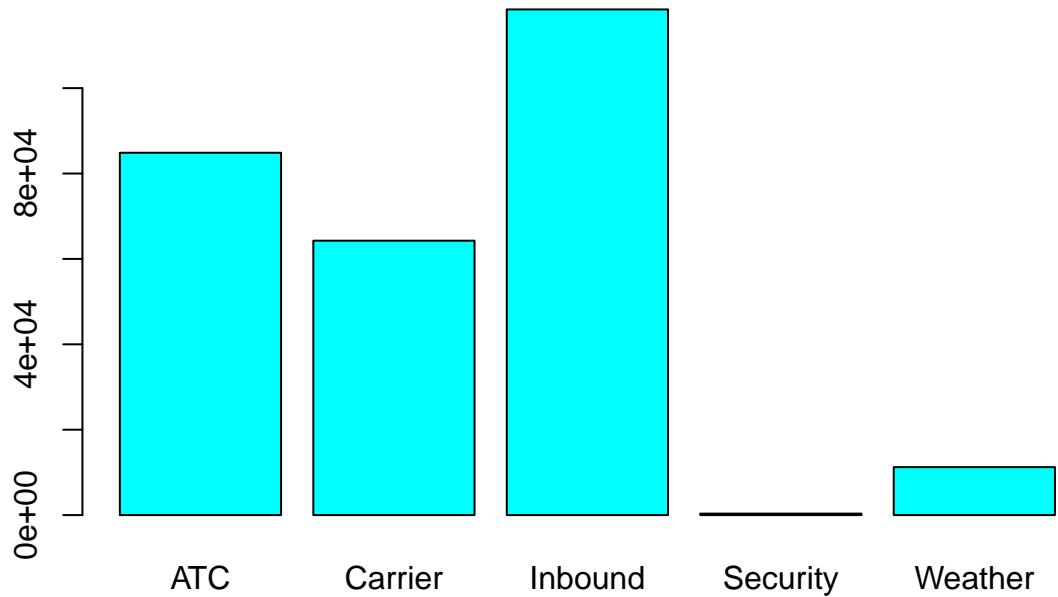
A continuación creamos un nuevo banco de datos *tabla* en la que guardamos las frecuencias y las ordenamos en orden alfabético:

```
frecuencia<-dfw[["Frecuencia"]]
tipo<-dfw[["Tipo"]]
tapply(frecuencia, tipo,sum)
tabla<-tapply(frecuencia, tipo, sum)
barplot(tabla)
```



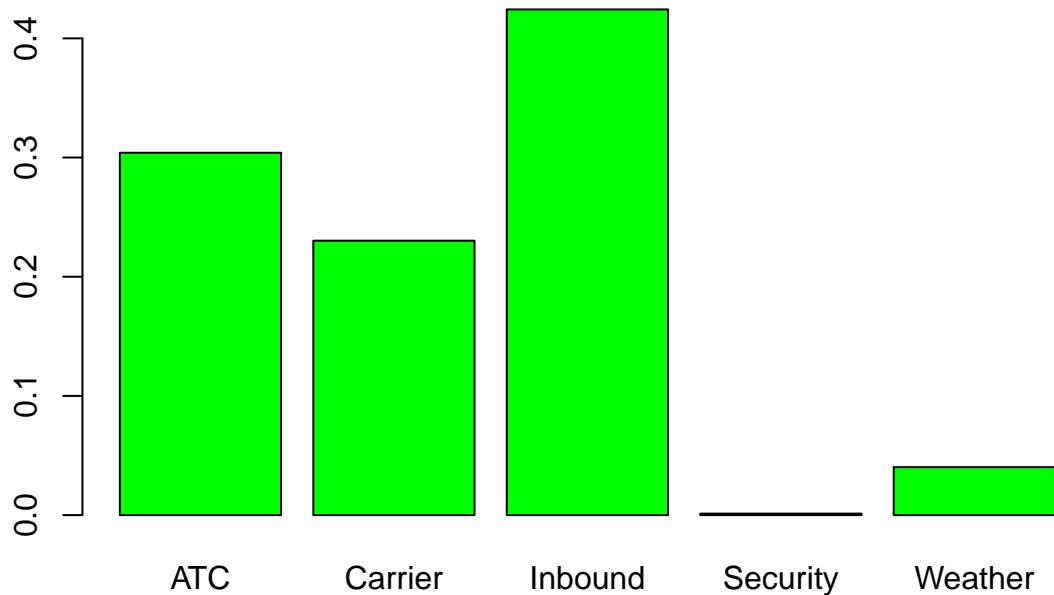
```
barplot(tabla, col="cyan", main="Tipo de retraso en los vuelos del aeropuerto de Dallas")
```

## Tipo de retraso en los vuelos del aeropuerto de Dallas



```
barplot(tabla/sum(frecuencia), col="green", main="Tipo de retraso en los vuelos del aeropuerto de Dallas")
```

## Tipo de retraso en los vuelos del aeropuerto de Dallas



Si queremos reordenar los niveles de un factor, partimos de una variable que está resordenada:

```
# Create a factor with the wrong order of levels
sizes <- factor(c("small", "large", "large", "small", "medium"))
sizes

## [1] small  large  large  small  medium
## Levels: large medium small
```

los niveles se pueden especificar como:

```
sizes <- factor(sizes, levels = c("small", "medium", "large"))
sizes

## [1] small  large  large  small  medium
## Levels: small medium large
```

Para reordenar los niveles utilizar el comando *ordered*:

```
sizes <- ordered(c("small", "large", "large", "small", "medium"))
sizes <- ordered(sizes, levels = c("small", "medium", "large"))
sizes

## [1] small  large  large  small  medium
## Levels: small < medium < large
```

## Análisis exploratorio de datos cuantitativos

El banco de datos que usaremos para este apartado es el del fichero *state.csv*, que contiene los datos de población y tasa de asesinatos (número de asesinatos por cada 100000 personas y año) en los diferentes

estados de USA.

En primer lugar leemos el fichero de datos:

```
state<-read.csv(file="./dades/state.csv", header=T, sep = ",", dec = ".", fileEncoding = "WINDOWS-1252")
state
```

Si queremos conocer el tipo de objeto, la dimensión del banco de datos, el nombre de las variables (columnas), una pequeña vista de los datos con información relevante (esa información mejor vista), estas son las funciones del R que podemos usar respectivamente:

```
# tipo de objeto  
class(state)
```

```
## [1] "data.frame"
```

```
# dimension del banco de datos  
dim(state)
```

```
## [1] 50 4
```

```
# nombres de las columnas/variables  
names(state)
```

```
## [1] "State"           "Population"      "Murder.Rate"    "Abbreviation"
```

```
# pequena vista de los datos con informacion relevante  
str(state)
```

```
## 'data.frame': 50 obs. of 4 variables:
```

```
## $ State          : chr "Alabama" "Alaska" "Arizona" "Arkansas" ...
```

## \$ Population : int 4779736 710231 6392017 2915918 37253956 502

```
## $ Murder.Rate : num 5.7 5.6 4.7 5.6 4.4 2.8
```

*# igual que la funcion anterior pero un poco mejor*

6

## Rows: 50

```
## Columns: 4
## $ State      <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "California", ~
## $ Population <int> 4779736, 710231, 6392017, 2915918, 37253956, 5029196, 357~
## $ Murder.Rate <dbl> 5.7, 5.6, 4.7, 5.6, 4.4, 2.8, 2.4, 5.8, 5.8, 5.7, 1.8, 2.~
## $ Abbreviation <chr> "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA", ~
```

Las siguientes instrucciones nos permiten obtener parte del banco de datos:

```
#imprime las primeras 10 filas del banco de datos
head(state, n=10)
```

##	State	Population	Murder.Rate	Abbreviation
## 1	Alabama	4779736	5.7	AL
## 2	Alaska	710231	5.6	AK
## 3	Arizona	6392017	4.7	AZ
## 4	Arkansas	2915918	5.6	AR
## 5	California	37253956	4.4	CA
## 6	Colorado	5029196	2.8	CO
## 7	Connecticut	3574097	2.4	CT
## 8	Delaware	897934	5.8	DE
## 9	Florida	18801310	5.8	FL
## 10	Georgia	9687653	5.7	GA

```

# imprime las ultimas filas del banco de datos
tail(state, n=5)

##           State Population Murder.Rate Abbreviation
## 46      Virginia     8001024       4.1          VA
## 47    Washington    6724540       2.5          WA
## 48 West Virginia   1852994       4.0          WV
## 49   Wisconsin    5686986       2.9          WI
## 50      Wyoming     563626        2.7          WY

state[1,3]

## [1] 5.7
state[1, ]

##           State Population Murder.Rate Abbreviation
## 1 Alabama     4779736       5.7          AL

state[, 1]

## [1] "Alabama"      "Alaska"       "Arizona"      "Arkansas"
## [5] "California"   "Colorado"     "Connecticut"  "Delaware"
## [9] "Florida"       "Georgia"      "Hawaii"       "Idaho"
## [13] "Illinois"     "Indiana"     "Iowa"         "Kansas"
## [17] "Kentucky"     "Louisiana"   "Maine"        "Maryland"
## [21] "Massachusetts" "Michigan"    "Minnesota"   "Mississippi"
## [25] "Missouri"     "Montana"     "Nebraska"    "Nevada"
## [29] "New Hampshire" "New Jersey"  "New Mexico"  "New York"
## [33] "North Carolina" "North Dakota" "Ohio"        "Oklahoma"
## [37] "Oregon"        "Pennsylvania" "Rhode Island" "South Carolina"
## [41] "South Dakota"   "Tennessee"   "Texas"       "Utah"
## [45] "Vermont"       "Virginia"    "Washington"  "West Virginia"
## [49] "Wisconsin"     "Wyoming"    
```

## Medidas de Localización

Una parte importante de la exploración de datos cuantitativos es obtener valores que resuman la parte central en la que se sitúan los datos, que representaremos por  $\{x_1, x_2, \dots, x_n\}$ .

Media, media recortada, mediana, media ponderada y mediana ponderada muestrales:

```

mean(state[["Population"]])

## [1] 6162876
mean(state[["Population"]], trim=0.1)

## [1] 4783697
median(state[["Population"]])

## [1] 4436370
weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])

## [1] 4.445834
library(matrixStats)
weightedMedian(state[["Murder.Rate"]], w=state[["Population"]]) 
```

```
## [1] 4.4
```

## Medidas de dispersión

Cálculo del mínimo, máximo, rango, desviación típica ( $sd$ ), desviación media absoluta que se define como:

$$mad = \frac{\sum_{i=1}^{i=n} |x_i - \bar{x}|}{n}$$

, el rango intercuartílico ( $IQR = Q_3 - Q_1$ ) y cuantiles muestrales.

```
min(state[["Population"]])
```

```
## [1] 563626
```

```
max(state[["Population"]])
```

```
## [1] 37253956
```

```
range(state[["Population"]])
```

```
## [1] 563626 37253956
```

```
sd(state[["Population"]])
```

```
## [1] 6848235
```

```
IQR(state[["Population"]])
```

```
## [1] 4847308
```

```
mad(state[["Population"]])
```

```
## [1] 3849870
```

```
quantile(state[["Murder.Rate"]], p=c(0.05, 0.25, 0.50, 0.75, 0.95))
```

```
##      5%    25%    50%    75%   95%
```

```
## 1.600 2.425 4.000 5.550 6.510
```

Resumen de la descripción de los datos

```
summary(state)
```

```
##           State          Population       Murder.Rate      Abbreviation
## Length:50          Min.   : 563626   Min.   : 0.900   Length:50
## Class :character  1st Qu.:1833004  1st Qu.: 2.425   Class :character
## Mode  :character  Median :4436370  Median : 4.000   Mode  :character
##                   Mean   :6162876  Mean   : 4.066
##                   3rd Qu.:6680312  3rd Qu.: 5.550
##                   Max.   :37253956  Max.   :10.300
```

## Visualización de los datos cuantitativos

Vamos a ver la tabla de frecuencias, histograma, diagrama de tallo y hojas y diagrama de cajas:

```
breaks<- seq(from=min(state[["Population"]]), to=max(state[["Population"]]), length=11)
pop_freq <- cut(state[["Population"]], breaks, right=TRUE, include.lowest=TRUE)
table(pop_freq)
```

```
## pop_freq
## [5.64e+05,4.23e+06]  (4.23e+06,7.9e+06]  (7.9e+06,1.16e+07] (1.16e+07,1.52e+07]
```

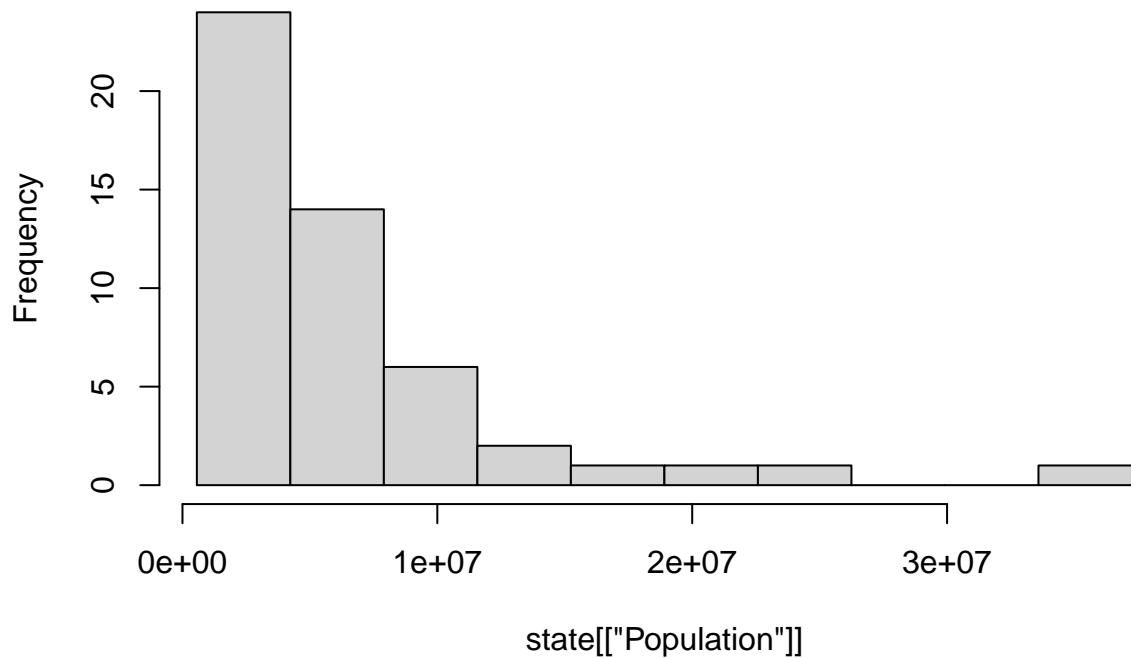
```

##          24          14          6          2
## (1.52e+07,1.89e+07] (1.89e+07,2.26e+07] (2.26e+07,2.62e+07] (2.62e+07,2.99e+07]
##          1           1           1           0
## (2.99e+07,3.36e+07] (3.36e+07,3.73e+07]
##          0           1

hist(state[["Population"]], breaks=breaks)

```

**Histogram of state[["Population"]]**

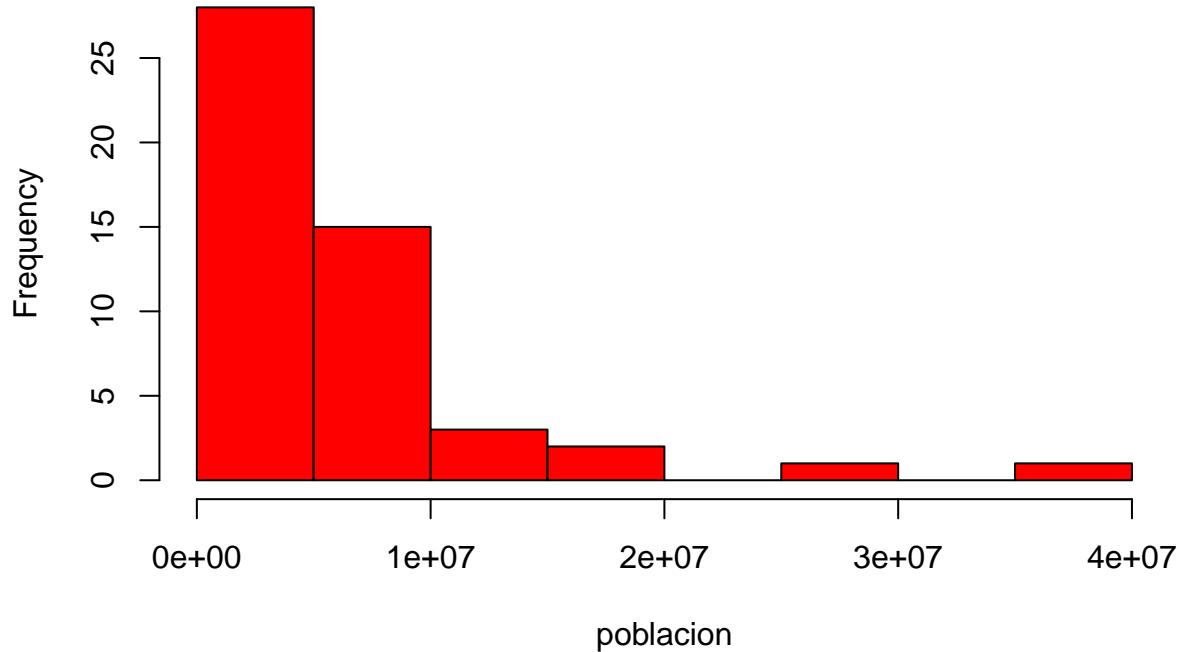


```

poblacion<-state[["Population"]]
hist(poblacion, col="red", breaks=12, freq=TRUE, main="Población de los estados de USA")

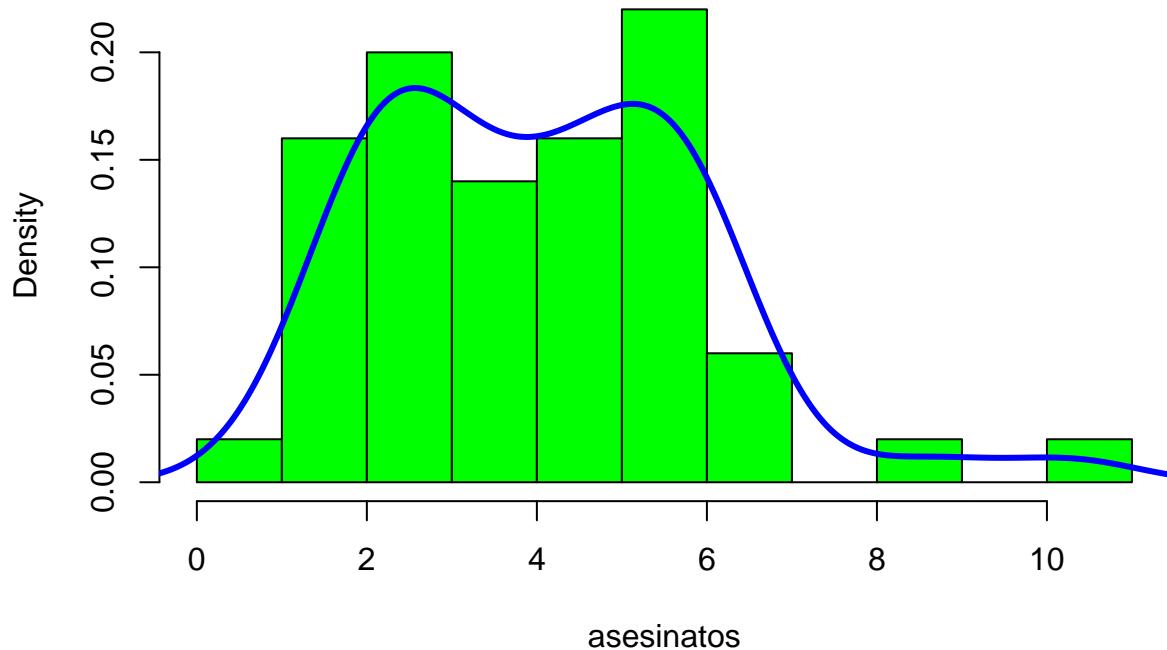
```

## Población de los estados de USA



```
asesinatos<-state[["Murder.Rate"]]
hist(asesinatos, col="green", breaks=12, freq=FALSE, main="Número de asesinatos por 100 mil habitantes")
lines(density(asesinatos), lwd=3, col="blue")
```

## Número de asesinatos por 100 mil habitantes

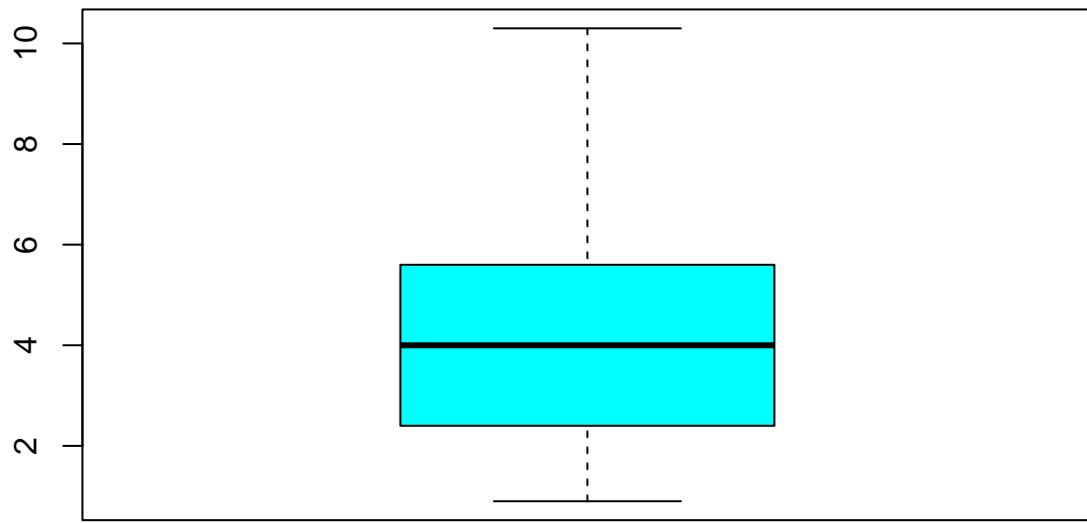


```
stem(asesinatos)

##
##      The decimal point is at the |
##
##      0 | 9
##      1 | 66689
##      2 | 000334457899
##      3 | 011669
##      4 | 001445788
##      5 | 01346677788
##      6 | 0146
##      7 |
##      8 | 6
##      9 |
##     10 | 3

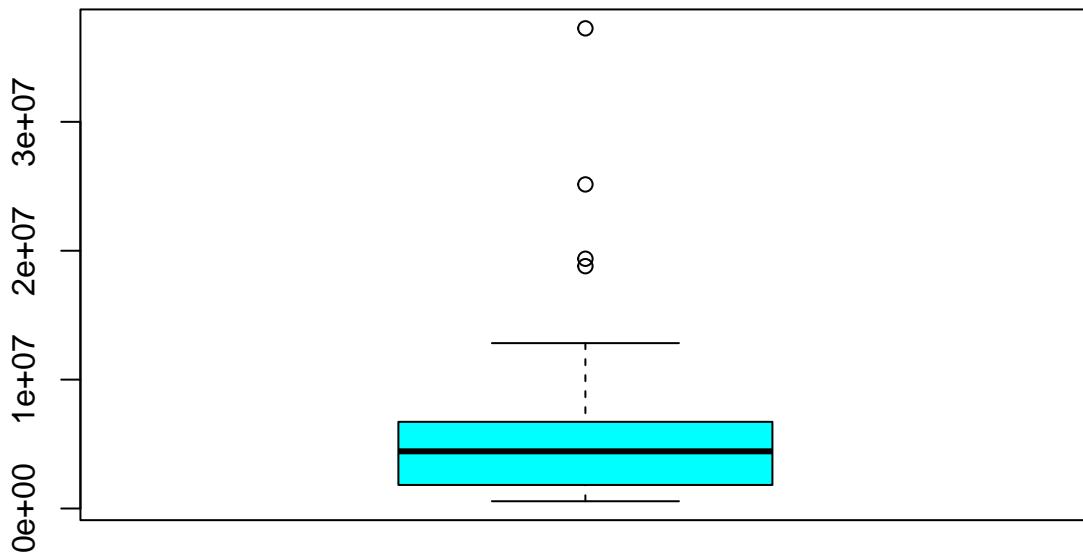
boxplot(asesinatos, col="cyan", main="Número de asesinatos por cada 100 mil habitantes")
```

## Número de asesinatos por cada 100 mil habitantes



```
boxplot(poblacion, col="cyan", main="Población de los estados de USA")
```

## Población de los estados de USA



### Explorando dos o más muestras

El análisis de datos en muchos proyectos involucra examinar la correlación entre variables. Decimos que están correladas si valores altos de  $X$  van a valores altos de  $Y$ , y valores bajos de  $X$  van a valores bajos de  $Y$ . Si valores altos de  $X$  van a valores bajos de  $Y$ , y viceversa, diremos que las variables están correladas negativamente.

El *coeficiente de correlación* nos mide la correlación entre dos variables, es una medida adimensional entre  $-1$  y  $1$ :  $r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$

Muestras de variables cuantitativas. Datos de la grasa corporal y medidas de los pliegues del bíceps, tríceps y muslo de un grupo de 20 mujeres.

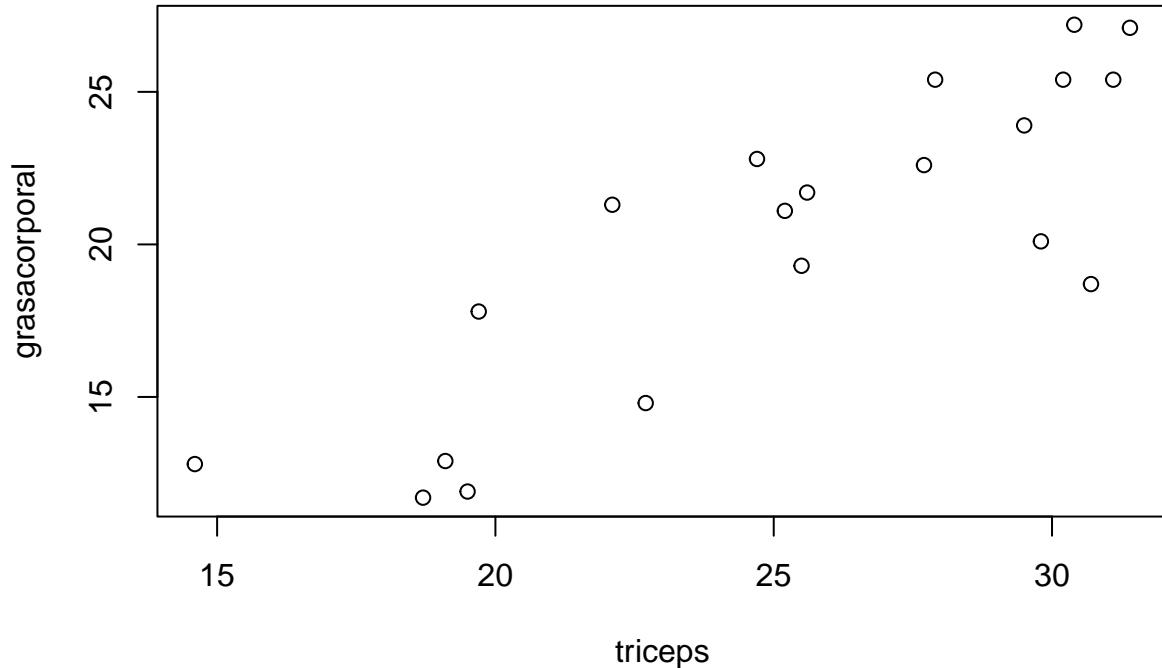
```
body.fat<-read.table("./dades/body.fat.txt", header=T)
dim(body.fat)

## [1] 20  4
attach(body.fat)
names(body.fat)

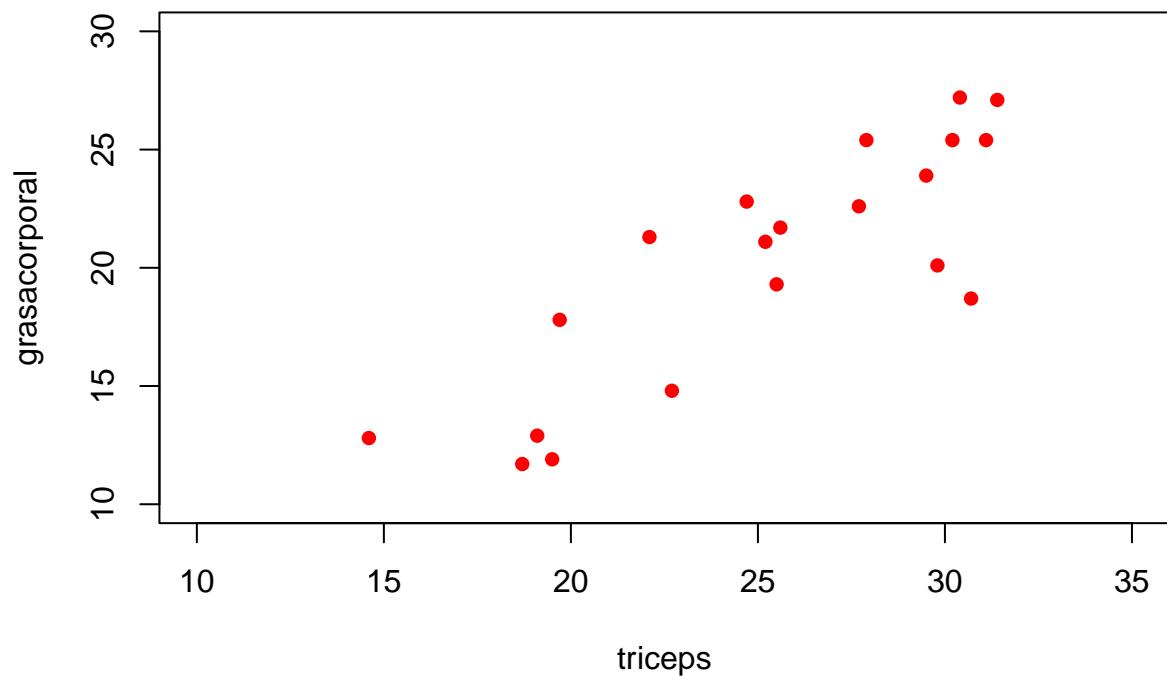
## [1] "triceps"      "muslo"        "biceps"       "grasacorporal"
cor(body.fat)

##          triceps    muslo     biceps  grasacorporal
## triceps   1.0000000 0.9238425 0.4577772  0.8432654
## muslo     0.9238425 1.0000000 0.0846675  0.8780896
## biceps    0.4577772 0.0846675 1.0000000  0.1424440
```

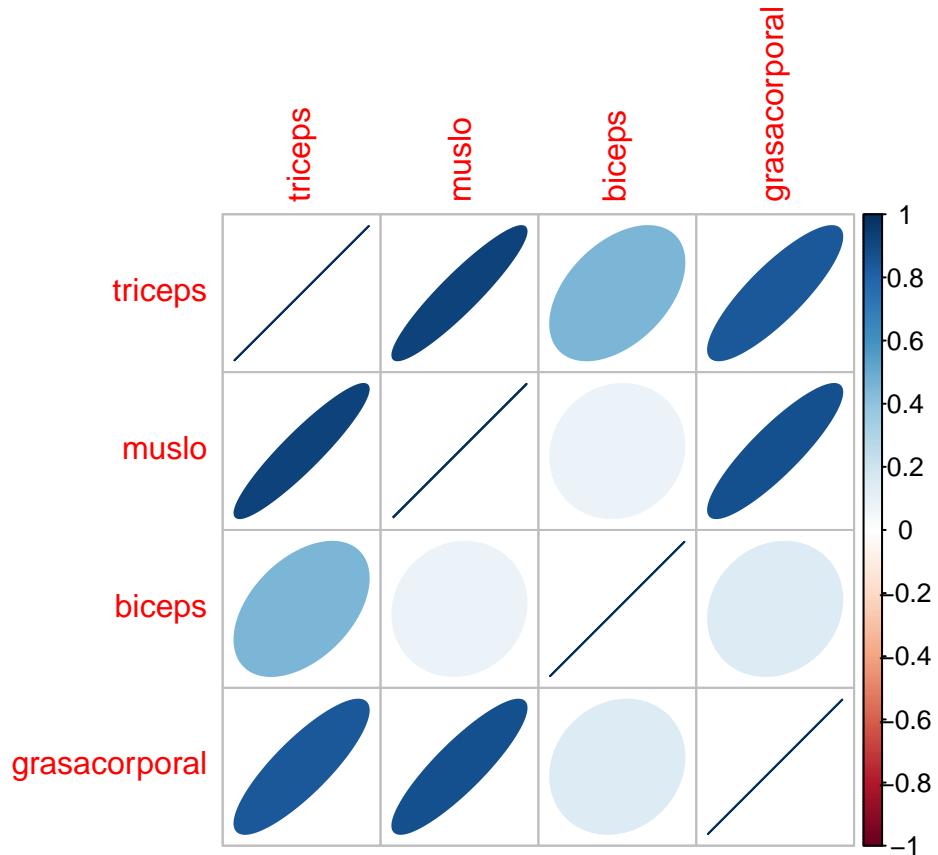
```
## grasacorporal 0.8432654 0.8780896 0.1424440      1.0000000
plot(triceps, grasacorporal)
```



```
plot(triceps, grasacorporal, col="red", xlim=c(10,35), ylim=c(10,30), pch=16)
```



```
library(corrplot)
corrplot(cor(body.fat), method="ellipse")
```



Datos de la tasa de impuestos, dimensión y códigos zip de 432,693 propiedades residenciales de King County, Washington.

```

kc_tax<- read.table("./dades/kc_tax.txt", sep = ",", header=T )
dim(kc_tax)

## [1] 498249      3
names(kc_tax)

## [1] "tasa_impuestos" "dimension"      "codigo_zip"
head(kc_tax, n=10)

##   tasa_impuestos dimension codigo_zip
## 1             NA       1730    98117
## 2        206000       1870    98002
## 3        303000       1530    98166
## 4        361000       2000    98108
## 5        459000       3150    98108
## 6        223000       1570    98032
## 7        259000       1770    98168
## 8        175000       1150    98168
## 9        178000       1980    98168
## 10       186000       1490    98168

kc_tax0 <- subset(kc_tax, tasa_impuestos < 750000 & dimension>100 &
dimension<3500)
nrow(kc_tax0)

```

```
## [1] 432693
```

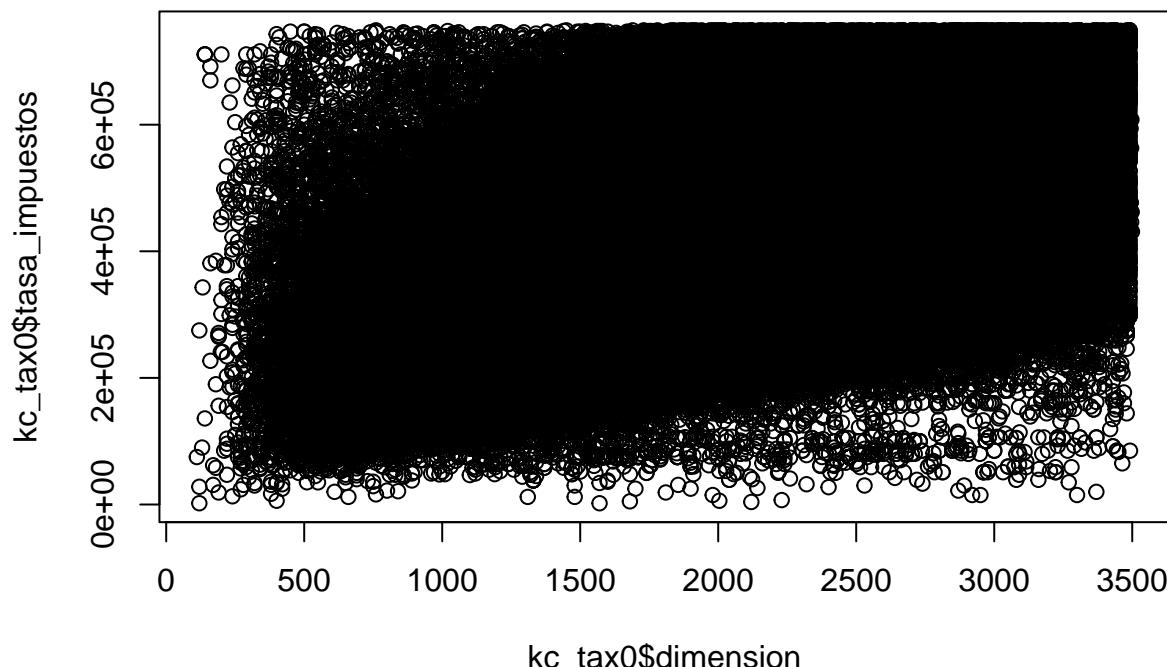
Gráficos que muestran la relación entre dos variables numéricas:

- *scatterplot* (Nube de puntos),
- *Hexagonal binning plot*, las gráficas *scatterplot* están bien cuando hay un número relativamente pequeño de datos. Cuando hay muchos datos los *scatterplots* no tienen sentido porque son demasiado densas. En el Hexagonal binning en lugar de pintar puntos, se agrupan las observaciones en contenedores hexagonales y se pintan los hexágonos en un color indicando el número de observaciones en cada contenedor.

En el gráfico se ve clara la relación entre la dimensión y la tasa de impuestos. Una interesante característica escondida, es la segunda nube arriba de la primera nube más marcada, indicando las residencias con la misma dimensión pero con una tasa de impuestos mayor. Característica que se ve más claramente en el siguiente gráfico.

- *Contour plot* (diagrama de puntos con curvas de nivel), figura que muestra la densidad de dos variables numéricas como en un mapa topográfico. Mismas conclusiones que en el hexagonal binninb, se ve más claramente el segundo pico.
- *Violin plots*, similar al boxplot pero mostrando la estimación de la densidad, en el que se pueden apreciar matices no apreciables en el boxplot. Para combinar un violin plot con un boxplot añadir *geom\_boxplot* al gráfico.

```
plot(kc_tax0$dimension, kc_tax0$tasa_impuestos)
```

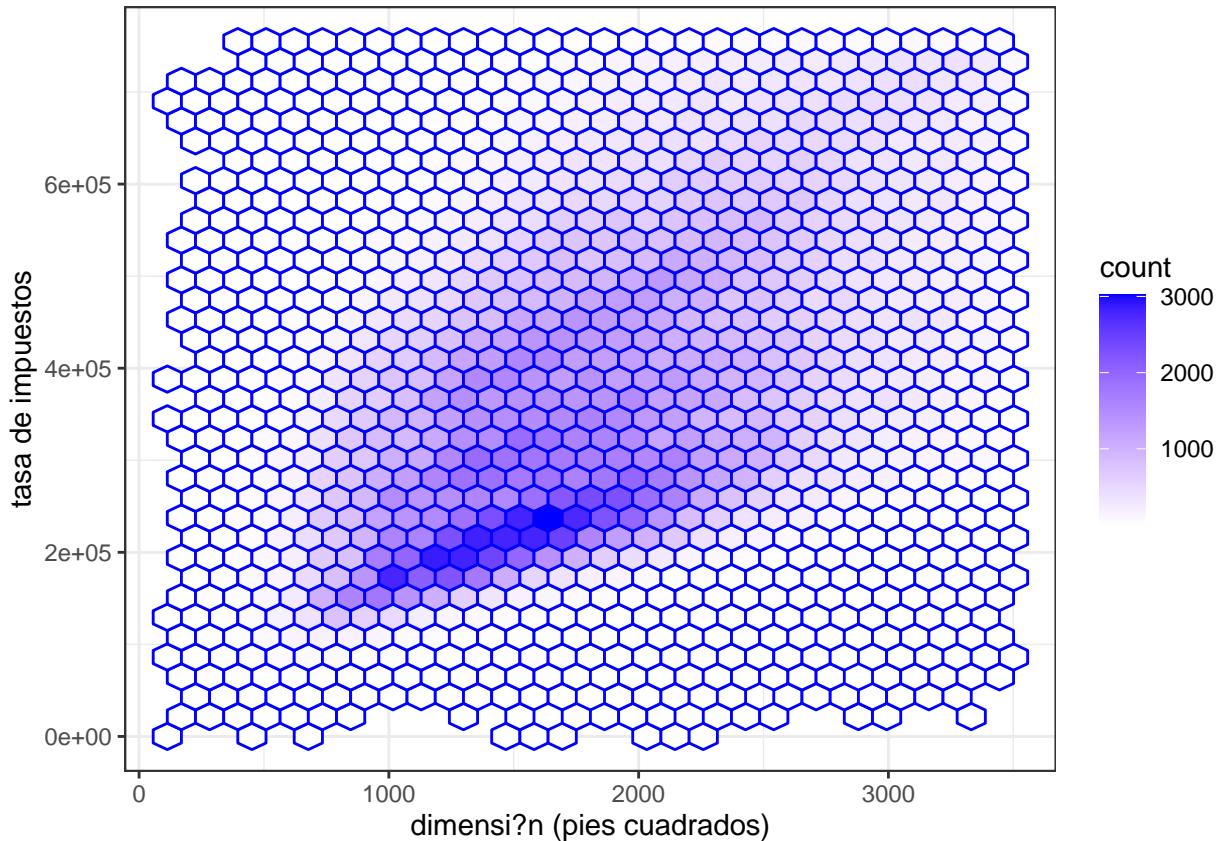


```
library(ggplot2)
ggplot(kc_tax0, (aes(x=dimension, y=tasa_impuestos))) +
  stat_binhex(colour="blue") +
```

```

theme_bw() +
scale_fill_gradient(low="white", high="blue") +
labs(x="dimensi?n (pies cuadrados)", y="tasa de impuestos")

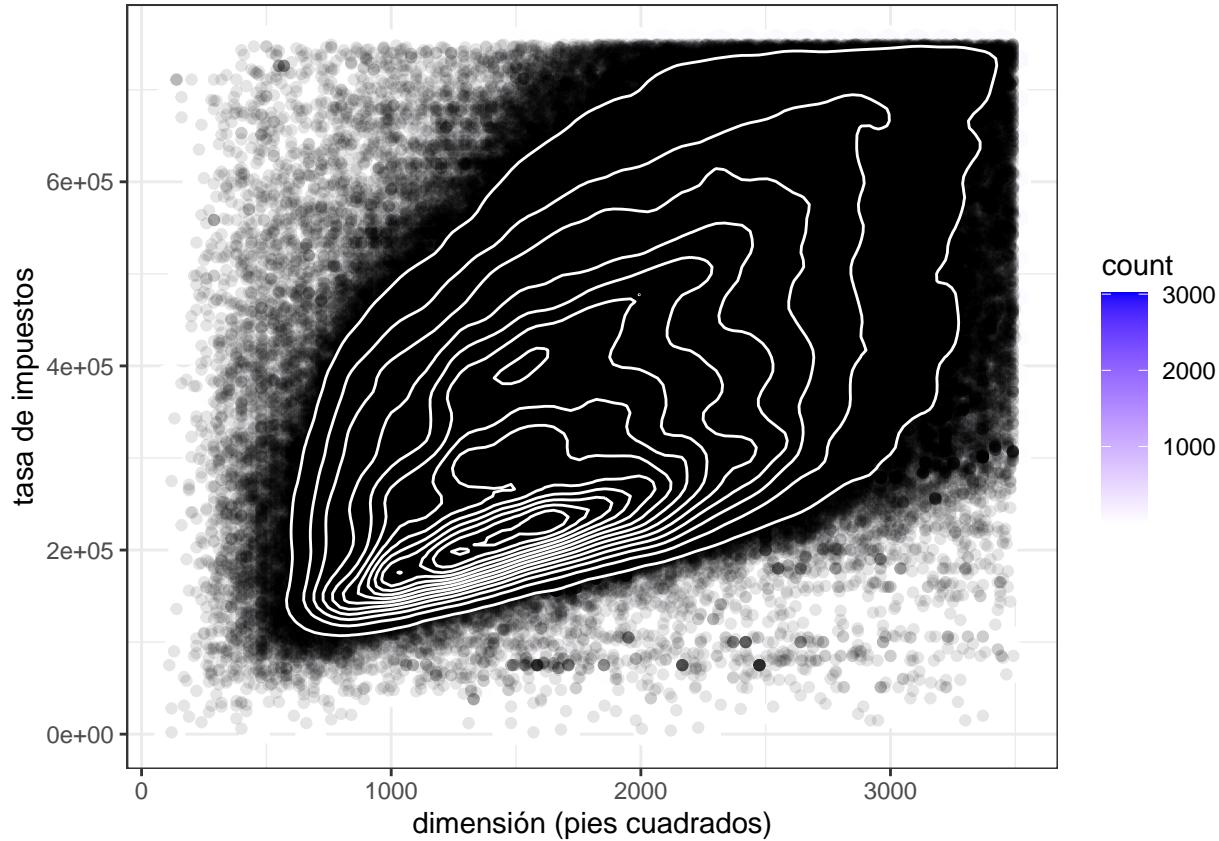
```



```

ggplot(kc_tax0, (aes(x=dimension, y=tasa_impuestos))) +
stat_binhex(colour="white") +
theme_bw() +
geom_point(alpha=0.1) +
geom_density2d(colour="white") +
scale_fill_gradient(low="white", high="blue") +
labs(x="dimensi?n (pies cuadrados)", y="tasa de impuestos")

```



## Datos de variables cuantitativas y categóricas

### Datos de dos variables categóricas

Datos de préstamos de una determinada institución

```
loan<- read.table("./dades/loan.txt", sep = ",", header=T )
dim(loan)
```

```
## [1] 45343     21
names(loan)

## [1] "X"                  "status"              "loan_amnt"
## [4] "term"                "annual_inc"          "dti"
## [7] "payment_inc_ratio"   "revol_bal"           "revol_util"
## [10] "purpose"             "home_ownership"      "delinq_2yrs_zero"
## [13] "pub_rec_zero"        "open_acc"            "grade"
## [16] "outcome"             "emp_length"         "purpose_"
## [19] "home_"               "emp_len_"           "borrower_score"
```

Imprime las primeras 10 filas del fichero

```
head(loan, n=10)
```

	X	status	loan_amnt	term	annual_inc	dti
## 1	1	Charged Off	2500	60 months	30000	1.00
## 2	2	Charged Off	5600	60 months	40000	5.55
## 3	3	Charged Off	5375	60 months	15000	18.08

```

## 4 4 Charged Off 9000 36 months 30000 10.08
## 5 5 Charged Off 10000 36 months 100000 7.06
## 6 6 Charged Off 21000 36 months 105000 13.22
## 7 7 Charged Off 6000 36 months 76000 2.40
## 8 8 Charged Off 15000 36 months 60000 15.22
## 9 9 Charged Off 5000 60 months 50004 13.97
## 10 10 Charged Off 5000 36 months 100000 16.33
## payment_inc_ratio revol_bal revol_util purpose home_ownership
## 1 2.39320 1687 9.4 car RENT
## 2 4.57170 5210 32.6 small_business OWN
## 3 9.71600 9279 36.5 other RENT
## 4 12.21520 10452 91.7 debt_consolidation RENT
## 5 3.90888 11997 55.5 other RENT
## 6 8.01977 32135 90.3 debt_consolidation RENT
## 7 3.13358 5963 29.7 major_purchase RENT
## 8 10.29280 5872 57.6 debt_consolidation RENT
## 9 2.96736 4345 59.5 other RENT
## 10 1.90524 74351 62.1 debt_consolidation MORTGAGE
## delinq_2yrs_zero pub_rec_zero open_acc grade outcome emp_length
## 1 1 1 3 4.8 default 1
## 2 1 1 11 1.4 default 5
## 3 1 1 2 6.0 default 1
## 4 1 1 4 4.2 default 1
## 5 1 1 14 5.4 default 4
## 6 1 1 7 5.8 default 11
## 7 1 1 7 5.6 default 2
## 8 1 1 7 4.4 default 10
## 9 0 1 14 3.4 default 3
## 10 1 1 17 7.0 default 11
## purpose_ home_ emp_len_ borrower_score
## 1 major_purchase RENT > 1 Year 0.65
## 2 small_business OWN > 1 Year 0.80
## 3 other RENT > 1 Year 0.60
## 4 debt_consolidation RENT > 1 Year 0.50
## 5 other RENT > 1 Year 0.55
## 6 debt_consolidation RENT > 1 Year 0.40
## 7 major_purchase RENT > 1 Year 0.70
## 8 debt_consolidation RENT > 1 Year 0.50
## 9 other RENT > 1 Year 0.45
## 10 debt_consolidation MORTGAGE > 1 Year 0.50

library(descr)
x_tab<-CrossTable(loan$home_, loan$status, prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
x_tab

## Cell Contents
## |-----|
## | N |
## | N / Row Total |
## |-----|
##
## =====
## loan$status
## loan$home_ Charged Off Default Fully Paid Total
## -----

```

```

##    MORTGAGE          9673          106        11098  20877
##                  0.463       0.005       0.532  0.460
## -----
##    OWN              1870           18        1834  3722
##                  0.502       0.005       0.493  0.082
## -----
##    RENT             10907          97        9740  20744
##                  0.526       0.005       0.470  0.457
## -----
## Total            22450          221        22672 45343
## -----

```

Datos del % de retrasos debidos a diferentes causas en los vuelos que aterrizan en el aeropuerto de Dallas-Fort Worth

```
airline<- read.table("./dades/airline_delays.txt", sep = ", ", header=T, fileEncoding ="WINDOWS-1252")
dim(airline)
```

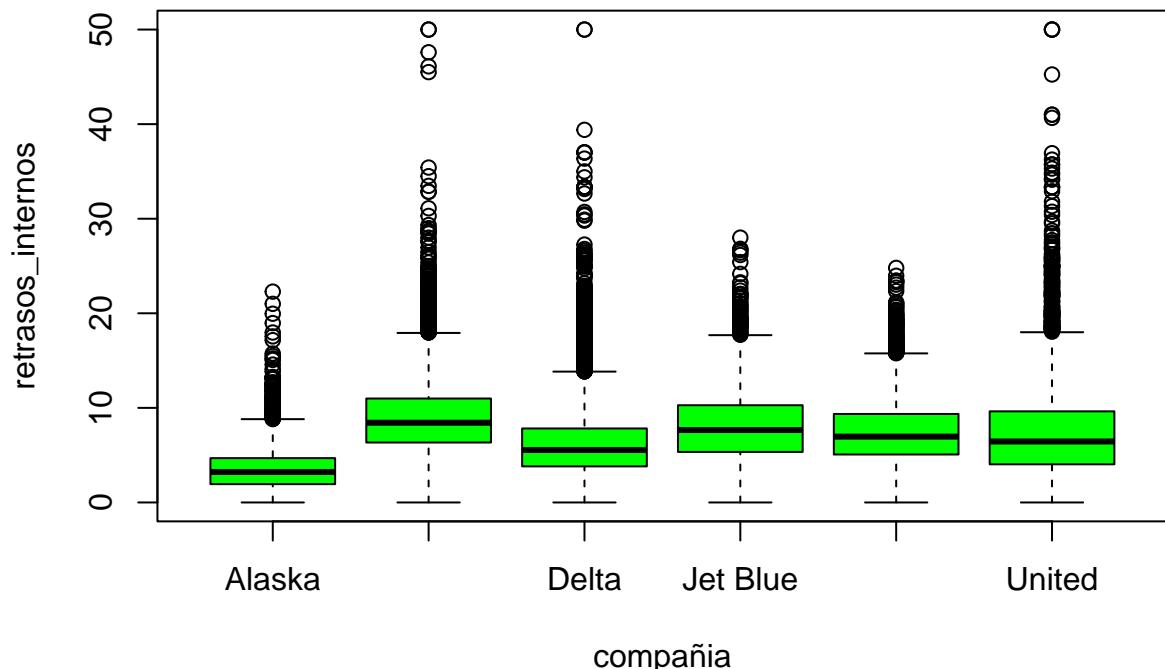
```
## [1] 33469      4
```

```
names(airline)
```

```
## [1] "retrasos_internos"      "retrasos_trafico"
## [3] "retrasos_cclimatologicas" "compañia"
```

Box plot:

```
boxplot(retrasos_internos ~ compañía, data=airline, ylim=c(0,50), col="green")
```



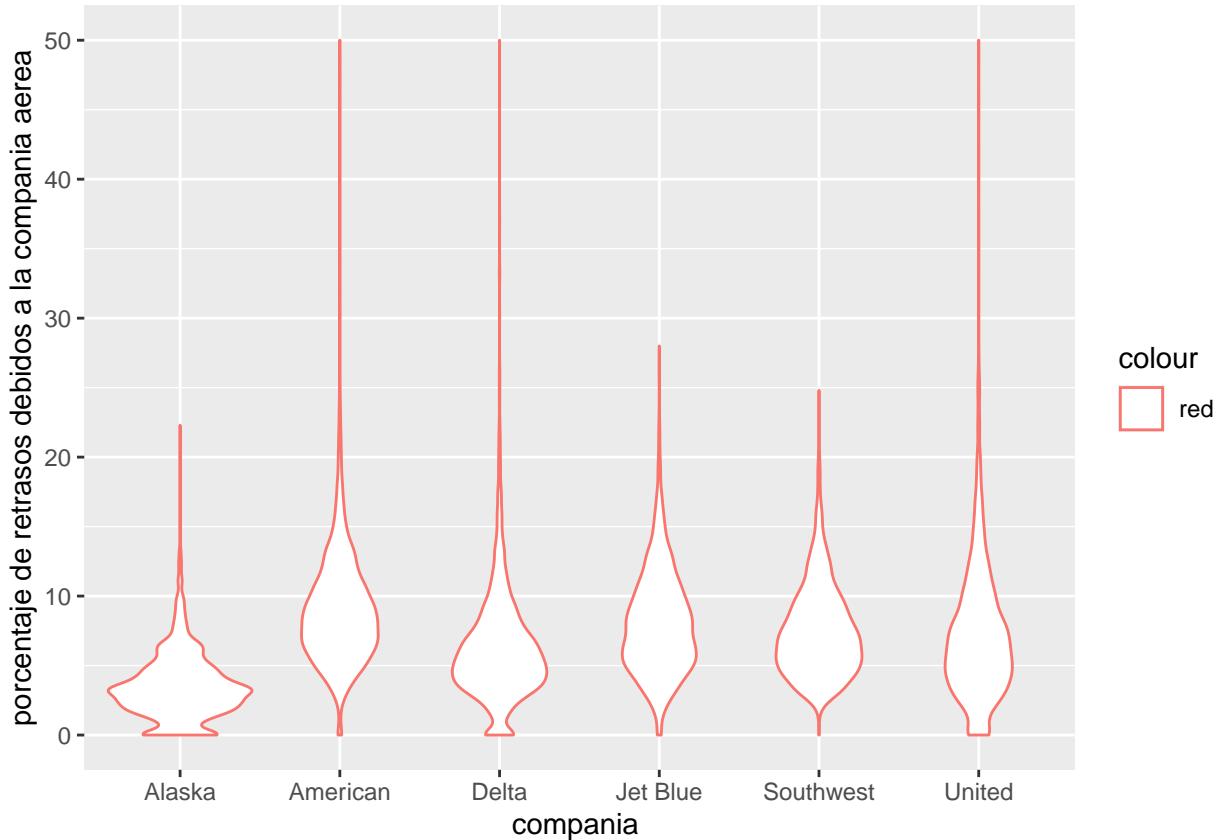
Violin plot

```

ggplot(data=airline, aes(compañia, retrasos_internos, color="red")) +
  ylim(0, 50) +
  geom_violin() +
  labs(x="compañia ", y="porcentaje de retrasos debidos a la compañía aérea")

## Warning: Removed 38 rows containing non-finite values (stat_ydensity).

```



Visualización de muchas variables

```

ggplot(subset(kc_tax0, codigo_zip %in% c(98188, 98105, 98108, 98126)),
aes(x=dimension, y=tasa_impuestos)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="blue") +
  labs(x="dimension (pies cuadrados)", y="tasa impuestos") +
  facet_wrap("codigo_zip")

```

