Máster en Ciencia de Datos

M2.951 – Tipología y ciclo de vida de los datos
Semestre 2019/2

Práctica 2
Limpieza y validación de datos

Emilio Arenas Tienda

Tabla de contenidos

1 Introducción.

1.1 Descripción del dataset.

Los datos escogidos para esta práctica es el dataset *Titanic: Machine Learning from Disaster*, disponible kaggle.

El dataset dispone de los siguientes atributos:

- **PassengerId**: Variable identificativa única para cada pasajero.
- **Name**: Nombre del pasajero.
- **Ticket**: Código identificativo del billete del pasajero.
- **Survived**: Variable respuesta del estudio.
  - 1 = *Sobrevivió*.
  - 0 = *No sobrevivió*.
- **Pclas**: Clase en la que viajaba el pasajero.
  - 1 = *Primera clase*.
  - 2 = *Segunda clase*.
  - 3 = *Tercera clase*.
- **Sex**: Sexo del pasajero.
- **Age**: Edad del pasajero (variable numérica continua ya que contiene fracciones).
- **SibSp**: No de hermanos/cónyuges acompañaba al pasajero.
- **Parch**: No de padres/hijos acompañaba al pasajero.
- **Fare**: Precio del billete.
- **Cabin**: Camarote en el que viajaba el pasajero.
- **Embarked**: Puerto de embarque del pasajero.
  - C = *Cherbourg*.
  - Q = *Queenstown*.
  - S = *Southampton*.

1.2 Motivación.

Los atributos del dataset presentado anteriormente, pertenecen a una competición para desarrollar un modelo predictivo que responda a la siguiente pregunta ¿Qué tipo de personas tenían más probabilidades de vivir?

Para dar una respuesta a esta pregunta, en un primer lugar y con estas variables, realizaremos un análisis exploratorio que nos sirva para familiarizarnos con los datos, que rango de valores tienen, como son su distribución, la existencia de valores atípicos, etc.

En un segundo lugar, analizaremos las relaciones entre variables y evaluaremos si estas son significativas para decidir si alguien sobrevive a la tragedia.

En último lugar plantearemos diferentes modelos para escoger finalmente aquel que tenga un mejor desempeño.

2 Limpieza de los datos

2.1 Carga de datos.

Los datos para descargar están disponibles en tres ficheros *.csv*, el *train*, *test* y su *target*. Lo que haremos ahora, será cargar todos los ficheros y unirlos en uno solo para su posterior procesado conservando la procedencia de cada caso.

```
#Cargo los tres ficheros en una lista.
datos<-sapply(list.files(pattern=".csv"),
function(k)read.csv(k,na.strings ="",stringsAsFactors=F))

names(datos)<-c("target_test","test","train")

#Fusiono la información del test con la información del target
datos$test<-merge(datos$target_test,datos$test,by="PassengerId")

#Creo una variable sample para el data.frame test y train que
#me indica la procedencia de cada caso.
datos$test$sample<-1
datos$train$sample<-0

#Finalmente junto los data.frame test y train.
df<-rbind(datos$test,datos$train)
rm(datos)

#Comprobamos la correcta importación de los datos.
knitr::kable(head(df[,c(-4,-13)]))
```

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
892	0	3	male	34.5	0	0	330911	7.8292	NA	Q
893	1	3	female	47.0	1	0	363272	7.0000	NA	S
894	0	2	male	62.0	0	0	240276	9.6875	NA	Q
895	0	3	male	27.0	0	0	315154	8.6625	NA	S
896	1	3	female	22.0	1	1	3101298	12.2875	NA	S
897	0	3	male	14.0	0	0	7538	9.2250	NA	S

Comprobada la correcta importación y fusión de los datos los guardaremos en un fichero *.csv* para no tener que realizar el mismo proceso cada vez que necesitemos los datos.

```
write.table(df,"titanic_original.csv",col.names = T,row.names =F,
sep=",",dec=".")
```

Ahora pasaremos a analizar su estructura del archivo de datos.

```
str(df)
```

```
> 'data.frame': 1309 obs. of  13 variables:
>  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
>  $ Survived   : int  0 1 0 0 1 0 1 0 1 0 ...
>  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
>  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)"
"Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...
>  $ Sex        : chr  "male" "female" "male" "male" ...
>  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
>  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
>  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
>  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
>  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
>  $ Cabin      : chr  NA NA NA NA ...
>$ Embarked   : chr  "Q" "S" "Q" "S" ...
>  $ sample     : num  1 1 1 1 1 1 1 1 1 1 ...
```

El data.frame está formado por 1309 observaciones (pasajeros) y 13 variables de tipos diferentes: enteros (*int*), numéricas (*num*) y texto (*chr*).

Las variables *Survived*, *Pclass*, *Sex*, *Cabin*, *Embarked*, de naturaleza categórica, serán transformadas en *factor*.

```
df$Survived<-factor(df$Survived,levels=c(0,1),labels=c("No Surv","Surv"))

df$Pclass<-factor(df$Pclass,levels=c(3,2,1),labels=c("3rd","2nd","1st"))

df$Sex<-factor(df$Sex,levels=c("male","female"),labels=c("male","female"))

df$Embarked<-factor(df$Embarked,levels=c("C","Q","S"),
labels=c("Cherbourg","Queenstown","Southampton"))

df$Cabin<-as.factor(df$Cabin)
```

Los atributos *PassengerId*, *Name* y *Ticket*, por su naturaleza identificativa de la observación no nos será de gran utilidad, así que por el momento las aislaremos del data.frame principal. Del mismo modo, la variable *sample*, solo nos será de utilidad llegado el momento de construir un modelo, por lo tanto, también la aislamos.

```
#descarto variables un data.frame diferente al principal
drops<-df[,c("PassengerId","Name","Ticket")]
sample<-df$sample

df[,c("PassengerId","Name","Ticket","sample")]<-NULL
```

2.2 Calidad de los datos: Valores ausentes.

Para una primera aproximación extraemos los principales estadísticos descriptivos de cada atributo.

```
for (i inlist(c(1:5),c(6:9))){
print(summary(df[,i]))
cat("\n")
}
```

```
>     Survived   Pclass        Sex           Age            SibSp
>  No Surv:815   3rd:709   male  :843   Min.   : 0.17   Min.   :0.0000
>  Surv   :494   2nd:277   female:466   1st Qu.:21.00   1st Qu.:0.0000
>                1st:323                Median :28.00   Median :0.0000
>                                       Mean   :29.88   Mean   :0.4989
>                                       3rd Qu.:39.00   3rd Qu.:1.0000
>                                       Max.   :80.00   Max.   :8.0000
>                                       NA's   :263
>
>      Parch            Fare                      Cabin             Embarked
>  Min.   :0.000   Min.   :  0.000   C23 C25 C27    :   6   Cherbourg  :270
>  1st Qu.:0.000   1st Qu.:  7.896   B57 B59 B63 B66:   5   Queenstown :123
>  Median :0.000   Median : 14.454   G6             :   5   Southampton:914
>  Mean   :0.385   Mean   : 33.295   B96 B98        :   4   NA's       :  2
>  3rd Qu.:0.000   3rd Qu.: 31.275   C22 C26        :   4
>  Max.   :9.000   Max.   :512.329   (Other)        : 271
>                  NA's   :1         NA's           :1014
```

Del resumen anterior podemos ver entre otras cosas que:

- El 37.7% del pasaje sobrevivió al desastre.
- El 54.2% del pasaje viajaba en tercera clase, mientras que el 24.7% lo hacía en primera.
- El 64.4% del pasaje eran hombres.

En lo que respecta a la calidad de los datos, podemos apreciar que las variables *Embarked*, *Fare*, *Age* y *Cabin* padecen de valores faltantes (*NA's*). Las dos primeras no parece nada preocupante, es una cantidad muy pequeña de casos, aunque hay que señalar que la variable *Fare* tiene casos con valores a *0* que habrá que considerar como valores ausentes. En cambio las variables *Age* y *Cabin* tienen una cantidad importante de valores ausentes.

```
#Paso a missing los valores 0
df$Fare[df$Fare==0]<-NA

tabla<-sapply(df[,c("Embarked","Fare","Age","Cabin")],function(k) sum(is.na(k)))
tabla<-cbind("Freq"=tabla,
"Pct"=round(tabla/nrow(df)*100,1))

knitr::kable(tabla)
```

	Freq	Pct
Embarked	2	0.2
Fare	18	1.4
Age	263	20.1
Cabin	1014	77.5

Los valores ausentes de *Embarked* solo suponen un 0.2% del total, y después de pasar a *missing* los *0's* en *Fare* tenemos un total de 17 valores ausentes, lo que supone un 1.4% del total. En ambos casos, al ser un valor tan residual, las decisiones que tomemos respecto a solventar estos inconvenientes tendrán muy poco impacto en el resultado de los análisis.

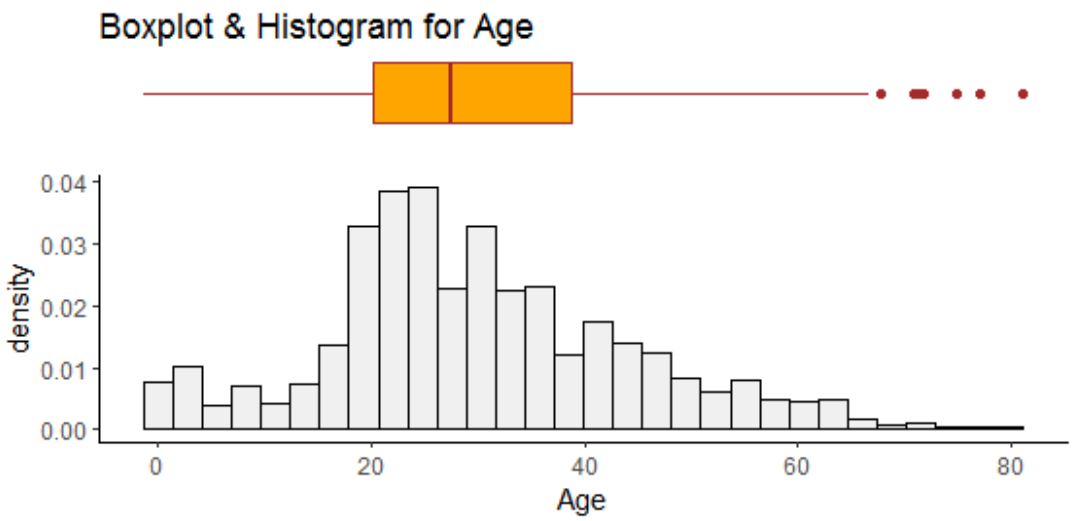
Por otro lado las variables *Age* y *Cabin* tienen una importante cantidad de valores ausente. En concreto, *Cabin* tiene el 77.5% de los casos sin información, así que optamos por prescindir de la variable.

```
drops$Cabin<-df$Cabin
df$Cabin<-NULL
```

En cambio *Age* tiene un 20% de casos a missing, una cantidad considerable a imputar que pueden causar sesgos en los análisis venideros. Pero si optamos por eliminarlos perdemos una cantidad importante de información.

De manera contraria que hemos hecho con *Cabin* vamos a optar por no deshacernos de nada y vamos imputar valores, así que antes de cualquier modificación veremos la distribución de la variable.

```
g1<-ggplot(df,aes(y=Age))+
geom_boxplot(color="brown",fill="orange")+coord_flip()+
labs(y=element_blank())+
ggtitle("Boxplot & Histogram for Age")+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
panel.background =element_blank())

g2<-ggplot(df,aes(x=Age))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
labs(x ="Age")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

(g1/g2)+plot_layout(heights=c(1, 4))
```

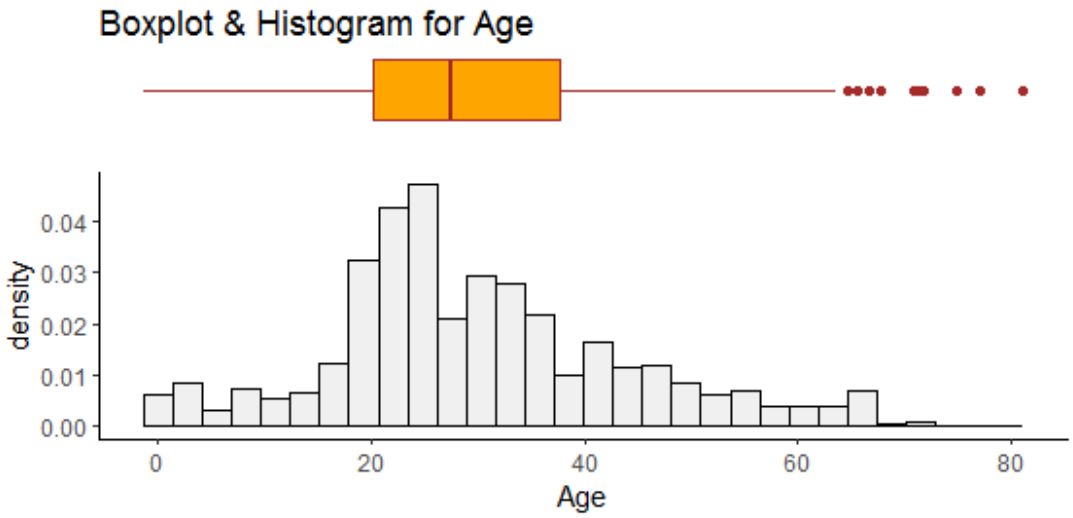


Apreciamos que hay cierta asimetría debido a que la cola de la derecha se prolonga debido en parte a la presencia de unos valores atípicos.

La imputación de estos valores ausentes la realizaremos mediante criterios de similitud entre los casos: la imputación basada en k vecinos más próximos (en inglés, knn-imputation). La elección de esta alternativa se realiza bajo la hipótesis de que nuestros registros guardan cierta relación.

```
temp<-df


#La imputacion la hacemos de las variables Age, Fare y Embarked
temp<-VIM::knn(temp,
variable=c("Age","Fare","Embarked"),
k=3,
dist_var=c("Survived","Pclass",
"Sex","SibSp","Parch","Fare","Embarked"))

#Conservo la variable edad original
drops$Age<-df$Age

#Reemplazo las variables con valores ausentes.
df[,c("Age","Fare","Embarked")]<-temp[,c("Age","Fare","Embarked")]
rm(temp)

g1<-ggplot(df,aes(y=Age))+
geom_boxplot(color="brown",fill="orange")+coord_flip()+
labs(y=element_blank())+
ggtitle("Boxplot & Histogram for Age")+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
panel.background =element_blank())
```

```
g2<-ggplot(df,aes(x=Age))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
labs(x ="Age")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

(g1/g2)+plot_layout(heights=c(1, 4))
```

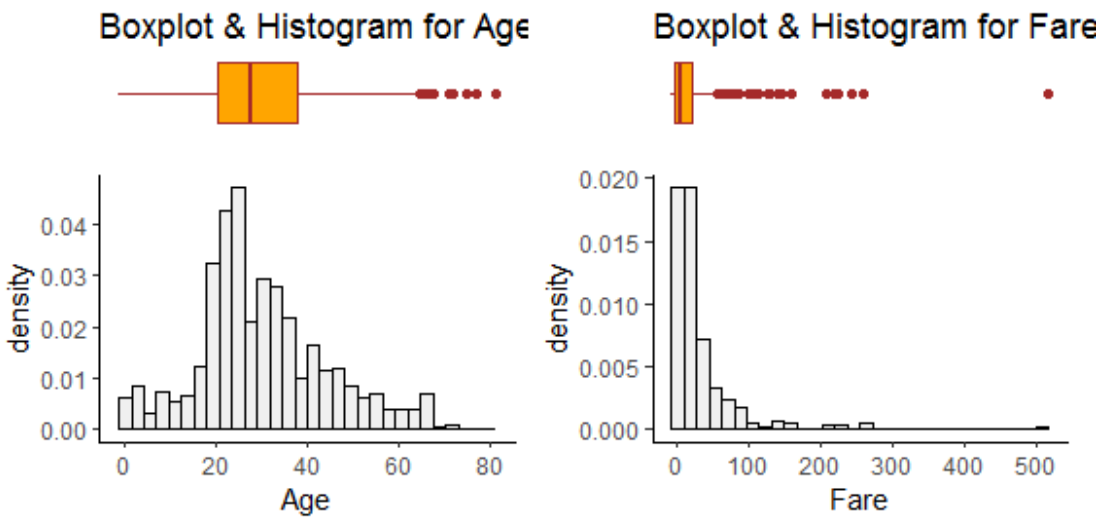


La imputación de los valores en la variable *Age* no ha causado un sesgo importante en su distribución.

2.3 Calidad de los datos: Valores extremos.

Ahora pasaremos a valorar la presencia de valores extremos que puedan existir en las variables numéricas como *Age* y *Fare*.

```
g3<-ggplot(df,aes(y=Fare))+
geom_boxplot(color="brown",fill="orange")+coord_flip()+
labs(y=element_blank())+
ggtitle("Boxplot & Histogram for Fare")+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
panel.background =element_blank())

g4<-ggplot(df,aes(x=Fare))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
labs(x ="Fare")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())


((g1/g2)+plot_layout(heights=c(1, 4)))|
((g3/g4)+plot_layout(heights=c(1, 4)))
```



La gráfica de edad no es nueva y ya sabíamos que tenía valores extremos. Dichos valores no parecen que puedan proceder de algún tipo de error, ya que son edades aberrantes, a decir verdad, son valores plausibles, simplemente son valores que se alejan del centro de la distribución.

La variable *Fare* si que acusa más de valores extremos. A partir del histograma como del boxplot podemos apreciar la asimetría de la distribución con una larga cola que se extiende hacia la derecha donde se hallan los valores extremos. Hay unos puntos en concretos que se no solo se alejan del centro de la distribución, sino del resto de los outliers, que son los que tienen un valor superior a 100 en *Fare*. Los seleccionamos y miramos con algo más de detenimiento.

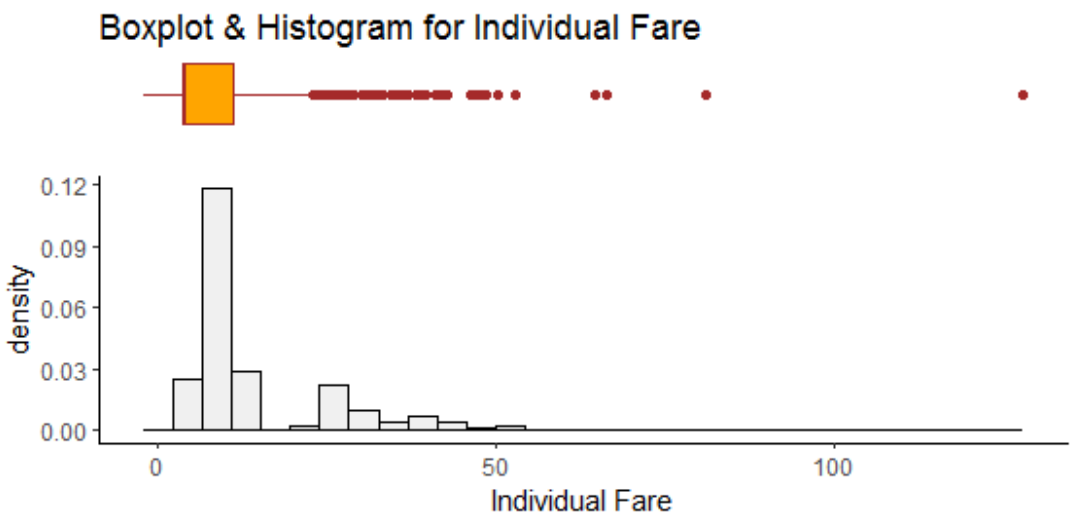
```
knitr::kable(head(df[df$Fare>100,],10),row.names=F)
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
Surv	1st	female	48.0	1	3	262.3750	Cherbourg
Surv	1st	female	28.0	3	2	263.0000	Southampton
Surv	1st	female	36.0	0	0	262.3750	Cherbourg
No Surv	1st	male	13.0	2	2	262.3750	Cherbourg
Surv	1st	female	60.0	1	4	263.0000	Southampton
Surv	1st	female	35.0	0	0	211.5000	Cherbourg
No Surv	1st	male	32.5	0	0	211.5000	Cherbourg
No Surv	1st	male	67.0	1	0	221.7792	Southampton
Surv	1st	female	63.0	1	0	221.7792	Southampton
Surv	1st	female	33.0	0	0	151.5500	Southampton

Aunque solo son los 10 primeros de 84 casos, podemos ver que hay más mujeres que hombres, también vemos que son de primera clase, lo cual parece razonable, pues el

precio debería estar relacionado con el status. También llama la atención, que el atributo *Fare* al ser continuo tenga valores repetidos al decimal, esto junto a que las mujeres de la época era raro que viajaran solas, hace pensar que *Fare* registra el precio de todos los acompañantes. Por lo tanto, pasaremos a calcular tarifa individual.

```
n_pasajes<-table(drops$Ticket)
n_pasajes<-as.vector(n_pasajes[drops$Ticket])
df$Fare_unit<-df$Fare/n_pasajes

g1<-ggplot(df,aes(y=Fare_unit))+
geom_boxplot(color="brown",fill="orange")+coord_flip()+
labs(y=element_blank())+ggtitle("Boxplot & Histogram for Individual Fare")+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
panel.background =element_blank())

g2<-ggplot(df,aes(x=Fare_unit))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
labs(x ="Individual Fare")+theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

(g1/g2)+plot_layout(heights=c(1, 4))
```




```
drops$Fare<-df$Fare
df$Fare<-NULL
```

Vemos que hay dos poblaciones, los que han pagado menos de 25 (la inmensa mayoría) y los que han pagado entre 25 y 60. Los que ya superan los 60 son valores atípicos, que consideraremos legítimos ya que pertenecen a primera clase.

3 Análisis de datos.

El objetivo de esta sección es la de plantear hipótesis que permitan determinar que características de los pasajeros hacen más probable la supervivencia al accidente.

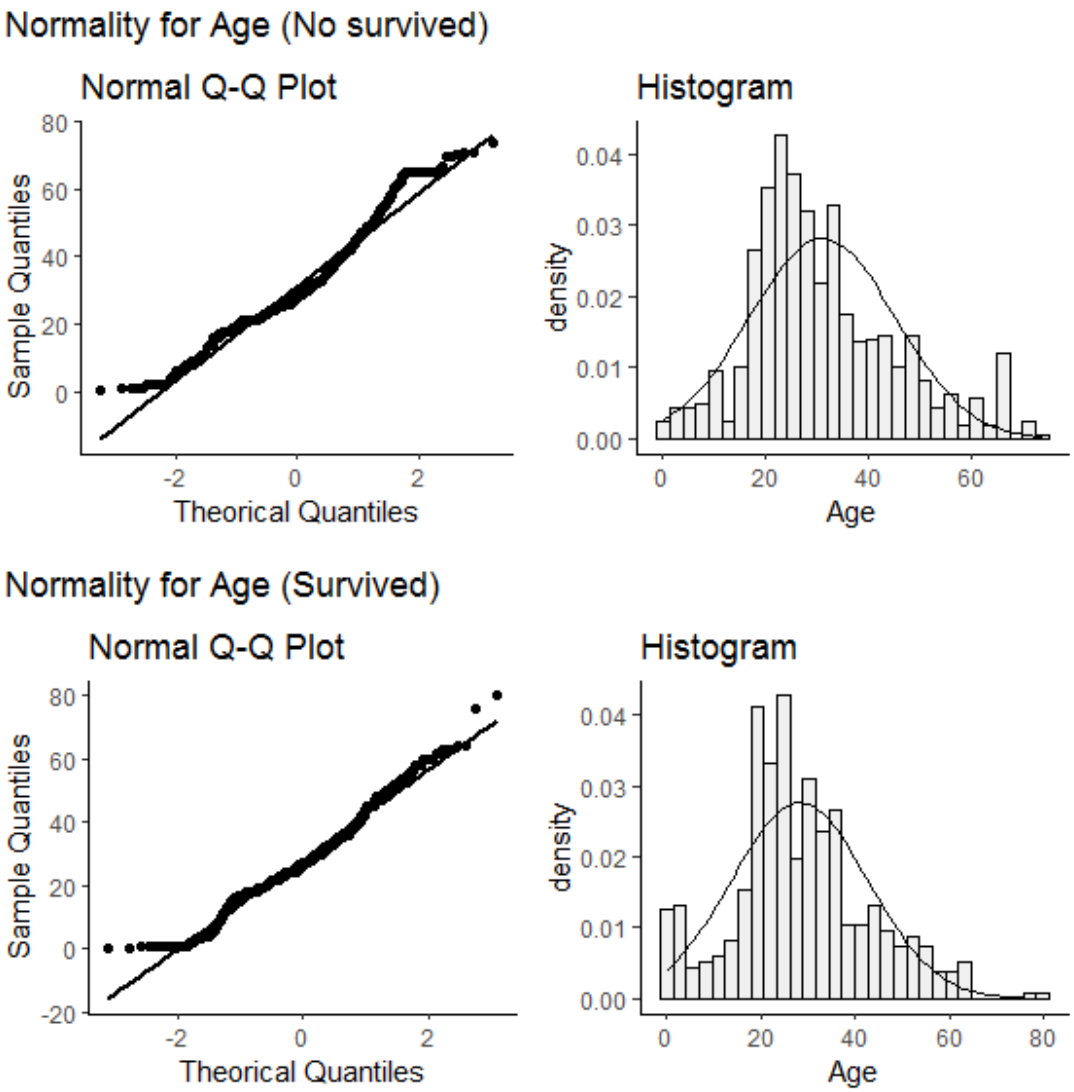
Para las variables categóricas como *Pclass* o *Sex* realizaremos test de independencia, que permitan valorar si la frecuencia de observaciones es significativamente distinta entre los grupos de estudio.

Para las variables continuas como *Age* y *Fare_unit* nos gustaría comprobar si hay diferencias en media entre las dos poblaciones de interés, supervivientes y no supervivientes.

3.1 Comprobación de la normalidad y de homogeneidad de la varianza.

Antes de testear ninguna hipótesis que nos podamos plantear, debemos de comprobar las suposiciones de normalidad y homogeneidad de varianzas de los grupos para poder llevar a cabo el test más adecuado.

```
grupos<-split(df[,"Age",drop=F],df$Survived)
names(grupos)<-c("No survived","Survived")

for (i inc("No survived","Survived")){

k<-grupos[[i]]

datos<-as.data.frame(qqnorm(k$Age,plot=F))

g1<-ggplot(datos,aes(x=x,y=y))+
geom_point() +
geom_smooth(method="lm",se=FALSE,col="black")+
labs(x="Theorical Quantiles",y="Sample Quantiles")+
ggtitle("Normal Q-Q Plot")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

g2<-ggplot(k,aes(x=Age))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
stat_function(fun = dnorm,args =list(mean =mean(k$Age),sd =sd(k$Age)))+
labs(x ="Age")+ggtitle("Histogram")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

print((g1|g2)+plot_annotation(title=paste0("Normality for Age (",i,")")))
}
```



Las gráficas de distribución con la curva de una distribución normal muestran como las curvas de densidad quedan por debajo del histograma en los valores centrales y por lo tanto no podemos asumir normalidad.

El grafico Q-Q plot es una representación gráfica que sirve para comparar dos distribuciones, la de nuestros datos y la de los valores teóricos de una distribución normal estándar. Si ambas distribuciones coinciden, es decir, si nuestros datos se distribuyen normalmente, veremos un grafico muy parecido a una línea recta. En nuestro caso, vemos como los extremos de la línea de puntos se aleja de la línea recta, lo que nos hace rechazar la hipótesis de normalidad.

A la misma conclusión con test más formales como el test de normalidad de Shapiro-Wilk.

```
lapply(grupos,function(k){
shapiro.test(k$Age)
})
```

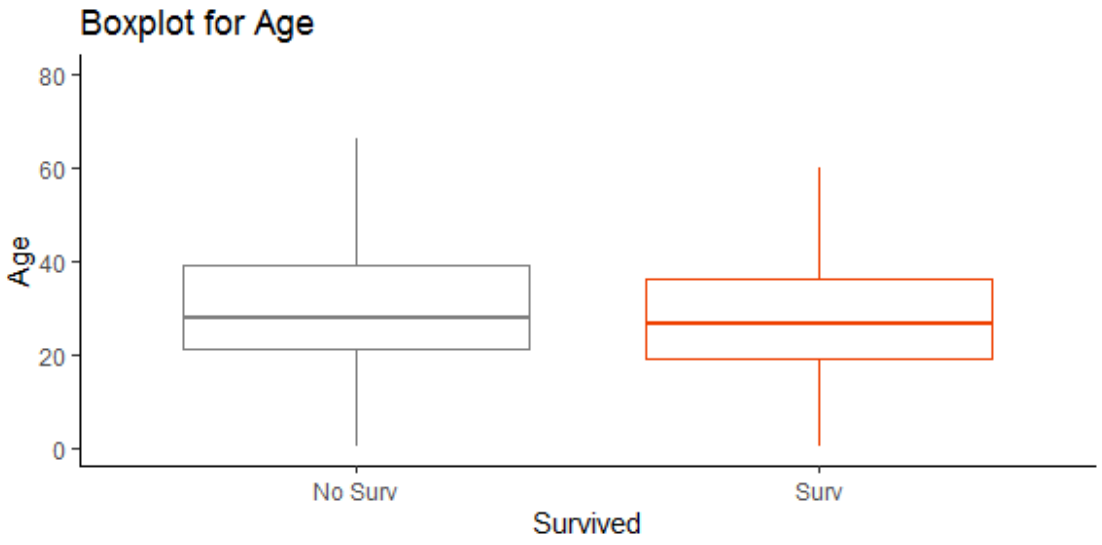
```
> $`No survived`
>
>   Shapiro-Wilk normality test
>
> data:  k$Age
> W = 0.95857, p-value = 2.089e-14
>
>
> $Survived
>
>   Shapiro-Wilk normality test
>
> data:  k$Age
> W = 0.97608, p-value = 3.122e-07
```

Con un p.valor inferior a 0.05, rechazamos la hipótesis nula de normalidad.

Ahora comprobaremos la hipótesis de igualdad de varianzas. Del mismo modo que antes, nos basaremos en gráficos y en test formales.

Por medio de un boxplot podemos ver si hay igualdad de varianzas.

```
ggplot(df, aes(x = Survived, y = Age, color = Survived)) +
geom_boxplot(outlier.shape =NA) +ggtitle("Boxplot for Age")+
scale_color_manual(values =c("gray50", "orangered2")) +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
legend.position="none",
panel.background =element_blank())
```



Podemos apreciar que las cajas tienen el mismo tamaño, esto nos hace sospechar que exista igualdad de varianzas. De todos modos aplicaremos un test formal que nos de mas seguridad a la hora de responder. Al no admitir la normalidad de la variable en ambas poblaciones, recomendamos aplicar el test no perimétrico *Fligner-Killeen*.

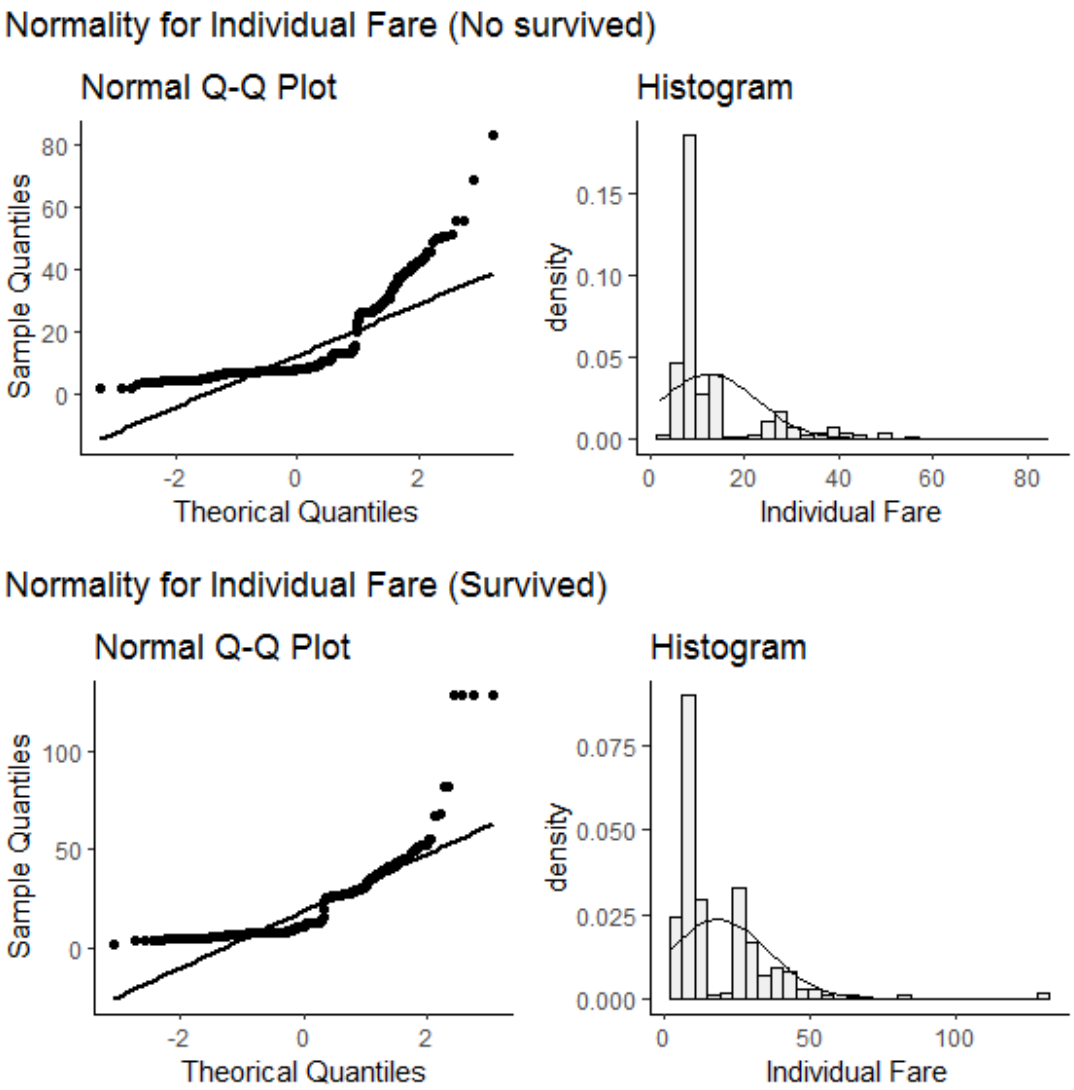
```
fligner.test(Age~Survived,data=df)
```

```
>
>   Fligner-Killeen test of homogeneity of variances
>
> data:  Age by Survived
> Fligner-Killeen:med chi-squared = 0.40313, df = 1, p-value = 0.5255
```

Con un p.valor superior a 0.05 no podemos rechazar la hipótesis nula de igualdad de varianzas.

Ahora realizamos lo propio con la variable *Fare_unit*.

```
grupos<-split(df[,"Fare_unit",drop=F],df$Survived)
names(grupos)<-c("No survived","Survived")

for (i inc("No survived","Survived")){

k<-grupos[[i]]

datos<-as.data.frame(qqnorm(k$Fare_unit,plot=F))

g1<-ggplot(datos,aes(x=x,y=y))+
geom_point() +
geom_smooth(method="lm",se=FALSE,col="black")+
labs(x="Theorical Quantiles",y="Sample Quantiles")+
ggtitle("Normal Q-Q Plot")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

g2<-ggplot(k,aes(x=Fare_unit))+
geom_histogram(aes(y =..density..),color="black",fill="gray94")+
stat_function(fun = dnorm,
args =list(mean =mean(k$Fare_unit),sd =sd(k$Fare_unit)))+
labs(x ="Individual Fare")+ggtitle("Histogram")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

print((g1|g2)+plot_annotation(
title=paste0("Normality for Individual Fare (",i,")")))

}
```



Gráficamente llegamos a la conclusión de que la variable no sigue una distribución normal en ninguna de las dos poblaciones, aunque si sigue la misma distribución, el histograma de ambas poblaciones tienen, en mayor o menor medida, la misma forma.

A la misma conclusión llegaremos con el test de *Shapiro-Wilk*.

```
lapply(grupos,function(k){
shapiro.test(k$Fare_unit)
})
```

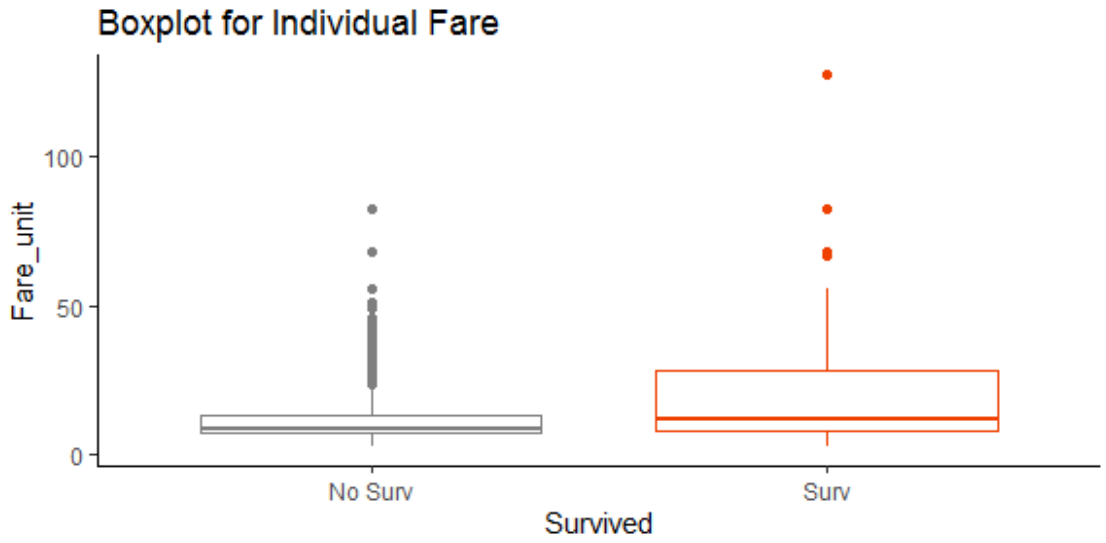
```
> $`No survived`
>
>   Shapiro-Wilk normality test
>
> data:  k$Fare_unit
> W = 0.65978, p-value < 2.2e-16
>
>
```

```
> $Survived
>
>   Shapiro-Wilk normality test
>
> data:  k$Fare_unit
> W = 0.72085, p-value < 2.2e-16
```

Con un p.valor inferior a 0.05, rechazamos la hipótesis nula de normalidad.

Ahora es el turno de comprobar la hipótesis de igualdad de varianzas.

```
ggplot(df, aes(x = Survived, y = Fare_unit, color = Survived)) +
geom_boxplot() +
ggtitle("Boxplot for Individual Fare")+
scale_color_manual(values =c("gray50", "orangered2")) +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
legend.position="none",
panel.background =element_blank())
```



Podemos apreciar que las cajas difieren mucho en tamaño, esto nos hace sospechar que no exista igualdad de varianzas. Al igual que antes realizamos el test no perimétrico *Fligner-Killeen*.

```
fligner.test(Fare_unit~Survived,data=df)
```

```
>
>   Fligner-Killeen test of homogeneity of variances
>
> data:  Fare_unit by Survived
> Fligner-Killeen:med chi-squared = 156.17, df = 1, p-value < 2.2e-16
```

Con un p.valor muy inferior a 0.05, rechazamos la hipótesis nula de igualdad de varianzas.

3.2 Contraste de hipótesis: Variables cuantitativas.

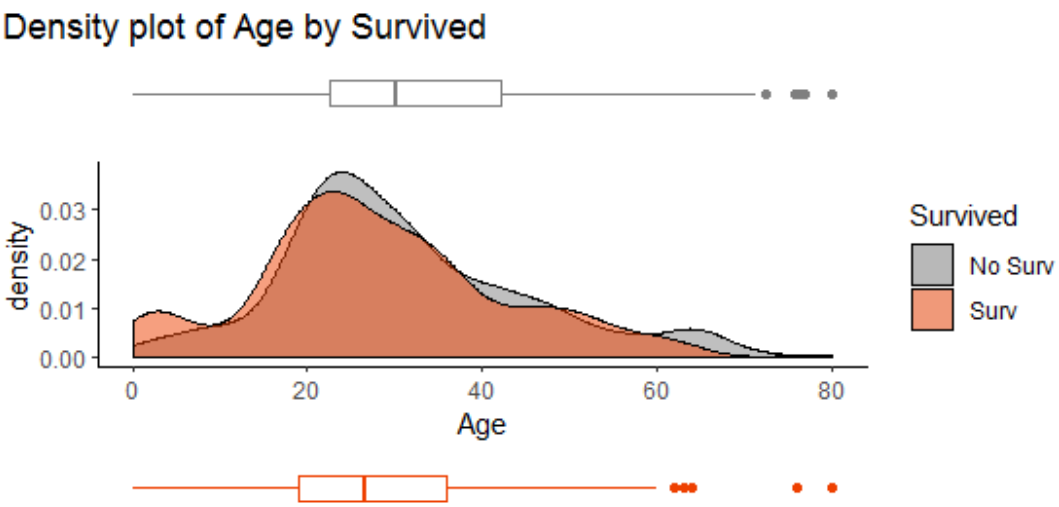
El objetivo del estudio es predecir que pasajeros sobrevivieron y cuáles no, el análisis de cada variable se hace en relación a la variable respuesta *Survived*.

Analizando los datos de esta forma, se pueden empezar a extraer ideas sobre que variables están más relacionadas con la supervivencia.

En esta sección pretendemos extraer ideas sobre que variables están más relacionadas con la supervivencia y las testearemos formalmente para ver si estas relaciones son significativas.

Anteriormente habíamos graficado por separado la distribución de la variable *Age* con la finalidad de ver si se distribuía como una normal en cada una de sus categorías. Como la finalidad ahora es ver si hay diferencias de edad entre las categorías de la variable objetivo las visualizaremos en el mismo grafico.

```
g1<-ggplot(df[df$Survived=="No Surv",],
aes(x = Survived, y = Age, color = Survived)) +
geom_boxplot() +scale_color_manual(values =c("gray50"))+
coord_flip()+
labs(y=element_blank(),x=element_blank())+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
legend.position="none",panel.background =element_blank())


g2<-ggplot(df, aes(x = Age, fill = Survived)) +
geom_density(alpha =0.5) +
scale_fill_manual(values =c("gray50", "orangered2")) +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())

g3<-ggplot(df[df$Survived=="Surv",],
aes(x = Survived, y = Age, color = Survived)) +
geom_boxplot() +scale_color_manual(values =c("orangered2"))+
coord_flip()+labs(y=element_blank(),x=element_blank())+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
legend.position="none",panel.background =element_blank())

(g1/g2/g3)+plot_layout(heights=c(1, 5, 1))+plot_annotation(title ="Density plot
of Age by Survived")
```



Aunque la distribución de edad son bastantes similares entre las categorías de la variable objetivo, se aprecia como el porcentaje de supervivientes es mayor en el intervalo de edad de 0 a 10 años, y que el porcentaje de fallecidos es mayor para los mayores de 60 años. Esto provoca la diferencia (en medianas) que señalan los boxplots.

Llegados a este punto ¿Podemos admitir que los que sobrevivieron al accidente eran en promedio más jóvenes que los que sucumbieron a la tragedia?

Formalmente esta hipótesis se plantea del siguiente modo:

$$H_0: \mu_{surv} - \mu_{no\ surv} = 0$$

$$H_1: \mu_{surv} - \mu_{no\ surv} < 0$$

Teniendo en cuenta que las poblaciones tienen la misma varianza y que la variable no se distribuye normalmente en ambas poblaciones podemos, podemos hacer uso del test no paramétrico *Mann-Whitney-Wilcoxon*.

```
wilcox.test(x = df$Age[df$Survived=="Surv"],
y = df$Age[df$Survived=="No Surv"],
alternative ="less", mu =0,
paired =FALSE, conf.int =0.95)
```

```
>
>   Wilcoxon rank sum test with continuity correction
>
> data:  df$Age[df$Survived == "Surv"] and df$Age[df$Survived == "No Surv"]
> W = 181043, p-value = 0.001116
> alternative hypothesis: true location shift is less than 0
> 95 percent confidence interval:
>       -Inf -1.000027
> sample estimates:
> difference in location
>              -2.000044
```

En vista de los resultados y con un p.valor de 0.001 (inferior a 0.05), rechazamos la hipótesis nula de igualdad a favor de que los supervivientes eran mas jóvenes que los que no sobrevivieron.

Podríamos haber hecho uso del contrate clásico de la *t.student* a pesar del no cumplimiento de normalidad. Este contraste es bastante robusto a la falta de normalidad y debido a la gran muestra que disponemos en ambas poblaciones y a lo que postula el *teorema central del limite* podríamos dar por fiables sus resultados.

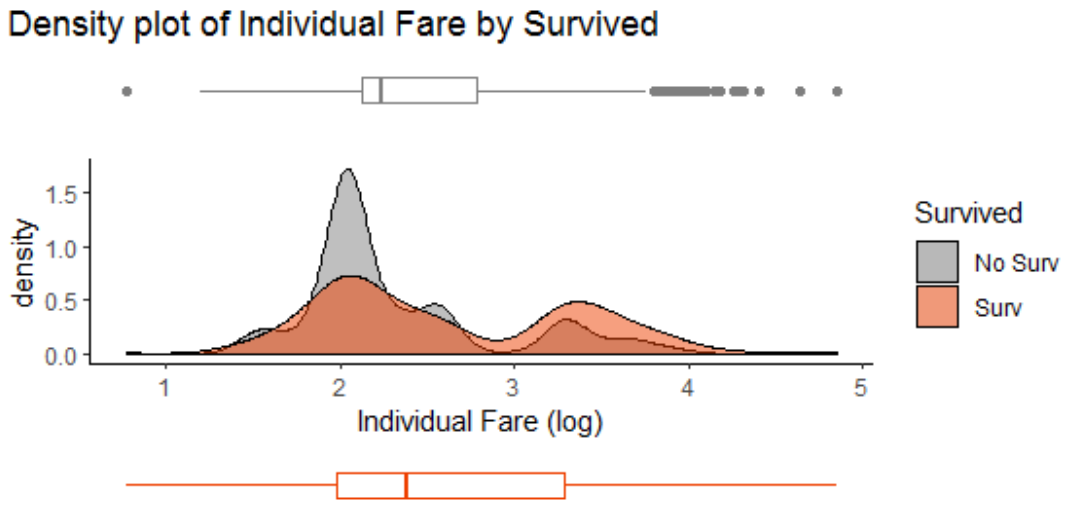
```
t.test(x = df$Age[df$Survived=="Surv"],
y = df$Age[df$Survived=="No Surv"],
alternative ="less", mu =0,
paired =FALSE, conf.int =0.95,var.equal = T)

>
>   Two Sample t-test
>
> data:  df$Age[df$Survived == "Surv"] and df$Age[df$Survived == "No Surv"]
> t = -3.3178, df = 1307, p-value = 0.0004662
> alternative hypothesis: true difference in means is less than 0
> 95 percent confidence interval:
>       -Inf -1.359981
> sample estimates:
> mean of x mean of y
>  28.22725  30.92627
```

Este test nos lleva a la misma conclusión. Por lo tanto concluimos que la media de edad de los que sobrevivieron al desastre era significativamente menor a los que perecieron.

Ahora hacemos lo propio con el precio del billete, pero antes, recordemos que *Fare_unit* tiene una distribución asimétrica, muchos billetes tenían un coste bajo y unos pocos un coste alto. Mediante una transformación logarítmica visualizaremos mejor las diferencias entre las categorías de la variable objetivo.

```
g1<-ggplot(df[df$Survived=="No Surv",],
aes(x = Survived, y =log(Fare_unit), color = Survived)) +
geom_boxplot() +scale_color_manual(values =c("gray50"))+
coord_flip()+labs(y=element_blank(),x=element_blank())+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
legend.position="none",panel.background =element_blank())


g2<-ggplot(df, aes(x =log(Fare_unit), fill = Survived)) +
geom_density(alpha =0.5) +labs(x="Individual Fare (log)")+
scale_fill_manual(values =c("gray50", "orangered2")) +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```

```
g3<-ggplot(df[df$Survived=="Surv",],
aes(x = Survived, y =log(Fare_unit), color = Survived)) +
geom_boxplot() +
scale_color_manual(values =c("orangered2"))+
coord_flip()+
labs(y=element_blank(),x=element_blank())+
theme(axis.line.x =element_blank(),axis.line.y =element_blank(),
axis.text.x=element_blank(),axis.text.y=element_blank(),
axis.ticks.x=element_blank(),axis.ticks.y=element_blank(),
legend.position="none",panel.background =element_blank())

(g1/g2/g3)+plot_layout(heights=c(1, 5, 1))+plot_annotation(title ="Density plot
of Individual Fare by Survived")
```



El grafico de densidades muestra que la proporción de supervivientes es mayor en aquellos que pagaron una cantidad mayor por sus billetes, y en media, parece, que los que sobrevivieron pagaron más por su billete. ¿Podemos admitir entonces que los que sobrevivieron al accidente pagaron más por sus billetes?

Formalmente esta hipótesis se plantea del siguiente modo:

$$H_0: \mu_{surv} - \mu_{no\ surv} = 0$$

$$H_1: \mu_{surv} - \mu_{no\ surv} > 0$$

En esta ocasión testearemos la hipótesis por medio del contraste clásico de la *t.student*, aunque la variable no se distribuya como una normal, si que disponemos de bastante muestra para obtener resultados robustos.

```
t.test(x = df$Fare_unit[df$Survived=="Surv"],
y = df$Fare_unit[df$Survived=="No Surv"],
alternative ="greater", mu =0,
paired =FALSE, conf.int =0.95,var.equal = F)
```

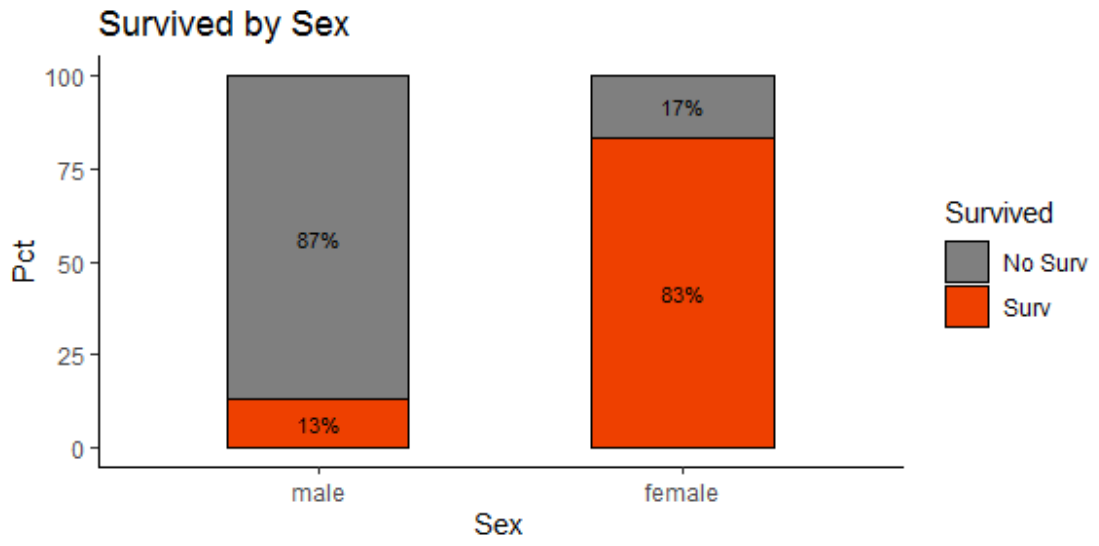
```
>
>   Welch Two Sample t-test
>
> data:  df$Fare_unit[df$Survived == "Surv"] and df$Fare_unit[df$Survived == "No
Surv"]
> t = 7.6576, df = 706.6, p-value = 3.123e-14
> alternative hypothesis: true difference in means is greater than 0
> 95 percent confidence interval:
>  5.063952      Inf
> sample estimates:
> mean of x mean of y
>  18.93800  12.48643
```

A la vista de los resultados podemos concluir que los que sobrevivieron al desastre pagaron más, en media, que los que no sobrevivieron.

3.3 Contraste de hipótesis: Variables cualitativas.

Ahora es el turno de probar la asociación entre las variables categóricas y la variable respuesta del dataset. Empezaremos por la variable *Sex*, lo que queremos vislumbrar es si las proporciones de la variable *Survived* son diferentes dependiendo del valor que adquiera la variable *Sex*.

```
tabla<-as.data.frame(round(prop.table(table(df$Survived,df$Sex),2)*100))
colnames(tabla)<-c("Survived","Sex","Pct")

ggplot(tabla, aes(x = Sex, y = Pct, fill = Survived, label = Pct)) +
geom_bar(stat ="identity",width =.5,color="black") +
geom_text(size =3,aes(label =paste0(Pct,"%")),
position =position_stack(vjust =0.5))+
scale_fill_manual(values =c("gray50", "orangered2"))+
ggtitle("Survived by Sex") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Recordemos que la proporción de pasajeros que sobrevivieron al desastre es del 37.7%, y en este grafico vemos como esta proporción aumenta hasta el 83% en el caso de las mujeres, y disminuye al 13% en caso de los hombres. A la vista de esto podríamos asumir que la variable *Survived* dependerá de las categorías de *Sex*. Pero realicemos la prueba formal del test de independencia basado en la χ^2, donde testeamos las siguientes hipotesis:

H_0: Supervivencia y Sexo son independientes, el porcentaje de supervivientes no varía entre las categorías de Sexo.

H_1: Supervivencia y Sexo son dependientes, el porcentaje de supervivientes varía entre las categorias de Sexo.

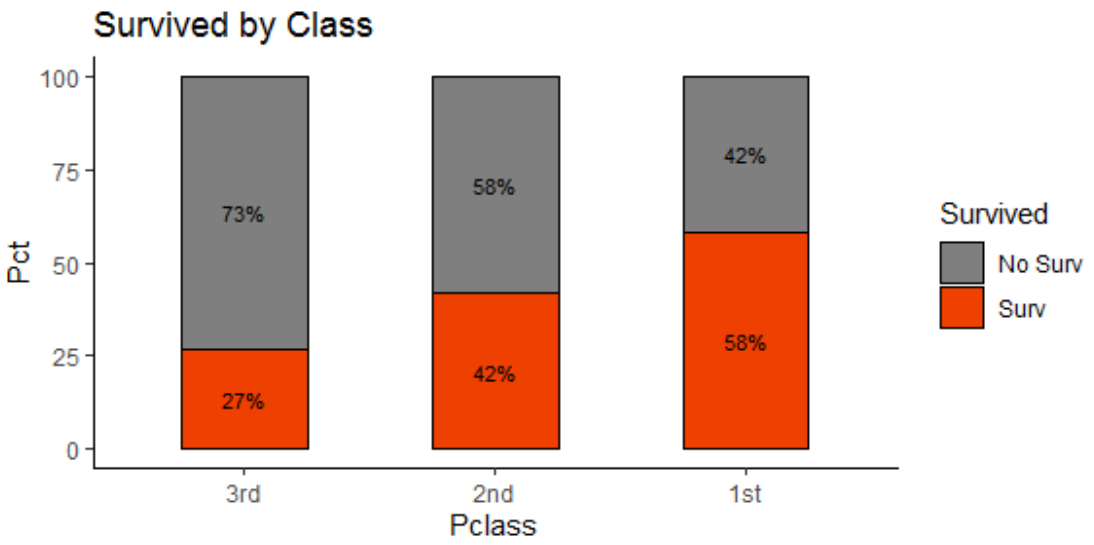
```
tabla<-table(df$Survived,df$Sex)
chisq.test(tabla,correct=F)
```

```
>
>   Pearson's Chi-squared test
>
> data:  tabla
> X-squared = 620.28, df = 1, p-value < 2.2e-16
```

Con estos resultados rechazamos la hipótesis nula de independencia y concluimos que existe una asociación significativa que implica que el porcentaje de supervivencia está asociado al *Sexo*.

Ahora haremos lo propio con la variable *Pclass*.

```
tabla<-as.data.frame(round(prop.table(table(df$Survived,df$Pclass),2)*100))
colnames(tabla)<-c("Survived","Pclass","Pct")

ggplot(tabla, aes(x = Pclass, y = Pct, fill = Survived, label = Pct)) +
geom_bar(stat ="identity",width =.5,color="black") +
geom_text(size =3,aes(label =paste0(Pct,"%")),
position =position_stack(vjust =0.5))+
scale_fill_manual(values =c("gray50", "orangered2"))+
ggtitle("Survived by Class") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Volviendo a tener en mente el 37.7% que sobrevivieron al desastre, entre clases podemos apreciar diferencias. Vemos como ese porcentaje aumenta hasta llegar al 58% entre los pasajeros de primera clase, y disminuye al 27% entre los pasajeros de tercera clase. La variación entre los de segunda clase no es tan llamativa pero sí parece significativa.

Si realizamos el mismo test formal anterior tenemos que:

```
tabla<-table(df$Survived,df$Pclass)
chisq.test(tabla,correct=F)

>
>   Pearson's Chi-squared test
>
> data:  tabla
> X-squared = 91.724, df = 2, p-value < 2.2e-16
```

Con un p.valor inferior a 0.05, concluimos que existe una asociación significativa que implica que el porcentaje de supervivencia está asociado a la clase del pasaje.

Este descubrimiento está muy en la línea de lo expuesto anteriormente, donde concluíamos que los que sobrevivieron al desastre habían pagado más (en media) por su pasaje, y cabe esperar que la clase este correlacionada al precio del pasaje.

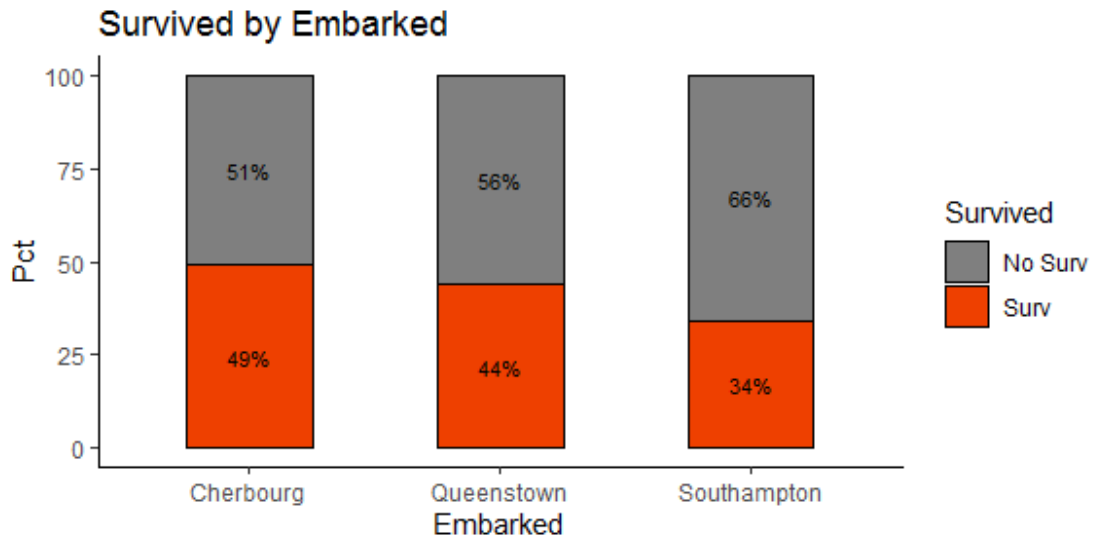
```
correlacion<-cor(as.numeric(df$Pclass),df$Fare_unit,method="spearman")
cat("Correlation Class with Fare_unit:",correlacion)

> Correlation Class with Fare_unit: 0.8644974
```

Vemos que el grado de correlación es elevado, cuanto más alto es la clase del pasaje, mas ha pagado por su billete.

Por último daremos un vistazo a la variable *Embarked*.

```
tabla<-as.data.frame(round(prop.table(
table(df$Survived,df$Embarked),2)*100))
```

```
colnames(tabla)<-c("Survived","Embarked","Pct")

ggplot(tabla, aes(x = Embarked, y = Pct, fill = Survived, label = Pct)) +
geom_bar(stat ="identity",width =.5,color="black") +
geom_text(size =3,aes(label =paste0(Pct,"%")),
position =position_stack(vjust =0.5))+
scale_fill_manual(values =c("gray50", "orangered2"))+
ggtitle("Survived by Embarked") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



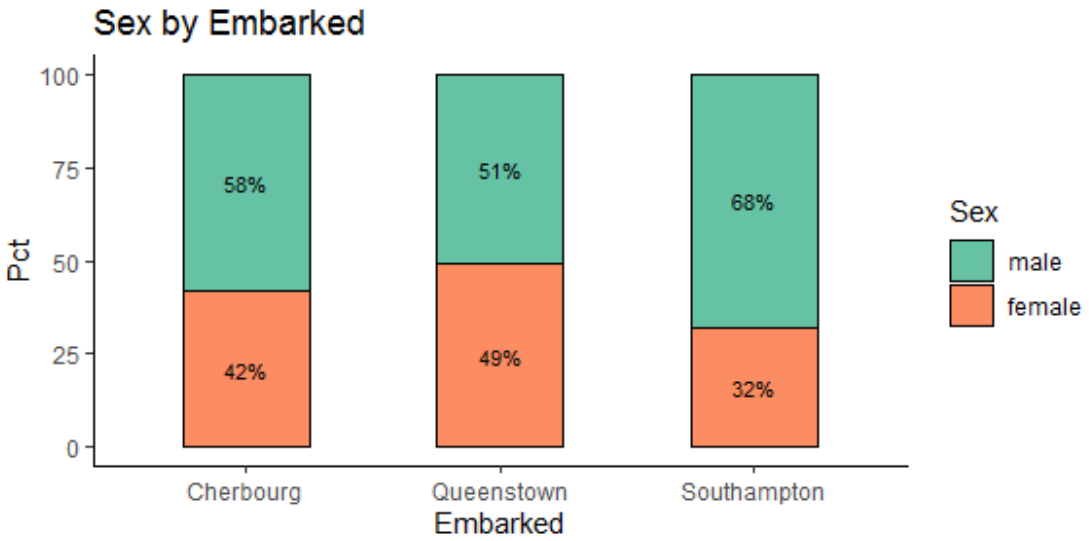
Se aprecian diferencias entre los pasajeros que embarcaron en *Cherbourg* y *Queenstown*, pero no tantas entre los que embarcaron en *Southampton*.

```
tabla<-table(df$Survived,df$Embarked)
chisq.test(tabla,correct=F)

>
>   Pearson's Chi-squared test
>
> data:  tabla
> X-squared = 24.194, df = 2, p-value = 5.577e-06
```

El test formal indica la existencia de asociación entre la variable supervivencia y la variable de embarque. Antes de terminar, queremos señalar que los que embarcaron en *Southampton* suponen una gran parte de la muestra, en concreto el 70%, lo que implica que tengan un perfil muy similar al del total de la muestra, por esa razón el porcentaje de supervivencia entre los que embarcaron en *Southampton* es tan similar.

Aun así, que el origen de embarque sea significativo nos hace sospechar que detrás de ella existan otras de las ya analizadas.

```
tabla<-as.data.frame(round(
prop.table(table(df$Sex,df$Embarked),2)*100))

colnames(tabla)<-c("Sex","Embarked","Pct")

ggplot(tabla, aes(x = Embarked, y = Pct, fill = Sex, label = Pct)) +
geom_bar(stat ="identity",width =.5,color="black") +
geom_text(size =3,aes(label =paste0(Pct,"%")),
position =position_stack(vjust =0.5))+
scale_fill_brewer(palette="Set2")+
ggtitle("Sex by Embarked") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```

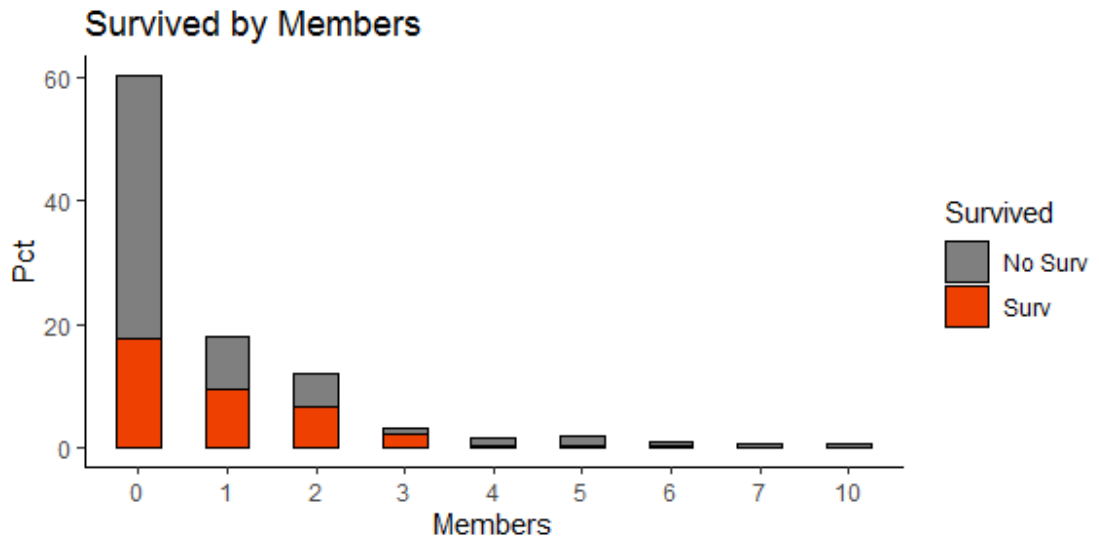


Del gráfico anterior podemos ver que en *Cherbourg* y *Queenstown* embarcaron más mujeres, recordemos que en la muestra que trabajamos el 35.6% son mujeres, y que es un segmento con una alta tasa de supervivencia. Por último señalar que *Southampton* tiene un porcentaje muy similar al total de la muestra.

Para terminar, nos queda por comprobar si existe algún tipo de asociación entre la supervivencia y el número de personas que acompañaban en el viaje. Para esto, tenemos disponible dos variables *SibSp* y *Parch*, que las sumaremos y obtendremos el número de personas que acompañaban a cada pasajero en su viaje.

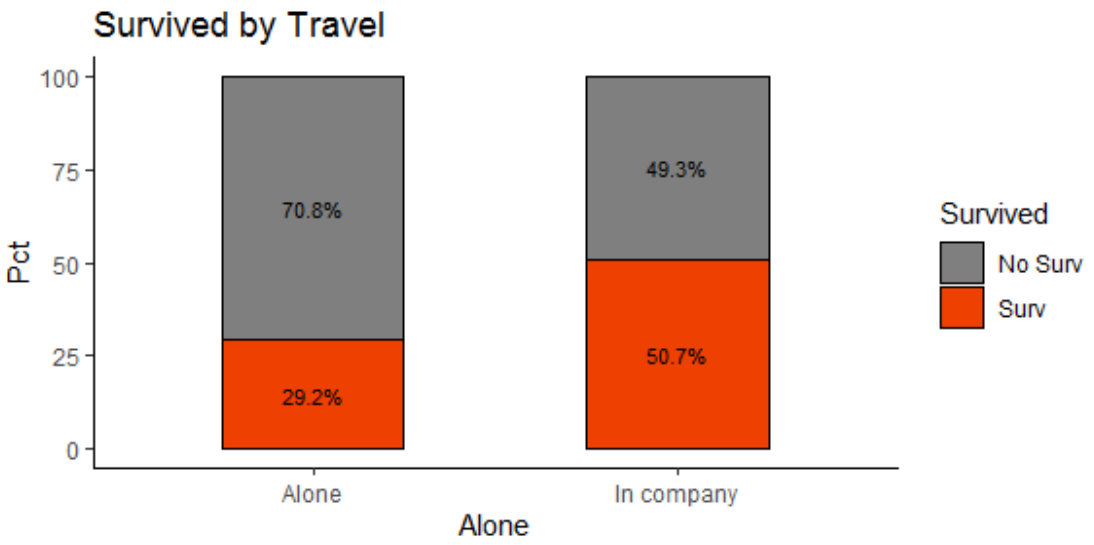
```
df$Members<-rowSums(df[,c("SibSp","Parch")])
drops[,c("SibSp","Parch")]<-df[,c("SibSp","Parch")]
df[,c("SibSp","Parch")]<-NULL

tabla<-as.data.frame(round(
prop.table(table(df$Survived,df$Members))*100,2))

colnames(tabla)<-c("Survived","Members","Pct")

ggplot(tabla, aes(x = Members, y=Pct, fill = Survived, label=Pct)) +
```

```
geom_bar(stat="identity",width =.5,color="black") +
scale_fill_manual(values =c("gray50", "orangered2"))+
ggtitle("Survived by Members") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



En este gráfico apreciamos un par de cosas. Primero que el 60% del pasaje viajaba solo y aproximadamente 1/3 de estos sobrevivieron al desastre. En contra, vemos como los que viajaban acompañados tuvieron más posibilidades de salir con vida (50% o más). Esto último no se cumple para aquellos que viajaban con más de tres miembros, pero estos solo suponen 6% del pasaje. Así que consideraremos la variable dicotómica viajaba solo.

```
df$Alone<-ifelse(df$Members==0,"Alone","In company")
df$Members<-NULL
df$Alone<-factor(df$Alone,
levels=c("Alone","In company"),
labels=c("Alone","In company"))


tabla<-as.data.frame(round(
prop.table(table(df$Survived,df$Alone),2)*100,1))

colnames(tabla)<-c("Survived","Alone","Pct")

ggplot(tabla, aes(x = Alone, y=Pct, fill = Survived, label=Pct)) +
geom_bar(stat="identity",width =.5,color="black") +
geom_text(size =3,aes(label =paste0(Pct,"%")),
position =position_stack(vjust =0.5))+
scale_fill_manual(values =c("gray50", "orangered2"))+
ggtitle("Survived by Travel") +
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```




```
tabla<-table(df$Survived,df$Alone)
chisq.test(tabla,correct=F)

>
>   Pearson's Chi-squared test
>
> data:  tabla
> X-squared = 61.242, df = 1, p-value = 5.047e-15
```

El test formal apoya lo que evidencia el grafico de barras, la asociación entre la variable supervivencia y la variable de si viajaba solo o en compañía.

Antes de pasar a la siguiente sección guardaremos los datos procesados que usaremos para modelar un clasificador en un fichero *.csv*.

```
write.table(df,"titanic_procesado.csv",col.names = T,
row.names = F,sep=",",dec=".")
```

4 Modelos supervisados.

Llegados a este punto y habiendo visto la relación de cada atributo con la variable respuesta estamos en disposición de buscar un modelo clasificador que obtenga un buen desempeño. Para evaluar este desempeño comprobaremos como de próximas son sus predicciones a los verdaderos valores de la variable respuesta. Para poder cuantificar de forma correcta este error, se necesita disponer de un conjunto de observaciones, de las que se conozca la variable respuesta, pero que el modelo no haya ?visto?, es decir, que no hayan participado en su ajuste. Con esta finalidad, utilizaremos la partición *sample* que divide los datos disponibles en un conjunto de entrenamiento y un conjunto de test, aproximadamente, en un 70%-30% respectivamente.

```
train<-df[sample==0,]
test<-df[sample==1,]
```

Es importante que la distribución de la variable respuesta sea similar en ambos conjuntos creados.

```
prop.table(table(train$Survived))

>
>   No Surv      Surv
> 0.6161616 0.3838384

prop.table(table(test$Survived))

>
>   No Surv      Surv
> 0.6363636 0.3636364
```

El entrenamiento del modelo se realizara por medio de una validación cruzada con 4 particiones evaluando diferentes métricas que cuantifiquen su desempeño. De esta manera, si estas métricas son muy superiores de las que obtendremos en la evaluación por medio del set *test* habremos incurrido en sobreajuste, es decir, que el algoritmo se ha especializado en los datos de entrenamiento y que no consigue generalizar bien casos nuevos, observaciones nunca vistas.

```
folds <-caret::createFolds(train$Survived, k =4)
```

4.1 Regresión logística.

Nuestro primer modelo se basara en la regresión logística.

```
cv_reglog<-sapply(folds,function(k){

  x<-train[-k,]
  y<-train[k,]

  clasificador<-glm(Survived~Pclass+Sex+Embarked+
Alone+Age+Fare_unit,data=x,family="binomial")

  probabilidad<-predict(clasificador,newdata = y,type ='response')
```

```
  prediccion<-ifelse(probabilidad<0.5,0,1)
  prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))

  matriz_conf<-table(y[,"Survived"],prediccion)
  accuracy<-sum(diag(prop.table(matriz_conf)))
  tpr<-prop.table(matriz_conf,1)[2,2]
  fpr<-prop.table(matriz_conf,1)[1,2]
  metric<-c(accuracy,tpr,fpr)

  roc<-ROCR::prediction(probabilidad,y[,"Survived",drop=F])
  auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

  metric<-c(metric,auc)
names(metric)<-c("Accuracy","TPR","FPR","AUC")
return(metric)

  })

metricas<-apply(cv_reglog,1,mean)

cat("Promedios validación cruzada regresión logística:\n",
"\nAccuracy:",metricas[1],
"\nTrue Positive Rate:",metricas[2],
"\nFalse Positive Rate:",metricas[3],
"\nAUC:",metricas[4])
```

```
> Promedios validación cruzada regresión logística:
>
> Accuracy: 0.7889982
> True Positive Rate: 0.6899453
> False Positive Rate: 0.1493309
> AUC: 0.8512044
```

Más adelante explicaremos con más detalle estas métricas. Ahora pasamos a evaluar la capacidad predictiva del modelo a partir de estas mismas métricas pero en el set de observaciones test y esperar que estas métricas no sean muy inferiores a las obtenidas en el entrenamiento, pues significaría que padecemos de sobreajuste.

```
modelo_reglog<-glm(Survived~Pclass+Sex+Embarked+Alone+Age+Fare_unit,
data=train,family="binomial")

probabilidad<-predict(modelo_reglog,type ='response',newdata = test)

prediccion<-ifelse(probabilidad<0.5,0,1)
prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))

matriz_conf<-table(test[,"Survived"],prediccion)
accuracy<-sum(diag(prop.table(matriz_conf)))
tpr<-prop.table(matriz_conf,1)[2,2]
fpr<-prop.table(matriz_conf,1)[1,2]

roc<-ROCR::prediction(probabilidad,test[,"Survived",drop=F])
auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]
auc<-round(auc,3)
curva_roc <-ROCR::performance(roc, "tpr", "fpr")
```

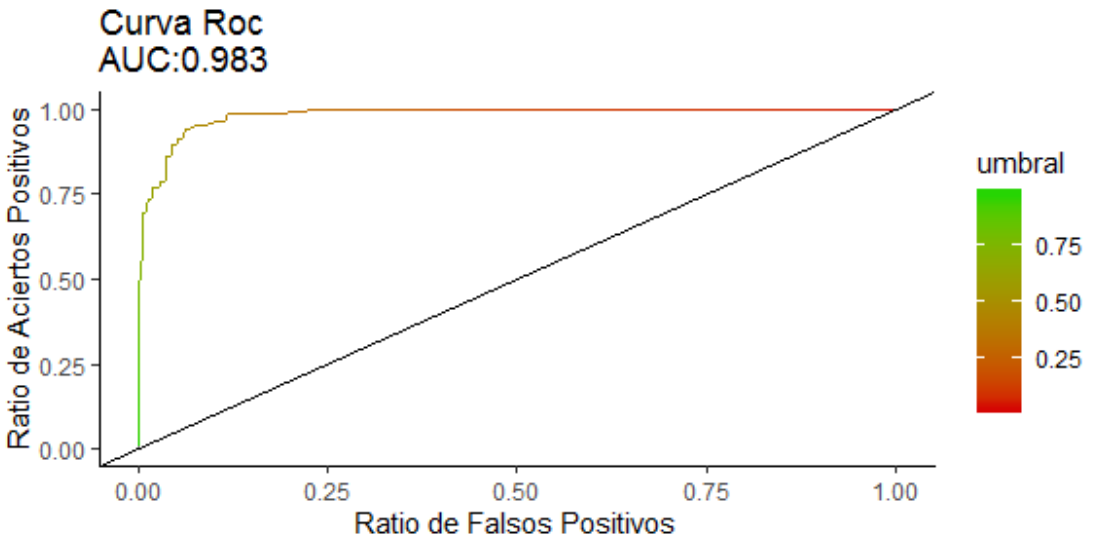
```
curva_roc<-data.frame("TPR"=unlist(curva_roc@y.values),
"FPR"=unlist(curva_roc@x.values),
"umbral"=unlist(curva_roc@alpha.values))

cat("Desempeño modelo regresión logística:\n",
"\nAccuracy:",accuracy,
"\nTrue Positive Rate:",tpr,
"\nFalse Positive Rate:",fpr)
```

```
> Desempeño modelo regresión logística:
>
> Accuracy: 0.9330144
> True Positive Rate: 0.9210526
> False Positive Rate: 0.06015038
```

El modelo logístico consigue un buen desempeño, pues su ajuste es del 93.3%, lo que supone un error del 6.7%. Además, el acierto dentro de cada categoría también es muy bueno, acierta con una precisión del 92.1% de los que realmente sobrevivieron (*True Positive Rate*) y no solo eso, además el error dentro de los que realmente no sobrevivieron es del 6% (*False Positive Rate*).

Estas métricas son las que se obtienen cuando aplicamos un umbral del 50% en las probabilidades calculadas. Para valorar el rendimiento del modelo también se utiliza el grafico de la curva ROC, que dibuja la tasa de true positives ante la tasa false positive. Cada punto de la curva corresponde a un nivel de umbral de la matriz de confusión. De este modo un buen modelo seria aquel que tuviera una curva lo próxima a la esquina superior izquierda (lo ideal es obtener a partir de cierto umbral el punto donde la tasa falsos positivos es 0 y la de verdaderos positivos es 1), mientras que una plana próxima a la diagonal correspondería a modelos pocos discriminantes. Por lo tanto calculando el área que hay debajo de la curva podríamos medir el desempeño de nuestro modelo.

```
ggplot(curva_roc,aes(y=TPR,x=FPR,color=umbral))+
geom_line()+
geom_abline()+
labs(title=paste0("Curva Roc\n",paste0("AUC:",auc)),
y="Ratio de Aciertos Positivos",x="Ratio de Falsos Positivos")+
scale_color_gradient(low="#d50603",high="#26d503")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Con un *AUC* del 0.98, consideramos que el modelo es muy bueno. Ahora vamos a interpretarlo.

```
summary(modelo_reglog)
```

```
>
> Call:
> glm(formula = Survived ~ Pclass + Sex + Embarked + Alone + Age +
>     Fare_unit, family = "binomial", data = train)
>
> Deviance Residuals:
>     Min       1Q   Median       3Q      Max
> -2.5833  -0.6468  -0.3898   0.6416   2.5058
>
> Coefficients:
>                       Estimate Std. Error z value Pr(>|z|)
> (Intercept)         -1.032661   0.332214  -3.108  0.00188 **
> Pclass2nd            1.370238   0.240887   5.688 1.28e-08 ***
> Pclass1st            2.240904   0.400682   5.593 2.24e-08 ***
> Sexfemale            2.542160   0.196725  12.922  < 2e-16 ***
> EmbarkedQueenstown   0.032276   0.384236   0.084  0.93306
> EmbarkedSouthampton -0.466238   0.238185  -1.957  0.05029 .
> AloneIn company     -0.136055   0.193409  -0.703  0.48177
> Age                 -0.037080   0.007490  -4.951 7.40e-07 ***
> Fare_unit            0.008976   0.011864   0.757  0.44933
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for binomial family taken to be 1)
>
>     Null deviance: 1186.66  on 890  degrees of freedom
> Residual deviance:  792.08  on 882  degrees of freedom
> AIC: 810.08
>
> Number of Fisher Scoring iterations: 5
```

Por un lado nos centraremos en la significación de los parámetros. Podemos ver que el modelo nos indica que los atributos *Embarked*, *Alone* y *Fare_unit* no son significativos.

Por otro lado, por el signo positivo de los coeficientes podemos decir que los pasajeros de primera y de segunda tienen menos riesgo de perecer en el accidente respecto a los de tercera clase (la categoría omitida). Del mismo modo los pasajeros de sexo femenino tienen menos riesgo de perecer en el accidente respecto a los hombres. Por último, el signo negativo de *Age* indica que cuanto más joven, más posibilidades de sobrevivir al accidente.

Hasta ahora sabemos que pasar de una clase a otra o ser de un sexo u otro, aumenta las probabilidades de sobrevivir al accidente, ?pero cuanto? En la regresión lineal, el propio parámetro estimado era lo que modificaba (en promedio) la variable respuesta a cada cambio unitario en la variable asociada a dicho parámetro. En la regresión logística binaria, lo que varía es el logaritmo del odds ratio de que un pasajero sobreviva. Esto significa que al pasar tercera clase a primera, se espera que el logaritmo del odds ratio de que un pasajero sobreviva a la tragedia sea (en promedio) 2.240904, por lo tanto, lo que varia el odds ratio de sobrevivir se incrementa en promedio $e^{2.240904} = 9.401827$.

```
odds<-exp(modelo_reglog$coefficients)
for (i inlist(c(1:4),c(5:7),c(8,9))){
print(odds[i])
cat("\n")
}
```

```
> (Intercept)   Pclass2nd   Pclass1st   Sexfemale
>   0.3560581   3.9362892   9.4018279  12.7070943
>
>  EmbarkedQueenstown EmbarkedSouthampton     AloneIn company
>           1.0328022           0.6273578           0.8727949
>
>       Age Fare_unit
> 0.9635995 1.0090160
```

Cuanto más alejados de la unidad se encuentren estos odds ratios mejor discrimina los estados/categorías de la variable predictora para determinar la categoría mas probable de la variable dependiente. Dicho esto podemos admitir que para este modelo, las variables más importantes para predecir la supervivencia son, sobre todo, *Sex* y *Pclass*.

Ahora podemos repetir el ejercicio eliminando las variables no significativas.

```
cv_reglog<-sapply(folds,function(k){

  x<-train[-k,]
  y<-train[k,]

  clasificador<-glm(Survived~Pclass+Sex+Age,data=x,family="binomial")

  probabilidad<-predict(clasificador,type ='response',newdata = y)

  prediccion<-ifelse(probabilidad<0.5,0,1)
  prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))
```

```
  matriz_conf<-table(y[,"Survived"],prediccion)
  accuracy<-sum(diag(prop.table(matriz_conf)))
  tpr<-prop.table(matriz_conf,1)[2,2]
  fpr<-prop.table(matriz_conf,1)[1,2]
  metric<-c(accuracy,tpr,fpr)

  roc<-ROCR::prediction(probabilidad,y[,"Survived",drop=F])
  auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

  metric<-c(metric,auc)
names(metric)<-c("Accuracy","TPR","FPR","AUC")
return(metric)

  })

metricas<-apply(cv_reglog,1,mean)

cat("Promedios validación cruzada regresión logística:\n",
"\nAccuracy:",metricas[1],
"\nTrue Positive Rate:",metricas[2],
"\nFalse Positive Rate:",metricas[3],
"\nAUC:",metricas[4])
```

```
> Promedios validación cruzada regresión logística:
>
> Accuracy: 0.7979569
> True Positive Rate: 0.7132695
> False Positive Rate: 0.1492912
> AUC: 0.8501512
```

Los valores son muy similares al obtenido en la validación cruzada considerando todos los atributos.

Ahora evaluaremos el desempeño por medio de las observaciones del set test.

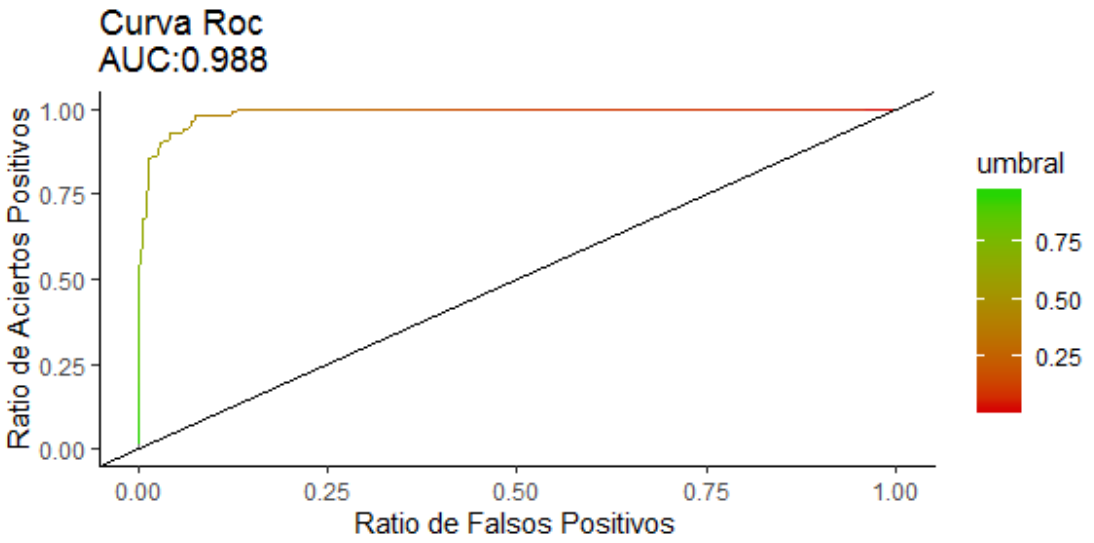
```
modelo_reglog<-glm(Survived~Pclass+Sex+Age,data=train,family="binomial")

probabilidad<-predict(modelo_reglog,type ='response',newdata = test)

prediccion<-ifelse(probabilidad<0.5,0,1)
prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))

matriz_conf<-table(test[,"Survived"],prediccion)
accuracy<-sum(diag(prop.table(matriz_conf)))
tpr<-prop.table(matriz_conf,1)[2,2]
fpr<-prop.table(matriz_conf,1)[1,2]

roc<-ROCR::prediction(probabilidad,
       test[,"Survived",drop=F])
auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]
auc<-round(auc,3)
curva_roc <-ROCR::performance(roc, "tpr", "fpr")
curva_roc<-data.frame("TPR"=unlist(curva_roc@y.values),
"FPR"=unlist(curva_roc@x.values),
"umbral"=unlist(curva_roc@alpha.values))
```

```
cat("Desempeño modelo regresión logística:\n",
"\nAccuracy:",accuracy,
"\nTrue Positive Rate:",tpr,
"\nFalse Positive Rate:",fpr)
```

```
> Desempeño modelo regresión logística:
>
> Accuracy: 0.9354067
> True Positive Rate: 0.9276316
> False Positive Rate: 0.06015038
```

Estos valores igualan (tal vez mejoran algo) los obtenidos con el modelo con todos los atributos.

```
ggplot(curva_roc,aes(y=TPR,x=FPR,color=umbral))+
geom_line()+
geom_abline()+
labs(title=paste0("Curva Roc\n",paste0("AUC:",auc)),
y="Ratio de Aciertos Positivos",x="Ratio de Falsos Positivos")+
scale_color_gradient(low="#d50603",high="#26d503")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Con el *AUC* pasa exactamente los mismo, un valor similar, aunque algo mejor al obtenido con el modelo con todos los atributos.

```
summary(modelo_reglog)
```

```
>
> Call:
> glm(formula = Survived ~ Pclass + Sex + Age, family = "binomial",
>     data = train)
>
> Deviance Residuals:
```

```
>     Min       1Q   Median       3Q      Max
> -2.6985  -0.6571  -0.3928   0.6357   2.4692
>
> Coefficients:
>             Estimate Std. Error z value Pr(>|z|)
> (Intercept) -1.34979    0.22905  -5.893 3.79e-09 ***
> Pclass2nd    1.30150    0.22837   5.699 1.20e-08 ***
> Pclass1st    2.49368    0.25424   9.808  < 2e-16 ***
> Sexfemale    2.54384    0.18691  13.610  < 2e-16 ***
> Age         -0.03667    0.00726  -5.051 4.39e-07 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> (Dispersion parameter for binomial family taken to be 1)
>
>     Null deviance: 1186.66  on 890  degrees of freedom
> Residual deviance:  799.41  on 886  degrees of freedom
> AIC: 809.41
>
> Number of Fisher Scoring iterations: 5
```

Por medio del resumen vemos que todos los regresores son significativos, conservando el signo del modelo anterior con todos los atributos, y los odds ratios tienen prácticamente los mismos valores.

```
exp(modelo_reglog$coefficients)
```

```
> (Intercept)   Pclass2nd   Pclass1st   Sexfemale         Age
>   0.2592945   3.6748115  12.1056875  12.7284668   0.9639949
```

En definitiva, con este modelo reducido obtenemos un desempeño similar, aunque algo mejor, que el modelo con todas ampliado.

4.2 K-Vecinos próximos (Knn).

Nuestro segundo modelo se en el algoritmo de los K-Vecinos próximos. En esta ocasión realizaremos un procesado previo antes de pasar los datos por el algoritmo. Las variables categóricas las convertiremos en *dummy*, es decir, cada categoría se convertirá en una variable binaria 0-1, donde el 0 representa la ausencia y el 1 la presencia de la categoría. Para terminar, resulta conveniente centrar los atributos, pues están registrados en diferentes escalas, y por lo tanto atributos como *Age* y *Fare_unit* que son de una magnitud elevada, pueden determinar en gran medida el valor de distancia/similitud obtenida al comparar las observaciones, infravalorando la aportación del resto de atributos. Escalar y centrar las variables de forma que todas ellas tengan media 0 y desviación estándar 1 antes de calcular la matriz de distancias asegura que todas las variables tengan el mismo peso.

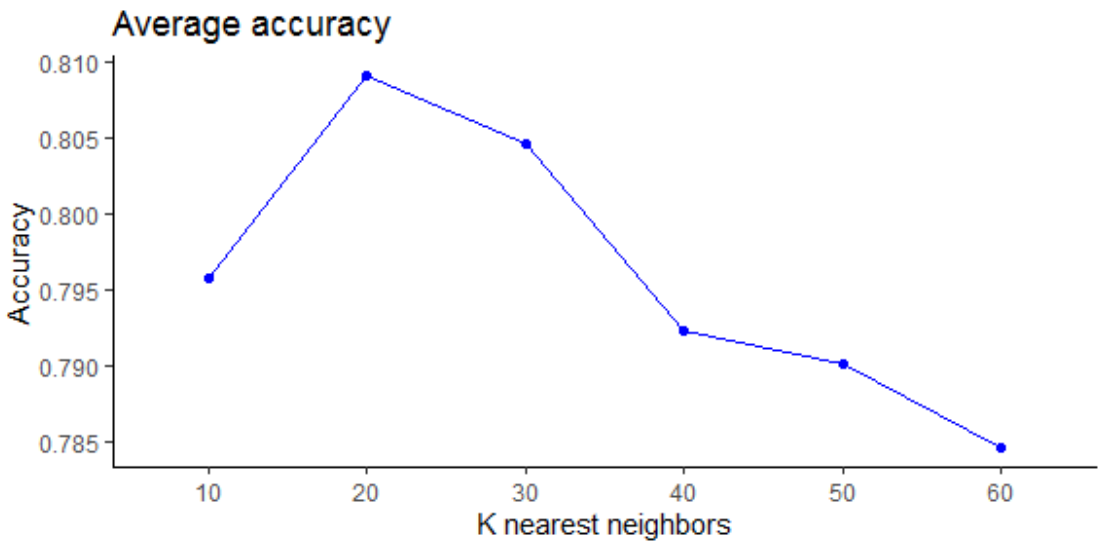
```
#Preparamos el set train
#Creacion de las dummys
train_dummys<-NULL
for (i inc("Pclass","Sex","Embarked","Alone")){
  niveles<-levels(df[sample==0,i])
```

```
  dummy<-t(apply(df[sample==0,i,drop=F],1,function(k)niveles%in%k))*1
colnames(dummy)<-niveles
  train_dummys<-cbind(train_dummys,dummy)
}

#Fusiono las dummys creadas con las variable continuas.
train<-cbind(train_dummys,df[sample==0,c("Age","Fare_unit")])

#Escalo y centro todas las variables.
train<-sapply(train,function(k)(k-mean(k))/sd(k))
train<-data.frame(train)

#Preparamos el set test
#Creacion de las dummys
test_dummys<-NULL
for (i inc("Pclass","Sex","Embarked","Alone")){
  niveles<-levels(df[sample==1,i])
  dummy<-t(apply(df[sample==1,i,drop=F],1,function(k)niveles%in%k))*1
colnames(dummy)<-niveles
  test_dummys<-cbind(test_dummys,dummy)
}

#Fusiono las dummys creadas con las variable continuas.
test<-cbind(test_dummys,df[sample==1,c("Age","Fare_unit")])

#Escalo y centro todas las variables.
test<-sapply(test,function(k)(k-mean(k))/sd(k))
test<-data.frame(test)

#La variable respuesta de las observaciones del set train
target_train<-df[sample==0,"Survived"]
#La variable respuesta de las observaciones del set test
target_test<-df[sample==1,"Survived"]
```

Con los datos ya procesados para que el algoritmo los “digiera” correctamente, en esta ocasión y debido a que el algoritmo dispone del hiperparámetro *K* (número de vecinos) y para decidir cual será su valor optimo, entrenaremos el modelo con diferentes valores de *K*, y lo evaluaremos utilizando validación cruzada. Posteriormente, escogeremos el valor del hiperparámetro que mejor resultados consiga.

```
library(class)

vecinos<-c(10,20,30,40,50,60)
names(vecinos)<-c(10,20,30,40,50,60)

cv_knn<-lapply(vecinos,function(v){

rejilla<-sapply(folds,function(k){

  x<-train[-k,]
  cl<-target_train[-k]
  y<-train[k,]

  prediccion<-knn(x,y,cl,k=v)
  matriz_conf<-table(target_train[k],prediccion)
```

```
  accuracy<-sum(diag(prop.table(matriz_conf)))
  tpr<-prop.table(matriz_conf,1)[2,2]
  fpr<-prop.table(matriz_conf,1)[1,2]
  metric<-c(accuracy,tpr,fpr)
names(metric)<-c("Accuracy","TPR","FPR")
return(metric)
  })
return(rejilla)
})

metricas<-round(sapply(cv_knn,function(k)apply(k,1,mean)),4)
metricas<-as.data.frame(t(metricas))
metricas<-cbind("k"=as.factor(row.names(metricas)),metricas)

ggplot(metricas,aes(y=Accuracy,x=k,group=1))+
geom_line(color="blue")+
geom_point(color="blue")+
ggtitle("Average accuracy")+
labs(x="K nearest neighbors")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



En esta ocasión el desempeño del modelo lo evaluaremos según el ajuste, ya que este algoritmo no proporciona la probabilidad de pertenencia a las categorías de la variable respuesta para ninguna de las observaciones. Dicho esto, podemos ver que el con *K=20* obtenemos el mejor ajuste.

Para una mejor decisión mostraremos los promedios de los *True Positive Rate* y de los *False Positive Rate*.

```
knitr::kable(metricas,row.names=F)
```

k	Accuracy	TPR	FPR
10	0.7957	0.6373	0.1056
20	0.8092	0.6226	0.0747
30	0.8047	0.6372	0.0911
40	0.7923	0.5874	0.0801
50	0.7901	0.5816	0.0801
60	0.7845	0.5465	0.0673

Con *k=20* escogemos el hiperparámetro con mejor ajuste y con mejor tasa de falsos positivos.

Para finalizar con este algoritmo, veremos que desempeño obtenemos con los datos del test fijando *k=20*.

```
prediccion<-knn(train,test,target_train,k=20)
matriz_conf<-table(target_test,prediccion)
accuracy<-sum(diag(prop.table(matriz_conf)))
tpr<-prop.table(matriz_conf,1)[2,2]
fpr<-prop.table(matriz_conf,1)[1,2]
metric<-c(accuracy,tpr,fpr)
names(metric)<-c("Accuracy","TPR","FPR")
metric

>   Accuracy        TPR        FPR
> 0.89952153 0.76973684 0.02631579
```

Obtenemos un ajuste muy bueno, pero inferior al obtenido por el modelo basado en la regresión logística.

4.3 Árboles de decisión.

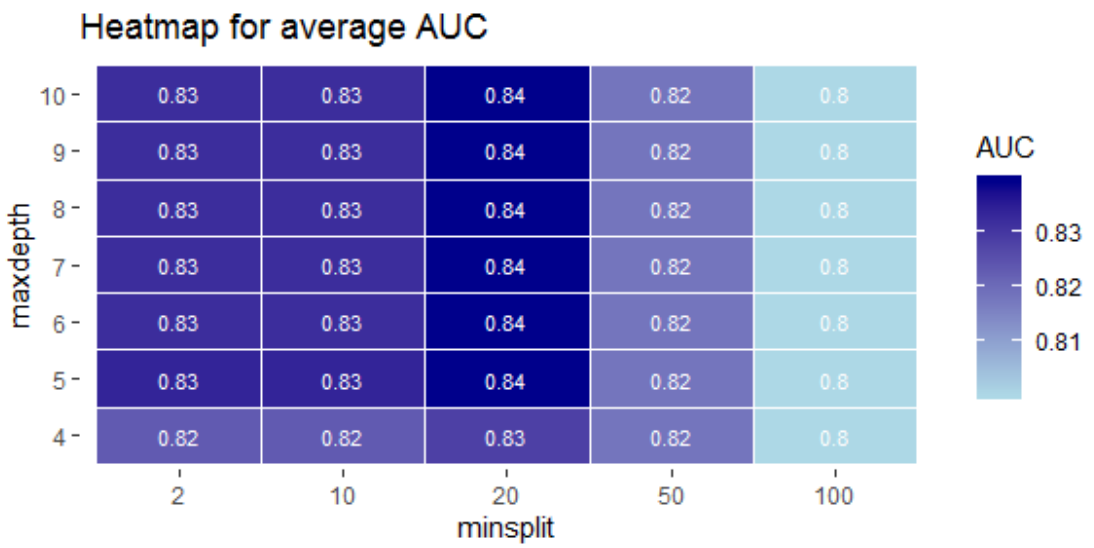
Para el tercer modelo optamos por uno basado en un árbol de decisión, ya que además de ser una técnica predictiva, su formulación no supone una “caja negra” y sus resultados en colecciones en reglas de clasificación aportan una interpretación natural, hacen que estos métodos sean versátiles y aplicables. Además, su representación gráfica aporta una mayor legibilidad. Un árbol de decisión es un clasificador en el que se particionan los datos de forma recursiva para formar grupos o clase.

En esta ocasión haremos una búsqueda de rejilla (*grid search*) para ajustar los hiperparámetros *maxdepth* (la profundidad máxima del árbol) y *minsplit* (el numero mínimo de observaciones que debe tener un nodo para poder dividirse).

```
train<-df[sample==0,]
test<-df[sample==1,]
```

```r
library(rpart)

hiperparametros<-expand.grid("minsplit"=c(2,10,20,50,100),
"maxdepth"=c(4:10))

cv_tree<-apply(hiperparametros,1,function(hp){

rejilla<-lapply(folds,function(k){

  x<-train[-k,]
  y<-train[k,]

  clasificador<-rpart(Survived~Pclass+Sex+Embarked+
Alone+Age+Fare_unit,x,method ='class',
control =rpart.control(
maxdepth =hp["maxdepth"],
minsplit = hp["minsplit"]))

  probabilidad<-predict(object=clasificador,newdata = y,
type ='prob')[,2]

  prediccion<-ifelse(probabilidad<0.5,0,1)
  prediccion<-factor(prediccion,levels=c(0,1),
labels=c("No surv","Surv"))

  matriz_conf<-table(y[,"Survived"],prediccion)
  accuracy<-sum(diag(prop.table(matriz_conf)))
  tpr<-prop.table(matriz_conf,1)[2,2]
  fpr<-prop.table(matriz_conf,1)[1,2]
  metric<-c(accuracy,tpr,fpr)

  roc<-ROCR::prediction(probabilidad,y[,"Survived",drop=F])
  auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

  metric<-c(metric,auc)
names(metric)<-c("Accuracy","TPR","FPR","AUC")
return(metric)

  })
return(rejilla)
})

metricas<-lapply(cv_tree,function(k)sapply(k,function(j)j))
metricas<-t(sapply(metricas,function(k)rowMeans(k)))
metricas<-cbind(metricas,hiperparametros)

g1<-ggplot(metricas, aes(x =as.factor(minsplit),
y =as.factor(maxdepth),
fill = AUC)) +
geom_tile(colour ="white")+
labs(x="minsplit",y="maxdepth")+
ggtitle("Heatmap for average AUC")+
scale_fill_gradient(low ="lightblue",high ="darkblue")+
geom_text(aes(label =round(AUC,2)),size=3, color ="white")+
theme(panel.background =element_blank())
```

```
g1
```



Por lo que vemos en el *heatmap* hay más variación en *AUC* cuando modificamos el número mínimo de observaciones necesarias para dividir un nodo. A partir de una profundidad de 5, el ajuste y el *AUC* permanecen constante fijado un *minsplit*.

Vemos que el mejor resultado lo obtenemos para *minsplit*=20 y *maxdepth*=5, pero como 20 observaciones puede dar a lugar a nodos terminales muy pequeños lo que puede derivar a que el modelo se especialice en el set train y acabemos padeciendo de sobreajuste. Por esta razón fijaremos un mínimo de 50 casos para dividir un nodo, que por lo que se ve proporciona un buen *AUC*. Dicho esto, pasamos obtener el modelo y evaluarlo con el set test.

```
modelo_tree<-rpart(Survived~Pclass+Sex+Embarked+Alone+
Age+Fare_unit,train,
method ='class',
control =rpart.control(maxdepth =5,
minsplit =50))

probabilidad<-predict(object=modelo_tree,newdata = test,type ='prob')[,2]
prediccion<-ifelse(probabilidad<0.5,0,1)
prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))
matriz_conf<-table(test[,"Survived"],prediccion)
accuracy<-sum(diag(prop.table(matriz_conf)))
tpr<-prop.table(matriz_conf,1)[2,2]
fpr<-prop.table(matriz_conf,1)[1,2]

roc<-ROCR::prediction(probabilidad,test[,"Survived",drop=F])
auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

auc<-round(auc,3)
curva_roc <-ROCR::performance(roc, "tpr", "fpr")
```

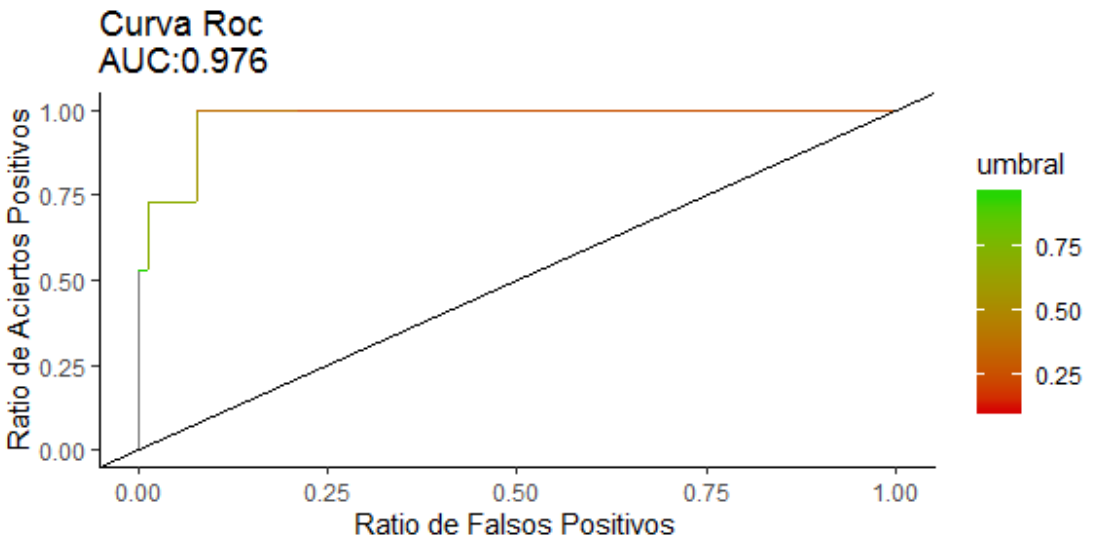
```
curva_roc<-data.frame("TPR"=unlist(curva_roc@y.values),
"FPR"=unlist(curva_roc@x.values),
"umbral"=unlist(curva_roc@alpha.values))

cat("Desempeño modelo árbol de decisión:\n",
"\nAccuracy:",accuracy,
"\nTrue Positive Rate:",tpr,
"\nFalse Positive Rate:",fpr)
```

```
> Desempeño modelo árbol de decisión:
>
> Accuracy: 0.8516746
> True Positive Rate: 0.7302632
> False Positive Rate: 0.07894737
```

Las métricas obtenidas son buenas. Tenemos un ajuste del 85.16%, lo que supone un error del 14.83%. Dentro de cada categoría observada apreciamos que el acierto de los que finalmente sobrevivieron es de un 73.02% y el error de los que realmente no sobrevivieron es de un 7.89%.

El *AUC* que obtiene este modelo es muy bueno, con un valor de 0.976.

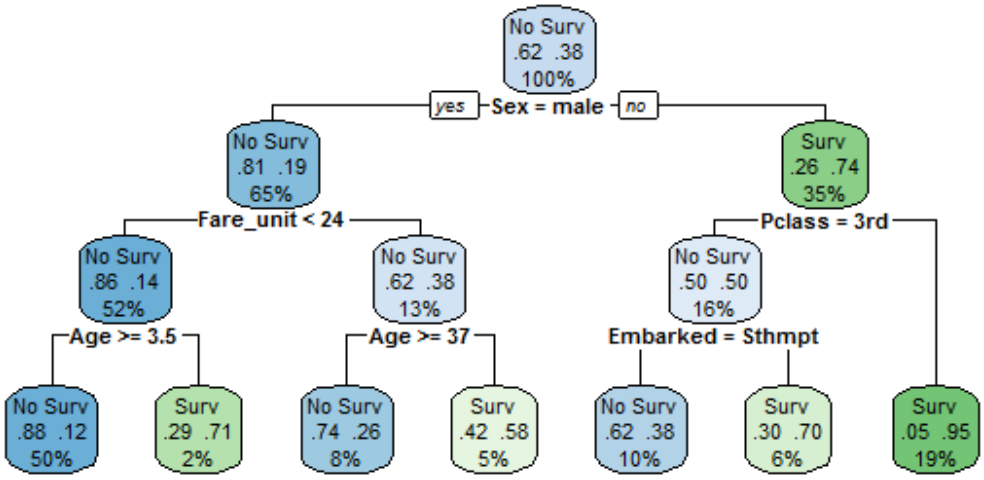
```
ggplot(curva_roc,aes(y=TPR,x=FPR,color=umbral))+
geom_line()+
geom_abline()+
labs(title=paste0("Curva Roc\n",paste0("AUC:",auc)),
y="Ratio de Aciertos Positivos",x="Ratio de Falsos Positivos")+
scale_color_gradient(low="#d50603",high="#26d503")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Todas estas métricas, a pesar de ser buenas, son algo inferiores a las que obtenemos por medio del modelo de regresión logística.

Para acabar, mostraremos el árbol generado con los datos de entrenamiento, con los que se obtienen las reglas para clasificar nuevas observaciones.

```
library(rpart.plot)
rpart.plot(modelo_tree, extra=104, cex=0.7, faclen =6, fallen.leaves=T)
```



Vemos un árbol de 3 niveles con 7 nodos terminales. Estos nodos terminales son los que finalmente nos proporcionan la probabilidad de cada individuo. Tenemos nodos muy homogéneos como el situado a la derecha que corresponde a la siguiente regla *No masculino (Femenino)->No 3º Clase (1ª o 2ª Clase)*, que de ser cierta para una observación, implicaría una probabilidad de sobrevivir del 95%. Por último, señalar que la primera variable para segmentar es el sexo, lo que indica que es la variable que en primera instancia crea segmentos mas homogéneos, algo que está muy en la línea de los odds ratios obtenidos del modelo de regresión logística, donde el sexo era el de mayor valor y por lo tanto el mas discriminaba entre las categorías de la variable respuesta.

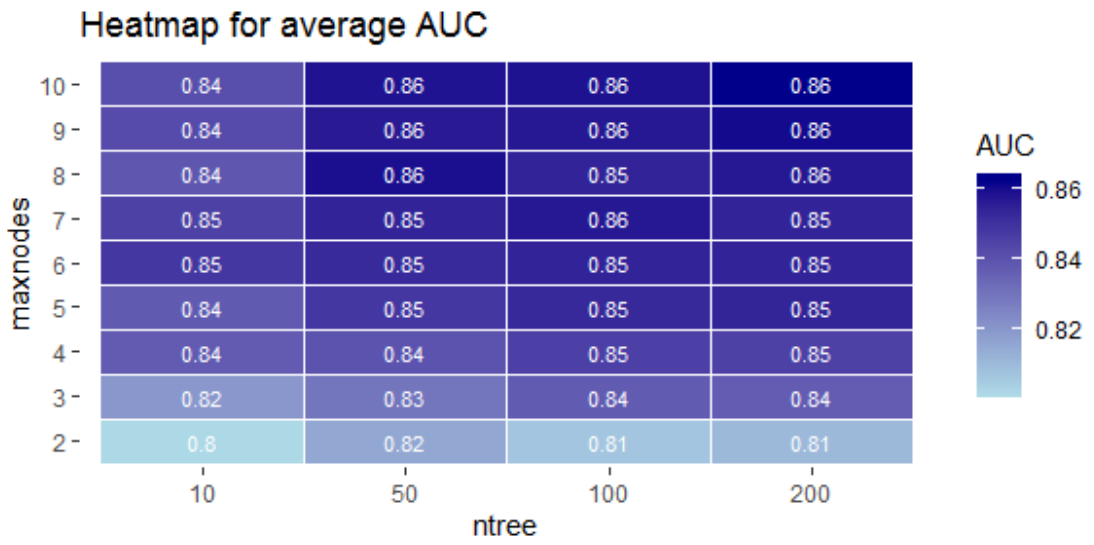
4.4 Random Forest.

Uno de los problemas en la creación de un árbol de decisión es que si le damos mucha profundidad, el árbol tiende a especializarse en los datos usados en vez de generalizar el aprendizaje. Es decir, a padecer de sobreajuste. La solución para evitar esto son los bosques aleatorios, es decir, crear muchos árboles y que trabajen en conjunto. Este algoritmo es un tipo de ensamble en machine learning donde se combinan diversos arboles y la salida de cada uno se contara cómo ?un voto? y la opción mas votada será la respuesta del bosque aleatorio. De este modo se entiende al bosque aleatorio como un metaestimador que se ajusta a varios clasificadores de arboles de decisión en varias submuestras del conjunto de datos y que utiliza el promedio para mejorar la precisión predictiva y evitar el sobreajuste.

Al igual que antes, utilizaremos un búsqueda por rejilla para ajustar los hiperparámetros *ntree* (numero de arboles) y *maxnodes* (número máximo de nodos terminales) y

entrenando el modelo por medio de validación cruzada, hasta encontrar una configuración de hiperparámetros que optimice el ajuste.

```
library(randomForest)

hiperparametros<-expand.grid("ntree"=c(10,50,100,200),"maxnodes"=c(2:10))

cv_forest<-apply(hiperparametros,1,function(hp){

rejilla<-lapply(folds,function(k){

  x<-train[-k,]
  y<-train[k,]

  clasificador<-randomForest(Survived~Pclass+Sex+Embarked+
Alone+Age+Fare_unit,x,
ntree =hp["ntree"],
maxnodes=hp["maxnodes"])

  probabilidad<-predict(object=clasificador,newdata = y,type ='prob')[,2]
  prediccion<-ifelse(probabilidad<0.5,0,1)
  prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))
  matriz_conf<-table(y[,"Survived"],prediccion)
  accuracy<-sum(diag(prop.table(matriz_conf)))
  tpr<-prop.table(matriz_conf,1)[2,2]
  fpr<-prop.table(matriz_conf,1)[1,2]
  metric<-c(accuracy,tpr,fpr)

  roc<-ROCR::prediction(probabilidad,y[,"Survived",drop=F])
  auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

  metric<-c(metric,auc)
names(metric)<-c("Accuracy","TPR","FPR","AUC")
return(metric)

  })
return(rejilla)
})

metricas<-lapply(cv_forest,function(k)sapply(k,function(j)j))
metricas<-t(sapply(metricas,function(k)rowMeans(k)))
metricas<-cbind(metricas,hiperparametros)

g1<-ggplot(metricas, aes(x =as.factor(ntree),
y =as.factor(maxnodes),
fill = AUC)) +
geom_tile(colour ="white")+
labs(x="ntree",y="maxnodes")+
ggtitle("Heatmap for average AUC")+
scale_fill_gradient(low ="lightblue",high ="darkblue")+
geom_text(aes(label =round(AUC,2)),size=3, color ="white")+
theme(panel.background =element_blank())

g1
```



Por el *hetamaps* se aprecia un comportamiento tendencioso, a más nodos y árboles mayor es el *AUC*, pero la mejor combinación la obtenemos con 50 árboles y 8 nodos terminales.

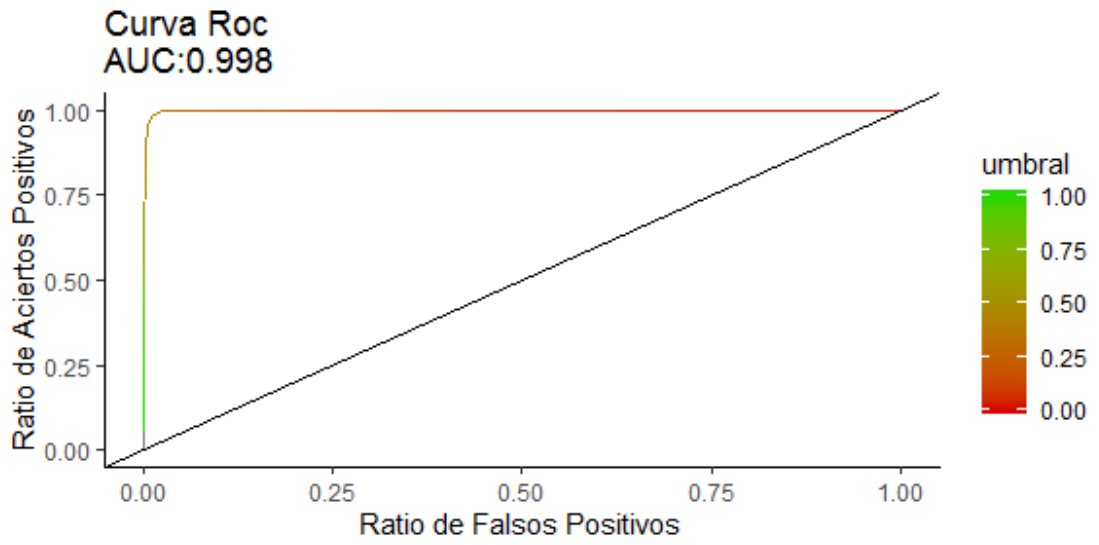
```
modelo_forest<-randomForest(Survived~Pclass+Sex+
Embarked+Alone+Age+Fare_unit,
                            train,
ntree =50,
maxnodes=8,
importance=T)

probabilidad<-predict(object=modelo_forest,
newdata = test,type ='prob')[,2]
prediccion<-ifelse(probabilidad<0.5,0,1)
prediccion<-factor(prediccion,levels=c(0,1),labels=c("No surv","Surv"))
matriz_conf<-table(test[,"Survived"],prediccion)
accuracy<-sum(diag(prop.table(matriz_conf)))
tpr<-prop.table(matriz_conf,1)[2,2]
fpr<-prop.table(matriz_conf,1)[1,2]

roc<-ROCR::prediction(probabilidad,test[,"Survived",drop=F])
auc<-ROCR::performance(roc, measure ="auc")@y.values[[1]]

auc<-round(auc,3)
curva_roc <-ROCR::performance(roc, "tpr", "fpr")
curva_roc<-data.frame("TPR"=unlist(curva_roc@y.values),
"FPR"=unlist(curva_roc@x.values),
"umbral"=unlist(curva_roc@alpha.values))

cat("Desempeño modelo Random Forest:\n",
"\nAccuracy:",accuracy,
"\nTrue Positive Rate:",tpr,
"\nFalse Positive Rate:",fpr)
```

```
> Desempeño modelo Random Forest:
>
> Accuracy: 0.9186603
> True Positive Rate: 0.7828947
> False Positive Rate: 0.003759398
```

Las métricas obtenidas, en general, mejoran a las del árbol de decisión. Tenemos un ajuste del 91.86%, lo que supone un error del 8.13%. Dentro de cada categoría observada apreciamos que el acierto de los que finalmente sobrevivieron es de un 78.28% y el error de los que realmente no sobrevivieron es de un 0.38%.

El *AUC* que obtiene este modelo es muy bueno, con un valor de 0.998.

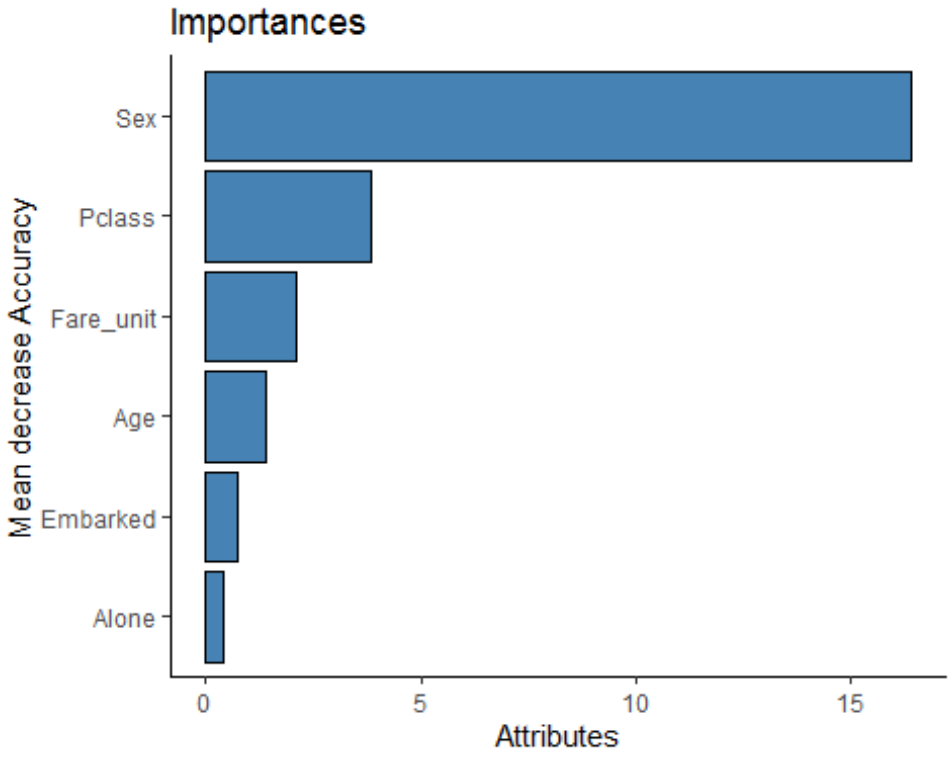
```
ggplot(curva_roc,aes(y=TPR,x=FPR,color=umbral))+
geom_line()+
geom_abline()+
labs(title=paste0("Curva Roc\n",paste0("AUC:",auc)),
y="Ratio de Aciertos Positivos",x="Ratio de Falsos Positivos")+
scale_color_gradient(low="#d50603",high="#26d503")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



Para terminar de analizar este modelo y al igual que hicimos en el modelo basado en la regresión logística vamos a ver qué variables son las más importantes para mejorar el ajuste.

```
importancias<-data.frame("Vars"=rownames(modelo_forest$importance),
"Mean_Decrease_Acc"=modelo_forest$importance[,3]*100)


ggplot(importancias,aes(x=reorder(Vars,Mean_Decrease_Acc),
y=Mean_Decrease_Acc))+
geom_bar(stat ="identity",fill="steelblue",col="black")+
coord_flip() +
```

```
labs(title="Importances",y="Attributes",x="Mean decrease Accuracy")+
theme(axis.line.x =element_line(size =.2),
axis.line.y =element_line(size =.2),
panel.background =element_blank())
```



El grafico muestra en qué medida decrece el ajuste cuando en los arboles no está presente la variable que hace referencia. En el destaca por encima del resto el sexo, por detrás de esta aparecen la clase del pasaje, la tarifa y la edad, algo muy en la línea de lo que hemos aprendido hasta ahora.

4 Conclusiones.

Como hemos podido ver a lo largo del documento y por medio de diferentes análisis, los atributos que discriminan mejor las categorías de superviviente o no son, en un primer lugar destacado, el sexo y luego ya la clase del pasaje y la edad. Parece ser que aquello de “mujeres y niños primero” fue cierto, pero también fue importante la clase, tal vez debido a que los camarotes de primera se situaban en los pisos más altos, mas lejos de la inundación y mas próximos del botes salvavidas.

De los modelos obtenidos el que está basado por random forest es el que ha obtenido mejor *AUC*, seguido (muy de cerca) por el modelo basado en regresión logística. Pero más allá de la pequeña diferencia en esta métrica, destacar que ambos son modelos viables para una buena generalización y lo que es más importante es que ambos coinciden en la importancia de los atributos.