Practica 2

Jonathan Rodriguez

5 de enero de 2019

Table of Contents

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading required package: reshape

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

##   
## Attaching package: 'rapportools'

## The following object is masked from 'package:dplyr':  
##   
## n

## The following objects are masked from 'package:stats':  
##   
## IQR, median, sd, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, range, sum

## Loading required package: ggplot2

## Loading required package: magrittr

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## corrplot 0.84 loaded

## Loading required package: lattice

##   
## Attaching package: 'psych'

## The following object is masked from 'package:car':  
##   
## logit

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

## Practica 2

### Enunciado

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle. Siguiendo las principales etapas de un proyecto analítico, realizar las diferentes tareas y justificar.

### Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos seleccionado se han obtenido desde <https://www.kaggle.com/aashishjhamtani/automobile-insurancerecoge>. Se compone de 40 variables de las cuales se ha descartado una al no contener ningún dato. Se pasan a describir las demás variables de las cuales se descartaran las que no sean objeto de estudio.

columna - Descripción

* months\_as\_customer-> almacena los meses como cliente
* age -> almacena la edad del cliente
* policy\_number -> almacena numero de poliza
* policy\_bind\_date -> almacena fecha formalización
* insured\_zip -> almacena codigo postal
* insured\_sex -> almacena sexo cliente
* insured\_education\_level-> almacena nivel educacion cliente
* insured\_occupation -> almacena ocupacion cliente
* insured\_relationship-> almacena relación cliente
* incident\_date -> almacena fecha incidente
* incident\_type -> almacena tipo de incidente
* collision\_type -> almacena tipo de incidente
* incident\_severity -> almacena tipo de severidad
* incident\_state -> almacena estado donde se produjo
* incident\_city -> almacena la ciudad incidente
* incident\_hour\_of\_the\_day -> almacena la hora incidente
* number\_of\_vehicles\_involved-> almacena vehiculos implicados
* total\_claim\_amount -> almacena cantidada economica total
* auto\_make -> almacena fabricante vehiculo
* auto\_model -> almacena modelo vehiculo
* auto\_year -> almacena año vehiculo
* fraud\_reported -> almacena si se ha reportado como fraude

Contiene 1000 registros que recogen las reclamaciones realizadas a seguros, perfil del demandante, tipo de vehículos, resultado de la demanda, costes, si ha sido declarado como fraude y demás datos que nos sirven para dar respuesta a distintos tipos de pregunta o problemas.

Pregunta que podemos resolver con este análisis son:

* Obtener un tipo de perfil de usuario para saber cúal es el que más demanda o defrauda, así ponemos un plus en la contratación del seguro o descuento en caso contrario.
* Si existe una relación entre tipo de accidente y los declarados como fraudes. Con esto podemos clasificar si es problable que un tipo de colisión pueda ser fraulenta y marcarla como revisable por un agente de seguros.
* Análisis de tipos de vehículos con más colisiones, estados, etc. Determina el precio del seguro.

Realizamos la carga de los datos.

df <- as.data.frame(read\_csv("C:/temp/insurance\_claims.xls",   
 col\_types = cols(`\_c39` = col\_skip()),   
 locale = locale()))

### 2. Integración y selección de los datos de interés a analizar.

* Son variables variables categóricas/cualitativas nominal(factor): insured\_sex,insured\_occupation,insured\_relationship,incident\_type,incident\_state,incident\_city,number\_of\_vehicles\_involved,total\_claim\_amount,number\_of\_vehicles\_involved,auto\_make,auto\_model,fraud\_reported
* Son variables cualitativas ordinales(ordered): insured\_education\_level,collision\_type,incident\_severity
* Son variables cuantitativas discretas(integer): months\_as\_customer,age,policy\_number,insured\_zip,incident\_hour\_of\_the\_day,total\_claim\_amount,auto\_year

# selección de los datos  
df\_claim <- subset(df,select=c("months\_as\_customer","age","policy\_number","policy\_bind\_date","insured\_zip","insured\_sex","insured\_education\_level","insured\_occupation","insured\_relationship","incident\_date","incident\_type","collision\_type","incident\_severity","incident\_state","incident\_city","incident\_hour\_of\_the\_day","number\_of\_vehicles\_involved","total\_claim\_amount","auto\_make","auto\_model","auto\_year","fraud\_reported"))  
dim(df\_claim)

## [1] 1000 22

Se selecciona un conjunto de datos para el estudio formado por 22 variables desechando aquellas variables que en el caso que nos ocupan no aportan información relevante. Nos centramos en las variables que contienen datos del tipo de cliente,incidente y vehiculos.

### 3. Limpieza de los datos.

#### 3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Verificadas las variables y con el tipo de formato que les corresponde pasamos a realizar una comprobación más profunda de los datos para detectar valores, nulos, NA, ausentes y tratarlos. Se observan en la algunos registros con ceros pero no son considerados valores faltantes. Al inspeccionar el conjunto de datos detectamos en la variable (collision\_type) existen varios 178 registros codificados con “?” reflejando la ausencia de variables. Para tomar una decisión sobre que hacer analizamos más profundamente a que tipo de datos se refiere y cuales son los valores existentes. Si nos fijamos, en la columna anterior todos estos datos con valores faltantes corresponden al tipo de incidente “vehículo robado” o “coche aparcado”. Los daños son menores o triviales lo que cabe pesar que no han tenido una colisión. Por ello, se decide codificarlos como “No collision”, ya que no han sufrido una colisión.

# Comprobar estructura conjunto de datos  
str(df\_claim)

## 'data.frame': 1000 obs. of 22 variables:  
## $ months\_as\_customer : num 328 228 134 256 228 256 137 165 27 212 ...  
## $ age : num 48 42 29 41 44 39 34 37 33 42 ...  
## $ policy\_number : num 521585 342868 687698 227811 367455 ...  
## $ policy\_bind\_date : Date, format: "2014-10-17" "2006-06-27" ...  
## $ insured\_zip : num 466132 468176 430632 608117 610706 ...  
## $ insured\_sex : chr "MALE" "MALE" "FEMALE" "FEMALE" ...  
## $ insured\_education\_level : chr "MD" "MD" "PhD" "PhD" ...  
## $ insured\_occupation : chr "craft-repair" "machine-op-inspct" "sales" "armed-forces" ...  
## $ insured\_relationship : chr "husband" "other-relative" "own-child" "unmarried" ...  
## $ incident\_date : Date, format: "2015-01-25" "2015-01-21" ...  
## $ incident\_type : chr "Single Vehicle Collision" "Vehicle Theft" "Multi-vehicle Collision" "Single Vehicle Collision" ...  
## $ collision\_type : chr "Side Collision" "?" "Rear Collision" "Front Collision" ...  
## $ incident\_severity : chr "Major Damage" "Minor Damage" "Minor Damage" "Major Damage" ...  
## $ incident\_state : chr "SC" "VA" "NY" "OH" ...  
## $ incident\_city : chr "Columbus" "Riverwood" "Columbus" "Arlington" ...  
## $ incident\_hour\_of\_the\_day : num 5 8 7 5 20 19 0 23 21 14 ...  
## $ number\_of\_vehicles\_involved: num 1 1 3 1 1 3 3 3 1 1 ...  
## $ total\_claim\_amount : num 71610 5070 34650 63400 6500 ...  
## $ auto\_make : chr "Saab" "Mercedes" "Dodge" "Chevrolet" ...  
## $ auto\_model : chr "92x" "E400" "RAM" "Tahoe" ...  
## $ auto\_year : num 2004 2007 2007 2014 2009 ...  
## $ fraud\_reported : chr "Y" "Y" "N" "Y" ...

head(df\_claim)

## months\_as\_customer age policy\_number policy\_bind\_date insured\_zip  
## 1 328 48 521585 2014-10-17 466132  
## 2 228 42 342868 2006-06-27 468176  
## 3 134 29 687698 2000-09-06 430632  
## 4 256 41 227811 1990-05-25 608117  
## 5 228 44 367455 2014-06-06 610706  
## 6 256 39 104594 2006-10-12 478456  
## insured\_sex insured\_education\_level insured\_occupation  
## 1 MALE MD craft-repair  
## 2 MALE MD machine-op-inspct  
## 3 FEMALE PhD sales  
## 4 FEMALE PhD armed-forces  
## 5 MALE Associate sales  
## 6 FEMALE PhD tech-support  
## insured\_relationship incident\_date incident\_type  
## 1 husband 2015-01-25 Single Vehicle Collision  
## 2 other-relative 2015-01-21 Vehicle Theft  
## 3 own-child 2015-02-22 Multi-vehicle Collision  
## 4 unmarried 2015-01-10 Single Vehicle Collision  
## 5 unmarried 2015-02-17 Vehicle Theft  
## 6 unmarried 2015-01-02 Multi-vehicle Collision  
## collision\_type incident\_severity incident\_state incident\_city  
## 1 Side Collision Major Damage SC Columbus  
## 2 ? Minor Damage VA Riverwood  
## 3 Rear Collision Minor Damage NY Columbus  
## 4 Front Collision Major Damage OH Arlington  
## 5 ? Minor Damage NY Arlington  
## 6 Rear Collision Major Damage SC Arlington  
## incident\_hour\_of\_the\_day number\_of\_vehicles\_involved total\_claim\_amount  
## 1 5 1 71610  
## 2 8 1 5070  
## 3 7 3 34650  
## 4 5 1 63400  
## 5 20 1 6500  
## 6 19 3 64100  
## auto\_make auto\_model auto\_year fraud\_reported  
## 1 Saab 92x 2004 Y  
## 2 Mercedes E400 2007 Y  
## 3 Dodge RAM 2007 N  
## 4 Chevrolet Tahoe 2014 Y  
## 5 Accura RSX 2009 N  
## 6 Saab 95 2003 Y

tail(df\_claim)

## months\_as\_customer age policy\_number policy\_bind\_date insured\_zip  
## 995 141 30 681486 2007-03-24 430665  
## 996 3 38 941851 1991-07-16 431289  
## 997 285 41 186934 2014-01-05 608177  
## 998 130 34 918516 2003-02-17 442797  
## 999 458 62 533940 2011-11-18 441714  
## 1000 456 60 556080 1996-11-11 612260  
## insured\_sex insured\_education\_level insured\_occupation  
## 995 MALE High School sales  
## 996 FEMALE Masters craft-repair  
## 997 FEMALE PhD prof-specialty  
## 998 FEMALE Masters armed-forces  
## 999 MALE Associate handlers-cleaners  
## 1000 FEMALE Associate sales  
## insured\_relationship incident\_date incident\_type  
## 995 own-child 2015-01-22 Parked Car  
## 996 unmarried 2015-02-22 Single Vehicle Collision  
## 997 wife 2015-01-24 Single Vehicle Collision  
## 998 other-relative 2015-01-23 Multi-vehicle Collision  
## 999 wife 2015-02-26 Single Vehicle Collision  
## 1000 husband 2015-02-26 Parked Car  
## collision\_type incident\_severity incident\_state incident\_city  
## 995 ? Minor Damage SC Northbend  
## 996 Front Collision Minor Damage NC Northbrook  
## 997 Rear Collision Major Damage SC Northbend  
## 998 Side Collision Minor Damage NC Arlington  
## 999 Rear Collision Major Damage NY Arlington  
## 1000 ? Minor Damage WV Columbus  
## incident\_hour\_of\_the\_day number\_of\_vehicles\_involved  
## 995 6 1  
## 996 20 1  
## 997 23 1  
## 998 4 3  
## 999 2 1  
## 1000 6 1  
## total\_claim\_amount auto\_make auto\_model auto\_year fraud\_reported  
## 995 6480 Honda Civic 1996 N  
## 996 87200 Honda Accord 2006 N  
## 997 108480 Volkswagen Passat 2015 N  
## 998 67500 Suburu Impreza 1996 N  
## 999 46980 Audi A5 1998 N  
## 1000 5060 Mercedes E400 2007 N

# verificamos valores  
print(paste("Valores Null", is.null(df\_claim),sep = " "))

## [1] "Valores Null FALSE"

print(is.null(df\_claim))

## [1] FALSE

print(paste("Valores Vacios por columnas:" ))

## [1] "Valores Vacios por columnas:"

colSums(sapply(df\_claim, is.empty))

## months\_as\_customer age   
## 1 0   
## policy\_number policy\_bind\_date   
## 0 0   
## insured\_zip insured\_sex   
## 0 0   
## insured\_education\_level insured\_occupation   
## 0 0   
## insured\_relationship incident\_date   
## 0 0   
## incident\_type collision\_type   
## 0 0   
## incident\_severity incident\_state   
## 0 0   
## incident\_city incident\_hour\_of\_the\_day   
## 0 52   
## number\_of\_vehicles\_involved total\_claim\_amount   
## 0 0   
## auto\_make auto\_model   
## 0 0   
## auto\_year fraud\_reported   
## 0 0

print(paste("Valores NA por columnas:"))

## [1] "Valores NA por columnas:"

colSums(sapply(df\_claim, is.na))

## months\_as\_customer age   
## 0 0   
## policy\_number policy\_bind\_date   
## 0 0   
## insured\_zip insured\_sex   
## 0 0   
## insured\_education\_level insured\_occupation   
## 0 0   
## insured\_relationship incident\_date   
## 0 0   
## incident\_type collision\_type   
## 0 0   
## incident\_severity incident\_state   
## 0 0   
## incident\_city incident\_hour\_of\_the\_day   
## 0 0   
## number\_of\_vehicles\_involved total\_claim\_amount   
## 0 0   
## auto\_make auto\_model   
## 0 0   
## auto\_year fraud\_reported   
## 0 0

print(paste("valores con ?"))

## [1] "valores con ?"

table(df\_claim$collision\_type)

##   
## ? Front Collision Rear Collision Side Collision   
## 178 254 292 276

head(df\_claim %>% filter(df\_claim$collision\_type=="?"))

## months\_as\_customer age policy\_number policy\_bind\_date insured\_zip  
## 1 228 42 342868 2006-06-27 468176  
## 2 228 44 367455 2014-06-06 610706  
## 3 121 34 626808 2012-10-26 464652  
## 4 257 43 863236 1990-09-20 436984  
## 5 202 34 608513 2002-07-18 607730  
## 6 289 49 933293 1993-02-03 620757  
## insured\_sex insured\_education\_level insured\_occupation  
## 1 MALE MD machine-op-inspct  
## 2 MALE Associate sales  
## 3 FEMALE MD armed-forces  
## 4 MALE High School prof-specialty  
## 5 MALE JD exec-managerial  
## 6 FEMALE JD priv-house-serv  
## insured\_relationship incident\_date incident\_type collision\_type  
## 1 other-relative 2015-01-21 Vehicle Theft ?  
## 2 unmarried 2015-02-17 Vehicle Theft ?  
## 3 wife 2015-01-08 Parked Car ?  
## 4 own-child 2015-01-28 Parked Car ?  
## 5 not-in-family 2015-01-07 Vehicle Theft ?  
## 6 unmarried 2015-01-18 Parked Car ?  
## incident\_severity incident\_state incident\_city incident\_hour\_of\_the\_day  
## 1 Minor Damage VA Riverwood 8  
## 2 Minor Damage NY Arlington 20  
## 3 Minor Damage SC Springfield 5  
## 4 Minor Damage PA Arlington 4  
## 5 Minor Damage VA Northbrook 5  
## 6 Minor Damage WV Arlington 16  
## number\_of\_vehicles\_involved total\_claim\_amount auto\_make auto\_model  
## 1 1 5070 Mercedes E400  
## 2 1 6500 Accura RSX  
## 3 1 7280 Toyota Highlander  
## 4 1 9020 Toyota Camry  
## 5 1 5720 Suburu Forrestor  
## 6 1 5330 Suburu Legacy  
## auto\_year fraud\_reported  
## 1 2007 Y  
## 2 2009 N  
## 3 2010 N  
## 4 2005 N  
## 5 2003 Y  
## 6 2001 N

# registros detectados procedemos a codificar   
i <- which(df\_claim$collision\_type=="?")  
df\_claim[i,"collision\_type"]<-"No collision"  
# comprobamos  
head(df\_claim %>% filter(df\_claim$collision\_type=="No collision"))

## months\_as\_customer age policy\_number policy\_bind\_date insured\_zip  
## 1 228 42 342868 2006-06-27 468176  
## 2 228 44 367455 2014-06-06 610706  
## 3 121 34 626808 2012-10-26 464652  
## 4 257 43 863236 1990-09-20 436984  
## 5 202 34 608513 2002-07-18 607730  
## 6 289 49 933293 1993-02-03 620757  
## insured\_sex insured\_education\_level insured\_occupation  
## 1 MALE MD machine-op-inspct  
## 2 MALE Associate sales  
## 3 FEMALE MD armed-forces  
## 4 MALE High School prof-specialty  
## 5 MALE JD exec-managerial  
## 6 FEMALE JD priv-house-serv  
## insured\_relationship incident\_date incident\_type collision\_type  
## 1 other-relative 2015-01-21 Vehicle Theft No collision  
## 2 unmarried 2015-02-17 Vehicle Theft No collision  
## 3 wife 2015-01-08 Parked Car No collision  
## 4 own-child 2015-01-28 Parked Car No collision  
## 5 not-in-family 2015-01-07 Vehicle Theft No collision  
## 6 unmarried 2015-01-18 Parked Car No collision  
## incident\_severity incident\_state incident\_city incident\_hour\_of\_the\_day  
## 1 Minor Damage VA Riverwood 8  
## 2 Minor Damage NY Arlington 20  
## 3 Minor Damage SC Springfield 5  
## 4 Minor Damage PA Arlington 4  
## 5 Minor Damage VA Northbrook 5  
## 6 Minor Damage WV Arlington 16  
## number\_of\_vehicles\_involved total\_claim\_amount auto\_make auto\_model  
## 1 1 5070 Mercedes E400  
## 2 1 6500 Accura RSX  
## 3 1 7280 Toyota Highlander  
## 4 1 9020 Toyota Camry  
## 5 1 5720 Suburu Forrestor  
## 6 1 5330 Suburu Legacy  
## auto\_year fraud\_reported  
## 1 2007 Y  
## 2 2009 N  
## 3 2010 N  
## 4 2005 N  
## 5 2003 Y  
## 6 2001 N

Comprobamos la asignación correcta de las clases en la carga de los datos, en caso contrario serán clasificadas con el tipo apropiado.

# inicializar tipos  
n.var <- names(df\_claim)  
var.factor <- c(6,8,9,11,14,15,17,18,19,20,22)  
var.integer <- c(1,2,3,5,16,21)  
var.ordered <- c(7,12,13)  
var.date <- c(4,10)  
var.tipus <- vector(mode="character",length=ncol(df\_claim))  
var.tipus[var.factor] <- "factor"  
var.tipus[var.integer] <- "integer"  
var.tipus[var.ordered] <- "ordered"  
var.tipus[var.date] <- "date"  
  
# Revisar clases de la selección  
res <- sapply(df\_claim,class)   
kable(data.frame(variables=names(res),clase=as.vector(res)))

|  |  |
| --- | --- |
| variables | clase |
| months\_as\_customer | numeric |
| age | numeric |
| policy\_number | numeric |
| policy\_bind\_date | Date |
| insured\_zip | numeric |
| insured\_sex | character |
| insured\_education\_level | character |
| insured\_occupation | character |
| insured\_relationship | character |
| incident\_date | Date |
| incident\_type | character |
| collision\_type | character |
| incident\_severity | character |
| incident\_state | character |
| incident\_city | character |
| incident\_hour\_of\_the\_day | numeric |
| number\_of\_vehicles\_involved | numeric |
| total\_claim\_amount | numeric |
| auto\_make | character |
| auto\_model | character |
| auto\_year | numeric |
| fraud\_reported | character |

var\_wrong <- n.var[res != var.tipus]  
var\_wrong

## [1] "months\_as\_customer" "age"   
## [3] "policy\_number" "policy\_bind\_date"   
## [5] "insured\_zip" "insured\_sex"   
## [7] "insured\_education\_level" "insured\_occupation"   
## [9] "insured\_relationship" "incident\_date"   
## [11] "incident\_type" "collision\_type"   
## [13] "incident\_severity" "incident\_state"   
## [15] "incident\_city" "incident\_hour\_of\_the\_day"   
## [17] "number\_of\_vehicles\_involved" "total\_claim\_amount"   
## [19] "auto\_make" "auto\_model"   
## [21] "auto\_year" "fraud\_reported"

# Asignar como factor  
for (i in var.factor) {df\_claim[,i] <- as.factor(df\_claim[,i])}  
# Asignar como integer  
for (i in var.integer) {df\_claim[,i] <- as.integer(df\_claim[,i])}  
# Asignar como ordered  
for (i in var.ordered) {df\_claim[,i] <- as.ordered(df\_claim[,i])}  
# Asignar como Date  
for (i in var.date) {df\_claim[,i] <- as.Date(df\_claim[,i])}  
  
str(df\_claim)

## 'data.frame': 1000 obs. of 22 variables:  
## $ months\_as\_customer : int 328 228 134 256 228 256 137 165 27 212 ...  
## $ age : int 48 42 29 41 44 39 34 37 33 42 ...  
## $ policy\_number : int 521585 342868 687698 227811 367455 104594 413978 429027 485665 636550 ...  
## $ policy\_bind\_date : Date, format: "2014-10-17" "2006-06-27" ...  
## $ insured\_zip : int 466132 468176 430632 608117 610706 478456 441716 603195 601734 600983 ...  
## $ insured\_sex : Factor w/ 2 levels "FEMALE","MALE": 2 2 1 1 2 1 2 2 1 2 ...  
## $ insured\_education\_level : Ord.factor w/ 7 levels "Associate"<"College"<..: 6 6 7 7 1 7 7 1 7 7 ...  
## $ insured\_occupation : Factor w/ 14 levels "adm-clerical",..: 3 7 12 2 12 13 10 13 8 9 ...  
## $ insured\_relationship : Factor w/ 6 levels "husband","not-in-family",..: 1 3 4 5 5 5 1 5 4 6 ...  
## $ incident\_date : Date, format: "2015-01-25" "2015-01-21" ...  
## $ incident\_type : Factor w/ 4 levels "Multi-vehicle Collision",..: 3 4 1 3 4 1 1 1 3 3 ...  
## $ collision\_type : Ord.factor w/ 4 levels "Front Collision"<..: 4 2 3 1 2 3 1 1 1 3 ...  
## $ incident\_severity : Ord.factor w/ 4 levels "Major Damage"<..: 1 2 2 1 2 1 2 3 3 3 ...  
## $ incident\_state : Factor w/ 7 levels "NC","NY","OH",..: 5 6 2 3 2 5 2 6 7 1 ...  
## $ incident\_city : Factor w/ 7 levels "Arlington","Columbus",..: 2 6 2 1 1 1 7 2 1 3 ...  
## $ incident\_hour\_of\_the\_day : int 5 8 7 5 20 19 0 23 21 14 ...  
## $ number\_of\_vehicles\_involved: Factor w/ 4 levels "1","2","3","4": 1 1 3 1 1 3 3 3 1 1 ...  
## $ total\_claim\_amount : Factor w/ 763 levels "100","1920","2160",..: 564 45 144 452 82 464 643 278 122 188 ...  
## $ auto\_make : Factor w/ 14 levels "Accura","Audi",..: 11 9 5 4 1 11 10 2 13 11 ...  
## $ auto\_model : Factor w/ 39 levels "3 Series","92x",..: 2 13 31 34 32 4 30 6 9 2 ...  
## $ auto\_year : int 2004 2007 2007 2014 2009 2003 2012 2015 2012 1996 ...  
## $ fraud\_reported : Factor w/ 2 levels "N","Y": 2 2 1 2 1 2 1 1 1 1 ...

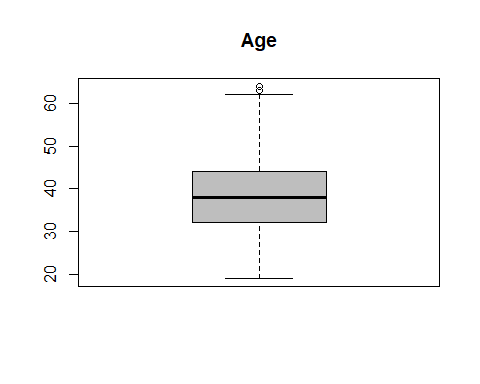
#### 3.2. Identificación y tratamiento de valores extremos.

Vamos a identificar valores atípicos en variables cuantitativas, para ellos se presenta un boxplot de cada variable cuantitativa. Primero revisamos si existen valores y en caso afirmativo se representa gráficamente. Se han detectado valores extremos en la variable Age, sin embargo son valores totalmente aceptables dentro del rango de edad, nada significativo. Las demás variables no tienen sentido revisar valores extremos.

r <- sapply(df\_claim,class)   
r <- which(r=="integer")  
for (i in r){print(boxplot.stats(df\_claim[,i])$out)}

## integer(0)  
## [1] 64 63 64 63  
## integer(0)  
## integer(0)  
## integer(0)  
## integer(0)

# visualizamos los datos extremos   
boxplot(df\_claim$age, main="Age",col="gray")



### 4. Análisis de los datos.

#### 4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Nos interesa seleccionar principalmente dos grupos de datos y un tercero que también puede aportar información que se transforme en conocimiento. Los grupos son cliente,incidente y vehiculos. Grupo de cliente:

* agrupación por sexo (2 niveles)

claim\_male <- df\_claim %>% filter(df\_claim$insured\_sex=="MALE")  
claim\_female <- df\_claim %>% filter(df\_claim$insured\_sex=="FEMALE")

\*agrupación por fraude (2 niveles)

claim\_fraudy <- df\_claim %>% filter(df\_claim$fraud\_reported=="Y")  
claim\_fraudn <- df\_claim %>% filter(df\_claim$fraud\_reported=="N")

Grupo de incidentes: \* agrupacion por tipo de incidente (4 niveles)

claim\_multi <- df\_claim %>% filter(df\_claim$incident\_type =="Multi-vehicle Collision")  
claim\_parked <- df\_claim %>% filter(df\_claim$incident\_type =="Parked Car")  
claim\_single <- df\_claim %>% filter(df\_claim$incident\_type =="Single Vehicle Collision")  
claim\_theft <- df\_claim %>% filter(df\_claim$incident\_type =="Vehicle Theft")

* agrupacion numero de vehiculos implicados (4 niveles)

claim\_one <- df\_claim %>% filter(df\_claim$number\_of\_vehicles\_involved=="1")  
claim\_two <- df\_claim %>% filter(df\_claim$number\_of\_vehicles\_involved=="2")  
claim\_three <- df\_claim %>% filter(df\_claim$number\_of\_vehicles\_involved=="3")  
claim\_four <- df\_claim %>% filter(df\_claim$number\_of\_vehicles\_involved=="4")

* agrupacion por tipo de colision (4 niveles)

claim\_Front <- df\_claim %>% filter(df\_claim$collision\_type =="Front Collision")  
claim\_No <- df\_claim %>% filter(df\_claim$collision\_type =="No collision")  
claim\_Rear <- df\_claim %>% filter(df\_claim$collision\_type =="Rear Collision")  
claim\_Side <- df\_claim %>% filter(df\_claim$collision\_type =="Side Collision")

#### 4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Para comprobar la normalidad instalamos el paquete Nortest que se componen de diferentes pruebas para comprobar la normalidad en los datos. En ete caso utilizaremos la prueba de Lilliefors (Kolmogorov-Smirnov) es una prueba general para la hipótesis compuesta de normalidad con grupo de datos mayor a 30. El estadístico de prueba es la diferencia absoluta máxima entre empírico e hipotético.

res <- sapply(df\_claim,class)   
res <- which(res=="integer")  
# aplicamos el test a las variables las 7 numericas  
  
for (i in res){  
 p <- lillie.test(df\_claim[,i])  
 p$data.name <- colnames(df\_claim)[i]  
 print(p)  
}

##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: months\_as\_customer  
## D = 0.050508, p-value = 2.689e-06  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: age  
## D = 0.065863, p-value = 5.174e-11  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: policy\_number  
## D = 0.056882, p-value = 4.396e-08  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: insured\_zip  
## D = 0.3228, p-value < 2.2e-16  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: incident\_hour\_of\_the\_day  
## D = 0.083552, p-value < 2.2e-16  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: auto\_year  
## D = 0.081182, p-value < 2.2e-16

Como podemos comprobar ninguna variable cumple normalidad.

El supuesto de homogeneidad de varianzas, también conocido como supuesto de homocedasticidad, considera que la varianza es constante en los diferentes niveles de un factor, es decir, entre diferentes grupos. Prodecemos a revisar la homogeniedad de la varianza usando el test de Levene. El test de Levene se puede aplicar con la función leveneTest() del paquete car. Se caracteriza, además de por poder comparar 2 o más poblaciones, por permitir elegir entre diferentes estadísticos de centralidad :mediana (por defecto), media, media truncada. Esto es importante a la hora de contrastar la homocedasticidad dependiendo de si los grupos se distribuyen de forma normal o no.

leveneTest(y = as.integer(df\_claim$fraud\_reported), group = df\_claim$insured\_sex, center = "median")

## Levene's Test for Homogeneity of Variance (center = "median")  
## Df F value Pr(>F)  
## group 1 0.9521 0.3294  
## 998

Puesto que obtenemos un p-valor superior a 0,05, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

### 4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

Para encontrar la correlacion entre todas las variables de datos seleccionados realizaremos una transformación a numericas ya que la correlación trabaja solamente con valores numericos. Utilizaremos el paquete “carret” para crear variables dummys y convertir las variables categoricas a numericas.

# se transforman las variables  
dmy <- dummyVars(" ~ .", data = df\_claim)  
claimstrns <- data.frame(predict(dmy, newdata = df\_claim))  
# usamos dos funciones para crear la matix de correlacion  
cor.prob <- function (X, dfr = nrow(X) - 2) {  
 R <- cor(X, use="pairwise.complete.obs")  
 above <- row(R) < col(R)  
 r2 <- R[above]^2  
 Fstat <- r2 \* dfr/(1 - r2)  
 R[above] <- 1 - pf(Fstat, 1, dfr)  
 R[row(R) == col(R)] <- NA  
 R  
}  
  
flattenSquareMatrix <- function(m) {  
 if( (class(m) != "matrix") | (nrow(m) != ncol(m))) stop("Must be a square matrix.")  
 if(!identical(rownames(m), colnames(m))) stop("Row and column names must be equal.")  
 ut <- upper.tri(m)  
 data.frame(i = rownames(m)[row(m)[ut]],  
 j = rownames(m)[col(m)[ut]],  
 cor=t(m)[ut],  
 p=m[ut])  
}  
# representamos las correlaciones  
corMasterList <- flattenSquareMatrix (cor.prob(claimstrns))  
print(head(corMasterList,10))

## i j cor p  
## 1 months\_as\_customer age 0.9220983226 0.00000000  
## 2 months\_as\_customer policy\_number 0.0575550330 0.06886845  
## 3 age policy\_number 0.0594134783 0.06036358  
## 4 months\_as\_customer policy\_bind\_date -0.0481727485 0.12792525  
## 5 age policy\_bind\_date -0.0369407214 0.24316777  
## 6 policy\_number policy\_bind\_date -0.0401042604 0.20510821  
## 7 months\_as\_customer insured\_zip 0.0178947610 0.57192446  
## 8 age insured\_zip 0.0256041758 0.41863305  
## 9 policy\_number insured\_zip 0.0070832059 0.82298040  
## 10 policy\_bind\_date insured\_zip -0.0002101377 0.99470462

corList <- corMasterList[order(corMasterList$cor),]  
kable(head(corList,10))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | i | j | cor | p |
| 21 | insured\_sex.FEMALE | insured\_sex.MALE | -1.0000000 | 0 |
| 1746 | incident\_type.Multi.vehicle.Collision | number\_of\_vehicles\_involved.1 | -1.0000000 | 0 |
| 388521 | fraud\_reported.N | fraud\_reported.Y | -1.0000000 | 0 |
| 1890 | number\_of\_vehicles\_involved.1 | number\_of\_vehicles\_involved.3 | -0.8793364 | 0 |
| 665 | incident\_type.Multi.vehicle.Collision | incident\_type.Single.Vehicle.Collision | -0.6977250 | 0 |
| 1867 | incident\_type.Single.Vehicle.Collision | number\_of\_vehicles\_involved.3 | -0.6135350 | 0 |
| 387682 | incident\_severity.L | fraud\_reported.Y | -0.4059877 | 0 |
| 386803 | incident\_severity.Q | fraud\_reported.N | -0.3975927 | 0 |
| 903 | incident\_severity.L | incident\_severity.Q | -0.3610341 | 0 |
| 1174 | incident\_state.NY | incident\_state.SC | -0.3421680 | 0 |

Representamos las variables más correlacionadas con denuncia fraudulenta observando que las variables más relevante son sexo, tipo de colision, tipo de incidente y numero de vehículos involucrados.

# seleccionamos las variables que nos interesan  
  
df\_reg <- df %>% select("fraud\_reported","insured\_sex","collision\_type","incident\_type","number\_of\_vehicles\_involved")   
dmy <- dummyVars(" ~ .", data = df\_reg)  
claimreg <- data.frame(predict(dmy, newdata = df\_reg))  
  
# Aplicamos el modelo de regresion   
model <- (lm(formula = fraud\_reportedY ~ insured\_sexMALE + collision\_type. + number\_of\_vehicles\_involved + incident\_typeMulti.vehicle.Collision + incident\_typeParked.Car, data = claimreg))  
summary (model)

##   
## Call:  
## lm(formula = fraud\_reportedY ~ insured\_sexMALE + collision\_type. +   
## number\_of\_vehicles\_involved + incident\_typeMulti.vehicle.Collision +   
## incident\_typeParked.Car, data = claimreg)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.31334 -0.28768 -0.25825 -0.06945 0.93055   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.25145 0.05931 4.239 2.45e-05  
## insured\_sexMALE 0.02943 0.02707 1.087 0.277  
## collision\_type. -0.20765 0.04884 -4.252 2.32e-05  
## number\_of\_vehicles\_involved 0.02566 0.05459 0.470 0.638  
## incident\_typeMulti.vehicle.Collision -0.07017 0.11325 -0.620 0.536  
## incident\_typeParked.Car 0.01317 0.06401 0.206 0.837  
##   
## (Intercept) \*\*\*  
## insured\_sexMALE   
## collision\_type. \*\*\*  
## number\_of\_vehicles\_involved   
## incident\_typeMulti.vehicle.Collision   
## incident\_typeParked.Car   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4259 on 994 degrees of freedom  
## Multiple R-squared: 0.03055, Adjusted R-squared: 0.02567   
## F-statistic: 6.264 on 5 and 994 DF, p-value: 9.875e-06

El resultado indica que el modelo de regresión múltiple explica tan solo el 3% de la variabilidad de clasificar un accidente como fraudulento. No parece ser un modelo acertado. Por ello debemos volver a evaluar si el conjunto de datos es el adecuado. Evaluando el modelo obtenido con la funcion “step”.

step(object = model, direction = "both", trace = 1)

## Start: AIC=-1701.08  
## fraud\_reportedY ~ insured\_sexMALE + collision\_type. + number\_of\_vehicles\_involved +   
## incident\_typeMulti.vehicle.Collision + incident\_typeParked.Car  
##   
## Df Sum of Sq RSS AIC  
## - incident\_typeParked.Car 1 0.0077 180.32 -1703.0  
## - number\_of\_vehicles\_involved 1 0.0401 180.35 -1702.9  
## - incident\_typeMulti.vehicle.Collision 1 0.0696 180.38 -1702.7  
## - insured\_sexMALE 1 0.2144 180.52 -1701.9  
## <none> 180.31 -1701.1  
## - collision\_type. 1 3.2797 183.59 -1685.0  
##   
## Step: AIC=-1703.04  
## fraud\_reportedY ~ insured\_sexMALE + collision\_type. + number\_of\_vehicles\_involved +   
## incident\_typeMulti.vehicle.Collision  
##   
## Df Sum of Sq RSS AIC  
## - number\_of\_vehicles\_involved 1 0.0401 180.36 -1704.8  
## - incident\_typeMulti.vehicle.Collision 1 0.0697 180.39 -1704.7  
## - insured\_sexMALE 1 0.2112 180.53 -1703.9  
## <none> 180.32 -1703.0  
## + incident\_typeParked.Car 1 0.0077 180.31 -1701.1  
## - collision\_type. 1 5.0065 185.32 -1677.7  
##   
## Step: AIC=-1704.82  
## fraud\_reportedY ~ insured\_sexMALE + collision\_type. + incident\_typeMulti.vehicle.Collision  
##   
## Df Sum of Sq RSS AIC  
## - incident\_typeMulti.vehicle.Collision 1 0.0726 180.43 -1706.4  
## - insured\_sexMALE 1 0.2200 180.58 -1705.6  
## <none> 180.36 -1704.8  
## + number\_of\_vehicles\_involved 1 0.0401 180.32 -1703.0  
## + incident\_typeParked.Car 1 0.0078 180.35 -1702.9  
## - collision\_type. 1 5.0075 185.37 -1679.4  
##   
## Step: AIC=-1706.41  
## fraud\_reportedY ~ insured\_sexMALE + collision\_type.  
##   
## Df Sum of Sq RSS AIC  
## - insured\_sexMALE 1 0.2158 180.65 -1707.2  
## <none> 180.43 -1706.4  
## + incident\_typeMulti.vehicle.Collision 1 0.0726 180.36 -1704.8  
## + number\_of\_vehicles\_involved 1 0.0430 180.39 -1704.7  
## + incident\_typeParked.Car 1 0.0077 180.42 -1704.5  
## - collision\_type. 1 5.3838 185.81 -1679.0  
##   
## Step: AIC=-1707.22  
## fraud\_reportedY ~ collision\_type.  
##   
## Df Sum of Sq RSS AIC  
## <none> 180.65 -1707.2  
## + insured\_sexMALE 1 0.2158 180.43 -1706.4  
## + incident\_typeMulti.vehicle.Collision 1 0.0684 180.58 -1705.6  
## + number\_of\_vehicles\_involved 1 0.0377 180.61 -1705.4  
## + incident\_typeParked.Car 1 0.0046 180.64 -1705.2  
## - collision\_type. 1 5.3453 185.99 -1680.1

##   
## Call:  
## lm(formula = fraud\_reportedY ~ collision\_type., data = claimreg)  
##   
## Coefficients:  
## (Intercept) collision\_type.   
## 0.2810 -0.1911

Para el contraste de hipotesis se plantea si un accidente es fraulento o no dependiendo del sexo. Así, se plantea el siguiente contraste de hipótesis de dos muestras sobre la diferencia de medias, el cual es unilateral atendiendo a la formulación de la hipótesis alternativa: *H0 : µ1−µ2 =0* H1 : µ1−µ2 < 0 donde µ1 es la media de la población de la que se extrae la primera muestra y µ2 es la media de la población de la que extrae la segunda.Se toma α =0,05.

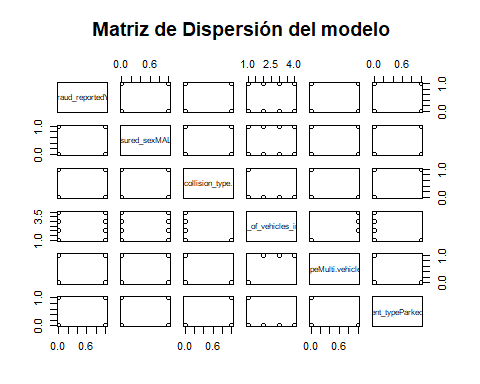
df\_claim$fraud\_reported <- if\_else(df\_claim$fraud\_reported=="Y", 1, 0)  
t.test(df\_claim$fraud\_reported, mu=TRUE, alternative = "less")

##   
## One Sample t-test  
##   
## data: df\_claim$fraud\_reported  
## t = -55.186, df = 999, p-value < 2.2e-16  
## alternative hypothesis: true mean is less than TRUE  
## 95 percent confidence interval:  
## -Inf 0.2694643  
## sample estimates:  
## mean of x   
## 0.247

Puesto que obtenemos un p-valor menor que el valor de signiﬁcación ﬁjado, rechazamos la hipótesis nula. Por lo cual, la declaración de un un accidente como fraude no depende del sexo.

### 5. Representación de los resultados a partir de tablas y gráficas.

# representacion grafica  
pairs(fraud\_reportedY ~ insured\_sexMALE + collision\_type. + number\_of\_vehicles\_involved + incident\_typeMulti.vehicle.Collision + incident\_typeParked.Car, data = claimreg, main=" Matriz de Dispersión del modelo")

 ###6.Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Las conclusiones obtenidas nos indican que debemos revisar desde las variables seleccionadas a los modelos aplicados. Los resultados no han sido nada concluyentes y no se ha podido responder con claridad a las cuestiones planteadas.