PRACTICA 2. TIPOLOGIA Y CICLO DE VIDA DE LOS DATOS

JUAN LARA CHUPS Y ORIOL MÖSSINGER

2023-06-13

Contents

Configuración entorno R

En este apartado aunque en PDF y html no se vea nada, en el fichero rmd que hemos usado para hacer esta practica se veían todos los instaladores y todas las librerías.

Descripción del dataset

A través del portal de datasets llamado Kaggle hemos encontrado el siguiente dataset para realizar esta práctica:

https://www.kaggle.com/datasets/uciml/adult-census-income

El conjunto de datos "adult.csv" es importante porque proporciona información relevante sobre los ingresos de los individuos en diferentes países del mundo y permite abordar diversas preguntas y problemas relacionados con la predicción de los ingresos altos o bajos de una persona. Algunas de las características clave del conjunto de datos incluyen:

- Edad: La edad del individuo.
- Educación: El nivel de educación alcanzado por el individuo.
- Ocupación: El tipo de ocupación del individuo.
- Estado civil: El estado civil del individuo.
- País de origen: El país de origen del individuo.

- Género: El género del individuo.
- Raza: La raza del individuo.
- Ganancias de capital: Las ganancias de capital del individuo.
- Pérdidas de capital: Las pérdidas de capital del individuo.
- Horas de trabajo: El número de horas de trabajo por semana.

El problema que este conjunto de datos pretende abordar es la predicción de si un individuo tiene ingresos altos o bajos basándose en las características mencionadas anteriormente. Esto puede ser útil para comprender los factores que influyen en los ingresos y para identificar patrones o tendencias que puedan ser utilizados en la toma de decisiones relacionadas con la política, el marketing, la segmentación de mercado, entre otros.

Integración y selección

Cargamos el dataset adult.csv:

```
library(readr)
df <- read_csv("adult.csv")
```

```
## Rows: 32561 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): workclass, education, marital.status, occupation, relationship, rac...
## dbl (6): age, fnlwgt, education.num, capital.gain, capital.loss, hours.per.week
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Mostramos las variables que incluyen el dataset importado:

```
names(df)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "income"
```

Una vez mostradas las variables, renombraremos las variables para que sean más fáciles de gestionar:

```
names(df)=c("edad","clas_trab","n_personas","educacion","id_educacion","estado_civil","ocupacion","rela
names(df)
```

```
##  [1] "edad"         "clas_trab"    "n_personas"   "educacion"    "id_educacion"
##  [6] "estado_civil" "ocupacion"    "relacion"     "raza"         "sexo"
## [11] "ganancias"    "perdidas"     "horas_semana" "nacion"       "ingresos"
```

Limpieza de los datos

Empezamos la limpieza de los datos revisando el tipo de dato que es cada variable:

```
str(df)
```

```
## spc_tbl_ [32,561 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ edad        : num [1:32561] 90 82 66 54 41 34 38 74 68 41 ...
##  $ clas_trab   : chr [1:32561] "?" "Private" "?" "Private" ...
##  $ n_personas  : num [1:32561] 77053 132870 186061 140359 264663 ...
##  $ educacion   : chr [1:32561] "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
```

```
##  $ id_educacion: num [1:32561] 9 9 10 4 10 9 6 16 9 10 ...
##  $ estado_civil: chr [1:32561] "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ ocupacion   : chr [1:32561] "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relacion    : chr [1:32561] "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ raza        : chr [1:32561] "White" "White" "Black" "White" ...
##  $ sexo        : chr [1:32561] "Female" "Female" "Female" "Female" ...
##  $ ganancias   : num [1:32561] 0 0 0 0 0 0 0 0 0 0 ...
##  $ perdidas    : num [1:32561] 4356 4356 4356 3900 3900 ...
##  $ horas_semana: num [1:32561] 40 18 40 40 40 45 40 20 40 60 ...
##  $ nacion      : chr [1:32561] "United-States" "United-States" "United-States" "United-States" ...
##  $ ingresos    : chr [1:32561] "<=50K" "<=50K" "<=50K" "<=50K" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   workclass = col_character(),
##   ..   fnlwgt = col_double(),
##   ..   education = col_character(),
##   ..   education.num = col_double(),
##   ..   marital.status = col_character(),
##   ..   occupation = col_character(),
##   ..   relationship = col_character(),
##   ..   race = col_character(),
##   ..   sex = col_character(),
##   ..   capital.gain = col_double(),
##   ..   capital.loss = col_double(),
##   ..   hours.per.week = col_double(),
##   ..   native.country = col_character(),
##   ..   income = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Mostramos las principales características de cada variable:

```
summary(df)
```

```
##       edad        clas_trab           n_personas       educacion
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :38.58                      Mean   : 189778
##  3rd Qu.:48.00                      3rd Qu.: 237051
##  Max.   :90.00                      Max.   :1484705
##   id_educacion   estado_civil        ocupacion           relacion
##  Min.   : 1.00   Length:32561       Length:32561       Length:32561
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.08
##  3rd Qu.:12.00
##  Max.   :16.00
##      raza               sexo             ganancias        perdidas
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1078   Mean   :  87.3
##                                        3rd Qu.:    0   3rd Qu.:   0.0
```

```
##                                      Max.   :99999   Max.   :4356.0
##   horas_semana      nacion            ingresos
##  Min.   : 1.00   Length:32561       Length:32561
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

Después de las dos previews anteriores, se observa que los datos de ingresos y id_educación se encuentran en formato númerico cuando deberían ser tratados como factores, realizamos el reemplazo:

```
df[, c("id_educacion","ingresos")] <- lapply(df[, c("id_educacion","ingresos")], as.factor)
levels(df$ingresos)<-c('bajos','altos')
ddply(df, .(ingresos), nrow)
```

```
##   ingresos    V1
## 1    bajos 24720
## 2    altos  7841
```

Comprobamos que se han realizado correctamente los cambios:

```
str(df)
```

```
## spc_tbl_ [32,561 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ edad        : num [1:32561] 90 82 66 54 41 34 38 74 68 41 ...
##  $ clas_trab   : chr [1:32561] "?" "Private" "?" "Private" ...
##  $ n_personas  : num [1:32561] 77053 132870 186061 140359 264663 ...
##  $ educacion   : chr [1:32561] "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ id_educacion: Factor w/ 16 levels "1","2","3","4",..: 9 9 10 4 10 9 6 16 9 10 ...
##  $ estado_civil: chr [1:32561] "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ ocupacion   : chr [1:32561] "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relacion    : chr [1:32561] "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ raza        : chr [1:32561] "White" "White" "Black" "White" ...
##  $ sexo        : chr [1:32561] "Female" "Female" "Female" "Female" ...
##  $ ganancias   : num [1:32561] 0 0 0 0 0 0 0 0 0 0 ...
##  $ perdidas    : num [1:32561] 4356 4356 4356 3900 3900 ...
##  $ horas_semana: num [1:32561] 40 18 40 40 40 45 40 20 40 60 ...
##  $ nacion      : chr [1:32561] "United-States" "United-States" "United-States" "United-States" ...
##  $ ingresos    : Factor w/ 2 levels "bajos","altos": 1 1 1 1 1 1 1 2 1 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   workclass = col_character(),
##   ..   fnlwgt = col_double(),
##   ..   education = col_character(),
##   ..   education.num = col_double(),
##   ..   marital.status = col_character(),
##   ..   occupation = col_character(),
##   ..   relationship = col_character(),
##   ..   race = col_character(),
##   ..   sex = col_character(),
##   ..   capital.gain = col_double(),
##   ..   capital.loss = col_double(),
##   ..   hours.per.week = col_double(),
##   ..   native.country = col_character(),
```

```
##   ..   income = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(df)
```

```
##       edad        clas_trab           n_personas       educacion
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :38.58                      Mean   : 189778
##  3rd Qu.:48.00                      3rd Qu.: 237051
##  Max.   :90.00                      Max.   :1484705
##
##   id_educacion   estado_civil        ocupacion           relacion
##  9      :10501   Length:32561       Length:32561       Length:32561
##  10     : 7291   Class :character   Class :character   Class :character
##  13     : 5355   Mode  :character   Mode  :character   Mode  :character
##  14     : 1723
##  11     : 1382
##  7      : 1175
##  (Other): 5134
##      raza               sexo             ganancias        perdidas
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1078   Mean   :  87.3
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##
##   horas_semana      nacion           ingresos
##  Min.   : 1.00   Length:32561       bajos:24720
##  1st Qu.:40.00   Class :character   altos: 7841
##  Median :40.00   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
##
```

Comprobación de resultados que contengan 0 o vacíos

Comprobamos si hay algún NA en el dataset:

```
na <- any(is.na(df))
na
```

```
## [1] FALSE
```

Comprobamos donde están los valores igual a 0:

```
# Identificar las columnas con valores cero y contar los ceros por columna
columnas_cero <- apply(df == 0, 2, sum)

# Filtrar las columnas con al menos un cero
columnas_con_ceros <- columnas_cero[columnas_cero > 0]

# Imprimir los resultados
```

```r
if (length(columnas_con_ceros) > 0) {
  print("Las siguientes columnas contienen valores cero:")
  print(names(columnas_con_ceros))
  print("Cantidad de ceros por columna:")
  print(columnas_con_ceros)
} else {
  print("No se encontraron valores cero en ninguna columna.")
}
```

```
## [1] "Las siguientes columnas contienen valores cero:"
## [1] "ganancias" "perdidas"
## [1] "Cantidad de ceros por columna:"
## ganancias  perdidas
##     29849     31042
```

En este caso, al ser en las columnas de ganancias y perdidas no debemos preocuparnos ya que es perfectamente plausible que existan.

Comprobamos que no haya celdas vacías:

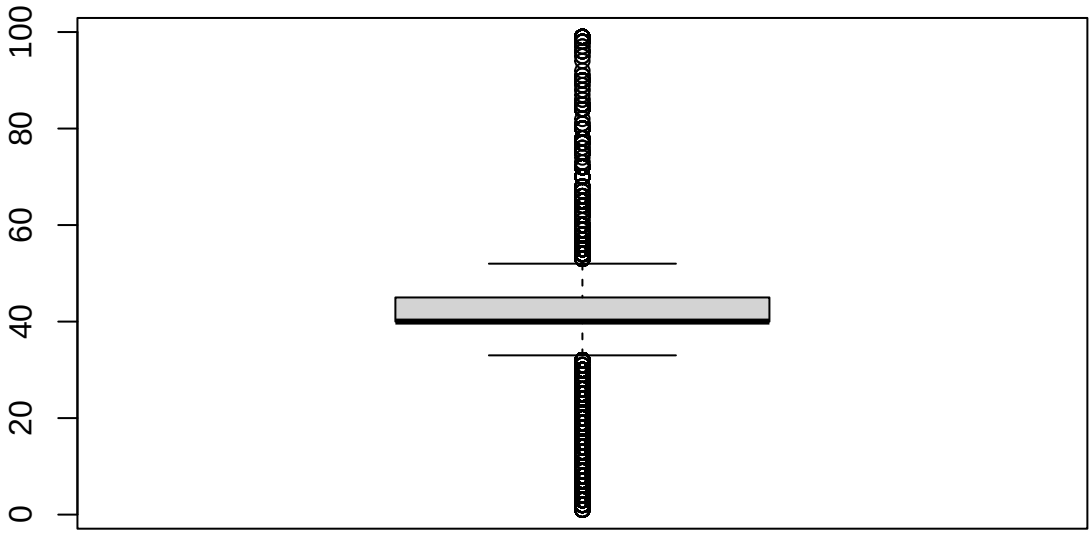
```r
vacios <- sum(df == '')
vacios
```

```
## [1] 0
```

Identificación de valores extremos

Podemos plantear un valor extremo cómo son las horas semanales:

```r
boxplot(df$horas_semana)
```



En este caso tenemos personas que trabajan más de 80 horas semanales, el doble de lo que permite el estatuto de los trabajadores de muchos países. Aunque sea un valor extremo no debemos quitarlo del dataset debido a que es un elemento realista.

Análisis de los datos

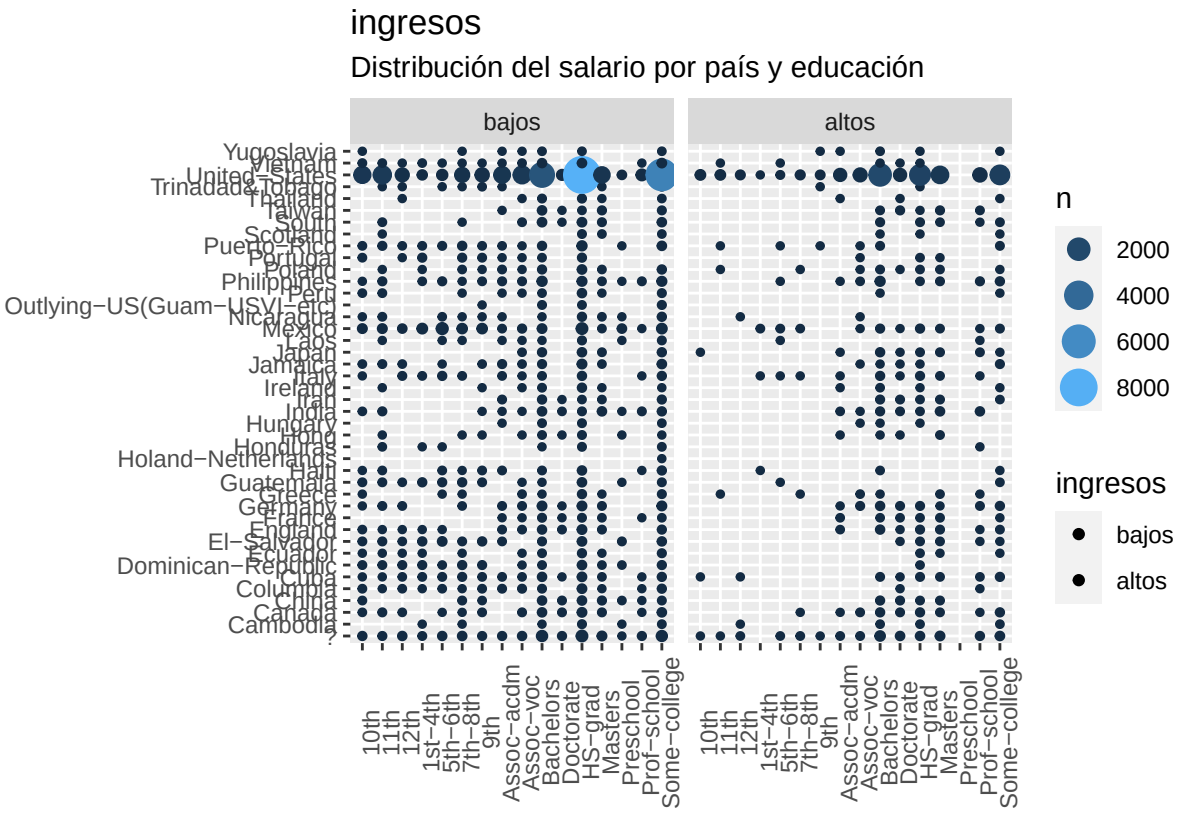
Selección de los grupos de datos que se quieren analizar/comparar

Comparativa estudios vs país

Empezamos este apartado realizando una visualización de la educación según la nación de la persona y educación:

```
ggplot(data = df, aes(x = educacion, y = nacion, fill = ingresos)) + geom_count(aes(color = ..n..,size =
  facet_wrap(~ingresos,ncol = 10)+labs(title='ingresos',
       subtitle='Distribución del salario por país y educación',
       x='', y='')+theme(axis.text.x = element_text(angle = 90)) +(guides(color = 'legend'))#gráfico no
```

```
## Warning: The dot-dot notation (`..n..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(n)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Como podemos observar de una forma poco detallada, los datos con los ingresos más bajos y más altos los mantienen los estadounidenses.

Debido a los distintos tipos que existen en las variables educación y nación no se observan bien los datos. Por tanto, agrupamos las naciones por continente y la educación por nivel educativo conseguido.

Vemos el listado total de países del dataset:

```
valores_unicos <- unique(df$nacion)
print(valores_unicos)
```

```
##  [1] "United-States"              "?"
```

```
##  [3] "Mexico"                    "Greece"
##  [5] "Vietnam"                   "China"
##  [7] "Taiwan"                    "India"
##  [9] "Philippines"               "Trinadad&Tobago"
## [11] "Canada"                    "South"
## [13] "Holand-Netherlands"        "Puerto-Rico"
## [15] "Poland"                    "Iran"
## [17] "England"                   "Germany"
## [19] "Italy"                     "Japan"
## [21] "Hong"                      "Honduras"
## [23] "Cuba"                      "Ireland"
## [25] "Cambodia"                  "Peru"
## [27] "Nicaragua"                 "Dominican-Republic"
## [29] "Haiti"                     "El-Salvador"
## [31] "Hungary"                   "Columbia"
## [33] "Guatemala"                 "Jamaica"
## [35] "Ecuador"                   "France"
## [37] "Yugoslavia"                "Scotland"
## [39] "Portugal"                  "Laos"
## [41] "Thailand"                  "Outlying-US(Guam-USVI-etc)"
```

Factorizamos los países por continentes y creamos la nueva variable llamada continente:

```
df$continente = df$nacion
df$continente = gsub("Cambodia","Asia",df$continente)
df$continente = gsub("Canada","N_America",df$continente)
df$continente = gsub("China","Asia",df$continente)
df$continente = gsub("Hong","Asia",df$continente)
df$continente = gsub("India","Asia",df$continente)
df$continente = gsub("Iran","Asia",df$continente)
df$continente = gsub("Japan","Asia",df$continente)
df$continente = gsub("Laos","Asia",df$continente)
df$continente = gsub("Philippines","Asia",df$continente)
df$continente = gsub("Taiwan","Asia",df$continente)
df$continente = gsub("Thailand","Asia",df$continente)
df$continente = gsub("Vietnam","Asia",df$continente)
df$continente = gsub("Cuba","N_America",df$continente)
df$continente = gsub(" Outlying-US(Guam-USVI-etc)","N_America",df$continente)
df$continente = gsub("United-States","N_America",df$continente)
df$continente = gsub("Columbia","S_America",df$continente)
df$continente = gsub("Dominican-Republic","S_America",df$continente)
df$continente = gsub("Ecuador","S_America",df$continente)
df$continente = gsub("El-Salvador","S_America",df$continente)
df$continente = gsub("Guatemala","S_America",df$continente)
df$continente = gsub("Haiti","S_America",df$continente)
df$continente = gsub("Honduras","S_America",df$continente)
df$continente = gsub("Jamaica","S_America",df$continente)
df$continente = gsub("Mexico","N_America",df$continente)
df$continente = gsub("Nicaragua","S_America",df$continente)
df$continente = gsub("Peru","S_America",df$continente)
df$continente = gsub("Puerto-Rico","S_America",df$continente)
df$continente = gsub("Trinadad&Tobago","S_America",df$continente)
df$continente = gsub("England","Europe",df$continente)
df$continente = gsub("France","Europe",df$continente)
df$continente = gsub("Germany","Europe",df$continente)
```

```
df$continente = gsub("Greece","Europe",df$continente)
df$continente = gsub("Hungary","Europe",df$continente)
df$continente = gsub("Ireland","Europe",df$continente)
df$continente = gsub("Italy","Europe",df$continente)
df$continente = gsub("Poland","Europe",df$continente)
df$continente = gsub("Portugal","Europe",df$continente)
df$continente = gsub("Scotland","Europe",df$continente)
df$continente = gsub("Yugoslavia","Europe",df$continente)
df$continente = gsub("South","Africa",df$continente)
df$continente = gsub("Holand-Netherlands","Europe",df$continente)
df$continente = as.factor(df$continente)
```

Eliminamos los valores que contengan un ? en la nueva variable contiente:

```
df = subset(df, continente != "?")
```

Comprobamos que se hayan creado los continentes correctamente:

```
valores_unicos <- unique(df$continente)
print(valores_unicos)
```

```
## [1] N_America                  Europe
## [3] Asia                       S_America
## [5] Africa                     Outlying-US(Guam-USVI-etc)
## 7 Levels: ? Africa Asia Europe N_America ... S_America
```

Antes de factorizar los estudios, pasamos a reconocerlos:

```
valores_unicos <- unique(df$educacion)
print(valores_unicos)
```

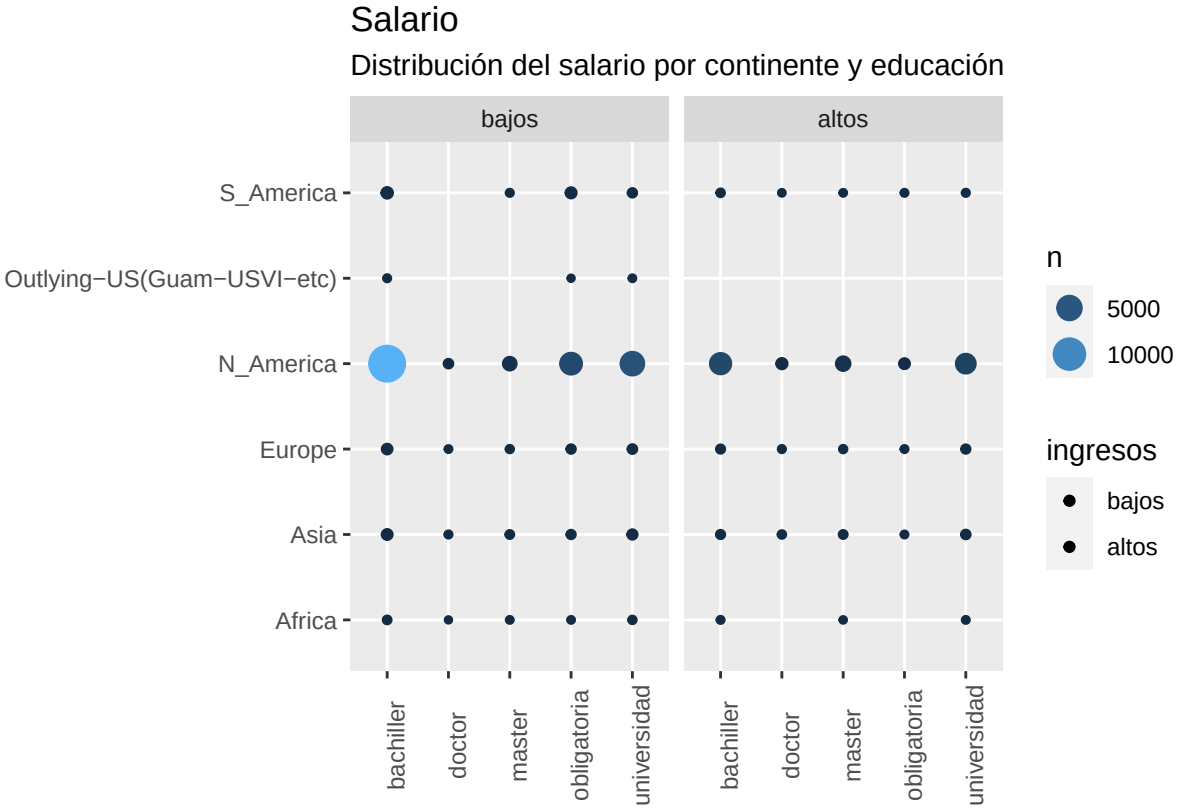
```
##  [1] "HS-grad"      "Some-college" "7th-8th"      "10th"         "Doctorate"
##  [6] "Prof-school"  "Bachelors"    "Masters"      "11th"         "Assoc-voc"
## [11] "1st-4th"      "5th-6th"      "Assoc-acdm"   "12th"         "9th"
## [16] "Preschool"
```

Factorizamos los estudios:

```
df$educacion2 = df$educacion
df$educacion2 = gsub("10th","obligatoria",df$educacion2)
df$educacion2 = gsub("11th","obligatoria",df$educacion2)
df$educacion2 = gsub("12th","obligatoria",df$educacion2)
df$educacion2 = gsub("1st-4th","obligatoria",df$educacion2)
df$educacion2 = gsub("5th-6th","obligatoria",df$educacion2)
df$educacion2 = gsub("7th-8th","obligatoria",df$educacion2)
df$educacion2 = gsub("9th","obligatoria",df$educacion2)
df$educacion2 = gsub("Assoc-acdm","universidad",df$educacion2)
df$educacion2 = gsub("Assoc-voc","universidad",df$educacion2)
df$educacion2 = gsub("Bachelors","universidad",df$educacion2)
df$educacion2 = gsub("Doctorate","doctor",df$educacion2)
df$educacion2 = gsub("HS-grad","bachiller",df$educacion2)
df$educacion2 = gsub("Masters","master",df$educacion2)
df$educacion2 = gsub("Preschool","obligatoria",df$educacion2)
df$educacion2 = gsub("Prof-school","bachiller",df$educacion2)
df$educacion2 = gsub("Some-college","bachiller",df$educacion2)
df$educacion2 = as.factor(df$educacion2)
```

Con ambas variables factorizadas, comprobamos cómo queda el nuevo análisis:

```
ggplot(data = df, aes(x = educacion2, y = continente, fill = ingresos)) + geom_count(aes(color = ..n..,
  facet_wrap(~ingresos,ncol = 10)+labs(title='Salario',
       subtitle='Distribución del salario por continente y educación',
       x='', y='')+theme(axis.text.x = element_text(angle = 90)) +(guides(color = 'legend'))
```

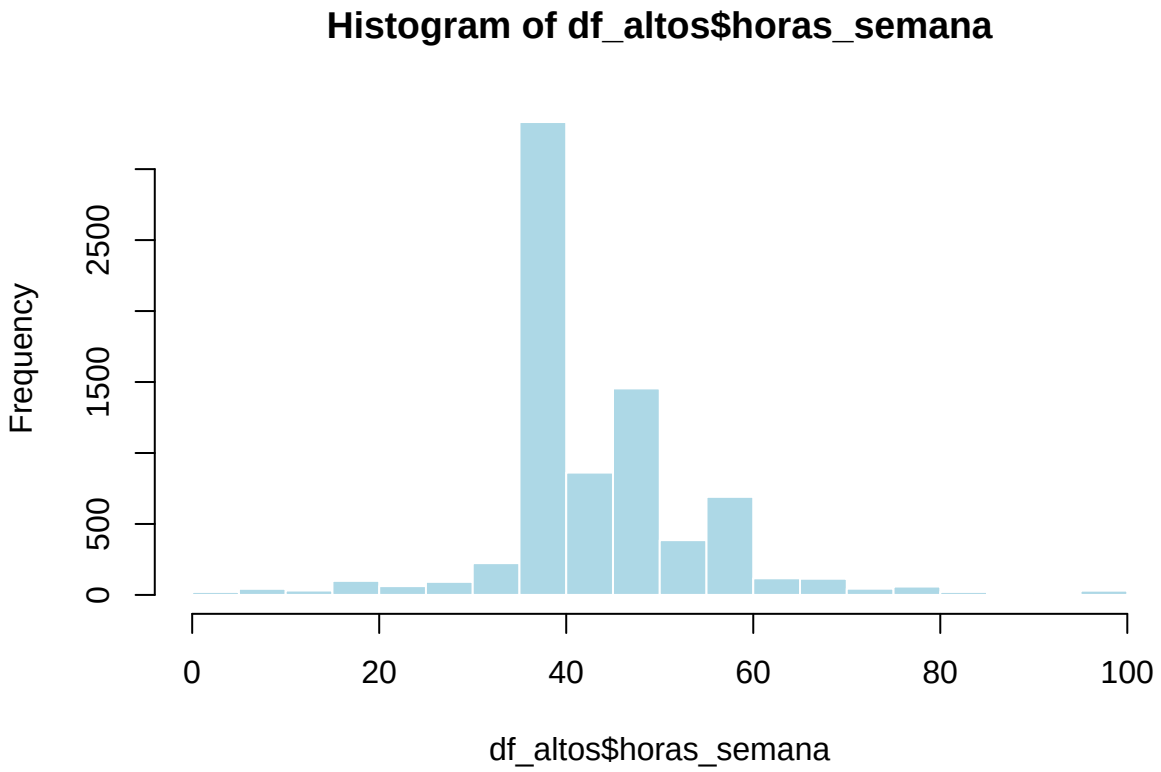


Comprobación de la normalidad y homogeneidad de la varianza

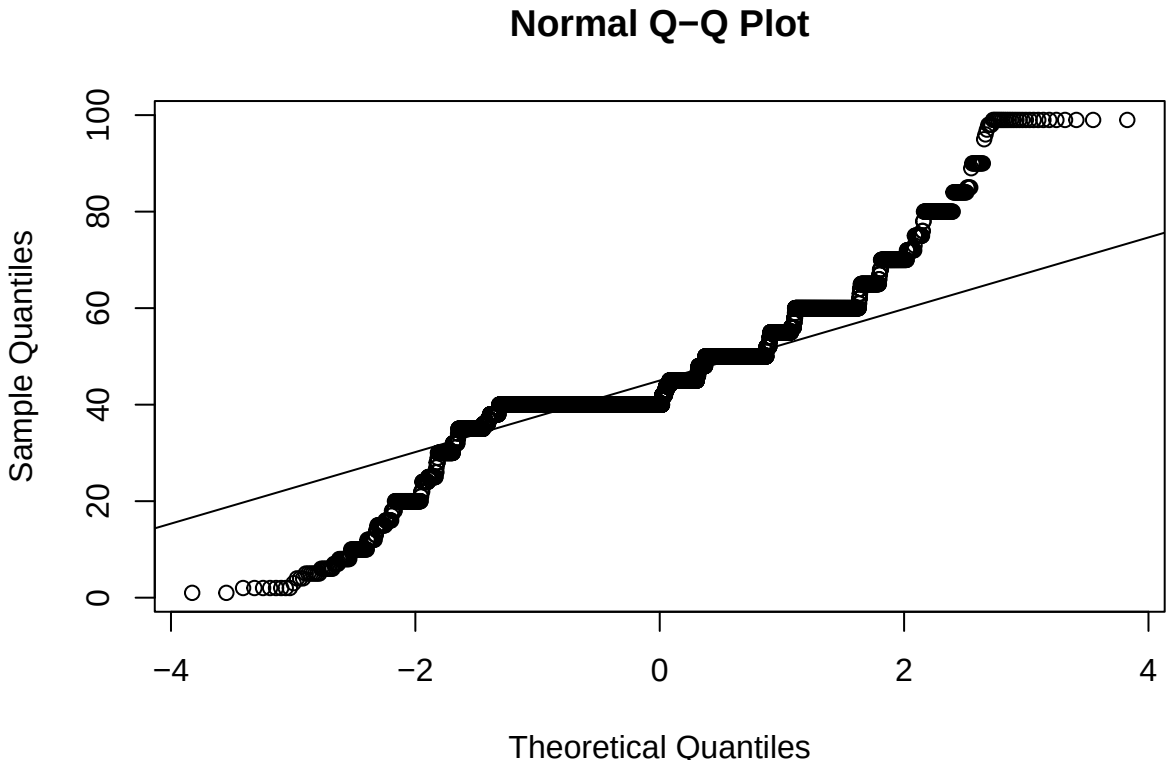
Comprobamos la normalidad de educacion_numerica y horas_semana para para ingresos altos e ingresos bajos:

```
df$educacion_numerica = df$id_educacion
df$educacion_numerica = as.numeric(df$educacion_numerica)
df_bajos <- df[df$ingresos =="bajos",]
df_altos <- df[df$ingresos =="altos",]
```

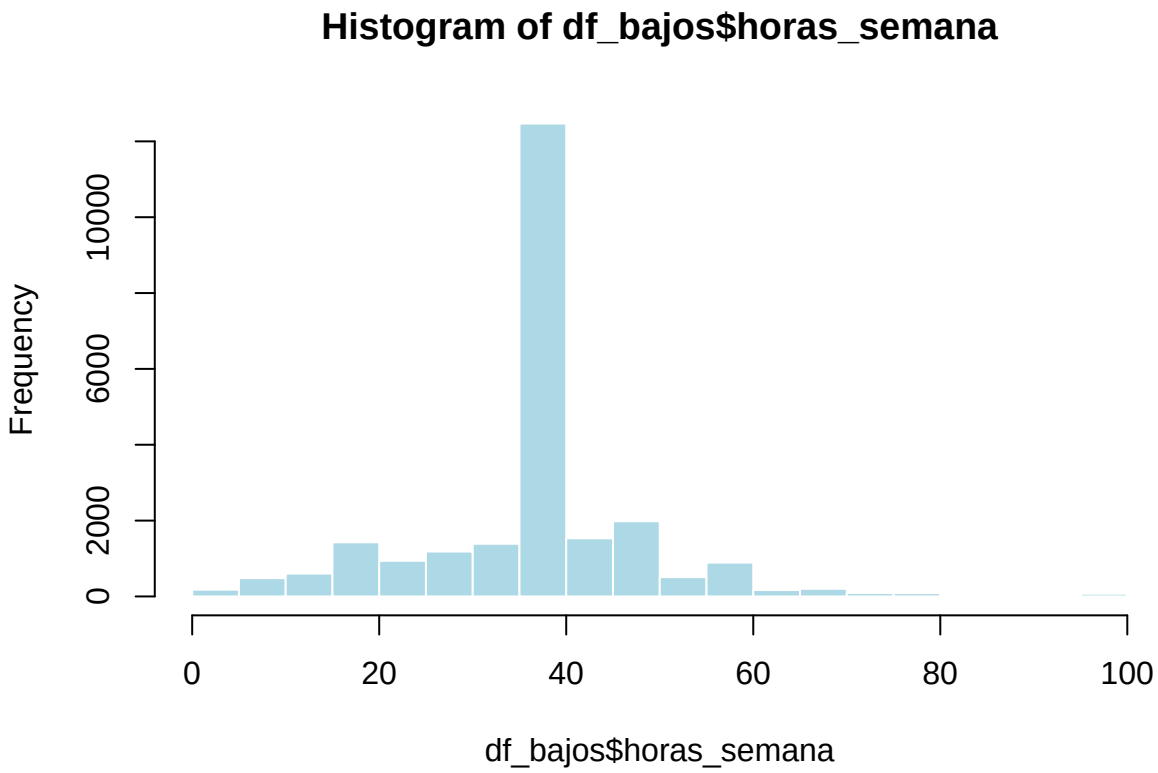
```
hist(df_altos$horas_semana, breaks = "Sturges", col = "lightblue", border = "white")
```



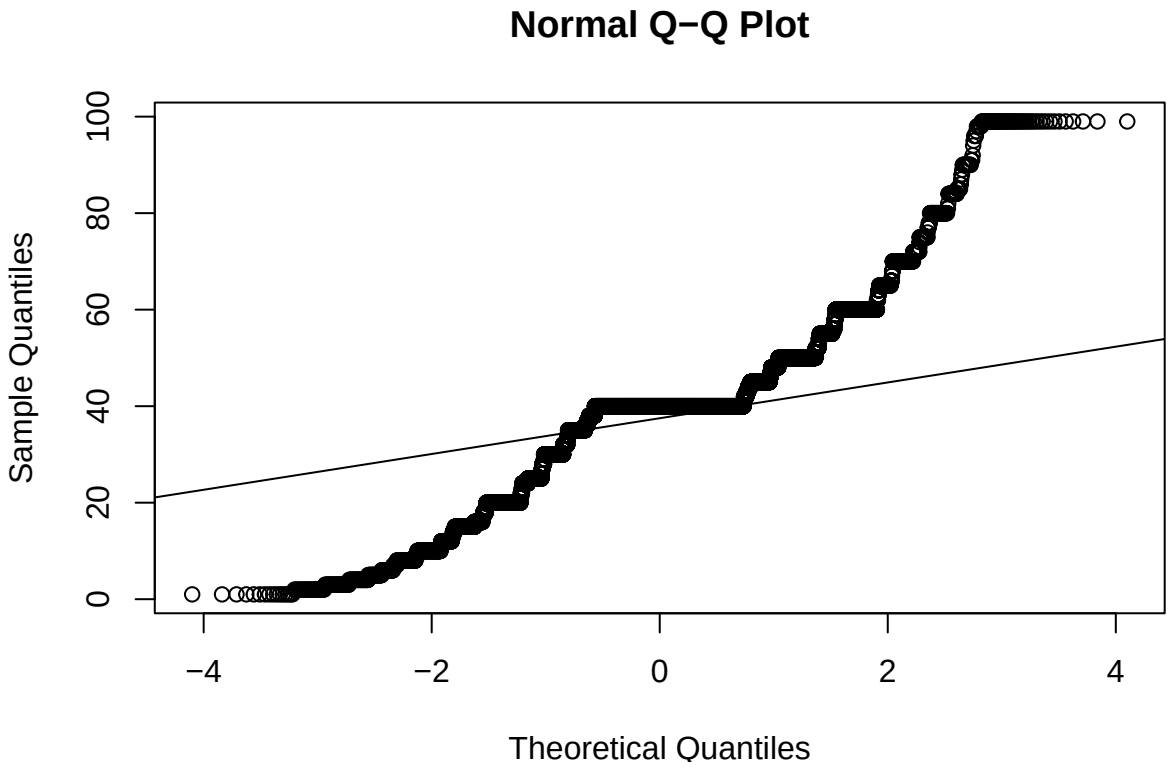

```r
# Gráfico de probabilidad normal (QQ plot)
qqnorm(df_altos$horas_semana)
qqline(df_altos$horas_semana)
```



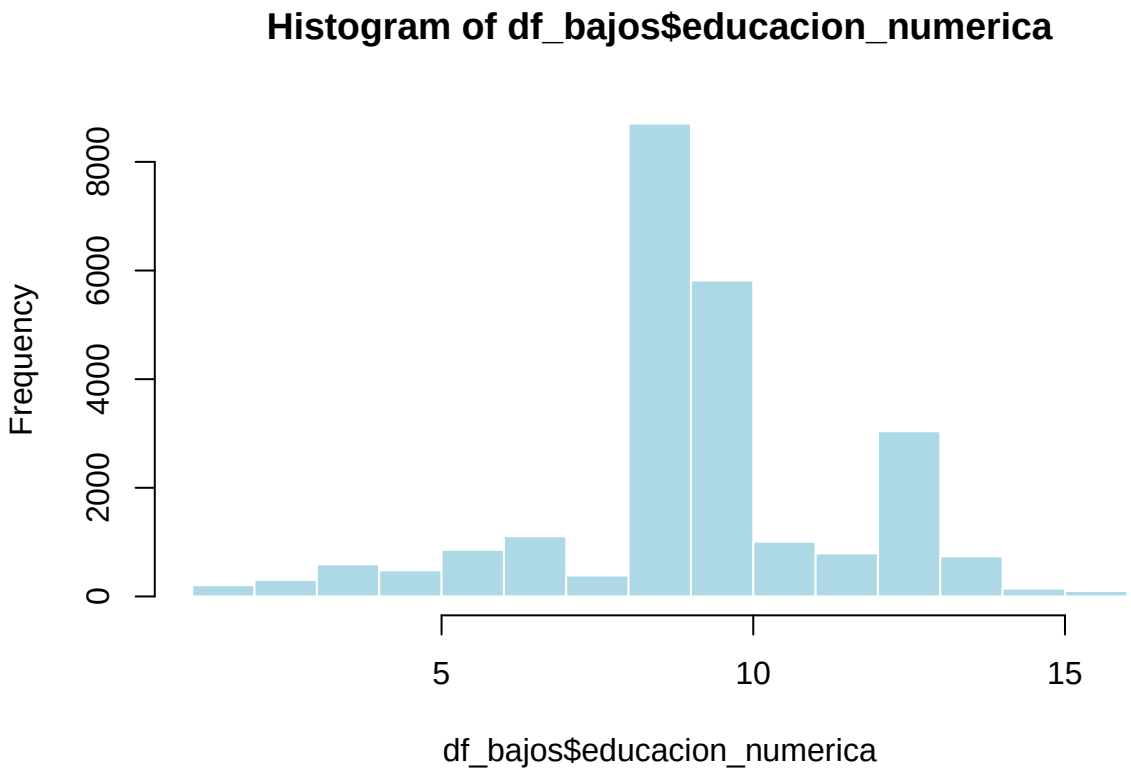
```r
hist(df_bajos$horas_semana, breaks = "Sturges", col = "lightblue", border = "white")
```



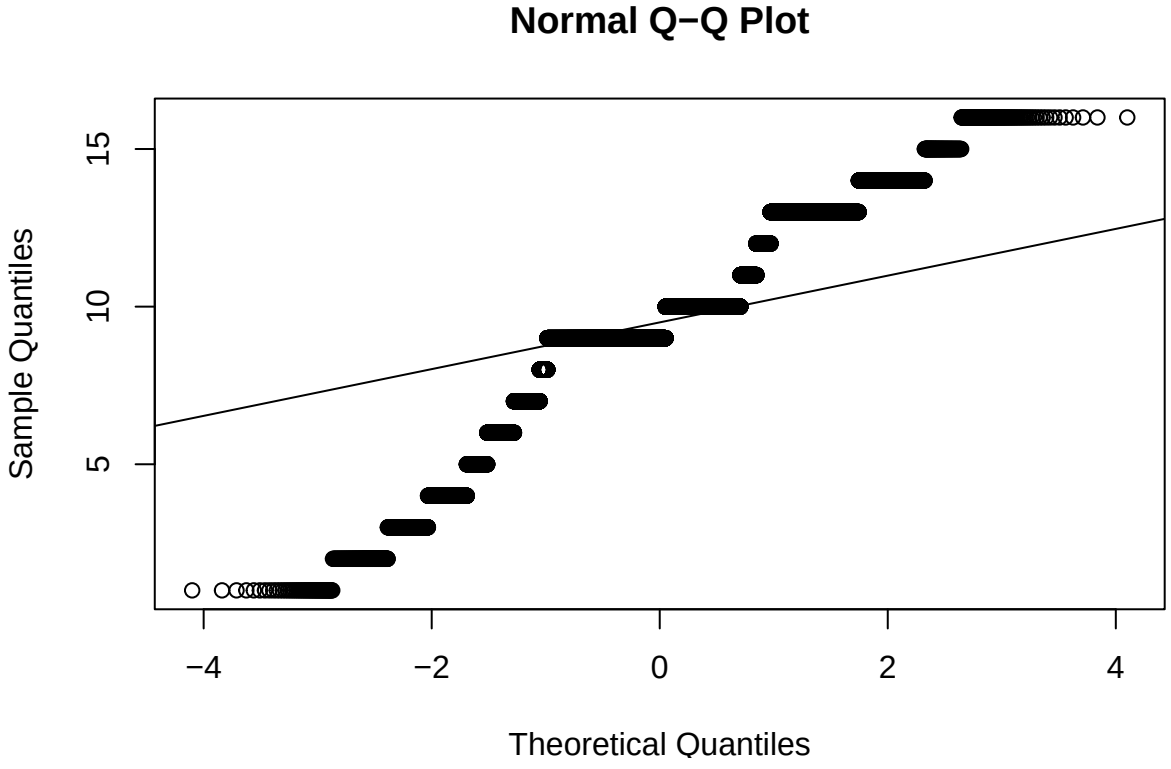

```r
# Gráfico de probabilidad normal (QQ plot)
qqnorm(df_bajos$horas_semana)
qqline(df_bajos$horas_semana)
```



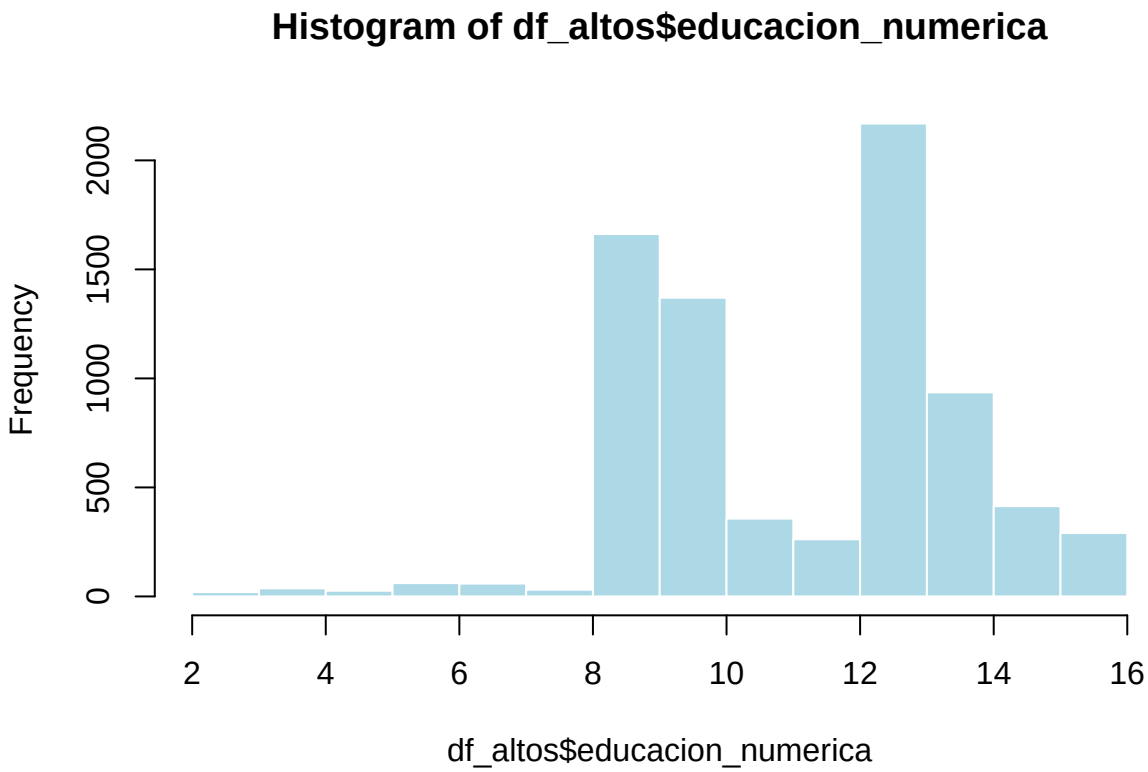
```r
hist(df_bajos$educacion_numerica, breaks = "Sturges", col = "lightblue", border = "white")
```



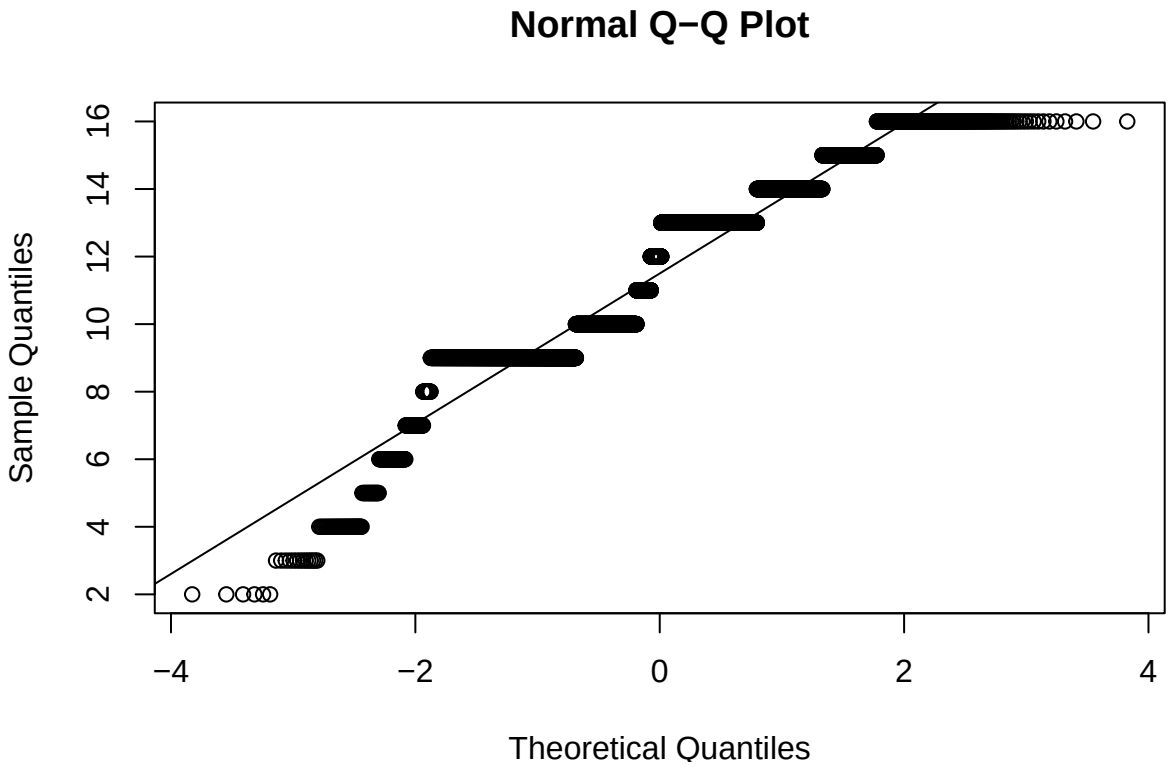

```r
# Gráfico de probabilidad normal (QQ plot)
qqnorm(df_bajos$educacion_numerica)
qqline(df_bajos$educacion_numerica)
```



```r
hist(df_altos$educacion_numerica, breaks = "Sturges", col = "lightblue", border = "white")
```




```r
# Gráfico de probabilidad normal (QQ plot)
qqnorm(df_altos$educacion_numerica)
qqline(df_altos$educacion_numerica)
```



Realizamos el test de varianza de las variables que hemos revisado la normalidad previamente:

```
var.test( df_bajos$educacion_numerica, df_altos$educacion_numerica)
```

```
##
##  F test to compare two variances
##
## data:  df_bajos$educacion_numerica and df_altos$educacion_numerica
## F = 1.0414, num df = 24282, denom df = 7694, p-value = 0.02926
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.004107 1.079640
## sample estimates:
## ratio of variances
##           1.041369
```

```
var.test( df_bajos$horas_semana, df_altos$horas_semana)
```

```
##
##  F test to compare two variances
##
## data:  df_bajos$horas_semana and df_altos$horas_semana
## F = 1.2457, num df = 24282, denom df = 7694, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.201116 1.291468
## sample estimates:
## ratio of variances
##           1.245688
```

Pruebas estadísticas

Test Chi Cuadrado

Antes de hacer el test hay que factorizar las variables:

```
df$clas_trab = as.factor(df$clas_trab)

df$estado_civil = as.factor(df$estado_civil)

df$ocupacion = as.factor(df$ocupacion)

df$relacion = as.factor(df$relacion)

df$raza = as.factor(df$raza)

df$sexo = as.factor(df$sexo)


df$edad_fact = factor(cut(df$edad,c(15,25,45,65,100),labels = c("joven","adulto_joven","adulto","ancian

df$horas_semana_fact = factor(cut(df$horas_semana,c(0,20,40,60,99),labels = c("parcial","completa","exc

df$n_personas_fact = factor(cut(df$edad,c(0,250,500,1000,1500),labels = c("poca","normal","elevada","ex

df$ganancias_fact = factor(cut(df[["ganancias"]],c(-Inf,0,median(df[["ganancias"]][df[["ganancias"]]>0]
```

```
df$perdidas_fact = factor(cut(df[["perdidas"]],c(-Inf,0,median(df[["perdidas"]][df[["perdidas"]]>0]),In
```

```
library(tidyr)

testt<-tidy(chiT<-chisq.test(table(df$ingresos,df$sexo)))
testt<-rbind(testt,tidy(chiT2<-chisq.test(table(df$ingresos,df$raza))))
testt<-rbind(testt,tidy(chiT3<-chisq.test(table(df$ingresos,df$educacion2))))
testt<-rbind(testt,tidy(chiT4<-chisq.test(table(df$ingresos,df$estado_civil))))
testt<-rbind(testt,tidy(chiT5<-chisq.test(table(df$ingresos,df$ocupacion))))
testt<-rbind(testt,tidy(chiT6<-chisq.test(table(df$ingresos,df$relacion))))
testt<-rbind(testt,tidy(chiT7<-chisq.test(table(df$ingresos,df$continente))))
testt<-rbind(testt,tidy(chiT8<-chisq.test(table(df$ingresos,df$horas_semana_fact))))
testt<-rbind(testt,tidy(chiT9<-chisq.test(table(df$ingresos,df$n_personas_fact))))
testt<-rbind(testt,tidy(chiT10<-chisq.test(table(df$ingresos,df$ganancias_fact))))
testt<-rbind(testt,tidy(chiT11<-chisq.test(table(df$ingresos,df$perdidas_fact))))
testt<-rbind(testt,tidy(chiT12<-chisq.test(table(df$ingresos,df$edad_fact))))
testt<-rbind(testt,tidy(chiT13<-chisq.test(table(df$ingresos,df$clas_trab))))

testt$variable<-c("sexo","raza","educacion","estado civil","ocupacion","relacion","continente","horas s
testt
```

```
## # A tibble: 13 x 5
##    statistic     p.value parameter method                                variable
##        <dbl>       <dbl>     <dbl> <chr>                                 <chr>
##  1     1492.   0                 1 Pearson's Chi-squared test with Yat~ sexo
##  2      323.   1.18e- 68         4 Pearson's Chi-squared test           raza
##  3     3153.   0                 4 Pearson's Chi-squared test           educaci~
##  4     6407.   0                 6 Pearson's Chi-squared test           estado ~
##  5     3973.   0                14 Pearson's Chi-squared test           ocupaci~
##  6     6588.   0                 5 Pearson's Chi-squared test           relacion
##  7      NaN  NaN                 6 Pearson's Chi-squared test           contine~
##  8     2140.   0                 3 Pearson's Chi-squared test           horas s~
##  9     8605.   0                 1 Chi-squared test for given probabil~ persona~
## 10     3670.   0                 2 Pearson's Chi-squared test           ganacias
## 11      814.   1.36e-177         2 Pearson's Chi-squared test           perdidas
## 12     2530.   0                 3 Pearson's Chi-squared test           edad
## 13     1023.   1.74e-215         8 Pearson's Chi-squared test           clase t~
```

Cómo podemos observar la hipótesis nula era que las variables eran independientes pero como da un valor menor de 0,05 significa que son independientes.

Test V Cramer

```
c1<-questionr::cramer.v(table(df$ingresos,df$sexo))
c2<-questionr::cramer.v(table(df$ingresos,df$raza))
c3<-questionr::cramer.v(table(df$ingresos,df$educacion2))
c4<-questionr::cramer.v(table(df$ingresos,df$estado_civil))
c5<-questionr::cramer.v(table(df$ingresos,df$ocupacion))
c6<-questionr::cramer.v(table(df$ingresos,df$relacion))
c7<-questionr::cramer.v(table(df$ingresos,df$continente))
c8<-questionr::cramer.v(table(df$ingresos,df$horas_semana))
c10<-questionr::cramer.v(table(df$ingresos,df$ganancias))
c11<-questionr::cramer.v(table(df$ingresos,df$perdidas))
c12<-questionr::cramer.v(table(df$ingresos,df$edad))
```

sexo	raza	educacion	estado_civil	ocupacion	relacion	continente	horas_semana
0.2160697	0.1005028	0.3139941	0.4476248	0.3524648	0.4538992	NaN	0.286145

ganancia_capital	perdida_capital	edad	clase_trabajador
0.4185457	0.2762997	0.3282446	0.178847

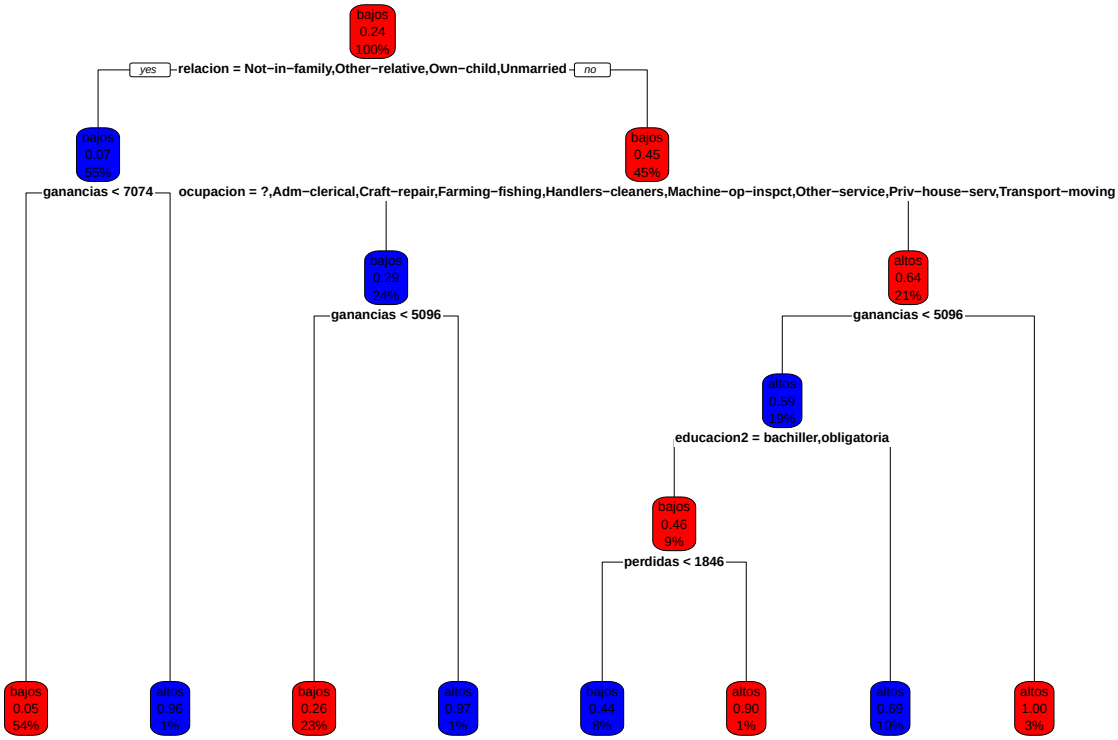
```
c13<-questionr::cramer.v(table(df$ingresos,df$clas_trab))
```

```
library(kableExtra)
cramer<-list(c1,c2,c3,c4,c5,c6,c7,c8)
cramer2 <- as.data.frame(cramer,col.names = c("sexo", "raza", "educacion","estado_civil","ocupacion","r

kbl(cramer2) %>%
    kable_styling()
```

```
cramer3<-list(c10,c11,c12,c13)
cramer4 <- as.data.frame(cramer3,col.names = c("ganancia_capital","perdida_capital","edad","clase_traba
kbl(cramer4) %>%
    kable_styling()
```

Árbol de decisión

```
modelo_arbol <- rpart(ingresos ~ edad + clas_trab + educacion2 + estado_civil + ocupacion + relacion + r
                data = df,
                method="class")

rpart.plot(modelo_arbol,box.col=c("red","blue"))
```



```r
prediccion_arb <- predict(modelo_arbol, newdata = df, type = "class")
confusionMatrix(prediccion_arb, df[["ingresos"]])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bajos altos
##      bajos 23248  3852
##      altos  1035  3843
##
##                Accuracy : 0.8472
##                  95% CI : (0.8432, 0.8511)
##     No Information Rate : 0.7594
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5221
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9574
##             Specificity : 0.4994
##          Pos Pred Value : 0.8579
##          Neg Pred Value : 0.7878
##              Prevalence : 0.7594
##          Detection Rate : 0.7270
##    Detection Prevalence : 0.8475
##       Balanced Accuracy : 0.7284
##
##        'Positive' Class : bajos
##
```

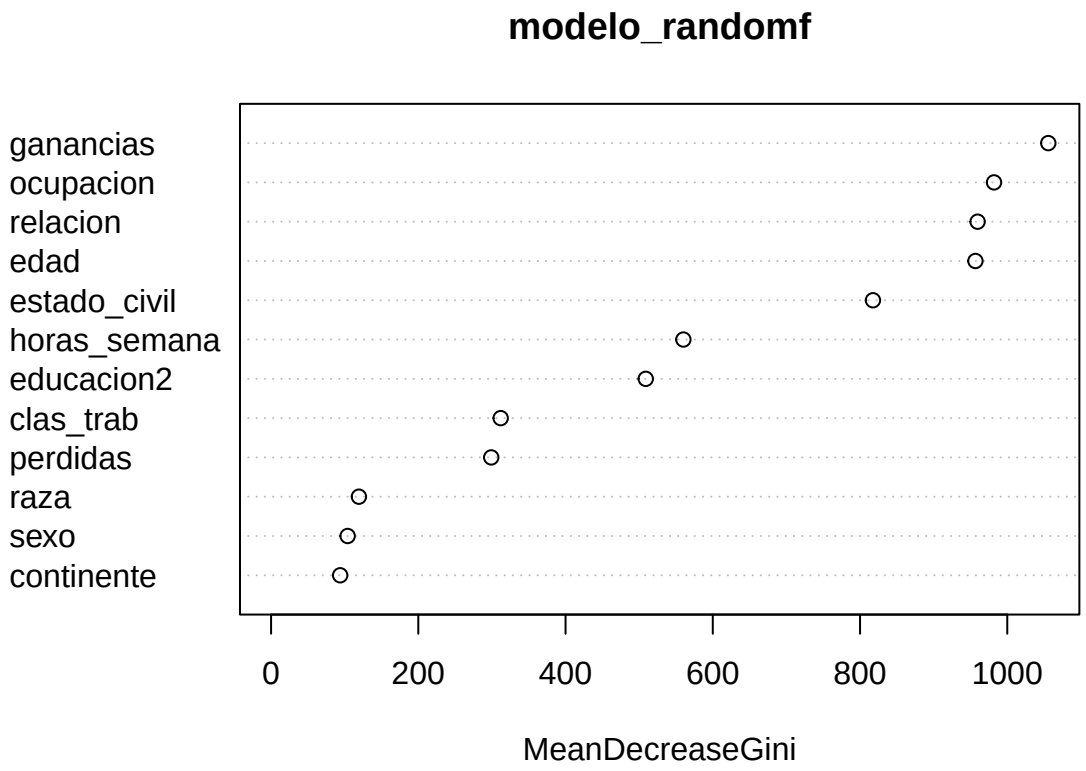
El modelo árbol también da buenos resultados con un accuracy por el que, el 84.72% de las veces, la clasificación que hace es correcta. Observando el gráfico, vemos que en principio las variables que más aportan al modelo de árbol son la relacion, la ocupación que tienen, la ganancia de capital y la edad.

Random forest

```r
df_forest<-subset(df, select = c(edad, clas_trab, educacion2, estado_civil, ocupacion, relacion, raza, s
set.seed(101)
df_tree<- nrow(df_forest)
tree_train <- round(df_tree*0.8)
indices_tree <- sample(1:df_tree , size=tree_train)
datos_train_tree <- df_forest[indices_tree,]
datos_test_tree <- df_forest[-indices_tree,]

modelo_randomf <- randomForest(ingresos ~ edad + clas_trab + educacion2 + estado_civil + ocupacion + rel

predicciones <- predict(modelo_randomf, datos_test_tree)
mc <- with(datos_test_tree,table(predicciones, ingresos))

accuracyRF<-100 * sum(diag(mc)) / sum(mc)

accuracyRF
```

```
## [1] 85.5222
```

```
varImpPlot(modelo_randomf)
```



Con el método de predicción random forest hemos conseguido un accuracy de aprox un 86%, por lo que podemos decir que la iteración de n modelos de árbol como el que hemos hecho, da un resultado bastante mejor que el de uno solo.

Resolución del problema

Tras realizar el análisis de este dataset llegamos a las siguientes conclusiones:

- Utilizando el Teorema del límite central hemos comprobado que las variables años de educación y horas trabajadas por semana, tanto en los grupos con ingresos mayores a 50k y menores, se comportan como distribuciones normales. A su vez, no existe homogeneidad en las varianzas de estas dos variables porque la p-value es menor a 0.05.
- Tras realizar las pruebas de Chi Cuadrado podemos afirmar que existen variables independientes explicativas a la variable ingreso (income).
- El resultado del test de Cramer nos indica que las variables independientes que más explican el ingreso son relación, estado civil y ganancias.
- Realizando un árbol de decisión, hemos llegado a un nivel de precisión del 84.72% lo que significa que el 84.72% de las veces el modelo clasifica correctamente la variable objetivo ingreso.
- Finalmente, realizando un random forest hemos conseguido aumentar la precisión del modelo a un 86%. Por lo que este modelo finalmente es el más preciso.

Contribuciones	Integrantes
Investigación previa	J.L.C, O.M.S
Redacción de las respuestas	J.L.C, O.M.S

Contribuciones	Integrantes
Desarrollo del código	J.L.C, O.M.S
Participación en el vídeo	J.L.C, O.M.S