Modelo para **A**juste del **TI**empo de **P**rescripción de **I**ncapacidades **C**on **O**utliers (**ATIPICO**)

Rodrigo Zepeda-Tello

Versión del 15/08/2021

Código en Github

Ir a https://github.com/RodrigoZepeda/ATIPICO

Descripción del problema

Objetivo

> Se tiene una base con el tiempo de incapacidad temporal en el trabajo que el equipo médico ha prescrito a los pacientes. Se desea modelar la distribución del tiempo de recuperación controlando por sexo y edad de los pacientes.

Consideraciones

El tiempo que prescriben las y los médicos para incapacidad suele estar redondeado a múltiplos de semana con los valores típicos siendo de 1 a 3 días, luego 7, 14 y 21. Esto no necesariamente está asociado a una razón biológica sino a convenciones sociales (el calendario). Más aún, una proporción pequeña de los datos (estimado $< 10\%$) contiene valores atípicos que pueden referir a errores de registro en la base, errores en el diagnóstico de la enfermedad o casos excepcionales donde por algún motivo la recuperación del paciente toma una cantidad distinta de tiempo.

> El objetivo entonces es reconstruir la distribución original de tiempo de convalecencia por una incapacidad considerando que los datos están fuertemente sesgados a múltiplos semanales y contienen un porcentaje de valores atípicos.

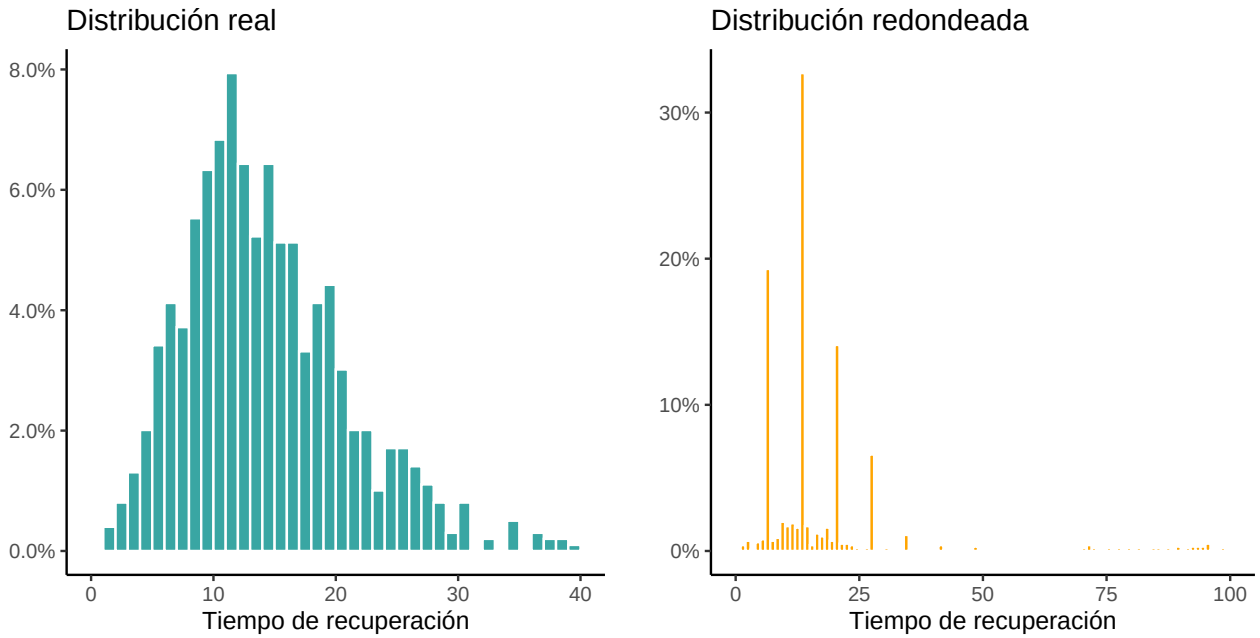
Métodos

La siguientes secciones exponen los métodos. En primer lugar se explica en formato de divulgación la metodología; luego se presenta una versión simplificada del método aplicado y finalmente se describe el método utilizado.

Idea del método

El modelo considera que todos los pacientes tienen un tiempo de incapacidad que sigue una distribución continua. Un paciente puede recuperarse tanto en el día 14 como en el 1.3 ó el 6.5 significando el primero una recuperación en la mañana del primer día y el segundo que la persona se sintió bien a la mitad de su sexto día. Se supone que dichos días son redondeados para su prescripción de manera artificial a números enteros (para prescribir en días) y, en cierta proporción (no todos), a múltiplos de 7 (para prescribir en semanas).

La imagen siguiente presenta el tiempo de recuperación real (gráfica izquierda) así como el tiempo de recuperación una vez una proporción se redondea en múltiplos de 7 (gráfica derecha):



En los datos observados (gráfica derecha) se modificaron, además, al 5% de los valores a fin de generar observaciones atípicas de ahí que la gráfica amarilla aparezca con un mayor sesgo a la derecha y valores cercanos a 100. Esto para reflejar de mejor mandera los datos reales.

El modelo funciona como sigue: para cada padecimiento, grupo de edad y sexo, se supone que una proporción (desconocida) de los datos corresponderá a atípicos. Para cada uno de los datos se analiza *qué es lo más probable*: que el valor sea atípico o que corresponda a los datos reales. Esto permite generar dos bases distintas: las de los probablemente atípicos y la de los probablemente reales. Sobre la base de aquellos clasificados como probablemente reales se busca entonces la mejor distribución de probabilidad que, al momento de muestrear de ella y redondear los resultados, se obtengan observaciones similares a los datos observados. Este proceso se repite múltiples veces para garantizar que los resultados no se deban sólo al azar.

De manera resumida el proceso está dado por:

1. Clasificar (de manera aleatoria) algunos de los datos como atípicos.
2. Ajustar un modelo de probabilidad a los datos restantes de tal forma que, al redondear de dicho modelo, se obtengan los resultados más parecidos posibles a lo observado.
3. Repetir.

La siguiente gráfica, resultante de simulaciones (ver más abajo), muestra bajo una base de datos sintética cómo se ve la verdadera distribución de los datos (*real*), cómo resultan los datos observados a partir de ella (*redondeado*) y cómo ajusta el modelo resultante de los pasos anteriores (*modelo*). Dicha base de datos sintética contiene además un 5% valores atípicos en la derecha que el modelo, adecuadamente, decide no ajustar.

El modelo que aquí se presenta es una adaptación del modelo de variables latentes para redondeo de Gelman *et al* (CITAR).

Ajuste simple

> Ésta es una explicación más sencilla del ajuste que el modelo completo pues no considera edad ni sexo.

Para cada paciente i, sea T_i^R el tiempo prescrito por el médico ya con redondeo. Dicho tiempo está en función

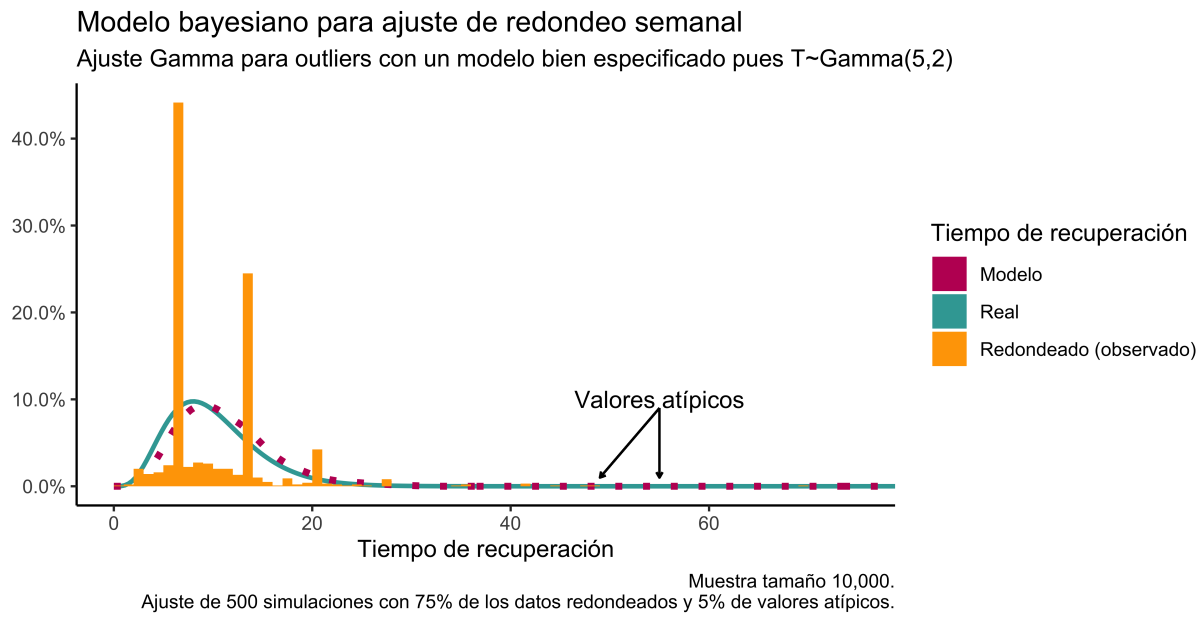


Figure 1: Distribución redondeada con valores atípicos. Comparativo con modelo original y modelo estimado.

del verdadero tiempo de recuperación (desconocido), T_i:

$$T_i^R = T_i + \epsilon_i$$

donde ϵ_i es el error de redondeo. Si $\epsilon_i = 0$ no hay redondeo en los datos y los observados corresponden a los teóricos. Una cierta proporción de los tiempos observados son en realidad valores atípicos. A estos, los denotamos T_i^A y suponemos que:

$$T_i^A = \mathcal{O}_i + \zeta_i$$

donde $\mathcal{O}_i$ es la distribución de los atípicos y ζ_i su error asociado.

Para cada paciente se observa un valor en la base T_i^{obs}, el cual, con probabilidad $1-\theta$ es un valor atípico:

$$T_i^{obs} = \begin{cases} T_i^R & \text{con probabilidad } \theta, \\ T_i^A & \text{con probabilidad } 1-\theta. \end{cases}$$

El ajuste se realiza de manera bayesiana donde se supone:

$$\begin{aligned} \theta &\sim \text{Beta}(0.25, 1) \\ T_i|\alpha_R, \beta_R &\sim \text{Gamma}(\alpha_R, \beta_R), \\ \mathcal{O}_i|\alpha_A, \beta_A &\sim \text{Gamma}(\alpha_A, \beta_A), \end{aligned} \tag{1}$$

donde $\alpha_k, \beta_j \sim \text{HalfCauchy}(0, 2.5)$ para $k, j \in \{R, A\}$. Se tienen además las siguientes identidades que relacionan los datos observados con el modelo teórico:

$$\begin{aligned} T_i &= T_i^R + \epsilon_i, \\ \mathcal{O}_i &= T_i^A + \zeta_i. \end{aligned} \tag{2}$$

donde $\epsilon_i \sim \text{Uniforme}(-3.5, 3.5)$ es el error de redondeo semanal y $\zeta_i \sim \text{Uniforme}(0, 1000)$ es el error correspondiente a un valor atípico.

El modelo puede ser ajustado en `Stan` de la siguiente forma:

```
data {
  int<lower=0> N;
  vector[N] Tobs;
}

parameters {
  real<lower=0> alpha[2];
  real<lower=0> beta[2];
  real<lower=0, upper=1> theta;
  vector<lower=-3.5, upper=3.5>[N] error1;
  vector<lower=0, upper=1000>[N] error2;
}

transformed parameters {
  vector[N] Treal;
  vector[N] Outlier;
  Treal   = Tobs + error1;
  Outlier = Tobs + error2;
}

model {
  alpha    ~ cauchy(0, 2.5);
  beta     ~ cauchy(0, 2.5);
  error1   ~ uniform(-3.5, 3.5);
  error2   ~ uniform(0, 1000);
  theta    ~ beta(0.25,1);
  for (n in 1:N)
  target += log_mix(theta,
                    gamma_lpdf(Treal[n] | alpha[1], beta[1]),
                    gamma_lpdf(Outlier[n] | alpha[2], beta[2]));
}

generated quantities {
  real Tpred       = gamma_rng(alpha[1], beta[1]);
  real Outlierpred = gamma_rng(alpha[2], beta[2]);
}
```

La siguiente gráfica muestra que incluso si especificamos de manera incorrecta la distribución a priori de T_i^R tomándola como Weibull en lugar de Gamma, el modelo de todas maneras ajusta correctamente si n es relativamente grande:

Código de `R` para generar datos que sigan esta distribución y validar el modelo puede encontrarse en el apéndice.

Ajuste controlando por edad, sexo y enfermedad.

Para el ajuste por sexo y edad, para cada enfermedad se establece una regresión donde la media de tiempo de recuperación (μ_R) es de la forma:

$$\mu_R = \nu_R + \gamma_R \cdot \text{Edad}_i + \eta_R \cdot \text{Sexo}_i,$$

mientras que la media de la distribución de valores atípicos está dada por

$$\mu_A = \nu_A + \gamma_A \cdot \text{Edad}_i + \eta_A \cdot \text{Sexo}_i.$$

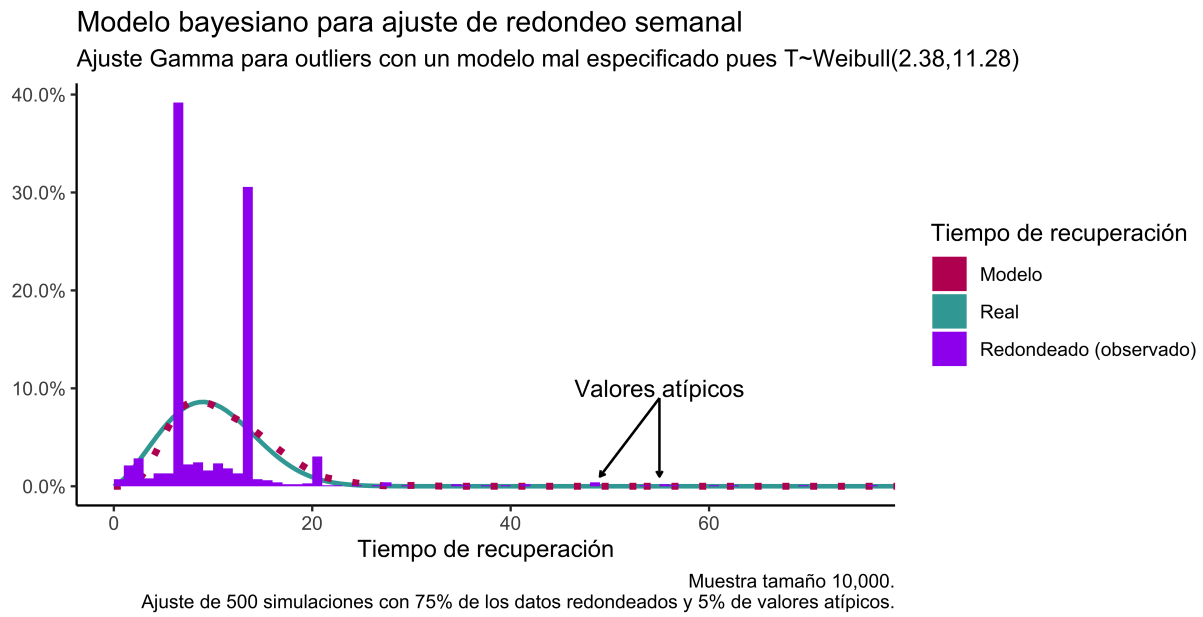


Figure 2: Distribución redondeada con valores atípicos especificando de manera incorrecta la distribución a priori. Comparativo con modelo original y modelo estimado.

En ambos casos se tiene la parametrización anterior:

$$\begin{aligned} T_i|\alpha_R, \beta_R &\sim \text{Gamma}(\alpha_R, \beta_R), \\ \mathcal{O}_i|\alpha_A, \beta_A &\sim \text{Gamma}(\alpha_A, \beta_A), \end{aligned} \tag{3}$$

donde

$$\alpha_j = \mu_j/\beta_j$$

para $j \in \{A, R\}$. Las distribuciones *a priori* de los nuevos parámetros son:

$$\begin{aligned} \nu_j, \beta_j &\sim \text{HalfCauchy}(0, 2.5), \\ \gamma_j, \eta_j &\sim \text{Normal}(0, 10000). \end{aligned} \tag{4}$$

para $k, j \in \{R, A\}$.

El modelo puede ser ajustado en `Stan` de la siguiente forma:

```
data {
  int<lower=0> N;
  vector[N] Tobs;
  vector[N] Edad;
  vector[N] Sexo;

  //Edades a interpolar en el resultado para reporte
  int<lower=0> M;
  vector[M] EdadesResultado;
}

parameters {
  vector[2] gamma;
  vector[2] eta;
  vector<lower=0>[2] nu;
```

```
  vector<lower=0>[2] beta;

  //Probability of outlier
  real<lower=0, upper=1> theta;

  //Roundong and outlier error as params (latent variables)
  vector<lower=-3.5, upper=3.5>[N] error1;
  vector<lower=0, upper=1000>[N] error2;
}

transformed parameters {
  vector[N] Treal;
  vector[N] Outlier;
  vector[N] mu_real;
  vector[N] mu_outlier;

  //Parámetro de medias de la regresión
  mu_real    = nu[1] + gamma[1]*Edad + eta[1]* Sexo;
  mu_outlier = nu[2] + gamma[2]*Edad + eta[2]* Sexo;

  //Datos con redondeo en función de los observados
  Treal   = Tobs + error1;
  Outlier = Tobs + error2;
}

model {
  nu       ~ cauchy(0, 2.5);
  beta     ~ cauchy(0, 2.5);
  gamma    ~ normal(0, 1000);
  eta      ~ normal(0, 1000);
  error1   ~ uniform(-3.5, 3.5);
  error2   ~ uniform(0, 1000);
  theta    ~ beta(0.25,1);
  for (n in 1:N)
  target += log_mix(theta,
                    gamma_lpdf(Treal[n] | mu_real[n] / beta[1], beta[1]),
                    gamma_lpdf(Outlier[n] | mu_outlier[n] / beta[2], beta[2]));
}

generated quantities {

  vector[M] Tpred_0;
  vector[M] Tpred_1;
  vector[M] Outlierpred_0;
  vector[M] Outlierpred_1;
  real mu_real_pred_0;
  real mu_real_pred_1;
  real mu_outlier_pred_0;
  real mu_outlier_pred_1;

  for (m in 1:M){
    //Sims
    mu_real_pred_0    = nu[1] + gamma[1]*EdadesResultado[m];
    mu_real_pred_1    = nu[1] + gamma[1]*EdadesResultado[m] + eta[1];
```

```
    mu_outlier_pred_0 = nu[2] + gamma[2]*EdadesResultado[m];
    mu_outlier_pred_1 = nu[2] + gamma[2]*EdadesResultado[m] + eta[2];

    Tpred_0[m]         = gamma_rng(mu_real_pred_0 ./ beta[1], beta[1]);
    Tpred_1[m]         = gamma_rng(mu_real_pred_1 ./ beta[1], beta[1]);
    Outlierpred_0[m]   = gamma_rng(mu_outlier_pred_0 ./ beta[2], beta[2]);
    Outlierpred_1[m]   = gamma_rng(mu_outlier_pred_1 ./ beta[2], beta[2]);
  }
}
```

El ajuste con datos redondeados se ve de la siguiente manera. Cabe recordar que el modelo opera conociendo sólo los datos que provienen del redondeo y desconoce la distribución real.

El código de `R` para generar datos que sigan esta distribución y validar el modelo puede encontrarse en el apéndice.

Apéndice

Código de R para generar ajuste simple

El siguiente código genera datos que se comportan de acuerdo al modelo y ajusta una distribución. Si la distribución de los datos coincide con la distribución *a priori* del modelo entonces el ajuste es perfecto. Si la distribución no coincide de manera completa (por ejemplo teniendo datos Weibull y un modelo Gamma) el ajuste de todas formas es muy bueno:

```
#Script para simular la estructura de datos y verificar
#si el modelo funciona para recuperarlos
rm(list = ls())

library(tidyverse)
library(cmdstanr)
library(scales)
library(posterior)
library(kdensity)
library(bayesplot)
library(mixdist)

set.seed(23476785)
stan_seed <- 23749
chains = 4; iter_warmup = 250; nsim = 500; pchains = 4;
threads_per_chain = 4; threads = 8; iter_variational = 10000
method = "variational" #faster (less accurate option) = "variational"

#Flags for my compiler faster
compiler_path_cxx <- "/usr/local/opt/llvm/bin/clang++"
options(mc.cores = parallel::detectCores())

#Simulation distribution
#gamma para que el modelo esté bien especificado;
#weibull para que no esté bien especificado
simdist <- "gamma" #weibull ó gamma
media    <- 10
varianza <- 20

#No todos los datos están redondeados así que se establece un porcentaje
#de cuántos tienen redondeo
```

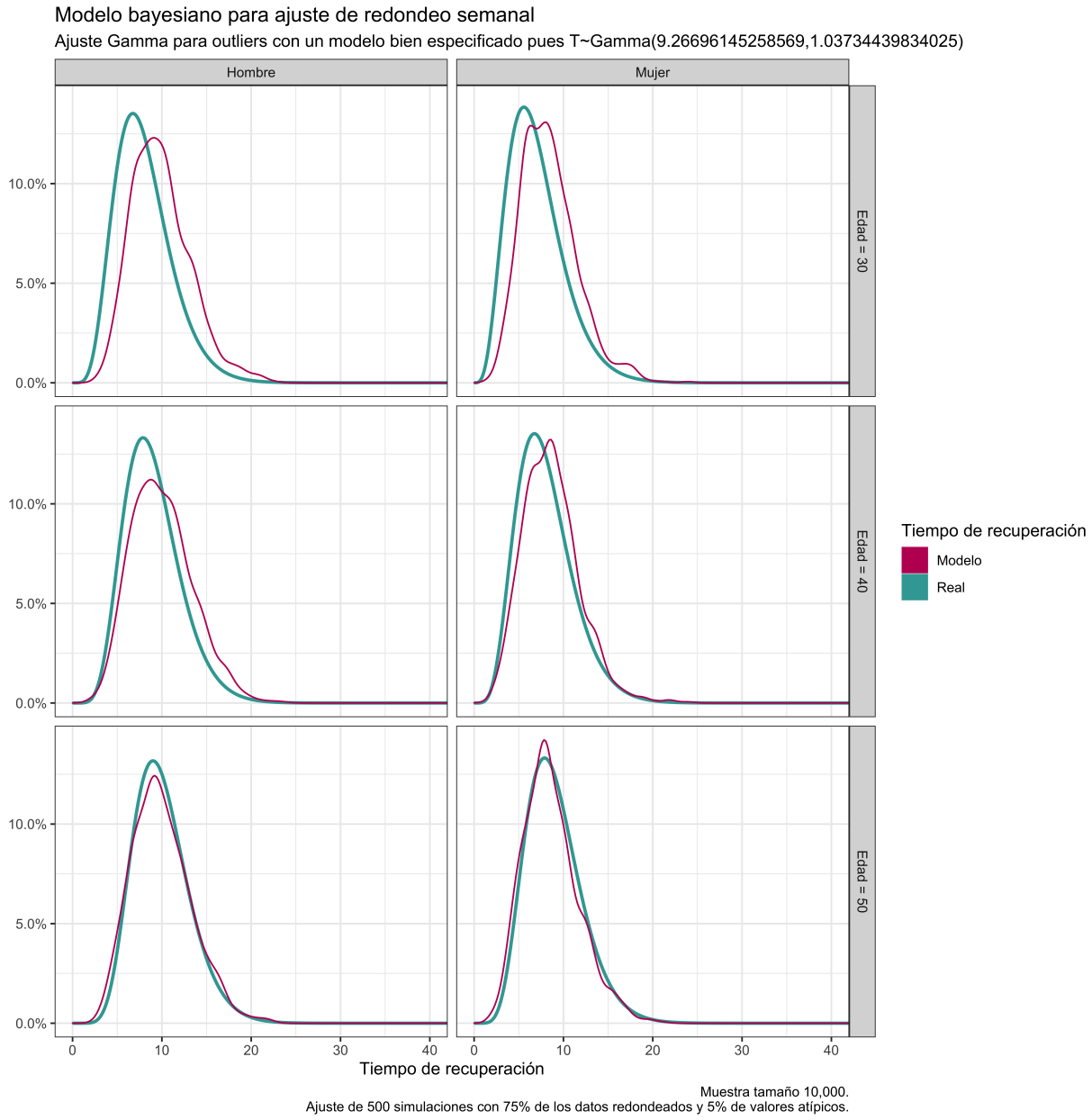


Figure 3: Ajuste de datos redondeados mediante modelo Gamma bayesiano controlando por sexo y edad. Comparativo con modelo original y modelo estimado usando sólo datos redondeados a múltiplos de 7.

```r
perc_redondeado <- 0.75  #Porcentaje de datos redondeados a 7
perc_outliers   <- 0.05  #Se agrega el 1% de outliers
nsims           <- 10000 #Número de simulaciones para el modelo

#Generamos las simulaciones:
if (simdist == "gamma"){
  message("Distribución Gamma")
  scale        <- varianza/media  %>% round(.,2)
  shape        <- media/scale     %>% round(.,2)
  col_redondeo <- "orange"
  tipo_modelo  <- paste0("bien especificado pues T~Gamma(",
                         shape, ",", scale,")")
  sample_dist  <- function(){rgamma(nsims, shape = shape, scale = scale)}
  dist_fun     <- function(x){dgamma(x, shape = shape, scale = scale)}
} else if (simdist == "weibull"){
  message("Distribución Weibull")
  params       <- weibullpar(media, sqrt(varianza))
  shape        <- params["shape"] %>% as.numeric() %>% round(.,2)
  scale        <- params["scale"] %>% as.numeric() %>% round(.,2)
  col_redondeo <- "purple"
  tipo_modelo  <- paste0("mal especificado pues T~Weibull(",
                         shape, ",", scale,")")
  sample_dist  <- function(x){rweibull(nsims, shape = shape, scale = scale)}
  dist_fun     <- function(x){dweibull(x, shape = shape, scale = scale)}
} else {
  stop("Distribución inválida selecciona 'gamma' o 'weibull'.")
}

#Simulamos los verdaderos datos filtrados como días (enteros)
datos_distribucion <- sample_dist()
datos_distribucion <- round(datos_distribucion,0)

#Agregamos outliers
outliers_id <- sample(1:nsims, ceiling(perc_outliers*nsims))
outliers    <- exp(rnorm(ceiling(perc_outliers*nsims), log(50), 1))
datos_distribucion[outliers_id] <- outliers

#Redondeamos la mayoría de los datos a múltiplos de 7 a partir de 3
redondear   <- datos_distribucion[datos_distribucion > 3]
a_redondear <- sample(ceiling(perc_redondeado*length(redondear)))
redondear[a_redondear]                    <- round(redondear[a_redondear]/7)*7
datos_distribucion[datos_distribucion > 3] <- redondear

#Modelo
message("Fitting model. Go grab a coffee this will take A LOT")
if (!is.null(compiler_path_cxx)){
  cpp_options <- list(cxx_flags = "-O3 -march=native",
                      cxx = compiler_path_cxx, stan_threads = TRUE)
} else {
  cpp_options <- list(cxx_flags = "-O3 -march=native",
                      stan_threads = TRUE)
}
```

```r
model_gamma_v1 <- cmdstan_model("models/Modelo_Gamma_Outliers.stan",
                                cpp_options = cpp_options)

datos <- list(N = length(datos_distribucion), Tobs = datos_distribucion)

initf2 <- function(chain_id = 1) {
  list(error1       = runif(length(datos_distribucion), 0, 3.5),
       error2       = runif(length(datos_distribucion), 0, 1000),
       alpha        = rnorm(2, 2.5, 1) %>% abs(),
       beta         = rnorm(2, 2.5, 1) %>% abs(),
       theta        = runif(1,0,1)
  )}
init_ll <- lapply(1:chains, function(id) initf2(chain_id = id))

if (!dir.exists("cmdstan")){dir.create("cmdstan")}
if (method == "HMC"){
  model_sample <- model_gamma_v1$sample(data = datos,
                                        chains = chains,
                                        seed = stan_seed,
                                        iter_warmup = iter_warmup,
                                        adapt_delta = 0.95,
                                        iter_sampling = nsim - iter_warmup,
                                        init = initf2,
                                        output_dir = "cmdstan",
                                        max_treedepth = 2^(11),
                                        threads_per_chain = threads_per_chain)
} else if (method == "variational"){
  model_sample <- model_gamma_v1$variational(data = datos,
                                        seed = stan_seed,
                                        iter= iter_variational,
                                        init = initf2,
                                        threads = threads,
                                        output_dir = "cmdstan")
} else {
  message(paste0("Method ", method, " not found. Try 'HMC' or 'variational'"))
}

# Herramienta de diagnósitco para verificar el ajuste
#model_sample$cmdstan_diagnose()

#Obtenemos la distribución posterior
ppdist    <- model_sample$draws(variables="Tpred") %>% as_draws_df()
prop_true <- model_sample$draws(variables="theta") %>% summarise_draws()

#Ajustamos densidades para graficar
#Si tenemos demasiados datos reducimos para la gráfica
if (nsims > 1000){datos_distribucion <- sample(datos_distribucion, 1000)}
x             <- seq(0, max(datos_distribucion), length.out = 1000)
densidad_real <- function(x){dist_fun(x)}
densidad_obs  <- kdensity(datos_distribucion, kernel = "gaussian")
densidad_pred <- kdensity(ppdist$Tpred + 0.01, start = "gamma",
                          kernel = "gaussian", support = c(0, Inf))
```

```r
#Gráfica de los datos
data.frame(x = x, Real = densidad_real(x), Observada = densidad_obs(x),
           Predicha = densidad_pred(x)) %>%
ggplot() +
  geom_line(aes(x = x, y = Real, color = "Real"), size = 1) +
  geom_line(aes(x = x, y = Predicha, color = "Modelo"), size = 1.5,
            linetype = "dotted") +
  geom_histogram(aes(x = x, y = ..density.., fill = "Redondeado (observado)"),
                 breaks = seq(0,100, by = 1),
                 data = data.frame(x = datos_distribucion)) +
  annotate("text", x = 55, y = 0.1, label = "Valores atípicos") +
  geom_segment(aes(x = 55, y = 0.09, xend = 55, yend = 0.01),
               arrow = arrow(length = unit(0.1, "cm"))) +
  geom_segment(aes(x = 55, y = 0.09, xend = 49, yend = 0.01),
               arrow = arrow(length = unit(0.1, "cm"))) +
  theme_classic() +
  scale_color_manual("Tiempo de recuperación",
                     values = c("Modelo" = "#BF1363",
                                "Real" = "#39A6A3",
                                "Redondeado (observado)" = col_redondeo)) +
  scale_fill_manual("Tiempo de recuperación",
                    values = c("Modelo" = "#BF1363",
                               "Real" = "#39A6A3",
                               "Redondeado (observado)" = col_redondeo)) +
  labs(
    x = "Tiempo de recuperación",
    y = "",
    title = "Modelo bayesiano para ajuste de redondeo semanal",
    subtitle = paste0("Ajuste Gamma para outliers con un modelo ", tipo_modelo),
    caption = paste0("Muestra tamaño ", comma(nsims), ".\nAjuste de ",
                     comma(nsim), " simulaciones con ",
                     percent(perc_redondeado), " de los datos redondeados y ",
                     percent(perc_outliers), " de valores atípicos.")
  ) +
  coord_cartesian(xlim = c(0, 75)) +
  scale_y_continuous(labels = scales::percent)
ggsave(paste0("images/Atipicos",simdist,".png"), width = 8,
       height = 4, dpi = 750)
```

Código de R para generar ajuste por edad y sexo

```r
#Script para simular la estructura de datos y verificar
#si el modelo funciona para recuperarlos
rm(list = ls())

library(tidyverse)
library(cmdstanr)
library(scales)
library(posterior)
library(kdensity)
library(bayesplot)
library(mixdist)
```

```r
set.seed(23476785)
stan_seed <- 23749
chains = 4; iter_warmup = 250; nsim = 500; pchains = 4;
threads_per_chain = 4; threads = 12; iter_variational = 50000;
adapt_iter = 1000;
method = "variational" #faster (less accurate option) = "variational"

#Flags for my compiler faster
compiler_path_cxx <- "/usr/local/opt/llvm/bin/clang++"
options(mc.cores = parallel::detectCores())

#Simulation distribution
#gamma para que el modelo esté bien especificado;
#weibull para que no esté bien especificado
simdist <- "gamma" #weibull ó gamma
media_baseline <- 5
varianza       <- 10
gamma_edad     <- 0.1
beta_sexo      <- -1

#Edades a interpolar
edades_interpol <- c(30,40,50)


#No todos los datos están redondeados así que se establece un porcentaje
#de cuántos tienen redondeo
perc_redondeado <- 0.75  #Porcentaje de datos redondeados a 7
perc_outliers   <- 0.05  #Se agrega el 1% de outliers
nsims           <- 10000 #Número de simulaciones para el modelo

#Simulamos las edades
edades <- rnorm(nsims, mean = 40, sd = 10)
edades[(edades < 18 | edades > 70)] <- runif(sum((edades < 18 | edades > 70)), 20, 70)

#Simulamos el sexo
psexo  <- 0.4
sexo   <- sample(c(0,1), nsims, replace = T, prob = c(psexo, 1-psexo))

#Generamos los parámetros
genera_media <- function(edad, sexo, sims = nsims, random = T){
  if (random){
    mu <- media_baseline + rnorm(sims, gamma_edad, 0.1)*edad + rnorm(sims, beta_sexo, 0.1)*sexo
    mu[mu < 0] <- media_baseline
  } else {
    mu <- media_baseline + gamma_edad*edad + beta_sexo*sexo
    mu[mu < 0] <- media_baseline
  }
  return(mu)
}
media <- genera_media(edades, sexo)

if (min(media < 0)){
  stop("Distribución no está bien especificada la media debe ser > 0")
```

```
}

#Generamos las simulaciones:
if (simdist == "gamma"){
  message("Distribución Gamma")
  scale        <- varianza/media  %>% round(.,2)
  shape        <- media/scale     %>% round(.,2)
  col_redondeo <- "orange"
  tipo_modelo  <- paste0("bien especificado pues T~Gamma(",
                         shape, ",", scale,")")
  sample_dist  <- function(){
    rgamma(nsims, shape = shape, scale = scale)
  }
  dist_fun     <- function(x, edad, sexo){
    scale <- varianza/genera_media(edad, sexo, 1, random = F)
    shape <- genera_media(edad, sexo, 1, random = F)/scale
    dgamma(x, shape = shape, scale = scale)
  }
} else if (simdist == "weibull"){
  message("Distribución Weibull")
  params       <- weibullpar(media, sqrt(varianza))
  shape        <- params["shape"]  %>% round(.,2)
  scale        <- params["scale"]  %>% round(.,2)
  col_redondeo <- "purple"
  tipo_modelo  <- paste0("mal especificado pues T~Weibull")
  sample_dist  <- function(x){
    rweibull(nsims, shape = unlist(shape), scale =  unlist(scale))
  }
  dist_fun     <- function(x, edad, sexo){
    params       <- weibullpar(genera_media(edad,sexo, 1, random = F), sqrt(varianza))
    shape        <- params["shape"] %>% round(.,2)
    scale        <- params["scale"] %>% round(.,2)
    dweibull(x, shape = unlist(shape), scale = unlist(scale))
  }
} else {
  stop("Distribución inválida selecciona 'gamma' o 'weibull'.")
}


#Simulamos los verdaderos datos filtrados como días (enteros)
datos_distribucion <- sample_dist()
datos_distribucion <- round(datos_distribucion,0)

#Agregamos outliers
outliers_id <- sample(1:nsims, ceiling(perc_outliers*nsims))
outliers    <- exp(rnorm(ceiling(perc_outliers*nsims), log(50), 1))
datos_distribucion[outliers_id] <- outliers

#Redondeamos la mayoría de los datos a múltiplos de 7 a partir de 3
redondear   <- datos_distribucion[datos_distribucion > 3]
a_redondear <- sample(ceiling(perc_redondeado*length(redondear)))
redondear[a_redondear]                    <- round(redondear[a_redondear]/7)*7
datos_distribucion[datos_distribucion > 3] <- redondear
```

```r
#Modelo
message("Fitting model. Go grab a coffee this will take A LOT")
if (!is.null(compiler_path_cxx)){
  cpp_options <- list(cxx_flags = "-O3 -march=native",
                      cxx = compiler_path_cxx, stan_threads = TRUE)
} else {
  cpp_options <- list(cxx_flags = "-O3 -march=native",
                      stan_threads = TRUE)
}

model_gamma_v1 <- cmdstan_model("models/Modelo_Gamma_Outliers_Edad.stan",
                                cpp_options = cpp_options)

datos <- list(N = length(datos_distribucion),
              Tobs = datos_distribucion, Edad = edades, Sexo = sexo,
              EdadesResultado = edades_interpol, M = length(edades_interpol))

initf2 <- function(chain_id = 1) {
  list(error1       = runif(length(datos_distribucion), 0, 3.5),
       error2       = runif(length(datos_distribucion), 0, 1000),
       beta         = (rnorm(2, 2.5, 1) %>% abs()) + 1,
       gamma        = rnorm(2, 0, 0.1),
       eta          = rnorm(2, 0, 0.1),
       nu           = rnorm(2, 100, 0.1) %>% abs(),
       theta        = runif(1,0,1)
  )}

if (!dir.exists("cmdstan")){dir.create("cmdstan")}
if (method == "HMC"){
  model_sample <- model_gamma_v1$sample(data = datos,
                                        chains = chains,
                                        seed = stan_seed,
                                        iter_warmup = iter_warmup,
                                        adapt_delta = 0.95,
                                        iter_sampling = nsim - iter_warmup,
                                        init = initf2,
                                        output_dir = "cmdstan",
                                        max_treedepth = 2^(11),
                                        threads_per_chain = threads_per_chain)
} else if (method == "variational"){
  model_sample <- model_gamma_v1$variational(data = datos,
                                             seed = stan_seed,
                                             iter= iter_variational,
                                             init = initf2,
                                             adapt_iter = adapt_iter,
                                             adapt_engaged = T,
                                             output_samples = 1000,
                                             threads = threads,
                                             output_dir = "cmdstan")
} else {
  message(paste0("Method ", method, " not found. Try 'HMC' or 'variational'"))
}
```

```r
# Herramienta de diagnósitco para verificar el ajuste
#model_sample$cmdstan_diagnose()

#Obtenemos la distribución posterior
ppdist_0  <- model_sample$draws(variables="Tpred_0") %>% as_draws_df()
ppdist_1  <- model_sample$draws(variables="Tpred_1") %>% as_draws_df()
ppdist_0  <- ppdist_0 %>% select(-starts_with("."))
ppdist_1  <- ppdist_1 %>% select(-starts_with("."))
colnames(ppdist_0) <- edades_interpol
colnames(ppdist_1) <- edades_interpol
ppdist_0  <- ppdist_0 %>% mutate("Sexo" = "Hombre")
ppdist_1  <- ppdist_1 %>% mutate("Sexo" = "Mujer")
ppdist    <- rbind(ppdist_0, ppdist_1)
ppdist    <- ppdist %>%
  pivot_longer(cols =  as.character(edades_interpol),
               values_to = "Tiempo", names_to = "Edad")
ppdist    <- ppdist %>% mutate(Edad = paste0("Edad = ", Edad))

#Proporción estimada de outliers
prop_true <- model_sample$draws(variables="theta") %>% summarise_draws()

#Ajustamos densidades para graficar
#Si tenemos demasiados datos reducimos para la gráfica
x             <- seq(0, 100, length.out = 1000)
for (edad in edades_interpol){
  for (sexo in c(0,1)){
    distribucion <- dist_fun(x, edad, sexo)
    if (edad == edades_interpol[1] & sexo == 0){
      datos_dist <- data.frame(Edad = paste0("Edad = ",edad), Sexo = "Hombre",
                               Tiempo = distribucion, x = x)
    } else {
      if (sexo == 0){sexname = "Hombre"}else{sexname="Mujer"}
      datos_dist <- data.frame(Edad = paste0("Edad = ",edad), Sexo = sexname,
                               Tiempo = distribucion, x = x) %>%
        bind_rows(datos_dist)
    }
  }
}

#Gráfica de los datos
ggplot(datos_dist) +
  geom_line(aes(x = x, y = Tiempo, color = "Real"), size = 1) +
  geom_density(aes(x = Tiempo, color = "Modelo"), data = ppdist) +
  coord_cartesian(xlim = c(0, 40)) +
  facet_grid(Edad ~ Sexo) +
  theme_bw() +
  scale_color_manual("Tiempo de recuperación",
                     values = c("Modelo" = "#BF1363",
                                "Real" = "#39A6A3")) +
  scale_fill_manual("Tiempo de recuperación",
                    values = c("Modelo" = "#BF1363",
                               "Real" = "#39A6A3")) +
  labs(
```

```
    x = "Tiempo de recuperación",
    y = "",
    title = "Modelo bayesiano para ajuste de redondeo semanal",
    subtitle = paste0("Ajuste Gamma para outliers con un modelo ", tipo_modelo),
    caption = paste0("Muestra tamaño ", comma(nsims), ".\nAjuste de ",
                     comma(nsim), " simulaciones con ",
                     percent(perc_redondeado), " de los datos redondeados y ",
                     percent(perc_outliers), " de valores atípicos.")
  ) +
  scale_y_continuous(labels = scales::percent)
ggsave(paste0("Atipicos_edad_",simdist,".png"), width = 10, height = 10, dpi = 750)
```