Guia 1 parte 2

Agustín Muñoz González

26/4/2020

1. Borrar todos los objetos existentes en el entorno de trabajo y establecer directorio de trabajo.

Procedemos a limpiar los registros. HAY QUE SETEAR EL DIRECTORIO DE TRABAJO CON EL ARCHIVO TITANIC.

```
rm(list=ls())
```

2. Leer el conjunto de titanic.csv teniendo en cuenta que en la primera línea del archivo figura el nombre de las variables y el tipo de separación de los datos y asignelo al data.frame titanic.

Leemos el archivo y lo guardamos en la variable titanic.

```
titanic=read.csv('titanic.csv', header=T)
```

3. Inspeccionar los primeros casos del archivo y los últimos.

Recordemos que los comandos para leer los primeros y los últimos datos son *head()* y *tail()* respectivamente. Específicamente *head()* devuelve los primeros 6 datos y *tail()* los últimos 6.

```
head(titanic)
```

```
##   pclass survived                                            name    sex    age
## 1      1        1                   Allen, Miss. Elisabeth Walton female     29
## 2      1        1                  Allison, Master. Hudson Trevor   male 0,9167
## 3      1        0                    Allison, Miss. Helen Loraine female      2
## 4      1        0            Allison, Mr. Hudson Joshua Creighton   male     30
## 5      1        0 Allison, Mrs. Hudson J C (Bessie Waldo Daniels) female     25
## 6      1        1                              Anderson, Mr. Harry   male     48
##   sibsp parch ticket     fare embarked
## 1     0     0  24160 211,3375        S
## 2     1     2 113781 151,5500        S
## 3     1     2 113781 151,5500        S
## 4     1     2 113781 151,5500        S
## 5     1     2 113781 151,5500        S
## 6     0     0  19952  26,5500        S
```

```
tail(titanic)
```

```
##      pclass survived                      name    sex  age sibsp parch ticket
## 1304      3        0     Yousseff, Mr. Gerious   male          0     0   2627
## 1305      3        0      Zabour, Miss. Hileni female 14,5     1     0   2665
## 1306      3        0     Zabour, Miss. Thamine female          1     0   2665
## 1307      3        0 Zakarian, Mr. Mapriededer   male 26,5     0     0   2656
## 1308      3        0       Zakarian, Mr. Ortin   male   27     0     0   2670
## 1309      3        0        Zimmerman, Mr. Leo   male   29     0     0 315082
##         fare embarked
## 1304 14,4583        C
## 1305 14,4542        C
## 1306 14,4542        C
## 1307  7,2250        C
## 1308  7,2250        C
## 1309  7,8750        S
```

4. Abrir con el editor al data.frame e inspeccionar el archivo.

Para esto usamos el comando *data.frame()*. ¿Qué hace este comando? Según el comando *help()*, el comando "Loads specified data sets, or list the available data sets." Guardamos el data.frame en una nueva variable 'tit'.

```
tit=data.frame(titanic)
```

Notar que tit==titanic y son datos de la misma clase (adelantandonos un poco al item 7), con lo cual *data.frame()* y *read()* devuelven el mismo tipo de datos.

5. Establecer el número de variables y de casos.

El comando *dim()* devuelve la cantidad de casos x la cantidad de variables del conjunto de datos que le ingresemos.

```
dim(tit)
```

```
## [1] 1309   10
```

6. Realizar un attach de titanic.

Attachamos la variable para que el manejo de los datos sea más práctico.

```
attach(titanic)
```

7. Inspeccionar los nombres de las variables de titanic e identificar de qué tipo de variable se trata cada una de ellas.

Para ver el nombre de las variables usamos el comando *names()* o *ls()*

```
names(tit)
```

```
##  [1] "pclass"   "survived" "name"     "sex"      "age"      "sibsp"
##  [7] "parch"    "ticket"   "fare"     "embarked"
```

```
ls(tit)
```

```
##  [1] "age"      "embarked" "fare"     "name"     "parch"    "pclass"
##  [7] "sex"      "sibsp"    "survived" "ticket"
```

La diferencia, como vemos, es que *ls()* devuelve ordenado alfabeticamente, es decir

```
ls(tit)==sort(names(tit),decreasing = F)
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Y para el tipo de variable usamos *class()*. Para recorrer todas las variables usamos la sintaxis for.

```
for (i in names(tit)){
print(class(get(i)))
}
```

```
## [1] "integer"
## [1] "integer"
## [1] "factor"
## [1] "factor"
## [1] "factor"
## [1] "integer"
## [1] "integer"
## [1] "factor"
## [1] "factor"
## [1] "factor"
```

```
# Debemos usar el comando get() para obtener los datos de la variable de nombre i.
# Digamos, si printeamos class(i) nos va a devolver "character".
```

El comando *str()* aporta un poco mas de información sobre la variable, pues no sólo aporta su clase sino los niveles de la misma (las categorías si se trata de una variable de tipo Factor) y los valores que toma

```
str(tit)
```

```
## 'data.frame':    1309 obs. of  10 variables:
##  $ pclass  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ survived: int  1 1 0 0 0 1 1 0 1 0 ...
##  $ name    : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age     : Factor w/ 99 levels "","0,1667","0,3333",..: 39 8 23 41 33 67 86 55 73 94 ...
##  $ sibsp   : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch   : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket  : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare    : Factor w/ 282 levels "","0,0000","10,1708",..: 79 58 58 58 58 103 236 2 157 153 ...
##  $ embarked: Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
```

Este comando es mas útil en tanto que vemos si tiene sentido el tipo de clase de cada variable y, en caso que sea incorrecto o inadecuado, podemos cambiarlo. Por ejemplo, las variables 'pclass' y 'survived' que reflejan la clase de cabina de los pasajeros y si sobrevivieron o no (1=sobrevivir, 0=morir) respectivamente, dado que

dividen el conjunto de datos en categorías deberían ser de tipo Factor. Por otro lado, las variables 'name', 'age', 'ticket' y 'fare' son variables que aportan datos únicos para cada persona y no dividen el conjunto de datos en grupos. Con lo cual tiene mas sentido que sean variables de tipo numérico que de tipo categórico. En general, si los niveles de las variables (los valores que toma) son 'muchos' (relativo a los datos en general y a la cantidad de niveles de las demás variables categóricas) tiene mas sentido que sea numérica, caracter, etc., que categórica.

Procedemos a arreglar todas estas variables

```
pclass=as.factor(pclass)
survived=as.factor(survived)
name=as.character(name)
age=as.numeric(age)
ticket=as.numeric(ticket)
fare=as.numeric(fare)
```

8. Calcule la chance de sobrevivir siendo hombre. Calcule la chance de sobrevivir siendo mujer.

Debemos calcular la proporción de supervivencia (o la tasa de mortalidad) de los hombres y las mujeres. Para eso filtraremos los datos de la variable 'survived' para hombres por un lado y para mujeres por otro.

```
male_survived = tit[sex == 'male','survived']
female_survived = tit[sex == 'female','survived']
```

Como los valores de 'survived' son 0 ó 1, calcular la media es calcular la proporción de supervivencia (equiv. 1-tasa de mortalidad). Mostramos de paso varios comandos para devolver por pantalla el dato junto con alguna frase

```
cat('La proporción de supervivencia de los hombres es de ', mean(male_survived))
```

```
## La proporción de supervivencia de los hombres es de  0.1909846
```

```
cat('La proporción de supervivencia de los mujeres es de ', mean(female_survived))
```

```
## La proporción de supervivencia de los mujeres es de  0.7274678
```

```
# message('La proporción de supervivencia de los hombres es de ', mean(male_survived))
# paste('La proporción de supervivencia de los hombres es de ', mean(male_survived))
# sprintf('La proporción de supervivencia de los hombres es de %f', mean(male_survived))
# como notamos en el output, los comandos sprintf() y paste()
# devuelven 'character' y message() devuelve un mensaje.
```

9. Cree que el tipo de ticket del pasajero (clase de cabina) esta asociado con su supervivencia?

Veamos las proporciones de supervivencia de cada clase

```
supervivencia_clase_1=mean(tit[pclass==1,'survived'])
supervivencia_clase_1
```

```
## [1] 0.619195
```

```
supervivencia_clase_2=mean(tit[pclass==2,'survived'])
supervivencia_clase_2
```

```
## [1] 0.4296029
```

```
supervivencia_clase_3=mean(tit[pclass==3,'survived'])
supervivencia_clase_3
```

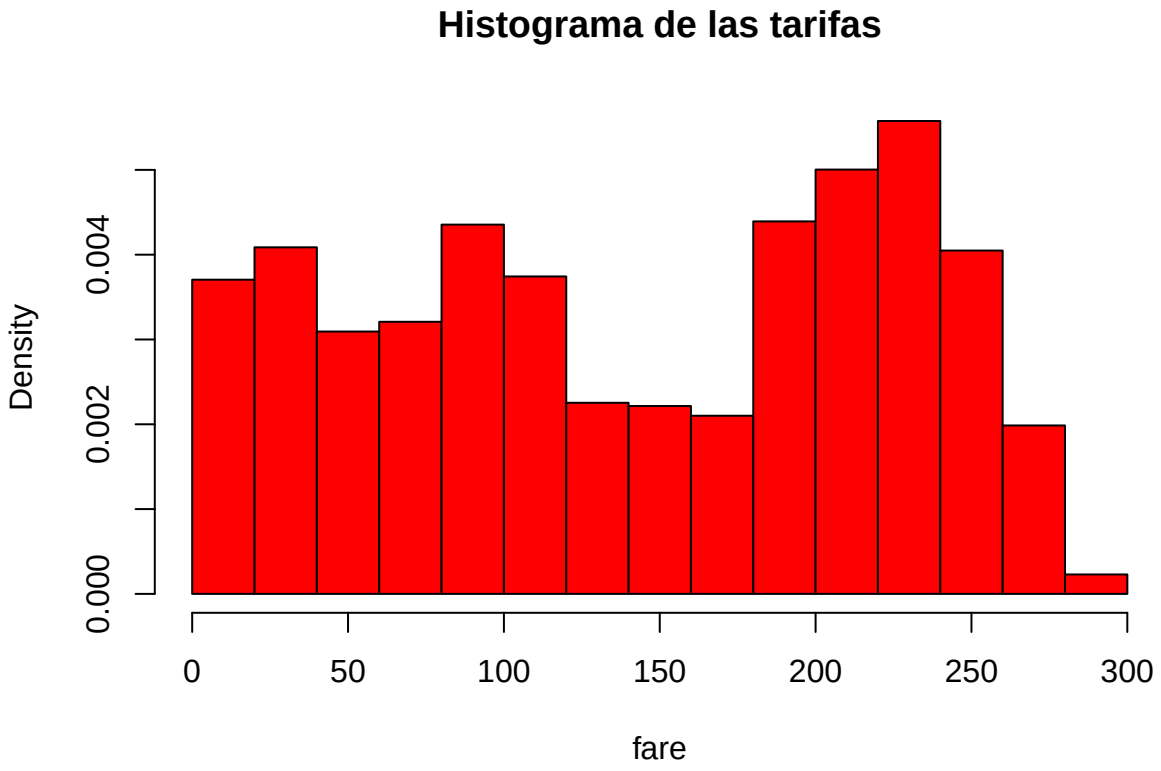
```
## [1] 0.2552891
```

Es evidente que la clase de la cabina no es independiente de las chances de sobrevivir porque se ve que en las mejores clases las chances fueron mas altas.

10. Estudie la distribución de las tarifas. Que observa? Le parece razonable suponer que la variable tarifa tenga distribución normal? Calcule la media y la mediana. Puede decidir de antemano quién es mas grande si la media o la mediana?

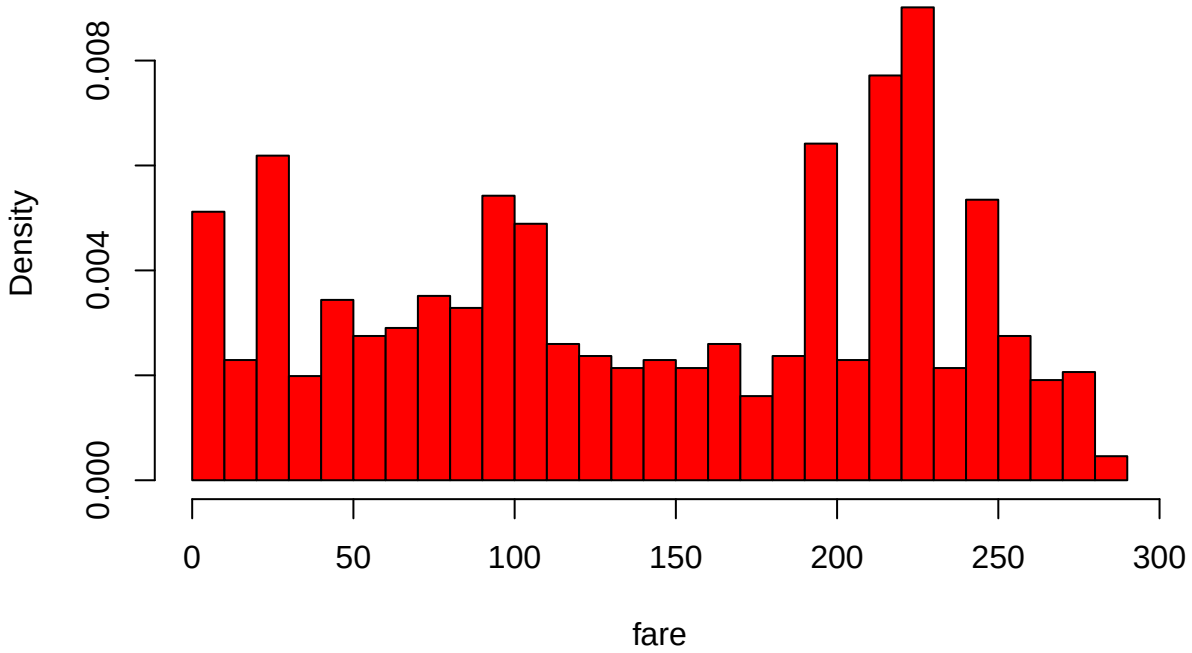
Analizaremos la información de las tarifas (variable 'fare') a través de diferentes tipos de gráficos. Como ya convertimos 'fare' en una variable numérica, podemos hacer un histograma (en general siempre podés hacer un histograma con cualquier tipo de variable pero es más adecuado para variables numéricas)

```
hist(fare,nclass=15,freq=F,col='red',
     main="Histograma de las tarifas")
```



Para agregarle un poco de precisión al gráfico veamos una variación del mismo pidiéndole una cantidad más grande de intervalos en los datos, es decir, establezcamos un 'nclass' (o un 'break', ya que tienen la 'misma' función) más alto

```
hist(fare,freq=F,col='red',
     #nclass = 30,
     #xlim=c(0,400),
     breaks = 30,
     main="Histograma de las tarifas")
```

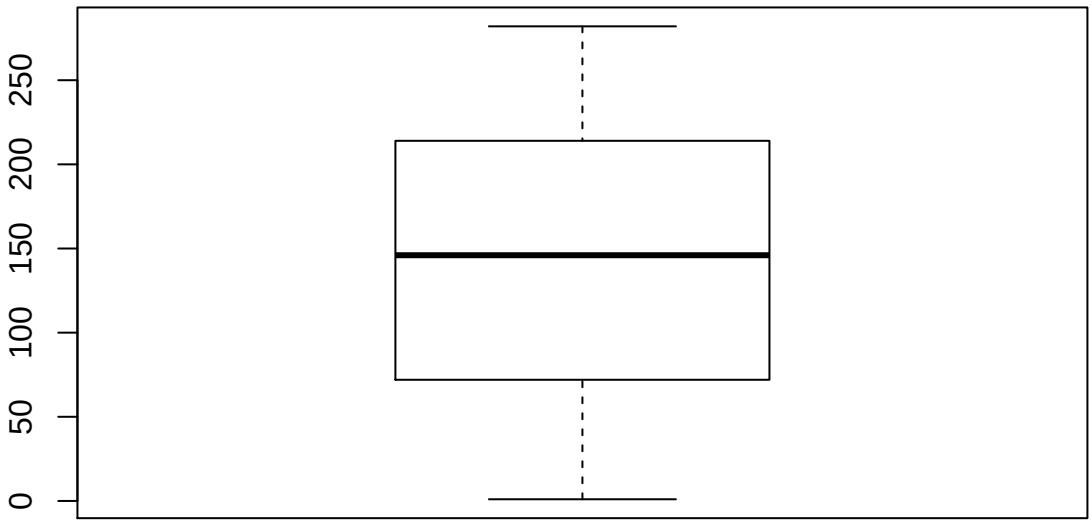



```
# break = int, establece la cantidad de divisiones del intervalo de los datos
# (i.e. la cantidad de columnas)
# break = vector, establece los cortes de cada intervalo.
# Ej: break= c(0,1,2), vamos a tener los intervalos [0,1] y [1,2].
```

Observamos que la mayor concentración de tickets se encuentra en las franjas de precios 0~100 y 200~250, siendo esta última la de mayor densidad. Con lo cual parece no tener una densidad normal.

Veamos ahora un gráfico de caja

```
boxplot(fare, xlab="Tickets",
        main="Tickets")
```



Tickets

Que el hecho de que la variable tenga una distribución normal sea razonable o no depende de la clase de pasajeros del Titanic, si es muy variada (y uniforme) podríamos pensar que sí es razonable que la distribución sea normal. Ahora bien, si la mayoría de personas son personas de bajos recursos probablemente tengamos una concentración en los tickets de menor costo y recíprocamente si la mayor parte de los pasajeros son de altos recursos económicos.

Calculemos media y mediana por medio de los comandos *mean()* y *median()* respectivamente.

Recordemos ambas nociones antes de responder de antemano que creemos lo que va a pasar. La media es el promedio, intuitivamente es el valor al que 'tiende' la muestra. La mediana es un valor respecto del cual los datos quedan divididos a la mitad en tanto que vamos a tener un 50% de la muestra en el intervalo [minimo, mediana] y la otra mitad en el intervalo [mediana, maximo]. Entonces si por ejemplo la muestra se trata de datos concentrados cerca del mínimo y el máximo no es alcanzado muchas veces pero es un número considerablemente superior al mínimo (tan superior que nos cambie mucho la media) entonces media y mediana serán muy distintas, con mediana << media. Es decir, la media no 've' la densidad, la mediana sí.

En nuestro caso concreto la verdad no se qué creer porque no termino de entender intuitivamente (porque teóricamente no hay mucho mas para decir que lo que dijimo antes) la información que aportan media y mediana.

Calculemos ambas y veamos

```
media_tickets=mean(fare)
media_tickets
```

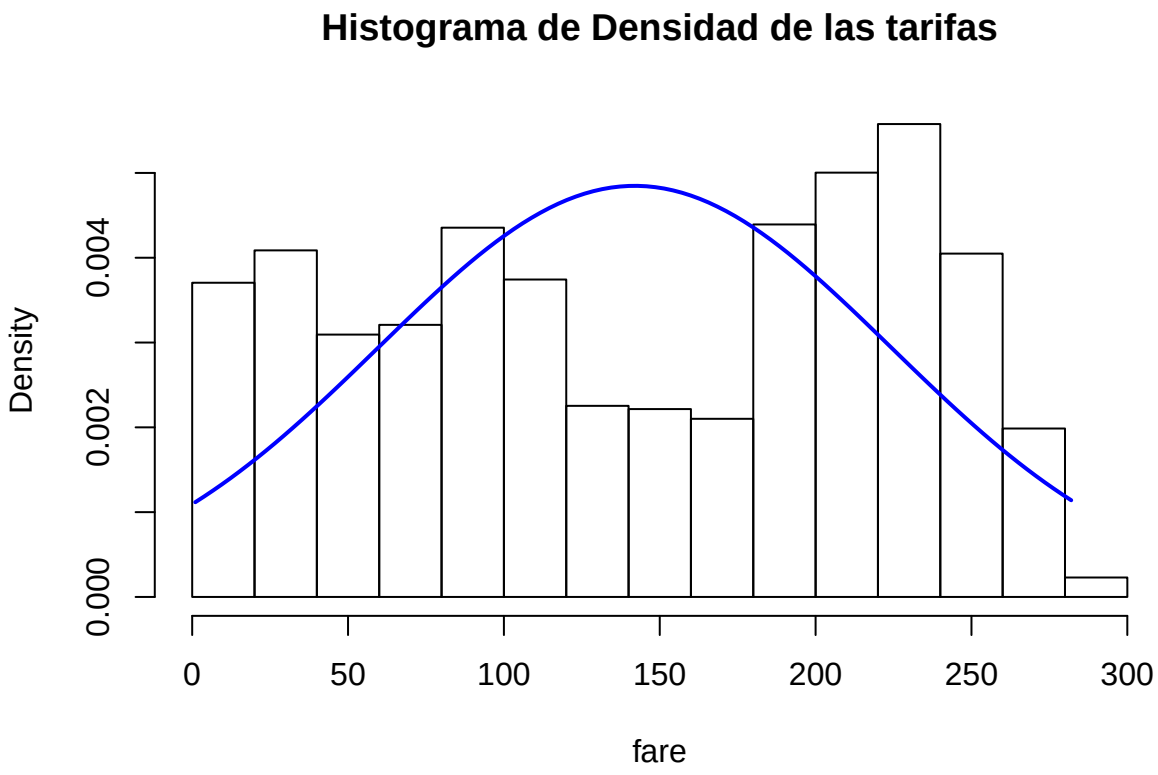
```
## [1] 141.984
```

```
mediana_tickets=median(fare)
mediana_tickets
```

```
## [1] 146
```

Algo que podemos hacer para ver mejor si se aleja de la distribución normal es dibujar el hisograma superpuesto con la curva de densidad (con entradas la media y la desviación de los tickets). Tenemos

```
media.es<-mean(fare)
desvio.es<-sd(fare)
grilla<-seq(range(fare)[1],
            range(fare)[2],length=100)
funn<-dnorm(grilla,media.es,desvio.es)
hist(fare,nclass=15,freq=F,
     main="Histograma de Densidad de las tarifas")
lines(grilla,funn,col="blue",lwd=2)
```



11. Estudie la relación entre tarifa y clase y entre edad y clase.

Antes que nada mencionar que no todos los tipos de gráficos sirven para lo mismo.

Por ejemplo si queremos ver relaciones entre variables de tipo numérico, un ġrafico de barras no va a aportar información muy clara, y en cambio un gráfico de dispersión es mas adecuado. Pero si nos interesa ver como se distribuyen los datos de cierta variable numerica en otra variabla categórica entonces el adecuado sería un gráfico de caja, ya que un diagrama de dispersión, de barras o de torta serían simplemente manchas negras (si los datos numéricos son muchos).

Como para tener en la cabeza, los distintos gráficos son adecuados para las siguientes situaciones (a grandes rasgos)

- Histograma: Para estudiar densidad de una variable numérica.
- Gráfico de caja: Para estudiar var categorica vs var numérica. Es decir, la disposición de la numérica en las categorías o grupos de la categórica.
- Diagrama de dispersión: Para estudiar var numérica vs var numérica.
- Gráfico de barras o de torta: Para estudiar la distribución de una (o más) var categórica en el total de datos.

Dicho esto y a riesgo de repetirnos, mostraremos gráficos adecuados e inadecuados en las relaciones que queremos estudiar para remarcar la diferencia.

Tarifa vs Clase

Empecemos con la relación tarifa vs clase. Donde tarifa debería ser de tipo numérico y clase debería ser de tipo factor. Como ya arreglamos la clase de estas variables en el Ejercicio 7 podemos realizar algún gráfico.

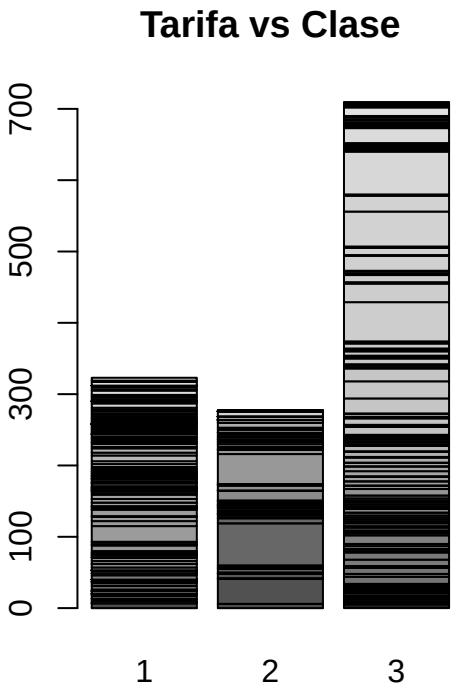
Como mencionamos al principio, para comparar una variable numérica vs una categórica el gráfico de barras no es adecuado. Veamos que esto es así.

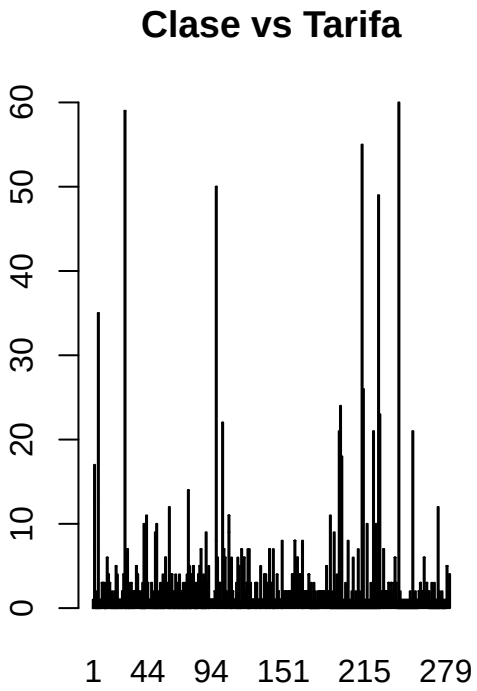
Hagamos una tabla asociada a la relación que queremos estudiar y la tabla transpuesta, para ver como cambia el output

```
tabla_tarifa_clase = table(fare,pclass)
tabla_clase_tarifa = table(pclass,fare)
```

Hagamos ambos gráficos de barras, que en general es adecuado para ver la distribución de una o mas variables categórica en el total de datos

```
par(mfrow=c(1,2))
barplot(tabla_tarifa_clase, main='Tarifa vs Clase')
barplot(tabla_clase_tarifa, main='Clase vs Tarifa')
```



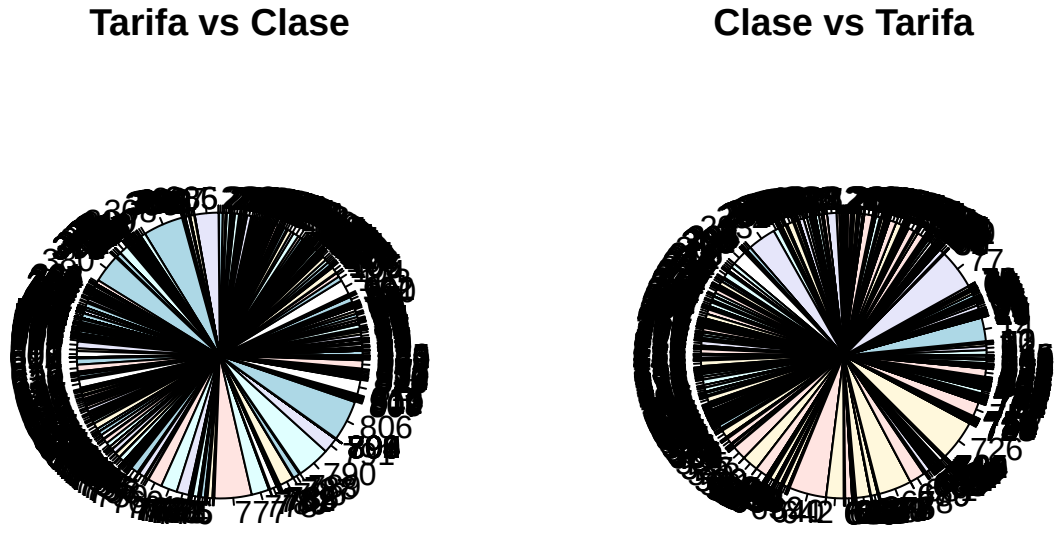



```
par(mfrow=c(1,1))
```

Se ve que como la variable tarifa tiene muchos datos este gráfico es caótico y está lejos de decirnos algo sobre la distribución de las tarifas en las distintas clases del barco.

Veamos el gráfico de torta que tiene en general la misma aplicación que el de barras

```
par(mfrow=c(1,2))
pie(tabla_tarifa_clase, main='Tarifa vs Clase')
pie(tabla_clase_tarifa, main='Clase vs Tarifa')
```

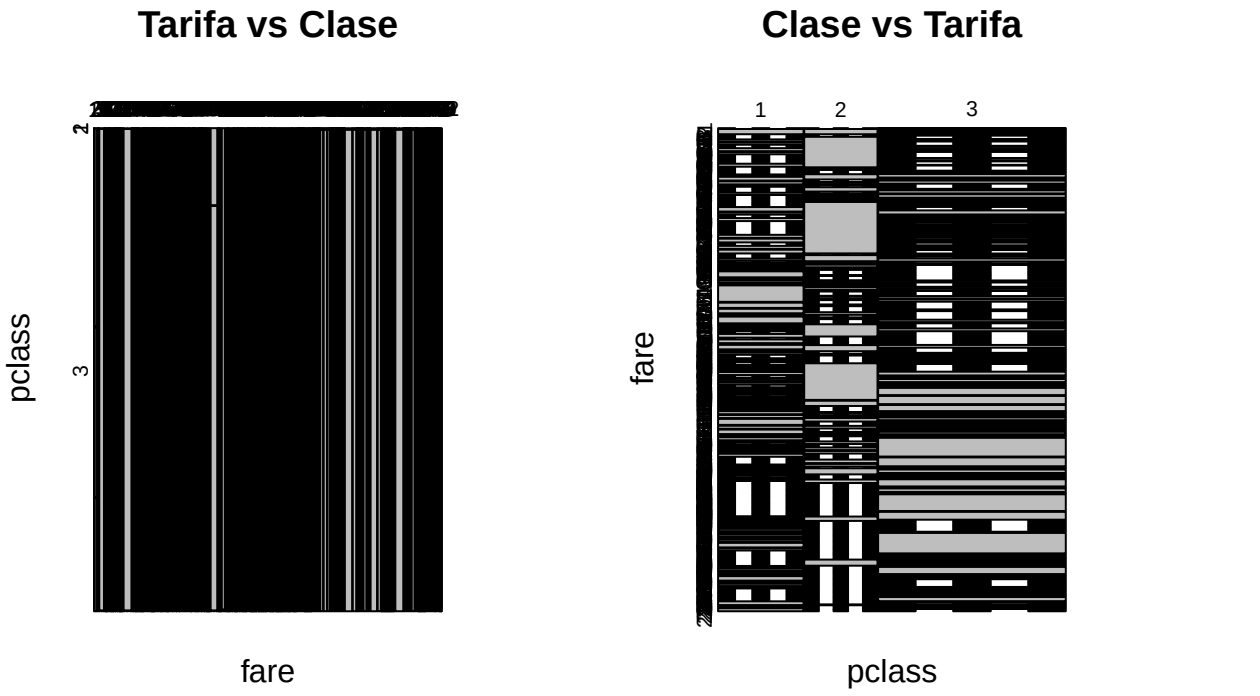



```
par(mfrow=c(1,1))
```

Que vuelve a ser caótico.

Por último hagamos un diagrama de dispersión, que se usa para comparar variables numéricas

```
par(mfrow=c(1,2))
plot(tabla_tarifa_clase, main='Tarifa vs Clase')
plot(tabla_clase_tarifa, main='Clase vs Tarifa')
```

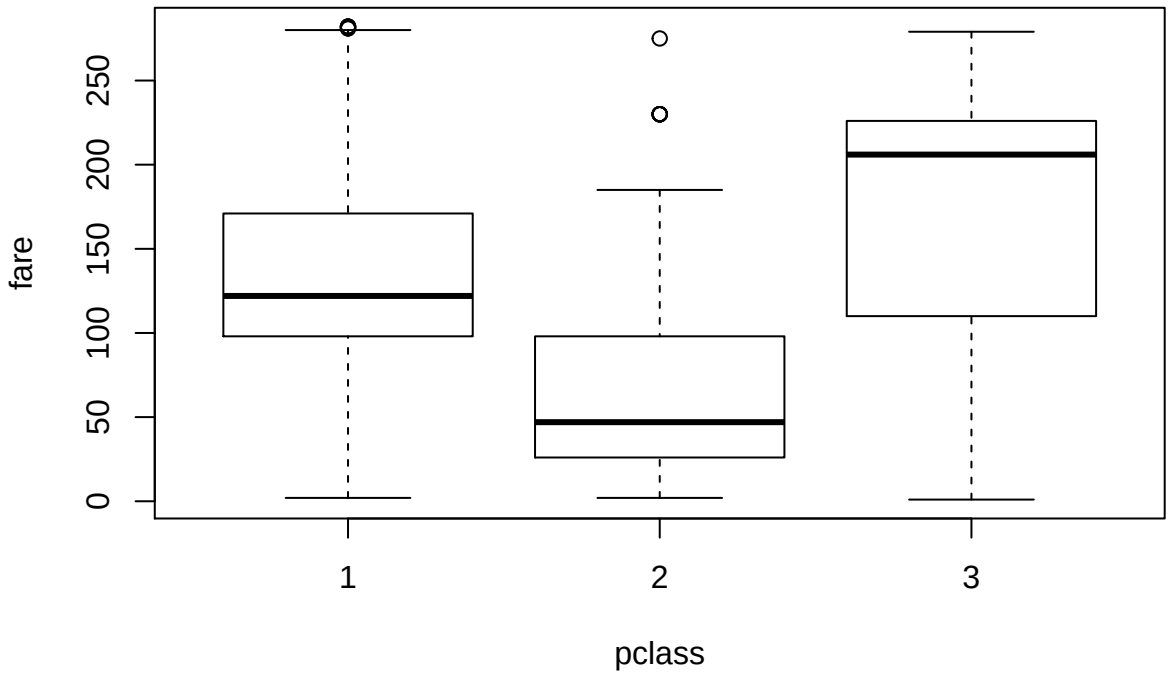



```
par(mfrow=c(1,1))
```

Veamos que efectivamente otros gráficos como los de caja son mas adecuados.

Veamos el gráfico de caja. El comando *boxplot()* no funciona por tablas sino por la asignación de una variable numérica y una categórica, de forma que los datos numéricos van a ser divididos en las categorías de la variable categórica.

```
boxplot(fare~pclass)
```




```
# donde: y ~ grp, where y is a numeric vector of data values to be
# split into groups according to the grouping variable grp
```

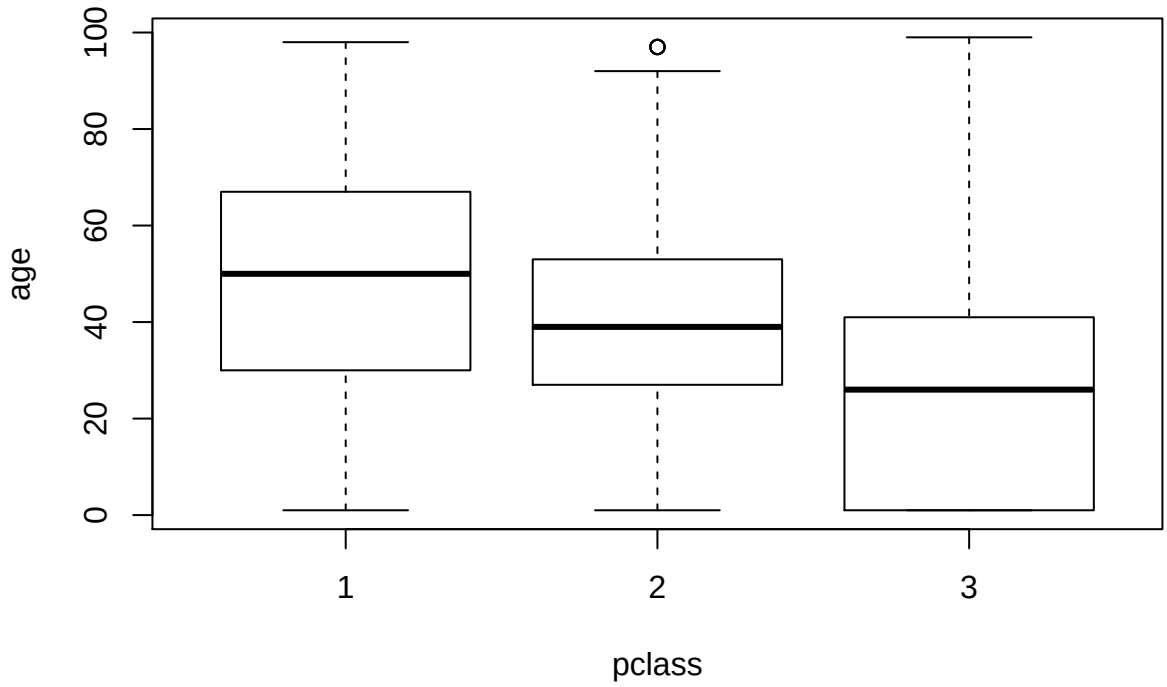
Este tipo de gráfico sí es mucho mas sugerente y adecuado que los demás ya que se ve mucho mas claramente la distribución de cada clase en cada cabina.

Edad vs Clase

Como ya convertimos las variables 'age' y 'pclass' a las clases correctas podemos realizar un gráfico adecuado para analizar esta relación.

Vamos a ver directamente los gráficos adecuados. Edad es una variable numérica y clase es categórica, con lo cual el adecuado sería el gráfico de caja.

```
boxplot(age~pclass)
```


age
0
20
40
60
80
100
1
2
3
pclass

12. Respecto a la relación entre la edad y la tarifa podemos pensar que las personas más jóvenes tenían menos dinero y por ende compraron los tiquetes más baratos. Puede confirmar esto en base a los datos?

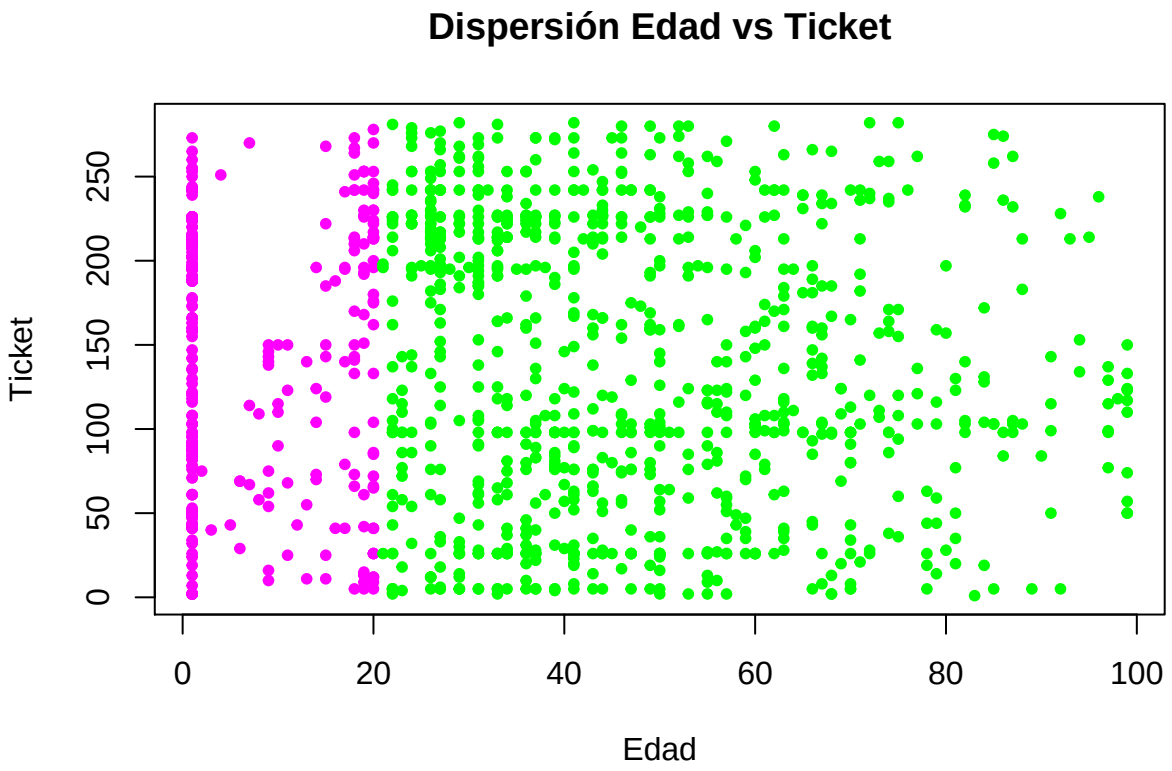
Edad vs Tarifa

Comparemos ahora estas dos variables numéricas en un gráfico de dispersión.

Primero notar que hacer una tabla con los datos que vamos a relacionar no tiene sentido porque como ambas son numéricas estaríamos creando filas y columnas para valores que no representan nada más que un sólo punto.

Realicemos un gráfico de dispersión

```
plot(age,fare, xlab='Edad',ylab='Ticket',
     main='Dispersión Edad vs Ticket',
     type='n')
# al poner type='n' sólo dibujamos la caja dónde vamos a agregar los puntos
# por separado a continuación.
points(age[age<=20], fare[age<=20], pch=20, col='magenta')
points(age[age>20],fare[age>20],pch=20,col='green')
```




```
# ahora agregamos estos puntos a la caja vacía
# pch (peach) establece el tipo de punto (circulo, triangulo, relleno, vacío, etc.) (Ver ?plot para mas
```

Observamos que hay una franja de menores de edad que compraron todas las variedades de precios de los tickets. Además hay compras en todas los tickets en la franja de edad de entre 0 y 20 años, con lo cual no podemos confirmar la idea de que los jóvenes tenían menos dinero y por tanto sus compras iban a ser de boletos baratos.