Tarea 1 CV2

Alejandro Brenes y Erick Venegas

2025-02-07

Análisis Exploratorio de los Datos

Descarga de datos

Como primer paso, se bajan las bases de datos necesarias, tanto la de los datos de los asegurados como la tabla de mortalidad.

```
library(readxl)
datos_asegurados <- read_excel("Datos_Final.xlsx")
tabla_mort <- read_excel("Tabla de mortalidad.xlsx")
```

Análisis para los datos de la mortalidad

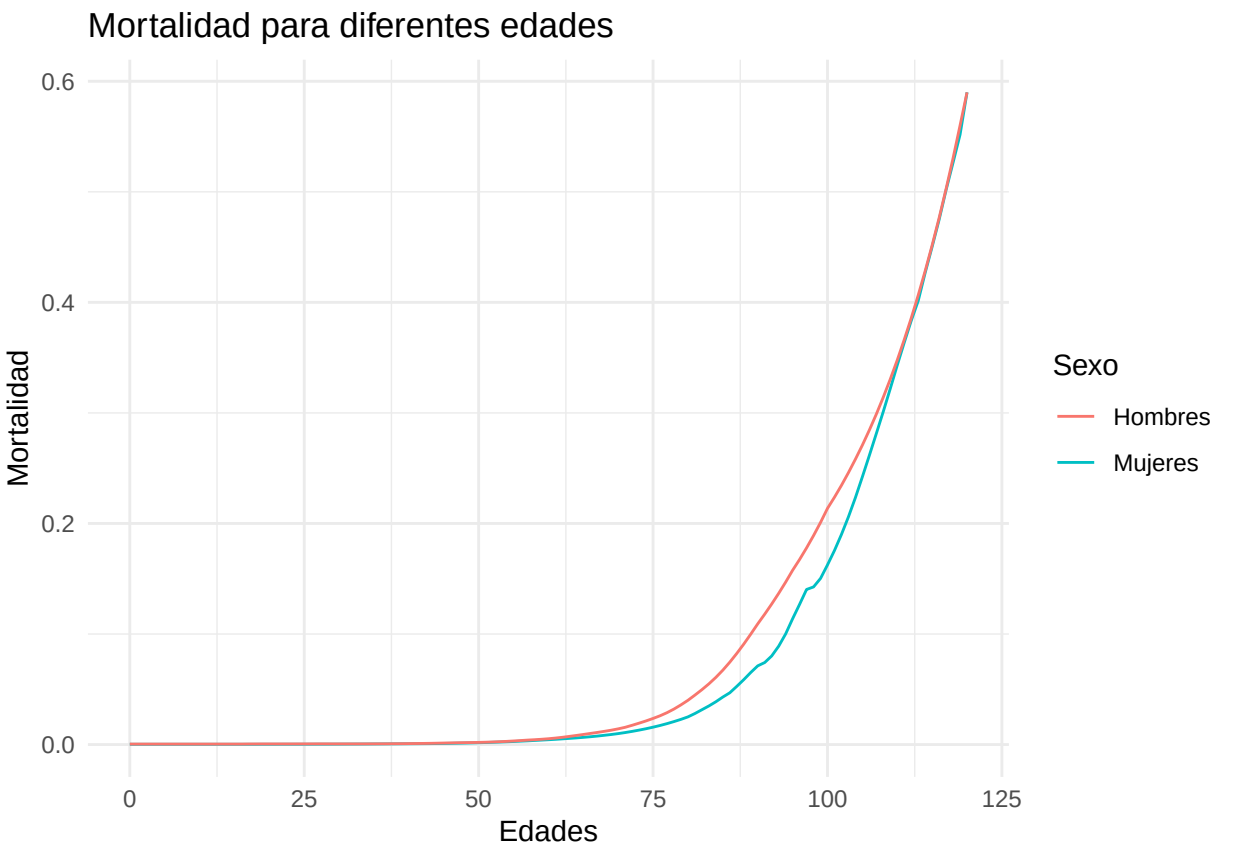
Se verifica si hay datos faltantes de algún tipo:

```
tabla_mort <- tabla_mort[, -c(3, 5)] #Se eliminan estas columnas correspondientes a los fumadores

any(is.na(tabla_mort))
```

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

Vemos que los datos están completos. Ahora podemos realizar un gráfico para visualizar de mejor manera los datos:

```
library(ggplot2)

ggplot(tabla_mort, aes(x = Edad)) +
  geom_line(aes(y = M_NF, color = "Mujeres")) +
  geom_line(aes(y = H_NF, color = "Hombres")) +
  labs(title = "Mortalidad para diferentes edades",
       x = "Edades",
       y = "Mortalidad",
       color = "Sexo") +
  theme_minimal()
```



Se puede recalcar es que la mortalidad se comporta bastante similar hasta llegar a los 70 años aproximadamente, ya que de ahí a aproximadamente los 110 años la mortalidad en los hombres siempre es superior. Algo a destacar es que se asume que en la edad máxima de esta base de datos, 120, la probabilidad de morir al año siguiente no es 1, por lo que no hay una edad Ω limite donde se "corta".

Además, pareciera que las probabilidades están bajo un enfoque más optimista, al ser estas bastante bajas. Por ejemplo, hasta la edad de 117, tanto en hombres como en mujeres, la probabilidad de que la persona sobreviva al año siguiente es inclusive mayor a la de morir. Esto es importante tomarlo en cuenta ya que puede repercutir en los montos de las reservas y cómo se comportan las personas a lo largo del tiempo. Otro ejemplo de ello es que las anualidades al solo considerar la probabilidad de vida tendrán valores más altos. De igual forma, tener probabilidades de muerte tan bajas, esto repercute a la hora de calcular los seguros, pues se podría estar haciendo una subestimación de estas probabilidades, provocando que las muertes esperadas sean mucho menores que de las que de verdad ocurren.

Análisis para los datos de los contratos

Ahora pasamos al análisis de la base de datos de los asegurados. Como primer paso, se verifica que cada id sea único:

```
length(unique(datos_asegurados$Id)) == nrow(datos_asegurados)
```

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

Vemos que la base esta completa:

```r
any(is.na(datos_asegurados))
```

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

Variables cualitativas

Ahora, se pueden analizar las columnas de datos cuantitativos:

```r
cat("Posibles sexos del asegurado:\n", unique(datos_asegurados$`Sexo del asegurado`), "\n")
```

```
## Posibles sexos del asegurado:
##  F M
```

```r
cat("Monedas utilizadas:\n", unique(datos_asegurados$Moneda), "\n")
```

```
## Monedas utilizadas:
##  $
```

```r
cat("Opciones para la periodicidad del pago:\n", unique(datos_asegurados$`Periodicidad del pago`), "\n")
```

```
## Opciones para la periodicidad del pago:
##  ANUAL MENSUAL SEMESTRAL TRIMESTRAL
```

```r
summary(datos_asegurados)
```

```
##        Id        Ultima renovación
##  Min.   :  1.0   Min.   :2023-01-03 00:00:00.00
##  1st Qu.:104.2   1st Qu.:2023-05-11 00:00:00.00
##  Median :207.5   Median :2023-08-07 00:00:00.00
##  Mean   :207.5   Mean   :2023-07-26 12:48:41.74
##  3rd Qu.:310.8   3rd Qu.:2023-10-24 00:00:00.00
##  Max.   :414.0   Max.   :2023-12-28 00:00:00.00
##  Fecha de nacimiento del asegurado Sexo del asegurado    Moneda
##  Min.   :1955-05-26 00:00:00.000   Length:414         Length:414
##  1st Qu.:1977-04-30 12:00:00.000   Class :character   Class :character
##  Median :1982-06-16 00:00:00.000   Mode  :character   Mode  :character
##  Mean   :1982-02-26 21:47:49.564
##  3rd Qu.:1988-02-28 00:00:00.000
##  Max.   :2003-07-02 00:00:00.000
##  Suma asegurada    Fecha de inicio de la póliza
##  Min.   :  17250   Min.   :2022-09-05 00:00:00.00
##  1st Qu.: 103500   1st Qu.:2023-01-21 12:00:00.00
##  Median : 115000   Median :2023-05-29 00:00:00.00
##  Mean   : 181074   Mean   :2023-05-12 14:15:39.13
##  3rd Qu.: 172500   3rd Qu.:2023-08-23 18:00:00.00
##  Max.   :2875000   Max.   :2023-12-22 00:00:00.00
##  Fecha de vencimiento de la póliza Periodicidad del pago Monto de la prima pura
##  Min.   :2032-09-14 00:00:00.00    Length:414            Min.   :   5.067
##  1st Qu.:2042-11-17 00:00:00.00    Class :character      1st Qu.:  29.857
```

```
##  Median :2043-05-26 12:00:00.00    Mode  :character      Median :  50.755
##  Mean   :2043-09-03 13:58:15.65                         Mean   : 145.421
##  3rd Qu.:2043-10-23 00:00:00.00                         3rd Qu.: 109.016
##  Max.   :2053-12-15 00:00:00.00                         Max.   :3688.769
##  Monto de la prima de inventario Monto de la prima comercial
##  Min.   :   9.211                Min.   :  20.46
##  1st Qu.:  57.323                1st Qu.: 154.59
##  Median : 100.776                Median : 279.27
##  Mean   : 234.330                Mean   : 555.68
##  3rd Qu.: 215.564                3rd Qu.: 576.26
##  Max.   :4320.717                Max.   :7832.50
```

Variables cuantitativas: fechas

Se continúa con una serie de observaciones de los datos cuantitativos. Para empezar, las fechas.

Se convierten las fechas a formato Date:

```
datos_asegurados$`Ultima renovación` <- as.Date(datos_asegurados$`Ultima renovación`)

datos_asegurados$`Fecha de nacimiento del asegurado` <- as.Date(datos_asegurados$`Fecha de nacimiento de

datos_asegurados$`Fecha de inicio de la póliza` <- as.Date(datos_asegurados$`Fecha de inicio de la póli

datos_asegurados$`Fecha de vencimiento de la póliza` <- as.Date(datos_asegurados$`Fecha de vencimiento
```

Ahora bien, las 4 fechas que se establecen para cada individuo deben de seguir un orden lógico. La fecha más antigua es la de nacimiento; de ahí en adelante, la fecha de inicio debe de ser mayor o igual de antigua que la última renovación, así como esta debe ser a lo sumo igual de antigua que la fecha de vencimiento. A continuación se verifican estas condiciones, mediante un cálculo vectorizado de condiciones. La idea es que en cada "which" se verifique fila por fila si las condiciones antes mencionadas sean falsas, en caso de serla, almacena la fila en la que se incumplió, obteniendo un vector con las filas que se incumplen las condiciones, en las que, por lo tanto, existe una inconsistencia.

```
# Cálculo vectorizado de las condiciones
falsos_1 <- which(datos_asegurados$`Fecha de nacimiento del asegurado` >= datos_asegurados$`Ultima renov
falsos_2 <- which(datos_asegurados$`Fecha de nacimiento del asegurado` >= datos_asegurados$`Fecha de ini
falsos_3 <- which(datos_asegurados$`Fecha de nacimiento del asegurado` >= datos_asegurados$`Fecha de ven
falsos_4 <- which(datos_asegurados$`Fecha de inicio de la póliza` >= datos_asegurados$`Fecha de vencimie
falsos_5 <- which(datos_asegurados$`Fecha de inicio de la póliza` >= datos_asegurados$`Ultima renovación
falsos_6 <- which(datos_asegurados$`Ultima renovación` >= datos_asegurados$`Fecha de vencimiento de la p

#Resultados
falsos_1
```

```
## integer(0)
```

```
falsos_2
```

```
## integer(0)
```

```
falsos_3
```

```
## integer(0)
```

```
falsos_4
```

```
## integer(0)
```

```
falsos_5
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  86  88  89  90  91  92  93  94  95
##  [73]  96  97  98  99 100 101 102 103 128 129 130 131 132 133 134 135 136 137
##  [91] 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155
## [109] 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173
## [127] 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
## [145] 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
## [163] 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227
## [181] 228 229 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 260
## [199] 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278
## [217] 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296
## [235] 297 299 300 301 302 303 304 305 306 307 308 310 311 312 313 314 315 316
## [253] 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334
## [271] 335 336 337 338 339 341 342 363 364 365 366 367 368 369 370 371 372 373
## [289] 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391
## [307] 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409
## [325] 410 411 412 413 414
```

```
falsos_6
```

```
## integer(0)
```

Se puede ver que todas las condiciones lógicas que se plantearon se cumplen, a excepción del caso en el que se verifica que las fechas de inicio de la póliza deben de ser a lo sumo igual de antiguas que la última renovación.

```
which(datos_asegurados$`Fecha de inicio de la póliza` > datos_asegurados$`Ultima renovación`)
```

```
## integer(0)
```

Cuando suprimimos el escenario en el que sean iguales, este comportamiento desaparece; de lo cual se deduce que en todos estos asegurados, no se ha dado ninguna renovación desde la fecha de inicio de la póliza.

Para todos los usuarios, la última renovación fue en 2023:

```
any(format(datos_asegurados$`Ultima renovación`, "%Y") != "2023")
```

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

Solo en el usuario 191 la fecha de vencimiento no es ciertos años exactos después de la fecha de inicio de la póliza, pero se tomará como que sí por simplicidad:

```
datos_asegurados$misma_fecha <- format(datos_asegurados$`Fecha de inicio de la póliza`, "%m-%d") == for

which(datos_asegurados$misma_fecha == FALSE)
```

```
## [1] 191
```

Sabemos que siempre la diferencia entre el 31 de diciembre 23 y la última evaluación es menor a un año:

```
fecha_referencia <- as.Date("2023-12-31")

diferencia <- as.numeric(difftime(fecha_referencia, datos_asegurados$`Ultima renovación`, units = "days

# Se verifica si hay alguna diferencia menor a 365 días
which(diferencia > 365 & diferencia >= 0)
```

```
## integer(0)
```

Todos los años de la fecha de inicio de la póliza son 2022 o 2023:

```
any(!(format(datos_asegurados$`Fecha de inicio de la póliza`, "%Y") %in% c("2022", "2023")))
```

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

Variables cuantitativas: montos

Como primer paso se podría hacer un resumen con los estadísticos más destacados de las columnas correspondientes a montos.

```
summary(datos_asegurados$`Suma asegurada`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17250  103500  115000  181074  172500 2875000
```

```
summary(datos_asegurados$`Monto de la prima pura`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    5.067   29.857   50.755  145.421  109.016 3688.769
```

```
summary(datos_asegurados$`Monto de la prima de inventario`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    9.211   57.323  100.776  234.330  215.564 4320.717
```

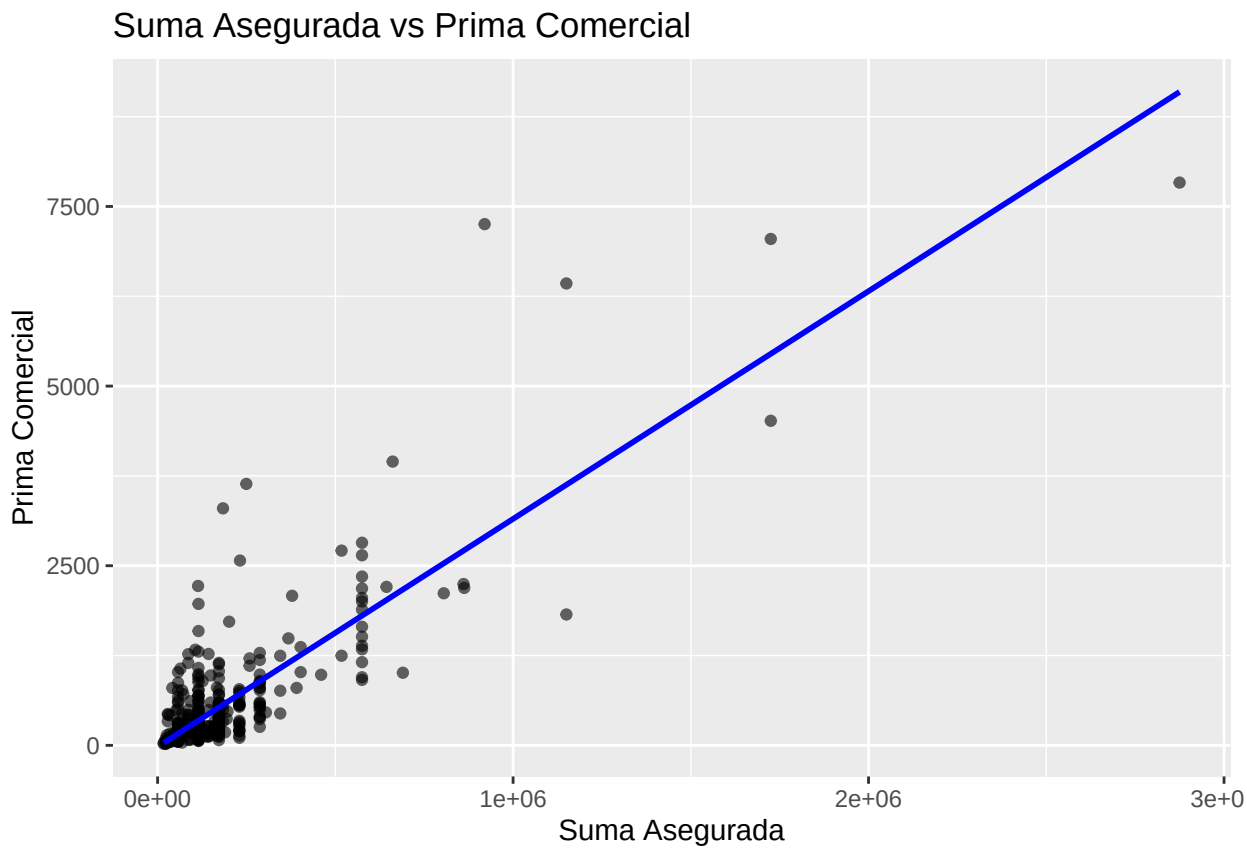
```r
summary(datos_asegurados$`Monto de la prima comercial`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.46  154.59  279.27  555.68  576.26 7832.50
```

Ahora, una serie de gráficos y medidas estadísticas que ayuden a entender la dinámica de las variables. Primero, un gráfico que ayude a visualizar la relación entre la suma asegurada y la prima comercial.

```r
library(ggplot2)

ggplot(datos_asegurados, aes(x = `Suma asegurada`, y = `Monto de la prima comercial`)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Suma Asegurada vs Prima Comercial",
       x = "Suma Asegurada",
       y = "Prima Comercial")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

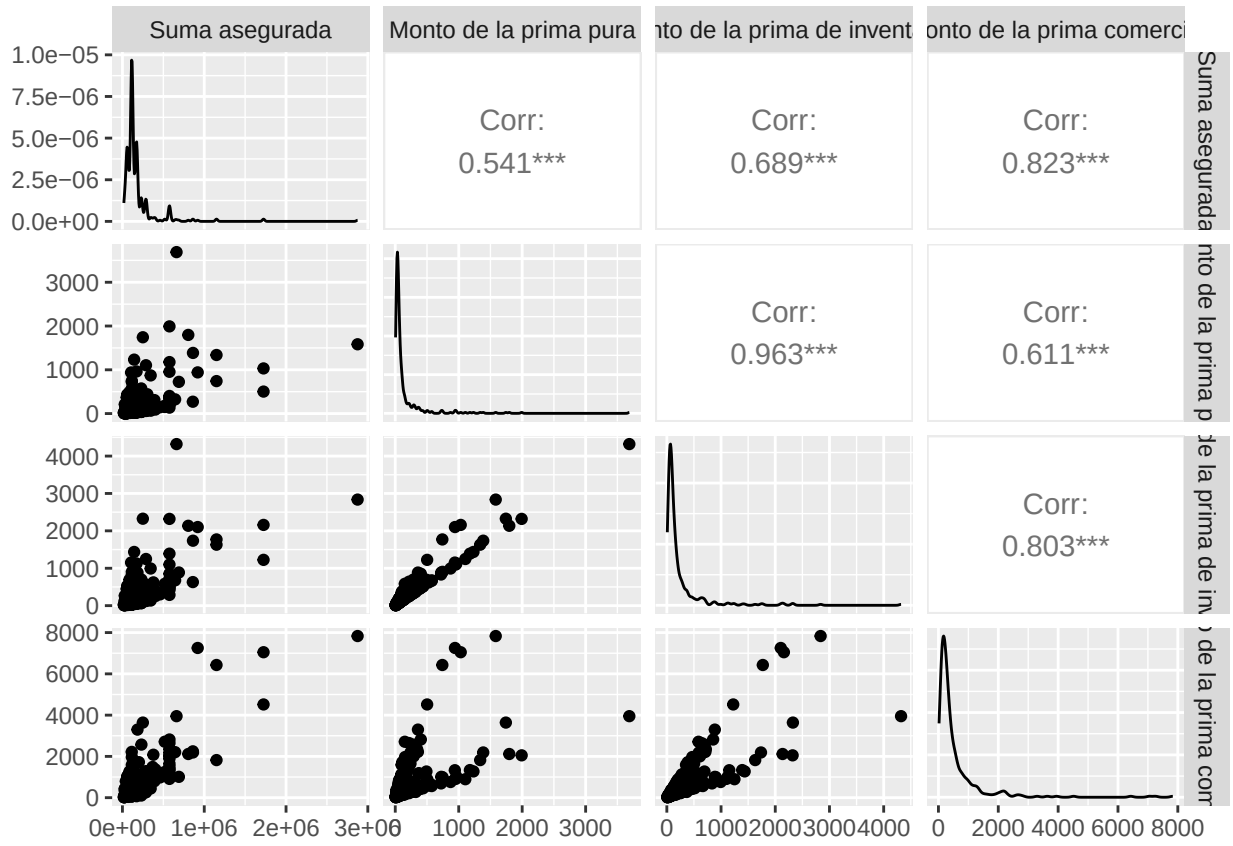


Ahora, con el paquete GGally, se nos permite observar la densidad de cada variable (en la diagonal), la correlación que existe entre ellas (a la derecha de la diagonal), y gráficos de dispersión (a la izquierda de la diagonal).

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

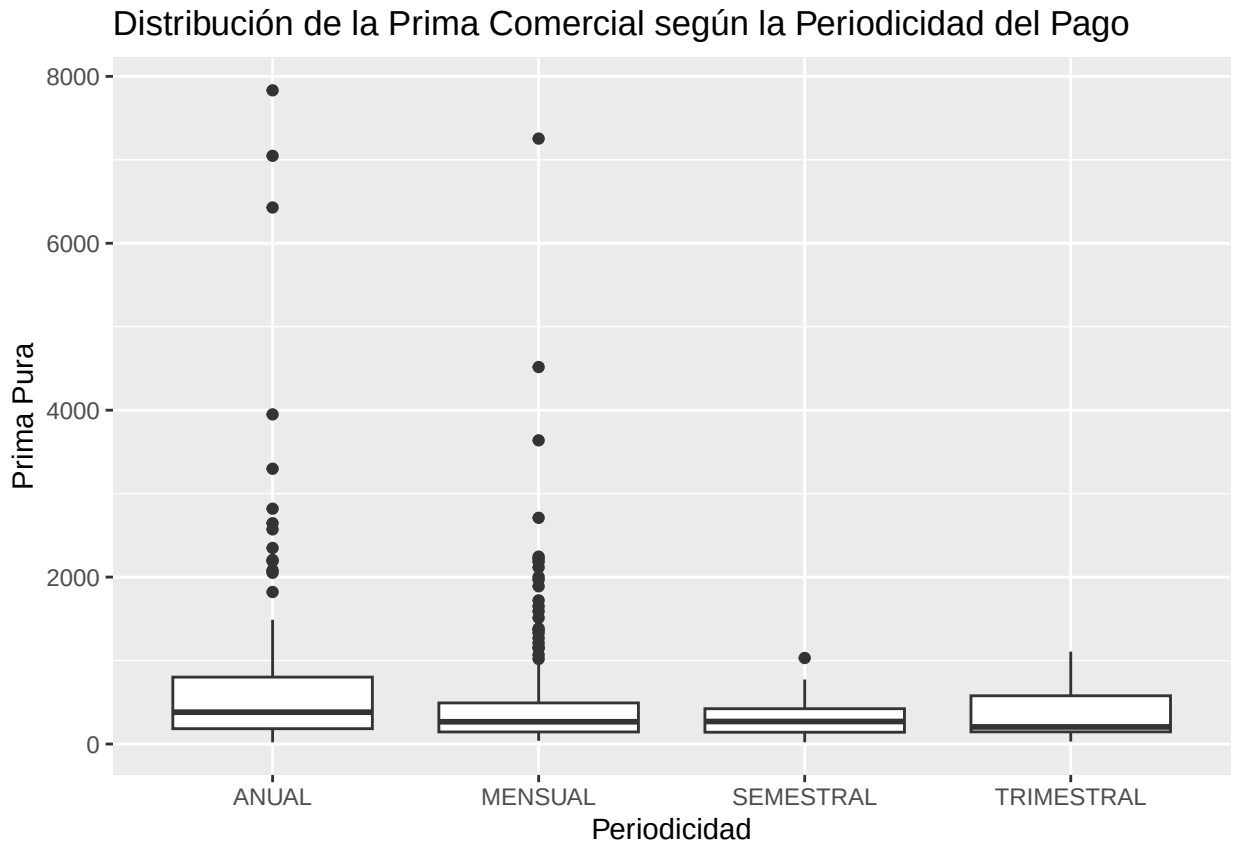
```r
montos <- datos_asegurados[, c(6,10,11,12)]
GGally::ggpairs(montos)
```



Algo a destacar es que, como es de esperar, a mayor detallada y especificada es la prima, es decir, a mayor se acerca la prima a su valor final, más se correlaciona con la suma asegurada.

Finalmente, se puede hacer un boxplot para dimensionar la relación que tienen una variable cuantitativa con una cualitativa. Por ejemplo, la prima comercial con la periodicidad de pago.

```r
ggplot(datos_asegurados, aes(x = `Periodicidad del pago`, y = `Monto de la prima comercial`)) +
  geom_boxplot() +
  labs(title = "Distribución de la Prima Comercial según la Periodicidad del Pago",
       x = "Periodicidad",
       y = "Prima Pura")
```



Vemos que en la periodicidad anual y mensual es donde se dan más valores atípicos fuera de la caja (correspondiente al rango intercuartílico).

Cálculo de las provisiones matemáticas

Creación de Variables

Como primer paso para calcular las provisiones matemáticas, se necesitan tomar en cuenta todas las variables necesarias. Primero, los datos a los que se pueden acceder directamente de la base de datos son: la suma asegurada, la prima comercial y la prima de inventario. Además, se conoce el interés técnico del 5% y los gastos de administración del 16 %. Por lo tanto, para calcular la provisión matemática en el momento t, tV, se necesitan calcular los seguros y anualidades asociadas a esta provisión.

Como primer paso, se podría obtener una columna nueva correspondiente a la edad de la persona en el momento en que inició la póliza. No se contabilizan meses ni días, únicamente los años cumplidos. Es decir, si alguien nació en agosto del año 2000, y el inicio de la póliza es en marzo de 2030, se toma como alguien de 29 años.

```
library(lubridate)
```

```
##
## Adjuntando el paquete: 'lubridate'
```

```
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

```
datos_asegurados$`Edad iniciando la póliza` <- as.integer(interval(datos_asegurados$`Fecha de nacimiento
```

Datos

```
# Establecemos los datos dados
tasa = 0.05
gastos_ad = 0.16
```

```
seguro_temporal <- function(tabla_mort, datos_asegurados, usuario,t){

  temporalidad <- ((as.numeric(format(datos_asegurados$`Fecha de vencimiento de la póliza`[usuario], "%Y

  if (temporalidad < 0){
    return("Se excedio la temporalidad del contrato")
  }

  edad <- as.numeric(datos_asegurados$`Edad iniciando la póliza`[usuario]) + t
  sexo <- datos_asegurados$`Sexo del asegurado`[usuario]

  valor_seguro <-0
  prob_sobrevivencia <- 1

  if (sexo == F){
    for (i in 0:(temporalidad-1)){
      edad_actual <- edad + i
      q_actualizado <- tabla_mort$M_NF[edad_actual + 1]
      valor_seguro <- valor_seguro + (1/(1+tasa))^(i+1)*prob_sobrevivencia*q_actualizado
      prob_sobrevivencia <- prob_sobrevivencia*(1 - q_actualizado)
    }
  }
  else {
    for (i in 0:(temporalidad-1)){
      edad_actual <- edad + i
      q_actualizado <- tabla_mort$H_NF[edad_actual + 1]
      valor_seguro <- valor_seguro + (1/(1+tasa))^(i+1)*prob_sobrevivencia*q_actualizado
      prob_sobrevivencia <- prob_sobrevivencia*(1 - q_actualizado)
    }
  }

  return(valor_seguro)
}
```

```
anualidad_temporal <- function(tabla_mort, datos_asegurados, usuario, t){

  temporalidad <- ((as.numeric(format(datos_asegurados$`Fecha de vencimiento de la póliza`[usuario], "%Y

  if (temporalidad < 0){
    return("Se excedio la temporalidad del contrato")
  }
```

```
  edad <- as.numeric(datos_asegurados$`Edad iniciando la póliza`[usuario]) + t
  sexo <- datos_asegurados$`Sexo del asegurado`[usuario]

  valor_anualidad <-1
  prob_sobrevivencia <- 1

  if (sexo == F){
    for (i in 1:(temporalidad-1)){
      edad_actual <- edad + i - 1
      p_actualizado <- 1 - tabla_mort$M_NF[edad_actual + 1]
      valor_anualidad <-valor_anualidad+(1/(1+tasa))^(i)*prob_sobrevivencia*p_actualizado
      prob_sobrevivencia <- prob_sobrevivencia*p_actualizado
    }
  }
  else {
      for (i in 1:(temporalidad-1)){
        edad_actual <- edad + i - 1
        p_actualizado <- 1 - tabla_mort$H_NF[edad_actual + 1]
        valor_anualidad <- valor_anualidad+(1/(1+tasa))^(i)*prob_sobrevivencia*p_actualizado
        prob_sobrevivencia <- prob_sobrevivencia*p_actualizado
    }
  }

  return(valor_anualidad)
}
```

```
reserva <- function(tabla_mort, datos_asegurados, usuario, t){

  if (t >0){
    t <-t-1  #la evaluación es al 31-12-2023
  }

  if (t >0) {
    if(format(datos_asegurados$`Fecha de inicio de la póliza`[usuario], "%Y") == "2022") {
        t <- t-1 #ya paso un año más
    }
  }


  suma_asegurada <- datos_asegurados$`Suma asegurada`[usuario]
  prima_comercial <- datos_asegurados$`Monto de la prima comercial`[usuario]
  prima_inventario <- datos_asegurados$`Monto de la prima de inventario`[usuario]

  # imprimir los valores antes de la multiplicación
  print(paste("Usuario:", usuario))
  print(paste("Suma asegurada:", suma_asegurada, "Clase:", class(suma_asegurada)))
  print(paste("Prima comercial:", prima_comercial, "Clase:", class(prima_comercial)))
  print(paste("Prima inventario:", prima_inventario, "Clase:", class(prima_inventario)))

  seg_temp <- seguro_temporal(tabla_mort, datos_asegurados, usuario, t)
  anu_temp <- anualidad_temporal(tabla_mort, datos_asegurados, usuario, t)

  print(paste("Seguro temporal:", seg_temp, "Clase:", class(seg_temp)))
```

```
  print(paste("Anualidad temporal:", anu_temp, "Clase:", class(anu_temp)))

  # Validación para evitar errores en la multiplicación
  suma_asegurada <- as.numeric(suma_asegurada)
  prima_comercial <- as.numeric(prima_comercial)
  prima_inventario <- as.numeric(prima_inventario)
  seg_temp <- as.numeric(seg_temp)
  anu_temp <- as.numeric(anu_temp)

  # Verificar si hay NA antes de la multiplicación
  if (any(is.na(c(suma_asegurada, prima_comercial, prima_inventario, seg_temp, anu_temp)))) {
    print("Valores NA en la multiplicación")
    return(NA)
  }

  valor_reserva <- suma_asegurada * seg_temp + anu_temp * (prima_comercial * gastos_ad - prima_inventari

  print("Valor de la reserva:")
  return(valor_reserva)
}
```

Así, se hace la fecha de evalación de la reserva el 31 de diciembre de 2023, tomando $t = 0$:

```
reservas_t0
```

```
## [1] 1680876
```

Interpretación de la Reserva Matemática

A modo general y matemáticamente hablando, la ecuación dada establece que la reserva en el momento t es igual a la suma asegurada multiplicada por el seguro temporal que contabiliza el t en sus cálculos, sumándose los gastos de administración, los cuales se pagan durante la vigencia de la póliza y equivale a un cierto porcentaje de la prima comercial. Lo anterior corresponde a los beneficios en valor presente más los gastos en valor presente que la persona asegurada debe cubrir, es decir significa el pasivo para la aseguradora. Por otro lado, hay que restar el valor presente de las primas, las cuales están sujetas a la sobrevivencia del sujero y se pagan durante la vigencia de la póliza, también contempla el t en sus cálculos y esta se considera como la prima de inventario. Esta parte de la ecuación representa los activos para la aseguradora. Por lo tanto, las reservas se definen como los pasivos de la aseguradora menos los activos de la misma.

Según el Reglamento sobre la solvencia de entidades de seguros y reaseguros 02-13, las reservas o provisiones matemáticas (PM) se calcula como "la diferencia entre el valor actuarial de las obligaciones futuras de la entidad y el valor actual actuarial de las obligaciones futuras del asegurador (primas)". Según esta misma normativa, el cálculo de esta se debe hacer de manera prospectiva. Es importante mencionar, que aunque matemáticamente y actuarialmente una reserva negativa se podría interpretar como que la entidad está obteniendo ganancias, según este mismo reglamento "la provisión matemática en ningún momento puede ser negativa".

$$ {}_tV = (S.A) \cdot A^{1}_{x+t:\overline{n-t}|} + P^{Com} \cdot \%GA \cdot \ddot{a}_{x+t:\overline{n-t}|} - P^{Inv} \cdot \ddot{a}_{x+t:\overline{n-t}|} $$

Figure 1: Fórmula Reserva

En donde ya establecimos que el lado izquierdo es la reserva en el momento t, S.A corresponde a la suma asegurada A es el seguro temporal, según el EDA realizado anteriormente, podemos ver que todos los seguros tienen duraciones de entre 10 a 30 años. Luego, a simboliza la anualidad, P son las primas que ya mencionamos y $G.A$ corresponde a los gastos de administración.

Recordemos que según el reglamento definimos la prima como "Aportación económica que ha de satisfacer el tomador o asegurado a la anteidad aseguradora en concepto de contraprestación por la cobertura del riesgo que ese le ofrece". A su vez, utilizamos la Prima de Inventario, la cual es según este mismo reglamento "el resultado de sumar a la prima pura el recargo para gastos de administración". Y por último la para la prima comercial esta se define como "Resultado de sumar a la prima de inventario los recargos para gastos de adquisición y la utilidad".

Interpolación de la Reserva

La interpolación sucede, según el Reglamento sobre la Solvencia de Entidades de Seguros y Reaseguros, cuando queremos calcular la reserva en un momento intermedio entre las reservas t y $t+1$, para ello usamos la expresión siguiente:

$$ {}_{t+(\frac{h}{T})}V = ({}_{t}V + P)\left(\frac{T-h}{T}\right) + {}_{t+1}V\left(\frac{h}{T}\right) $$

Donde:

${}_{t}V$= Provisión matemática en el año póliza t

P = prima de inventario

T = 365

h = número de días transcurridos desde el último aniversario de la póliza hasta el momento de la valuación.

Modificamos la fórmula de la reserva para obtener el valor directamente, pasando por alto los print anteriores.

```
reserva_nueva <- function(tabla_mort, datos_asegurados, usuario, t){

  if (t >0){
    t <-t-1
  }

  if (t >0) {
    if(format(datos_asegurados$`Fecha de inicio de la póliza`[usuario], "%Y") == "2022") {
        t <- t-1
    }
  }

  suma_asegurada <- datos_asegurados$`Suma asegurada`[usuario]
  prima_comercial <- datos_asegurados$`Monto de la prima comercial`[usuario]
  prima_inventario <- datos_asegurados$`Monto de la prima de inventario`[usuario]

  seg_temp <- seguro_temporal(tabla_mort, datos_asegurados, usuario, t)
  anu_temp <- anualidad_temporal(tabla_mort, datos_asegurados, usuario, t)
```

```
  suma_asegurada <- as.numeric(suma_asegurada)
  prima_comercial <- as.numeric(prima_comercial)
  prima_inventario <- as.numeric(prima_inventario)
  seg_temp <- as.numeric(seg_temp)
  anu_temp <- as.numeric(anu_temp)

  if (any(is.na(c(suma_asegurada, prima_comercial, prima_inventario, seg_temp, anu_temp)))) {
    return(NA)
  }

  valor_reserva <- suma_asegurada * seg_temp + anu_temp * (prima_comercial * gastos_ad - prima_inventari

  return(valor_reserva)
}
```

Una vez implementada la función anterior, la fórmula de interpolación es solo de implementar como sigue:

```
reserva_interpolada <- function(tabla_mort, datos_asegurados, usuario, t, h){
  # Obtenemos la reserva matemática en t y t+1
  reserva_t <- reserva_nueva(tabla_mort, datos_asegurados, usuario, t)
  reserva_t1 <- reserva_nueva(tabla_mort, datos_asegurados, usuario, t + 1)

  prima_inventario <- as.numeric(datos_asegurados$`Monto de la prima de inventario`[usuario])

  T <- 365
  # Calculamos el valor de la reserva interpolada
  reserva_interp <- ( (reserva_t + prima_inventario) * ((T - h) / T) ) + (reserva_t1 * (h / T))

  return(reserva_interp)
}
```