Universitat Oberta de Catalunya

Patricia Lázaro Tello
plazarotello@uoc.edu

A1 - Preprocesado de datos

Patricia Lázaro Tello

Índice general

1 Carga del archivo **2**

2 Duplicación de códigos **3**

3 Nombres de las variables **4**

4 Normalización de los datos cualitativos **5**
4.1 *Marital Status* 5
4.2 Género 6

5 Normalización de los datos cuantitativos **7**
5.1 `IniCost` y `UltCost` 7
5.2 Edad 9
5.3 *WeeklyWages*, *HoursWeek*, *DaysWeek* 9

6 Valores atípicos **10**
6.1 *Age* 11
6.2 *WeeklyWages* 13
6.3 *HoursWeek* 15

7 Imputación de valores **19**
7.1 *Age* 19
7.2 *WeeklyWages*, *HoursWeek*, `IniCost`, `UltCost` 19

8 Preparación de los datos **21**
8.1 Tiempo de abertura del expediente 21
8.2 Diferencia entre `IniCost` y `UltCost` 23

9 Estudio descriptivo **24**
9.1 Funciones de media robustas 24
9.2 Estudio descriptivo de las variables cuantitativas 27

10 Archivo final **34**

Patricia Lázaro Tello
plazarotello@uoc.edu

```
## Instalacion de paquetes y dependencias

if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('VIM')) install.packages('VIM'); library('VIM')
if (!require('psych')) install.packages('psych'); library('psych')
```

1 Carga del archivo

Se procede a la carga del archivo `train3.csv` y a la visualización de sus datos.

```
insurances <- read.csv2(file = "train3.csv", header = TRUE)
head(insurances, n = 3L)
```

```
##   ClaimNumber   DateTimeOfAccident         DateReported Age Gender
## 1   WC8285054 2002-04-09T07:00:00Z 2002-07-05T00:00:00Z  48      M
## 2   WC6982224 1999-01-07T11:00:00Z 1999-01-20T00:00:00Z  43      F
## 3   WC5481426 1996-03-25T00:00:00Z 1996-04-14T00:00:00Z  30      M
##   MaritalStatus DependentChildren DependentsOther WeeklyWages PartTimeFullTime
## 1             M                 0               0      500.00                F
## 2             m                 0               0      509.34                F
## 3             U                 0               0      709.10                F
##   HoursWorkedPerWeek DaysWorkedPerWeek
## 1               38.0                5
## 2               37.5                5
## 3               38.0                5
##                                       ClaimDescription
## 1          LIFTING TYRE INJURY TO RIGHT ARM AND WRIST INJURY
## 2 STEPPED AROUND CRATES AND TRUCK TRAY FRACTURE LEFT FOREARM
## 3                          CUT ON SHARP EDGE CUT LEFT THUMB
##   InitialIncurredCalimsCost UltimateIncurredClaimCost
## 1                      1500               4303.188001
## 2                      5500               6105.872938
## 3                      1700               2098.629955
```

```
summary(insurances)
```

```
##  ClaimNumber        DateTimeOfAccident DateReported            Age
##  Length:54000       Length:54000       Length:54000       Min.   : 13.00
##  Class :character   Class :character   Class :character   1st Qu.: 23.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 32.00
##                                                           Mean   : 34.06
##                                                           3rd Qu.: 43.00
##                                                           Max.   :999.00
```

```
##     Gender          MaritalStatus      DependentChildren DependentsOther
##  Length:54000       Length:54000       Min.   :0.0000    Min.   :0.000000
##  Class :character   Class :character   1st Qu.:0.0000    1st Qu.:0.000000
##  Mode  :character   Mode  :character   Median :0.0000    Median :0.000000
##                                        Mean   :0.1192    Mean   :0.009944
##                                        3rd Qu.:0.0000    3rd Qu.:0.000000
##                                        Max.   :9.0000    Max.   :5.000000
##   WeeklyWages     PartTimeFullTime   HoursWorkedPerWeek DaysWorkedPerWeek
##  Min.   :   1.0   Length:54000       Min.   :  0.00     Min.   :1.000
##  1st Qu.: 200.0   Class :character   1st Qu.: 38.00     1st Qu.:5.000
##  Median : 392.2   Mode  :character   Median : 38.00     Median :5.000
##  Mean   : 416.4                      Mean   : 37.74     Mean   :4.906
##  3rd Qu.: 500.0                      3rd Qu.: 40.00     3rd Qu.:5.000
##  Max.   :7497.0                      Max.   :640.00     Max.   :7.000
##  ClaimDescription   InitialIncurredCalimsCost UltimateIncurredClaimCost
##  Length:54000       Min.   :      1           Length:54000
##  Class :character   1st Qu.:    700           Class :character
##  Mode  :character   Median :   2000           Mode  :character
##                     Mean   :   7841
##                     3rd Qu.:   9500
##                     Max.   :2000000
```

El fichero contiene 54.000 registros con 15 atributos.

---

2 Duplicación de códigos

Primero se listarán los *ClaimNumber* duplicados y el número de registro al que pertenecen:

```
cn_dups <- duplicated(insurances$ClaimNumber)
insurances$ClaimNumber[cn_dups]
```

```
## [1] "WC8668542" "WC8668542" "WC3501716" "WC6383678"
```

```
which(cn_dups)
```

```
## [1]   111 40049 50091 50279
```

A continuación, se obtendrá el mayor valor de los códigos de *ClaimNumber* y se sustituirán los códigos duplicados por valores superiores al máximo código de la variable, para eliminar los códigos duplicados y evitar nuevos duplicados. Finalmente se comprueba que se han aplicado correctamente los nuevos códigos de *ClaimNumber*:

```
max_cn <- max(strtoi(sub("WC", "", insurances$ClaimNumber)))
insurances$ClaimNumber[cn_dups] <- paste(
  "WC",
  seq(from= max_cn + 1, length=sum(cn_dups)),
  sep="")
insurances$ClaimNumber[which(cn_dups)]
```

```
## [1] "WC9999962" "WC9999963" "WC9999964" "WC9999965"
```

---

3 Nombres de las variables

Primero se observan los nombres actuales de las columnas. A partir de esta información se van reemplazando los nombres conflictivos.

```
colnames(insurances)
```

```
##  [1] "ClaimNumber"              "DateTimeOfAccident"
##  [3] "DateReported"             "Age"
##  [5] "Gender"                   "MaritalStatus"
##  [7] "DependentChildren"        "DependentsOther"
##  [9] "WeeklyWages"              "PartTimeFullTime"
## [11] "HoursWorkedPerWeek"       "DaysWorkedPerWeek"
## [13] "ClaimDescription"         "InitialIncurredCalimsCost"
## [15] "UltimateIncurredClaimCost"
```

```
# InitialIncurredClaimCost -> IniCost
colnames(insurances)[14] <- "IniCost"

# UltimateIncurredClaimCost -> UltCost
colnames(insurances)[15] <- "UltCost"

# HourseWorkedPerWeek -> HoursWeek
colnames(insurances)[11] <- "HoursWeek"

# DaysWorkedPerWeek -> DaysWeek
colnames(insurances)[12] <- "DaysWeek"

colnames(insurances)
```

```
##  [1] "ClaimNumber"         "DateTimeOfAccident" "DateReported"
##  [4] "Age"                 "Gender"             "MaritalStatus"
##  [7] "DependentChildren"   "DependentsOther"    "WeeklyWages"
## [10] "PartTimeFullTime"    "HoursWeek"          "DaysWeek"
## [13] "ClaimDescription"    "IniCost"            "UltCost"
```

---

4 Normalización de los datos cualitativos

4.1 *Marital Status*

Para normalizar los datos cualitativos, el primer paso es averiguar cuántos valores tiene el atributo sin normalizar. Con estos valores se podrá decidir qué modificaciones llevar a cabo más adelante.

```
unique(insurances$MaritalStatus)
```

```
##  [1] "M"       "m"       "U"       "S"       "married" "W"       ""
##  [8] "d"       "D"       "w"
```

```
anyNA(insurances$MaritalStatus)
```

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

```
table(insurances$MaritalStatus)
```

```
##
##                 d       D       m       M married       S       U       w       W
##      29        13      63     242   21862     316   26161    5294       8      12
```

Los valores válidos en *marital status* son: M, S, U, D, W. Se puede observar que los valores aparecen también en minúscula — se convertirán los valores a mayúsculas, — así como valores vacíos que se convertirán a *status* U y valores "*married*" que pasarán a ser M. No existen valores nulos a cambiar.

```
insurances$MaritalStatus[insurances$MaritalStatus == "married"] <- "M"
insurances$MaritalStatus[insurances$MaritalStatus == ""] <- "U"
insurances$MaritalStatus <- as.factor(toupper(insurances$MaritalStatus))

unique(insurances$MaritalStatus)
```

```
## [1] M U S W D
```

```
## Levels: D M S U W
```

```
table(insurances$MaritalStatus)
```

```
##
##     D     M     S     U     W
##    76 22420 26161  5323    20
```

Después de las transformaciones se puede apreciar que *marital status* está normalizado y sus valores posibles son los expuestos en el enunciado: D, U, S, W, D.

4.2 Género

Como se indica en el enunciado, los posibles valores del atributo son `F` (femenino), `M` (masculino) y `U` (*unknown*).

Se procede en primer lugar a explorar los posibles valores que toma actualmente la variable y si hay valores nulos (NA).

```
unique(insurances$Gender)
```

```
## [1] "M"  "F"  "Fm" "f"  "U"
```

```
anyNA(insurances$Gender)
```

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

```
table(insurances$Gender)
```

```
##
##     f     F    Fm     M     U
##   534 11507   297 41660     2
```

No se aprecian valores nulos (NA). Como sucedía con *marital status*, hay valores en minúsculas que habrá que convertir a mayúsculas. Se considera el valor “Fm” como una errata y se convierte a U (unknown).

```
insurances$Gender[insurances$Gender == "Fm"] <- "U"
insurances$Gender <- as.factor(toupper(insurances$Gender))

table(insurances$Gender)
```

```
##
##     F     M     U
## 12041 41660   299
```

Después de las transformaciones se puede apreciar que *gender* está normalizado y sus valores posibles son los expuestos en el enunciado: F, M, U.

---

5 Normalización de los datos cuantitativos

5.1 `IniCost` y `UltCost`

Los criterios de normalización para `IniCost` y `UltCost` son los siguientes: deben ser variables númericas de tipo entero, en unidades (no miles) y sin decimales.

5.1.1 `IniCost`

Se inicia el análisis de `IniCost`. Se comprobará el tipo de variable inicialmente y si hay valores nulos (NA); y se mostrarán los primeros valores como análisis exploratorio inicial.

```
class(insurances$IniCost)
```

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

```
anyNA(insurances$IniCost)
```

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

```
head(insurances$IniCost)
```

```
## [1]  1500  5500  1700 15000  2800   500
```

De este análisis se puede extraer que el atributo `IniCost` ya se encuentra normalizado, sin valores nulos (NA) y cumple los criterios de tipo de dato (entero), unidades y sin decimales.

5.1.2 `UltCost`

A continuación se realizarán las mismas comprobaciones hechas sobre `IniCost` en el atributo `UltCost`.

```
class(insurances$UltCost)
```

```
## [1] "character"
```

```
anyNA(insurances$UltCost)
```

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

```
head(insurances$UltCost)
```

```
## [1] "4303.188001" "6105.872938" "2098.629955" "16282.94081" "3771.73258"
## [6] "746.6213097"
```

`UltCost` no se encuentra normalizado: no tiene valores nulos (NA), pero es de tipo string (character) y tiene decimales. Al tener decimales, debería ser de tipo double por defecto; al no serlo se puede inferir que no todos los valores son números con decimales. Se procede a comprobar si los datos tienen estructura de doubles.

```
uc_nd <- !grepl("^\\d+(\\.\\d+)*$", insurances$UltCost)
head(insurances$UltCost[uc_nd])
```

```
## [1] "1.277718363K"  "0.1077643807K" "1.445204861K"  "8.60810336K"
## [5] "2.597486704K"  "2.005016752K"
```

```
length(which(uc_nd))
```

```
## [1] 324
```

Hay 324 valores en `UltCost` que no cumple el formato integer/double. Estos valores distintos tienen en común la 'K' del final, que indica que están en miles. Por tanto, se habrá de eliminar la K y se multiplicarán los valores por 1.000.

Antes, sin embargo, habrá que convertir todos los valores a tipo double.

```
insurances$UltCost[uc_nd] <- gsub("K|k", "", insurances$UltCost[uc_nd])
insurances$UltCost <- as.double(insurances$UltCost)
insurances$UltCost[uc_nd] <- insurances$UltCost[uc_nd] * rep(
  c(1000), length(which(uc_nd)))
head(insurances$UltCost[uc_nd])
```

```
## [1] 1277.7184  107.7644 1445.2049 8608.1034 2597.4867 2005.0168
```

Por último se convertirán todos los datos a enteros. Así la variable `UltCost` cumplirá los criterios de normalización propuestos: será de tipo entero (integer), expresada en unidades y sin decimales.

```
insurances$UltCost <- as.integer(insurances$UltCost)
head(insurances$UltCost)
```

```
## [1]  4303  6105  2098 16282  3771   746
```

5.2 Edad

La variable *Age* debe ser un entero sin decimales para estar normalizada. A continuación se inspecciona el tipo de la variable y sus valores.

```
class(insurances$Age)
```

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

```
head(insurances$Age)
```

```
## [1] 48 43 30 41 36 50
```

Se comprueba que la variable *Age* ya se encuentra normalizada; es de tipo entero sin decimales.

5.3 *WeeklyWages*, *HoursWeek*, *DaysWeek*

5.3.1 *WeeklyWages*

WeeklyWages ha de ser una variable numérica y con decimales. Se procede a comprobar su tipo actual, si contiene algún valor perdido (NA) y una muestra de los datos.

```
class(insurances$WeeklyWages)
```

```
## [1] "numeric"
```

```
anyNA(insurances$WeeklyWages)
```

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

```
head(insurances$WeeklyWages)
```

```
## [1] 500.00 509.34 709.10 555.46 377.10 200.00
```

No contiene valores nulos (NA) y ya se trata de una variable numérica con decimales; por tanto se confirma que *WeeklyWages* ya se encuentra normalizada.

5.3.2 *HoursWeek*

HoursWeek, como *WeeklyWages*, tiene que ser una variable numérica con decimales. Se procede a realizar las mismas comprobaciones que en el atributo anterior.

```
class(insurances$HoursWeek)
```

```
## [1] "numeric"
```

```
anyNA(insurances$HoursWeek)
```

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

```
head(insurances$HoursWeek)
```

```
## [1] 38.0 37.5 38.0 38.0 38.0 38.0
```

HoursWeek también se encuentra normalizada y no tiene valores nulos (NA).

5.3.3 *DaysWeek*

DaysWeek ha de ser una variable numérica discreta, sin decimales, de tipo entero. Se procede a las comprobaciones pertinentes:

```
class(insurances$DaysWeek)
```

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

```
anyNA(insurances$DaysWeek)
```

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

```
head(insurances$DaysWeek)
```

```
## [1] 5 5 5 5 5 5
```

Se confirma que *DaysWeek* ya está normalizada según los criterios expuestos en el enunciado.

---

6 Valores atípicos

Los valores atípicos o *outliers* se pueden clasificar en dos tipos: **valores centinelas** — aquellos que, siendo atípicos, tienen el significado semántico de un valor perdido, — y **valores propiamente atípicos** — aquellos que son insconsistentes con el conjunto de datos.

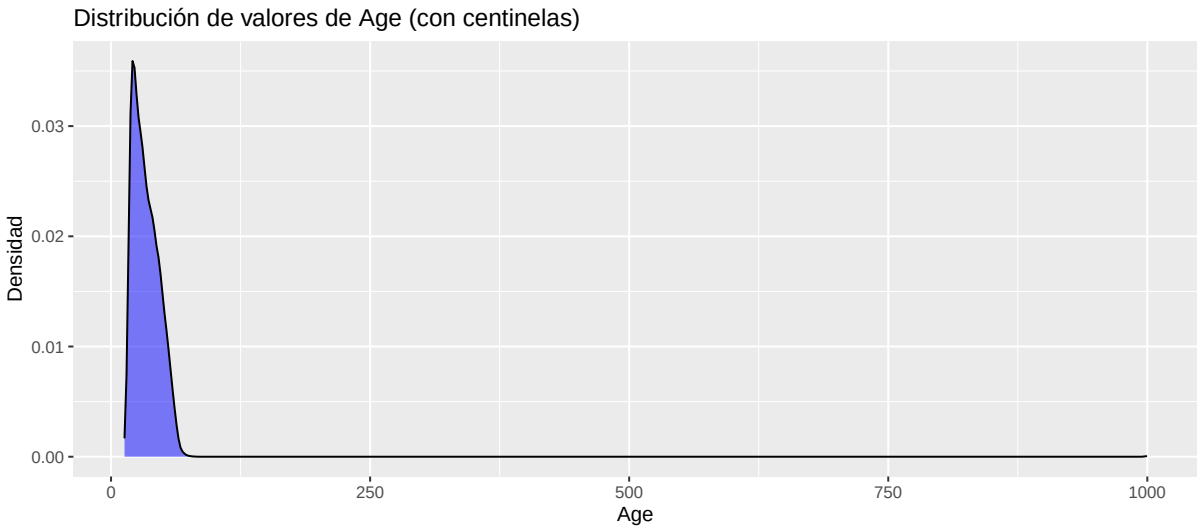
A continuación se comprobarán los valores centinelas primero y los *outliers* después

para cada variable (*Age*, *WeeklyWages*, *HoursWeek*, *DaysWeek*).

6.1 *Age*

A continuación se muestra la distribución de valores de la variable y un resumen de sus características.

```
ggplot(mapping= aes(x=insurances$Age)) +
  geom_density(alpha=0.5, fill="blue") +
  labs(title = "Distribución de valores de Age (con centinelas)",
       x = "Age", y = "Densidad")
```




```
summary(insurances$Age)
```

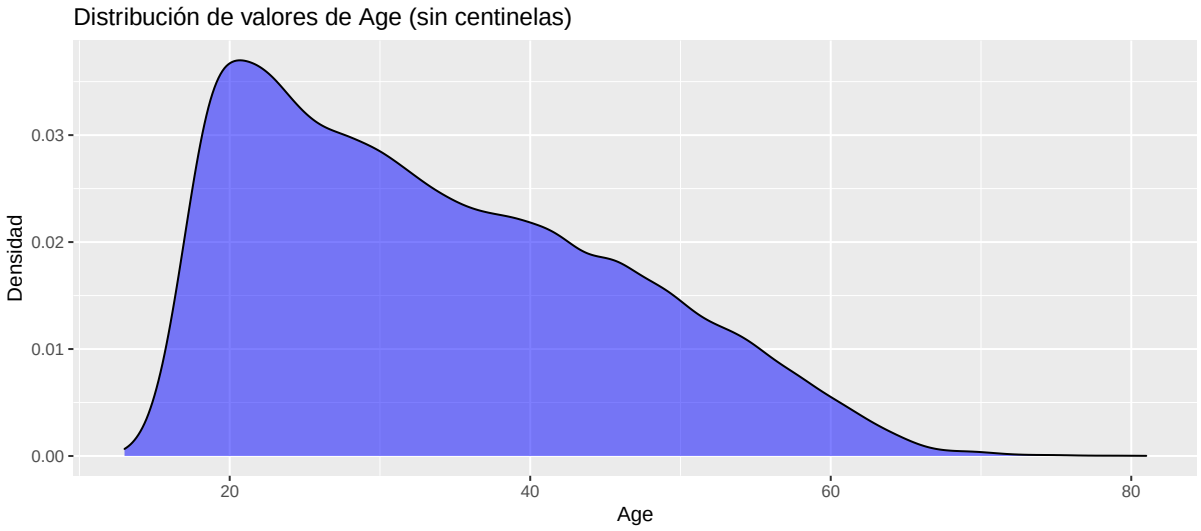
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   23.00   32.00   34.06   43.00  999.00
```

Aunque en el gráfico es difícil de percibir, se puede observar un pequeño pico en el valor 1.000. Con el valor máximo mostrado por las características (999) ese pico queda confirmado. Además, observando los percentiles y el gráfico se observa que los valores se concentran en torno a valores inferiores a 100 y solo hay unos pocos valores atípicos de 999.

Como no es posible vivir 999 años y encaja con la descripción de valor centinela vista anteriormente, se procede a sustituir estos valores por NA y rehacer el gráfico y resumen de características.

Patricia Lázaro Tello
plazarotello@uoc.edu

```
insurances$Age[insurances$Age == 999] <- NA

ggplot(mapping= aes(x=insurances$Age)) +
  geom_density(alpha=0.5, fill="blue", na.rm = TRUE) +
  labs(title = "Distribución de valores de Age (sin centinelas)",
    x = "Age", y = "Densidad")
```

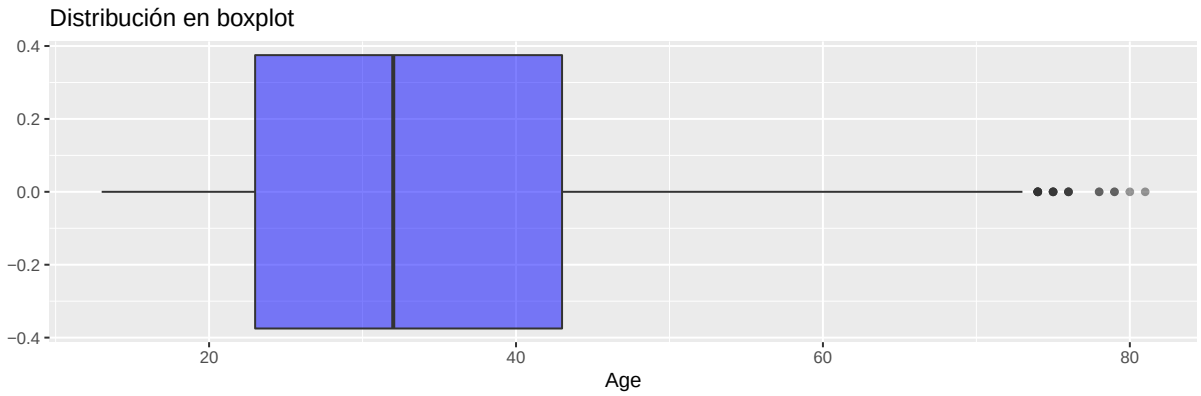



```
summary(insurances$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   13.00   23.00   32.00   33.84   43.00   81.00      12
```

Ahora la distribución de valores recorre el rango [13, 81], edades apropiadas para seres humanos. Se han eliminado (convertido a NAs) 12 valores. A continuación se muestra un *boxplot* de los datos para buscar valores atípicos.

```
ggplot(data = insurances, mapping = aes(x=Age, fill=Age)) +
  geom_boxplot(alpha=0.5, fill="blue", na.rm = TRUE) +
  labs(title = "Distribución en boxplot", x = "Age")
```




```
boxplot.stats(insurances$Age)$out
```

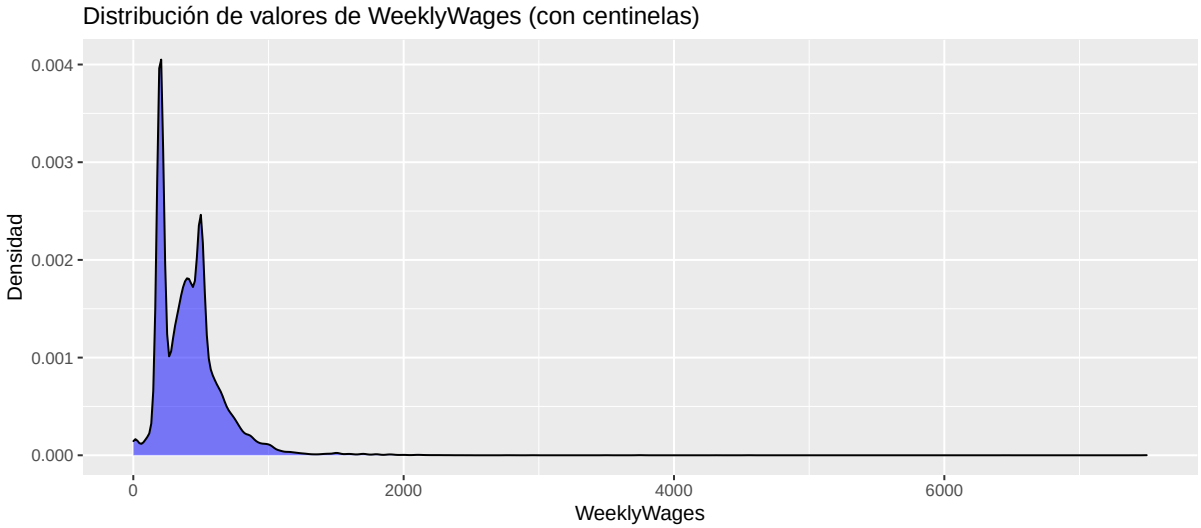
```
##  [1] 74 74 74 75 75 79 74 75 74 74 76 75 76 76 75 74 80 76 79 78 78 81
```

Los posibles valores atípicos mostrados por el *boxplot* son de personas de 74 años o más. Aún siendo atípicos, se consideran valores reales poco frecuentes y se decide no eliminarlos.

6.2 *WeeklyWages*

A continuación se muestra la distribución de valores de la variable y un resumen de sus características, así como los valores mínimos y máximos.

```
ggplot(mapping= aes(x=insurances$WeeklyWages)) +
  geom_density(alpha=0.5, fill="blue") +
  labs(title = "Distribución de valores de WeeklyWages (con centinelas)",
    x = "WeeklyWages", y = "Densidad")
```




```
summary(insurances$WeeklyWages)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   200.0   392.2   416.4   500.0  7497.0
```

```
tail(sort(unique(insurances$WeeklyWages)), 10)
```

```
##  [1] 2766.04 2817.92 2956.52 3500.00 3750.00 4311.30 4556.00 6453.00 7400.00
## [10] 7497.00
```

```
head(sort(unique(insurances$WeeklyWages)), 10)
```

```
##  [1] 1.00 1.91 3.59 3.95 4.61 4.73 5.00 5.25 5.49 5.78
```

Mirando la distribución de valores de la variable, se aprecian valores atípicos en su cola; sin embargo no parecen ser valores centinelas, dada la naturaleza de los valores (7497.00).

Los valores mínimos también son interesantes: existen personas que cobran 1ud. a la semana. Este valor sí que puede coincidir con un valor centinela, y por tanto se examinará en mayor profundidad.

```
wages.min <- head(sort(unique(insurances$WeeklyWages)), 10)
wages.freq <- table(insurances$WeeklyWages)
wages.freq[which(names(wages.freq) %in% wages.min)]
```

```
##
##    1 1.91 3.59 3.95 4.61 4.73    5 5.25 5.49 5.78
##  122    1    1    2    1    2   16    2    2    1
```

Analizando el número de ocurrencias de cada uno de los valores mínimos de *Weekly-Wages* se observa que el valor 1 concentra una cantidad anormal de ocurrencias; sin embargo, se decide no considerarlo valor centinela porque un valor así podría darse.

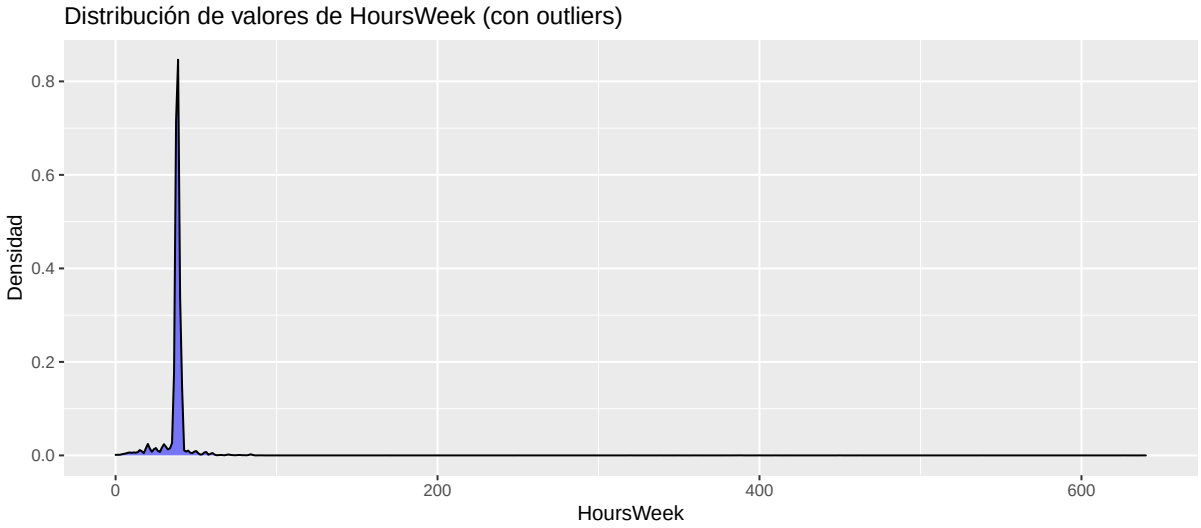
Respecto a los valores propiamente atípicos, los valores mínimos (1.0, 1.91, 3.59…) y los valores máximos (7497, 7400…) se consideran, aunque atípicos, posibles valores reales; sin más información del origen de los datos, se decide ser conservador y no eliminarlos.

6.3 *HoursWeek*

> *Una semana se compone de 7 días, y cada día tiene 24 horas. Por tanto, si una persona trabajara 24h/7 días, sus horas semanales ascenderían a **168 horas semanales**. Se considera este valor el máximo posible para el atributo.*

A continuación se muestra la distribución de valores de la variable y un resumen de sus características.

```
ggplot(mapping= aes(x=insurances$HoursWeek)) +
  geom_density(alpha=0.5, fill="blue") +
  labs(title = "Distribución de valores de HoursWeek (con outliers)",
       x = "HoursWeek", y = "Densidad")
```



```
summary(insurances$HoursWeek)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   38.00   38.00   37.74   40.00  640.00
```

Del resumen de las características de la variable se observa que hay trabajadores laborando 640 horas semanales, un dato imposible. Se procede a obtener los valores mínimos y máximos del atributo en busca de valores centinela.

```
hours.min <- head(sort(unique(insurances$HoursWeek)), 10)
hours.max <- tail(sort(unique(insurances$HoursWeek)), 10)

hours.freq <- table(insurances$HoursWeek)

hours.freq[which(names(hours.freq) %in% hours.min)]
```

```
##
##   0   1   2 2.1   3 3.5   4 4.1 4.5   5
##  29  31   6   1  26   5  34   1   4  50
```

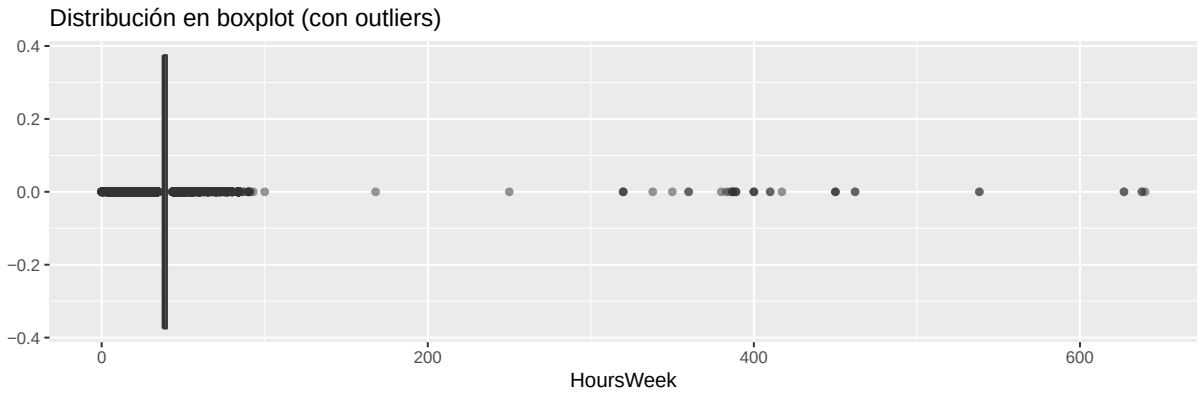
```
hours.freq[which(names(hours.freq) %in% hours.max)]
```

```
##
##   389   400   410 417.2   450 462.08 538.3   627   638   640
##     3     3     2     1     3     2     2     2     2     1
```

Las características de los valores mínimos y máximos no coinciden con las de un valor centinela. Se puede concluir que no existen valores centinelas en el atributo.

Respecto a los valores propiamente atípicos, se han tomado como atípicos los valores que son iguales o exceden de 168h semanales por las razones expuestas al inicio de esta sección.

```
ggplot(data = insurances, mapping = aes(x=HoursWeek, fill=Age)) +
  geom_boxplot(alpha=0.5, fill="blue", na.rm = TRUE) +
  labs(title = "Distribución en boxplot (con outliers)",
    x = "HoursWeek")
```



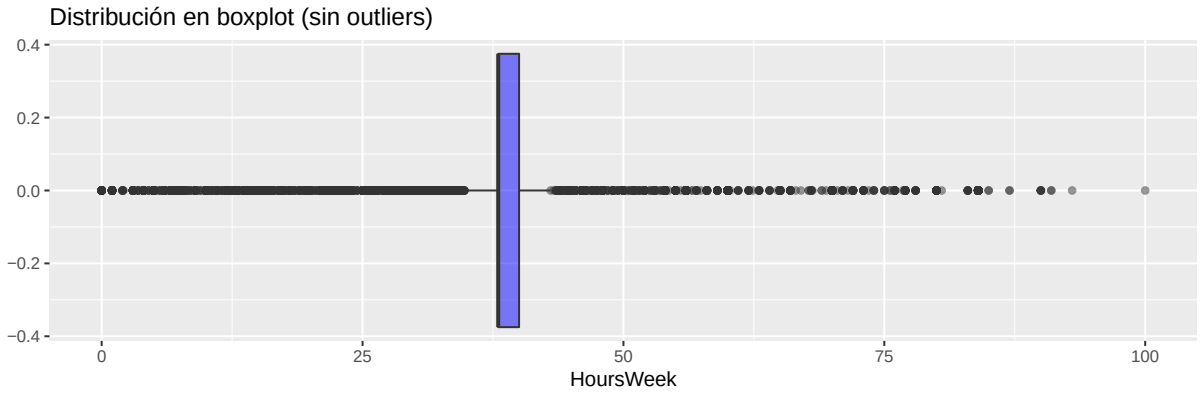

```
head(sort(unique(boxplot.stats(insurances$HoursWeek)$out)), 10)
```

```
##  [1] 0.0 1.0 2.0 2.1 3.0 3.5 4.0 4.1 4.5 5.0
```

```
tail(sort(unique(boxplot.stats(insurances$HoursWeek)$out)), 10)
```

```
##  [1] 389.00 400.00 410.00 417.20 450.00 462.08 538.30 627.00 638.00 640.00
```

```
insurances$HoursWeek[insurances$HoursWeek >= 168] <- NA

ggplot(data = insurances, mapping = aes(x=HoursWeek, fill=Age)) +
  geom_boxplot(alpha=0.5, fill="blue", na.rm = TRUE) +
  labs(title = "Distribución en boxplot (sin outliers)",
       x = "HoursWeek")
```



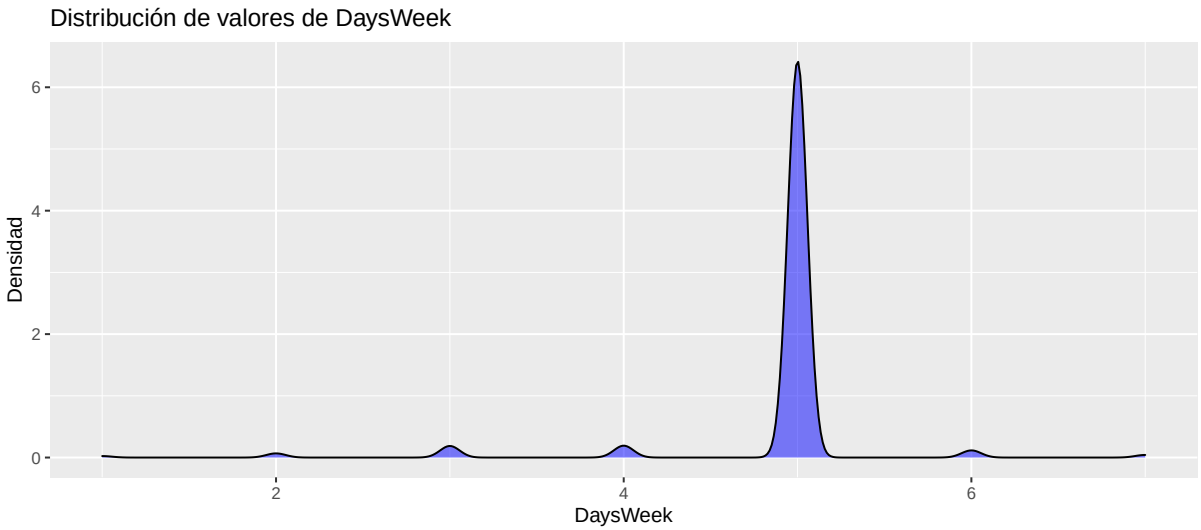
Se confirma lo ya visto en la distribución de valores de la variable: la mayoría de trabajadores laboran en torno a 38 horas semanales; esta gran concentración de registros en unos pocos valores hace que el *boxplot* trate a la mayoría de los datos que no se encuentran en torno a ese valor como atípicos.

Tras convertir a NA los valores atípicos, el gráfico de la distribución de los valores sigue marcando muchos valores como atípicos; sin embargo, se consideran valores reales pero poco frecuentes.

##DaysWeek

A continuación se muestra la distribución de valores de la variable y un resumen de sus características.

```
ggplot(mapping= aes(x=insurances$DaysWeek)) +
  geom_density(alpha=0.5, fill="blue") +
  labs(title = "Distribución de valores de DaysWeek",
       x = "DaysWeek", y = "Densidad")
```




```
summary(insurances$DaysWeek)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   5.000   4.906   5.000   7.000
```

Se sabe que la semana tiene 7 días; del resumen de características se extrae que el rango de la variable es [1, 7]. Se concluye que no existen valores propiamente atípicos ni valores centinelas.

7 Imputación de valores

A continuación se investiga la existencia de valores perdidos en las variables *Age*, *WeeklyWages*, *HoursWeek*, `IniCost` y `UltCost` y, si es necesario, se imputan según los criterios establecidos en el enunciado.

7.1 *Age*

```
age.nas <- is.na(insurances$Age)
length(which(age.nas))
```

```
## [1] 12
```

En el atributo *Age* hay 12 valores perdidos. Se procede a imputarlos por la media aritmética y mostrar el resultado de los datos afectados por la imputación.

```
insurances$Age[age.nas] <- as.integer(mean(insurances$Age, na.rm = TRUE))
as.integer(mean(insurances$Age, na.rm = TRUE))
```

```
## [1] 33
```

```
head(insurances$Age[age.nas])
```

```
## [1] 33 33 33 33 33 33
```

7.2 *WeeklyWages*, *HoursWeek*, `IniCost`, `UltCost`

```
wages.nas <- is.na(insurances$WeeklyWages)
length(which(wages.nas))
```

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

```
hours.nas <- is.na(insurances$HoursWeek)
length(which(hours.nas))
```

```
## [1] 37
```

```
inicost.nas <- is.na(insurances$IniCost)
length(which(inicost.nas))
```

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

```
ultcost.nas <- is.na(insurances$UltCost)
length(which(ultcost.nas))
```

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

Hay valores perdidos en *HoursWeek*. Se procede a imputar los valores perdidos utilizando `kNN()` con distinción por género, considerando las 4 variables para el cómputo de los vecinos más cercanos.

```
all.nas <- wages.nas + hours.nas + inicost.nas + ultcost.nas > 0

insurances[which(all.nas)[1:5], c("Gender", "WeeklyWages", "HoursWeek",
                                  "IniCost", "UltCost")]
```

```
##      Gender WeeklyWages HoursWeek IniCost UltCost
## 3692      M      338.00        NA     500     975
## 4672      M      584.78        NA    3500   33925
## 5233      F      200.00        NA     600    1866
## 5489      M      200.00        NA   10000    5055
## 6132      F      456.00        NA     800    1876
```

```
insurances[insurances$Gender == "M",] <- kNN(
  data = insurances[insurances$Gender == "M",],
  variable = c("HoursWeek"),
  dist_var = c("WeeklyWages", "HoursWeek", "IniCost", "UltCost"),
  impNA = TRUE, imp_var = FALSE)

insurances[insurances$Gender == "F",] <- kNN(
  data = insurances[insurances$Gender == "F",],
  variable = c("HoursWeek"),
  dist_var = c("WeeklyWages", "HoursWeek", "IniCost", "UltCost"),
  impNA = TRUE, imp_var = FALSE)

insurances[insurances$Gender == "U",] <- kNN(
  data = insurances[insurances$Gender == "U",],
  variable = c("HoursWeek"),
  dist_var = c("WeeklyWages", "HoursWeek", "IniCost", "UltCost"),
  impNA = TRUE, imp_var = FALSE)

insurances[which(all.nas)[1:5], c("Gender", "WeeklyWages", "HoursWeek",
                                  "IniCost", "UltCost")]
```

```
##      Gender WeeklyWages HoursWeek IniCost UltCost
## 3692      M      338.00        38     500     975
## 4672      M      584.78        40    3500   33925
## 5233      F      200.00        38     600    1866
## 5489      M      200.00        38   10000    5055
## 6132      F      456.00        40     800    1876
```

```
length(which(is.na(insurances$WeeklyWages)))
```

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

```
length(which(is.na(insurances$HoursWeek)))
```

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

```
length(which(is.na(insurances$IniCost)))
```

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

```
length(which(is.na(insurances$UltCost)))
```

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

---

8 Preparación de los datos

8.1 Tiempo de abertura del expediente

Para calcular el tiempo de abertura del expediente el primer paso a realizar es cambiar *DateOfTimeAccident* y *DateReported* a formato fecha.

```
class(insurances$DateReported)
```

```
## [1] "character"
```

```
anyNA(insurances$DateReported)
```

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

```
head(insurances$DateReported, 3)
```

```
## [1] "2002-07-05T00:00:00Z" "1999-01-20T00:00:00Z" "1996-04-14T00:00:00Z"
```

```
insurances$DateReported <- as.Date(insurances$DateReported)

class(insurances$DateReported)
```

```
## [1] "Date"
```

```
head(insurances$DateReported, 3)
```

```
## [1] "2002-07-05" "1999-01-20" "1996-04-14"
```

```
class(insurances$DateTimeOfAccident)
```

```
## [1] "character"
```

```
anyNA(insurances$DateTimeOfAccident)
```

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

```
head(insurances$DateTimeOfAccident, 3)
```

```
## [1] "2002-04-09T07:00:00Z" "1999-01-07T11:00:00Z" "1996-03-25T00:00:00Z"
```

```
insurances$DateTimeOfAccident <- as.Date(insurances$DateTimeOfAccident)

class(insurances$DateTimeOfAccident)
```

```
## [1] "Date"
```

```
head(insurances$DateTimeOfAccident, 3)
```

```
## [1] "2002-04-09" "1999-01-07" "1996-03-25"
```

A continuación se procede a calcular la diferencia entre la fecha en que se reportó el accidente (*DateReported*) y la fecha en que se sufrió dicho accidente (*DateTimeOfAccident*) y guardar en la variable *Time* del *dataset*.

```
insurances$Time <- insurances$DateReported - insurances$DateTimeOfAccident

head(insurances[, c("ClaimNumber", "DateReported", "DateTimeOfAccident",
                    "Time")])
```

```
##   ClaimNumber DateReported DateTimeOfAccident    Time
## 1   WC8285054   2002-07-05         2002-04-09 87 days
## 2   WC6982224   1999-01-20         1999-01-07 13 days
## 3   WC5481426   1996-04-14         1996-03-25 20 days
## 4   WC9775968   2005-07-22         2005-06-22 30 days
```

```
## 5   WC2634037   1990-09-27         1990-08-29 29 days
## 6   WC6828422   1999-09-09         1999-06-21 80 days
```

8.2 Diferencia entre `IniCost` y `UltCost`

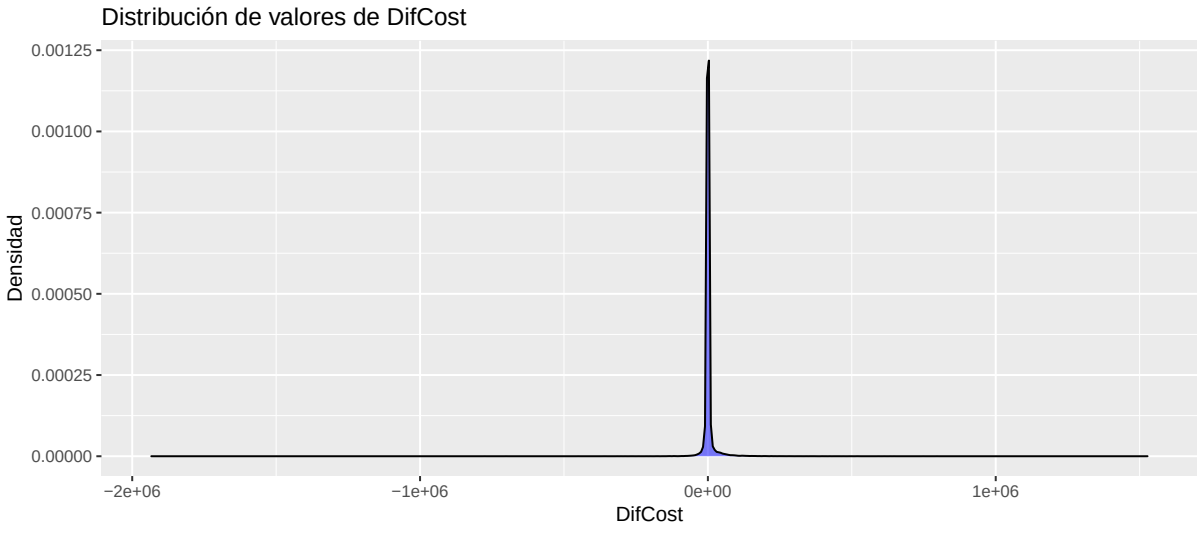
Se calcula la diferencia entre el coste final (`UltCost`) y el coste inicial estimado (`InitCost`) y se almacena en la variable `DifCost` del *dataset*.

```
insurances$DifCost <- insurances$UltCost - insurances$IniCost

head(insurances[, c("ClaimNumber", "IniCost", "UltCost", "DifCost")])
```

```
##   ClaimNumber IniCost UltCost DifCost
## 1   WC8285054    1500    4303    2803
## 2   WC6982224    5500    6105     605
## 3   WC5481426    1700    2098     398
## 4   WC9775968   15000   16282    1282
## 5   WC2634037    2800    3771     971
## 6   WC6828422     500     746     246
```

Se procede a visualizar la distribución de la nueva variable `DifCost` con un diagrama de densidad.

```
ggplot(mapping= aes(x=insurances$DifCost)) +
  geom_density(alpha=0.5, fill="blue") +
  labs(title = "Distribución de valores de DifCost",
    x = "DifCost", y = "Densidad")
```

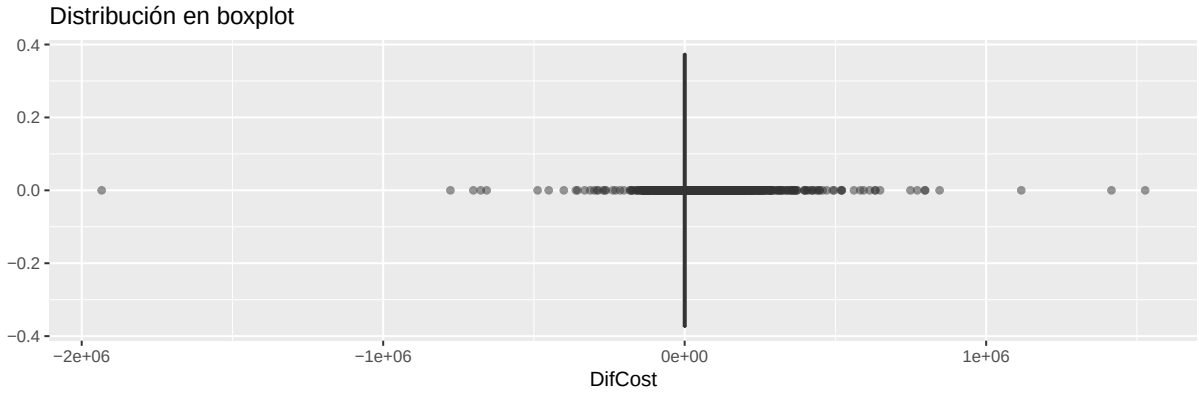


```
summary(insurances$DifCost)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1933783      -449       190      2354      1435   1527535
```

La mayoría de los valores se distribuyen en torno a 0; es decir, las estimaciones iniciales del coste de la indemnización son más o menos precisas. Sin embargo, mirando los valores mínimo y máximo, se aprecia la existencia de estimaciones extremas. Se comprueba con una visualización en *boxplot*.

```
ggplot(data = insurances, mapping = aes(x=DifCost, fill=Age)) +
  geom_boxplot(alpha=0.5, fill="blue") +
  labs(title = "Distribución en boxplot", x = "DifCost")
```



9 Estudio descriptivo

9.1 Funciones de media robustas

9.1.1 Media recortada

La **media recortada** es una función de media robusta que elimina un % de los extremos inferior y superior de los datos antes de calcular la media.

```
# x : vector de numerics
# perc : fraccion de los datos a recortar
# returns la media recortada
media.recortada <- function(x, perc=0.05){
```

```
  if(!is.numeric(x)){
    stop("X must be a <numeric> vector")
  }
  if(anyNA(x)){
    stop("X must not contain NA values")
  }
  if(perc < 0.0 || perc > 1.0){
    stop("perc must be a <numeric> in [0.0, 1.0] range")
  }

  x.sorted <- sort(x)
  perc.to_trim <- floor(perc * length(x))
  x.trimmed <- x.sorted[seq(perc.to_trim + 1,length(x) - perc.to_trim)]
  return(sum(x.trimmed) / length(x.trimmed))
}
```

9.1.2 Media winsorizada

La **media winsorizada** es una función de media robusta que reemplaza un % de los extremos inferior y superior de los datos con la observación más cercana a ellos antes de calcular la media.

```
# x : vector de numerics
# perc : fraccion de los datos a winsorizar
# returns la media winsorizada
media.winsor <- function(x, perc=0.05){
 if(!is.numeric(x)){
    stop("X must be a <numeric> vector")
  }
  if(anyNA(x)){
    stop("X must not contain NA values")
  }
  if(perc < 0.0 || perc > 1.0){
    stop("perc must be a <numeric> in [0.0, 1.0] range")
  }

  x.sorted <- sort(x)
```

```
  perc.to_winsor.low <- perc * length(x)
  perc.to_winsor.high <- (1 - perc) * length(x)

  winsor.quantiles <- quantile(x, probs=c(perc, 1.0 - perc))
  x.to_winsor.low <- winsor.quantiles[1]
  x.to_winsor.high <- winsor.quantiles[2]

  x.winsorized <- x.sorted
  x.winsorized[x.winsorized < x.to_winsor.low] <- x.to_winsor.low
  x.winsorized[x.winsorized > x.to_winsor.high] <- x.to_winsor.high

  return(sum(x.winsorized) / length(x.winsorized))
}
```

9.1.3 Pruebas

Se procede a crear un vector numérico de prueba para comprobar que las funciones de media recortada y winsorizada anteriormente creadas son correctas, comparando su resultado con el resultado que las funciones *built-in* de R ofrecen.

```
mean_test<-c(92, 19, 101, 58, 1053, 91, 26, 78, 10, 13, -40,
             101, 86, 85, 15, 89, 89, 28, -5, 41)

media.recortada(mean_test, 0.05)
```

```
## [1] 56.5
```

```
mean(mean_test, trim=0.05)
```

```
## [1] 56.5
```

```
media.winsor(mean_test, 0.05)
```

```
## [1] 57.9425
```

```
winsor.mean(mean_test, trim=0.05)
```

```
## [1] 57.9425
```

9.2 Estudio descriptivo de las variables cuantitativas

A continuaciónse realiza un estudio descriptivo de las variables cuantitativas *Age*, *WeeklyWages*, *DaysWeek*, *HoursWeek*, `IniCost` y `UltCost`. Para cada una de ellas, se calcula la media aritmética, la media recortada, la media winsorizada, la mediana y la desviación típica.

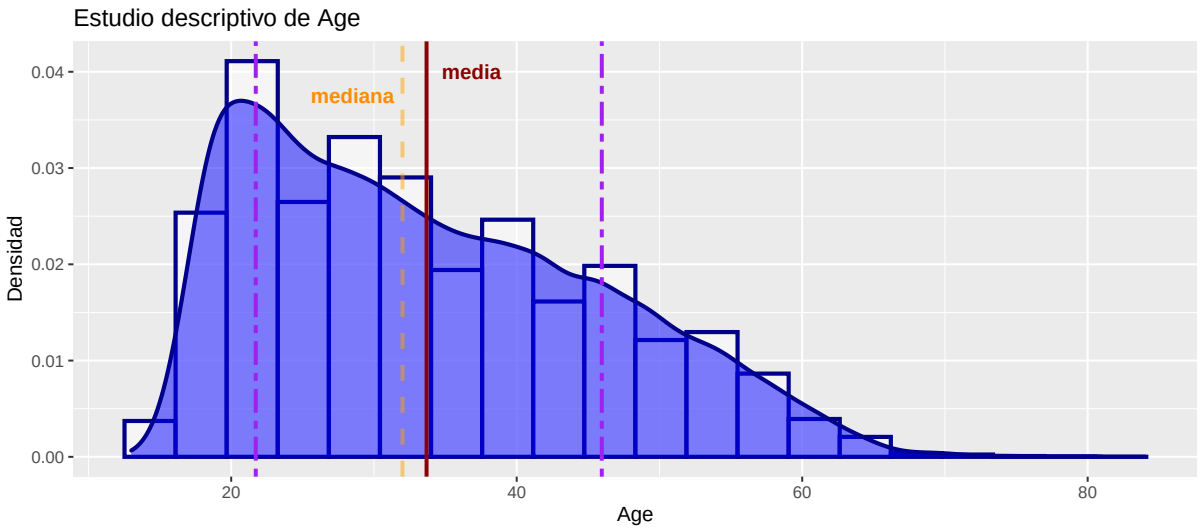
```
descr.study <- data.frame(
  "Media" = c(round(mean(insurances$Age), digits=2),
              round(mean(insurances$WeeklyWages), digits=2),
              round(mean(insurances$DaysWeek), digits=2),
              round(mean(insurances$HoursWeek), digits=2),
              round(mean(insurances$IniCost), digits=2),
              round(mean(insurances$UltCost), digits=2)),

  "MediaRec." = c(round(media.recortada(insurances$Age), digits=2),
               round(media.recortada(insurances$WeeklyWages), digits=2),
               round(media.recortada(insurances$DaysWeek), digits=2),
               round(media.recortada(insurances$HoursWeek), digits=2),
               round(media.recortada(insurances$IniCost), digits=2),
               round(media.recortada(insurances$UltCost), digits=2)),

  "MediaWins." = c(round(media.winsor(insurances$Age), digits=2),
                round(media.winsor(insurances$WeeklyWages), digits=2),
                round(media.winsor(insurances$DaysWeek), digits=2),
                round(media.winsor(insurances$HoursWeek), digits=2),
                round(media.winsor(insurances$IniCost), digits=2),
                round(media.winsor(insurances$UltCost), digits=2)),

  "Mediana" = c(round(median(insurances$Age), digits=2),
                round(median(insurances$WeeklyWages), digits=2),
                round(median(insurances$DaysWeek), digits=2),
                round(median(insurances$HoursWeek), digits=2),
                round(median(insurances$IniCost), digits=2),
                round(median(insurances$UltCost), digits=2)),

  "Desv.Tipica" = c(round(sd(insurances$Age), digits=2),
                    round(sd(insurances$WeeklyWages), digits=2),
                    round(sd(insurances$DaysWeek), digits=2),
```

```
                      round(sd(insurances$HoursWeek), digits=2),
                      round(sd(insurances$IniCost), digits=2),
                      round(sd(insurances$UltCost), digits=2)),

  row.names = c("Age", "WeeklyWages", "DaysWeek", "HoursWeek",
                "IniCost", "UltCost"))

descr.study
```

```
##                Media MediaRec. MediaWins. Mediana Desv.Tipica
## Age            33.84     33.32      33.69    32.0       12.12
## WeeklyWages   416.36    394.62     406.01   392.2      248.64
## DaysWeek        4.91      4.98       4.93     5.0        0.55
## HoursWeek      37.47     37.94      37.27    38.0        7.01
## IniCost      7841.15   5102.87    6108.34  2000.0    20584.08
## UltCost     10194.96   6195.80    7538.20  3179.0    29023.51
```

9.2.1 *Age*

```
ggplot(mapping= aes(x=insurances$Age)) +
  geom_histogram(aes(y=..density..), alpha=0.5, fill="white",
                 colour="darkblue", bins=20, size=1.05) +
  geom_density(alpha=0.5, fill="blue", colour="darkblue", size=1.05) +
  labs(title = "Estudio descriptivo de Age", x = "Age",
       y = "Densidad") +
  geom_vline(aes(xintercept=media.winsor(insurances$Age)),
             color="darkred", linetype="solid", size=1) +
  annotate("text", x= mean(insurances$Age) + 3, y = 0.04,
           label="media", color="darkred", size=4, fontface="bold") +
  geom_vline(aes(xintercept=median(insurances$Age)), color="orange",
             linetype="dashed", size=1, alpha=0.5) +
  annotate("text", x= median(insurances$Age) - 3.5, y = 0.0375,
           label="mediana", color="darkorange", size=4,
           fontface="bold") +
  geom_vline(aes(xintercept=mean(insurances$Age) + sd(insurances$Age)),
             color="purple", linetype="twodash", size=1) +
  geom_vline(aes(xintercept=mean(insurances$Age) - sd(insurances$Age)),
             color="purple", linetype="twodash", size=1)
```

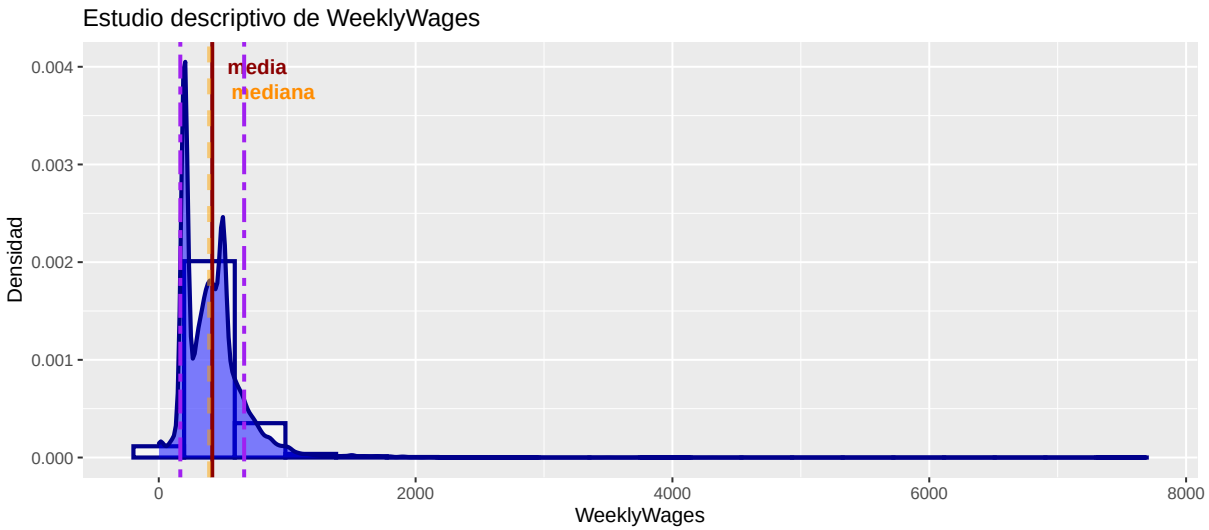


En la figura se puede comprobar que el atributo *Age* es asimétrico, con una cola a la derecha. La mayor parte de los valores se concentran en el rango [20, 45].

9.2.2 *WeeklyWages*

```
ggplot(mapping= aes(x=insurances$WeeklyWages)) +
  geom_histogram(aes(y=..density..), alpha=0.5, fill="white",
                 colour="darkblue", bins=20, size=1.05) +
  geom_density(alpha=0.5, fill="blue", colour="darkblue", size=1.05) +
  labs(title = "Estudio descriptivo de WeeklyWages",
       x = "WeeklyWages", y = "Densidad") +
  geom_vline(aes(xintercept=mean(insurances$WeeklyWages)),
             color="darkred", linetype="solid", size=1) +
  annotate("text", x= mean(insurances$WeeklyWages) + 350, y = 0.004,
           label="media", color="darkred", size=4, fontface="bold") +
  geom_vline(aes(xintercept=median(insurances$WeeklyWages)),
             color="orange", linetype="dashed", size=1, alpha=0.5) +
  annotate("text", x= median(insurances$WeeklyWages) + 500, y = 0.00375,
           label="mediana", color="darkorange", size=4,
           fontface="bold") +
  geom_vline(aes(xintercept=mean(insurances$WeeklyWages) +
                   sd(insurances$WeeklyWages)),
             color="purple", linetype="twodash", size=1) +
  geom_vline(aes(xintercept=mean(insurances$WeeklyWages) -
```

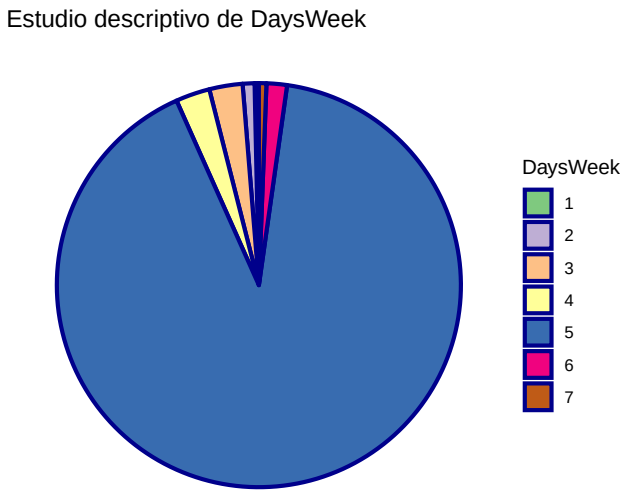
```
                    sd(insurances$WeeklyWages)),
              color="purple", linetype="twodash", size=1)
```



En la figura se puede comprobar que el atributo *WeeklyWages* es asimétrico, con una cola a la derecha muy larga. La mayor parte de los valores se concentran en el rango [200, 600].

9.2.3 *DaysWeek*

```
days_week.freqs <- aggregate(data.frame(value=insurances$DaysWeek),
                             list(group=insurances$DaysWeek), length)
days_week.freqs$value <- days_week.freqs$value /
  length(insurances$DaysWeek)

ggplot(days_week.freqs, aes(x="", y=value, fill=as.factor(group))) +
  geom_bar(width=1, stat="identity", color="darkblue", size=1.05) +
  coord_polar("y", start=0) +
  scale_fill_brewer(palette="Accent") +
  theme_void() +
  labs(title = "Estudio descriptivo de DaysWeek", fill="DaysWeek")
```

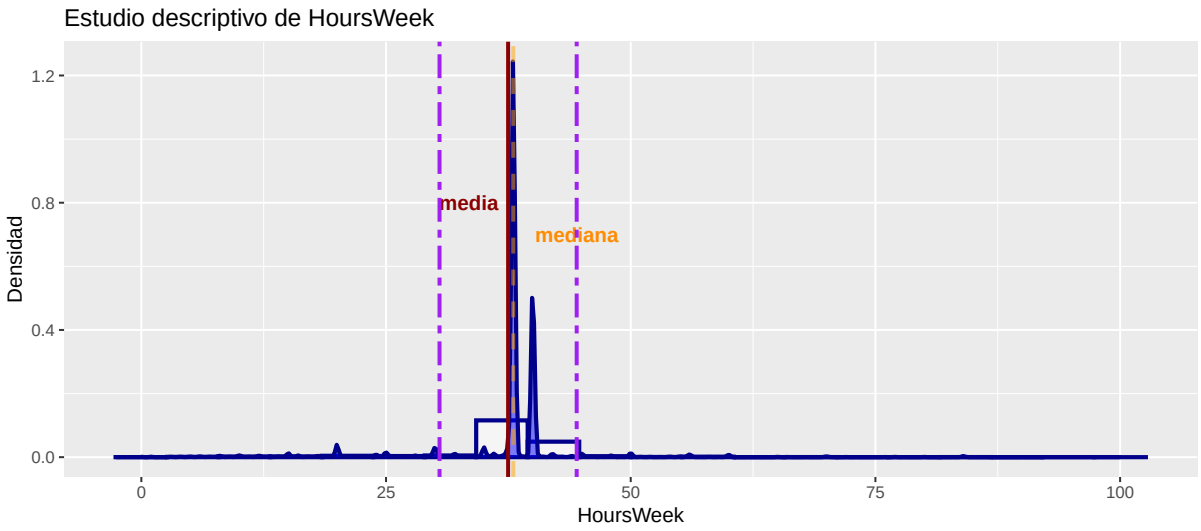


En la figura se comprueba que en *DaysWeek* la mayoría de los trabajadores laboran 5 días a la semana.

9.2.4 *HoursWeek*

```
ggplot(mapping= aes(x=insurances$HoursWeek)) +
  geom_histogram(aes(y=..density..), alpha=0.5, fill="white",
                 colour="darkblue", bins=20, size=1.05) +
  geom_density(alpha=0.5, fill="blue", colour="darkblue", size=1.05) +
  labs(title = "Estudio descriptivo de HoursWeek",
       x = "HoursWeek", y = "Densidad") +
  geom_vline(aes(xintercept=mean(insurances$HoursWeek)), color="darkred",
             linetype="solid", size=1) +
  annotate("text", x= mean(insurances$HoursWeek) - 4, y = 0.8,
           label="media", color="darkred", size=4, fontface="bold") +
  geom_vline(aes(xintercept=median(insurances$HoursWeek)), color="orange",
             linetype="dashed", size=1, alpha=0.5) +
  annotate("text", x= median(insurances$HoursWeek) + 6.5, y = 0.7,
           label="mediana", color="darkorange", size=4,
           fontface="bold") +
  geom_vline(aes(xintercept=mean(insurances$HoursWeek) +
                   sd(insurances$HoursWeek)),
             color="purple", linetype="twodash", size=1) +
  geom_vline(aes(xintercept=mean(insurances$HoursWeek) -
                   sd(insurances$HoursWeek)),
```

```
            color="purple", linetype="twodash", size=1)
```

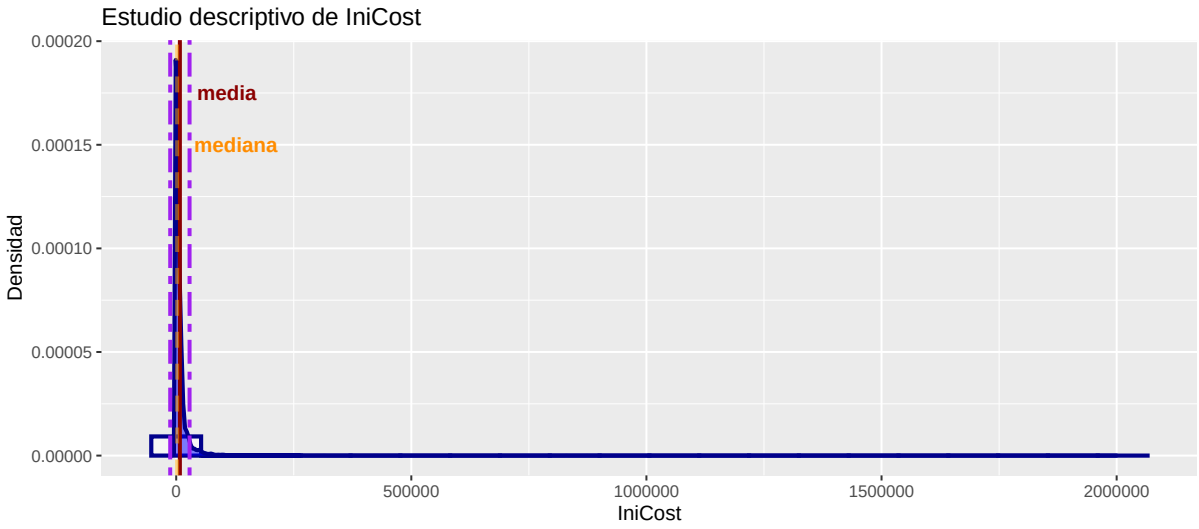


En la figura se puede comprobar que el atributo *HoursWeek* es simétrico, con colas a la izquierda y a la derecha muy largas. La mayor parte de los valores se concentran en el rango [30, 45].

9.2.5 `IniCost`

```
ggplot(mapping= aes(x=insurances$IniCost)) +
  geom_histogram(aes(y=..density..), alpha=0.5, fill="white",
                 colour="darkblue", bins=20, size=1.05) +
  geom_density(alpha=0.5, fill="blue", colour="darkblue", size=1.05) +
  labs(title = "Estudio descriptivo de IniCost",
       x = "IniCost", y = "Densidad") +
  geom_vline(aes(xintercept=mean(insurances$IniCost)),
             color="darkred", linetype="solid", size=1) +
  annotate("text", x= mean(insurances$IniCost) + 100000, y = 0.000175,
           label="media", color="darkred", size=4, fontface="bold") +
  geom_vline(aes(xintercept=median(insurances$IniCost)),
             color="orange", linetype="dashed", size=1, alpha=0.5) +
  annotate("text", x= median(insurances$IniCost) + 125000, y = 0.00015,
           label="mediana", color="darkorange", size=4,
           fontface="bold") +
  geom_vline(aes(xintercept=mean(insurances$IniCost) +
```

```
                    sd(insurances$IniCost)),
              color="purple", linetype="twodash", size=1) +
  geom_vline(aes(xintercept=mean(insurances$IniCost) -
                    sd(insurances$IniCost)),
              color="purple", linetype="twodash", size=1)
```

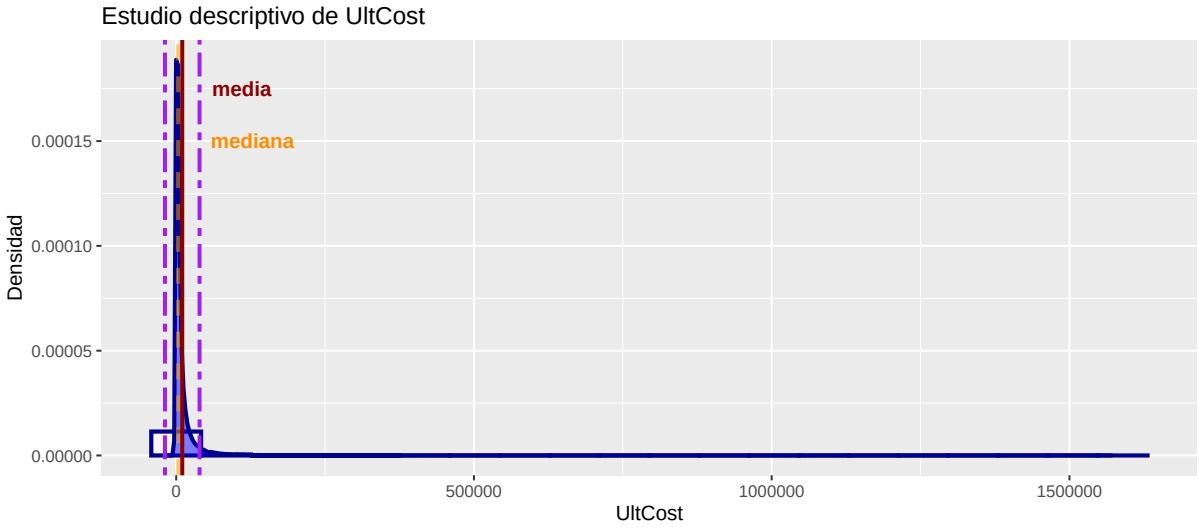


En la figura se puede comprobar que el atributo `IniCost` es asimétrico, con una cola a la derecha muy larga.

9.2.6 `UltCost`

```
ggplot(mapping= aes(x=insurances$UltCost)) +
  geom_histogram(aes(y=..density..), alpha=0.5, fill="white",
                 colour="darkblue", bins=20, size=1.05) +
  geom_density(alpha=0.5, fill="blue", colour="darkblue", size=1.05) +
  labs(title = "Estudio descriptivo de UltCost",
       x = "UltCost", y = "Densidad") +
  geom_vline(aes(xintercept=mean(insurances$UltCost)),
             color="darkred", linetype="solid", size=1) +
  annotate("text", x= mean(insurances$UltCost) + 100000, y = 0.000175,
           label="media", color="darkred", size=4, fontface="bold") +
  geom_vline(aes(xintercept=median(insurances$UltCost)),
             color="orange", linetype="dashed", size=1, alpha=0.5) +
  annotate("text", x= median(insurances$UltCost) + 125000, y = 0.00015,
```

```
          label="mediana", color="darkorange", size=4,
          fontface="bold") +
  geom_vline(aes(xintercept=mean(insurances$UltCost) +
                   sd(insurances$UltCost)),
             color="purple", linetype="twodash", size=1) +
  geom_vline(aes(xintercept=mean(insurances$UltCost) -
                   sd(insurances$UltCost)),
             color="purple", linetype="twodash", size=1)
```



En la figura se puede comprobar que el atributo `UltCost` es asimétrico, con una cola a la derecha muy larga.

---

10 Archivo final

Se procede a copiar el resultado del preprocesamiento —el *dataframe* `insurances` — en el archivo *Lazaro_fichero_clean.csv*.

```
head(insurances, n=3L)
```

```
##   ClaimNumber DateTimeOfAccident DateReported Age Gender MaritalStatus
## 1   WC8285054         2002-04-09   2002-07-05  48      M             M
## 2   WC6982224         1999-01-07   1999-01-20  43      F             M
## 3   WC5481426         1996-03-25   1996-04-14  30      M             U
```

```
##   DependentChildren DependentsOther WeeklyWages PartTimeFullTime HoursWeek
## 1                 0               0      500.00                F      38.0
## 2                 0               0      509.34                F      37.5
## 3                 0               0      709.10                F      38.0
##   DaysWeek                                        ClaimDescription IniCost
## 1        5          LIFTING TYRE INJURY TO RIGHT ARM AND WRIST INJURY    1500
## 2        5 STEPPED AROUND CRATES AND TRUCK TRAY FRACTURE LEFT FOREARM    5500
## 3        5                          CUT ON SHARP EDGE CUT LEFT THUMB    1700
##   UltCost    Time DifCost
## 1    4303 87 days    2803
## 2    6105 13 days     605
## 3    2098 20 days     398
```

```
write.csv(insurances, "Lazaro_fichero_clean.csv", row.names = FALSE)
```