Examen Parcial

Sara Luz Valenzuela Camacho (CU 204535)

Examen Parcial

Bibliotecas

```
library(tidyverse)
library(cmdstanr)
library(dplyr)
library('bayesplot')
```

```
This is bayesplot version 1.11.1

- Online documentation and vignettes at mc-stan.org/bayesplot

- bayesplot theme set to bayesplot::theme_default()

   * Does _not_ affect other ggplot2 plots

   * See ?bayesplot_theme_set for details on theme setting
```

Pregunta 1

Sea θ la tasa de créditos hipotecarios otorgados por un banco en Argentina. Durante el 2023 la tasa promedio fue de 60 % y la desviación estándar de la tasa fue de 0.04. En lo que va del año 2024 se han solicitado 100 créditos, de los cuales se han otorgado únicamente 50.

a. Usando la información del año pasado, encuentra la distribución beta que mejor describe el conocimiento inicial.

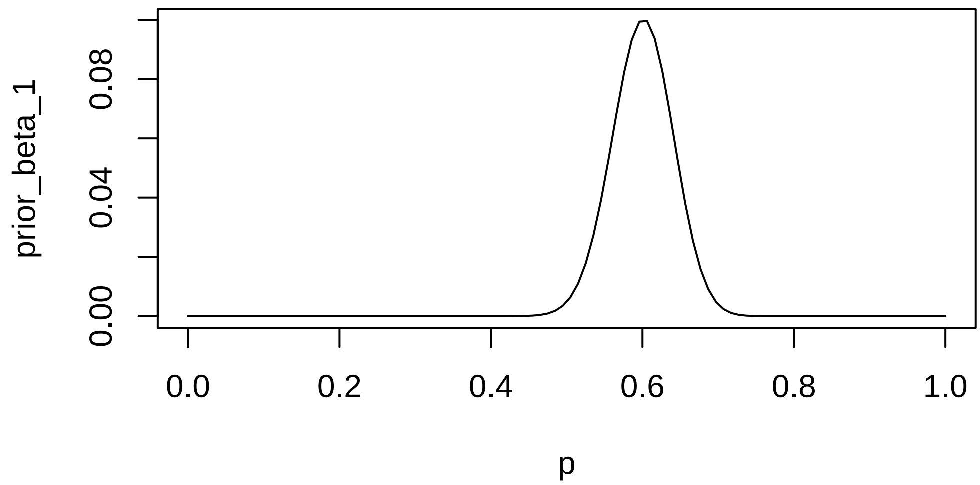
$$E[X] = \frac{\alpha}{\alpha+\beta} = 0.6$$
$$0.4\alpha = 0.6\beta$$
$$\frac{2}{3}\alpha = \beta$$

$$\text{var}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} = 0.04^2$$
$$\frac{\frac{2}{3}\alpha^2}{(\frac{5}{3}\alpha)^2(\frac{5}{3}\alpha+1)} = 0.04^2$$
$$\frac{2}{3}\alpha^2 = (0.04^2)(\frac{5^2}{3^2}\alpha^2)(\frac{5}{3}\alpha+1)$$
$$\frac{2}{3}\alpha^2 = (0.04^2)(\frac{5^3}{3^3}\alpha^3 + \frac{5^2}{3^2}\alpha)$$
$$\left[\frac{2}{3} - 0.04^2\frac{5^2}{3^2}\right]\alpha^2 = 0.04^2\frac{5^3}{3^3}\alpha^3$$

Si $\alpha \neq 0$

$$\alpha = \frac{2/3 - (0.04^2)(5^2/3^2)}{0.04^2(5^3/3^3)} = 89.4$$
$$\beta = \frac{2}{3}\alpha = \frac{2}{3}89.4 = 59.6$$

```
alpha_1 = (2/3-0.0016*(5^2/3^2))/(0.0016*(5^3/3^3))
beta_1 = 2/3*alpha_1
#define range
p = seq(0, 1, length=100)
prior_beta_1 <- dbeta(p, alpha_1, beta_1)/sum(dbeta(p, alpha_1, beta_1))
#create plot of Beta distribution with shape parameters 2 and 10
plot(p, prior_beta_1, type='l')
```



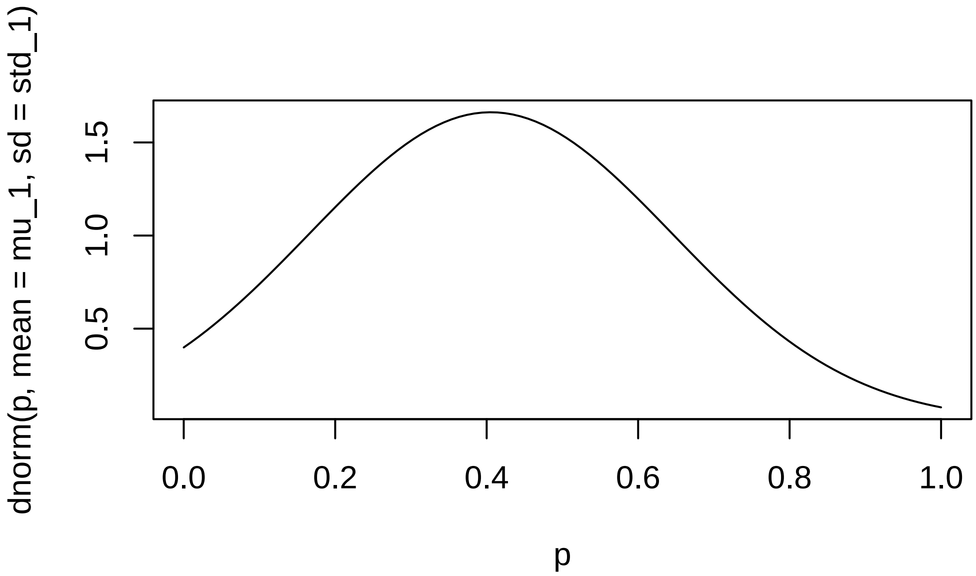
b. Usando la información del año pasado, encuentra la distribución normal transformada que mejor describa el conocimiento inicial.

Usando la liga canónica de la distribucón de Bernoulli:

$$logit(\theta) = \ln\Big(\frac{\theta}{1-\theta}\Big) = \ln\Big(\frac{0.6}{1-0.6}\Big) = 0.40546$$

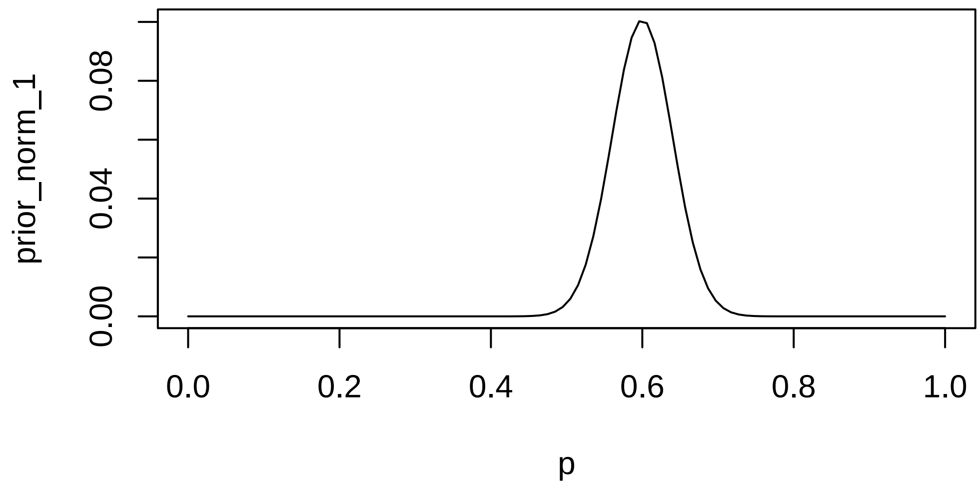
$$\frac{\partial^2 b}{\partial\theta^2} = \frac{\partial^2}{\partial\theta^2}\log(1+e^\theta) = \frac{\partial}{\partial\theta}\frac{e^\theta}{1+e^\theta} = \frac{e^\theta}{(1+e^\theta)^2} = \frac{e^{0.6}}{(1+e^{0.6})^2} = 0.24$$

```
mu_1 = log(0.6/(1-0.6))
std_1 = exp(mu_1)/(1+exp(mu_1))^2

# exp(mu_1)/(1+exp(mu_1))^2

#create plot of Beta distribution with shape parameters 2 and 10
plot(p, dnorm(p, mean=mu_1, sd=std_1), type='l')
```



OJO, acá el profesor comentó la hicieramos como una normal sin transformar

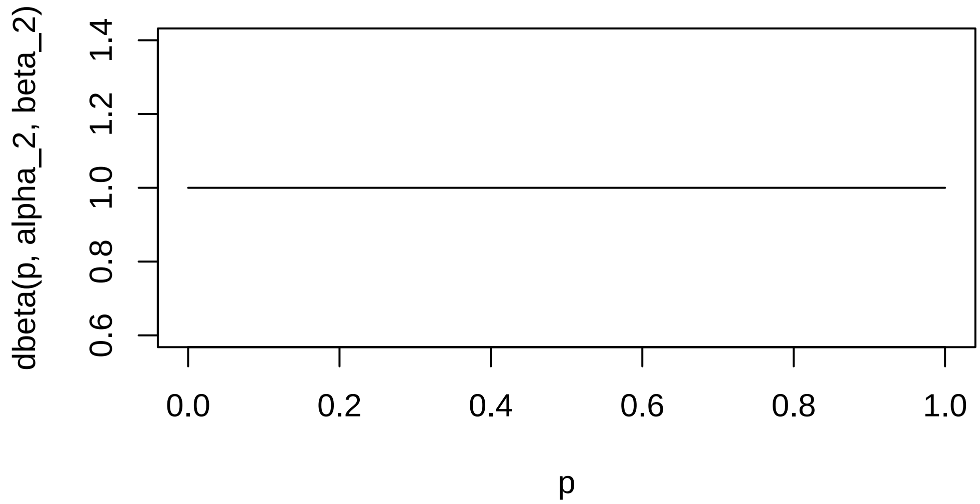
```
mu_1 = 0.6
std_1 = 0.04

# exp(mu_1)/(1+exp(mu_1))^2
prior_norm_1  <-dnorm(p, mean=mu_1, sd=std_1)
prior_norm_1 <- prior_norm_1 / sum(prior_norm_1)
#create plot of Beta distribution with shape parameters 2 and 10
plot(p, prior_norm_1 , type='l')
```



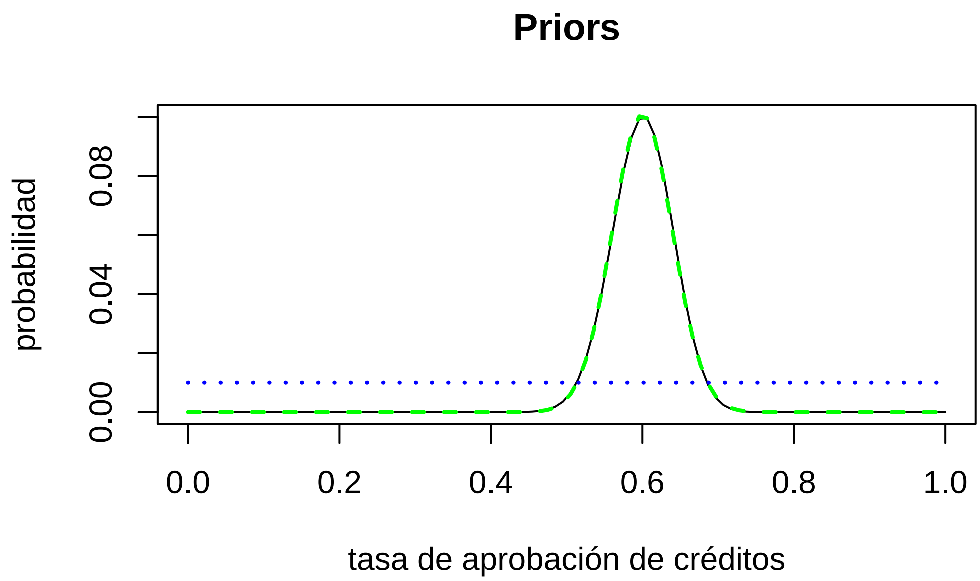
c. Determina la distribución inicial de referencia.

Elegimos una uniforme, sin embargo sabemos que una Beta cuyos parámetros son $\alpha = \beta = 1$ es igual a una uniforme en [0,1].

```
alpha_2 = 1
beta_2 = 1
#define range
p = seq(0, 1, length=100)
#create plot of Beta distribution with shape parameters 2 and 10
prior_unif_1 <- dbeta(p, alpha_2, beta_2)
prior_unif_1 <- prior_unif_1/ sum(prior_unif_1)
plot(p, dbeta(p, alpha_2, beta_2), type='l')
```




```
xlim <- range(c(0,1))
ylim <- range(c(0,0.1))
plot(p, prior_beta_1, type='l',xlim=xlim, ylim=ylim, main="Priors", #sub="Distribuciones i
  xlab="tasa de aprobación de créditos", ylab="probabilidad")
lines(p, prior_norm_1 ,lty=2,lwd=2,col="green")
lines(p, prior_unif_1,lty=3,lwd=2,col="blue")
```



d. Usando los datos del año 2024 encuentra la distribución final para cada una de las distribuciones iniciales de los incisos (a) – (c).

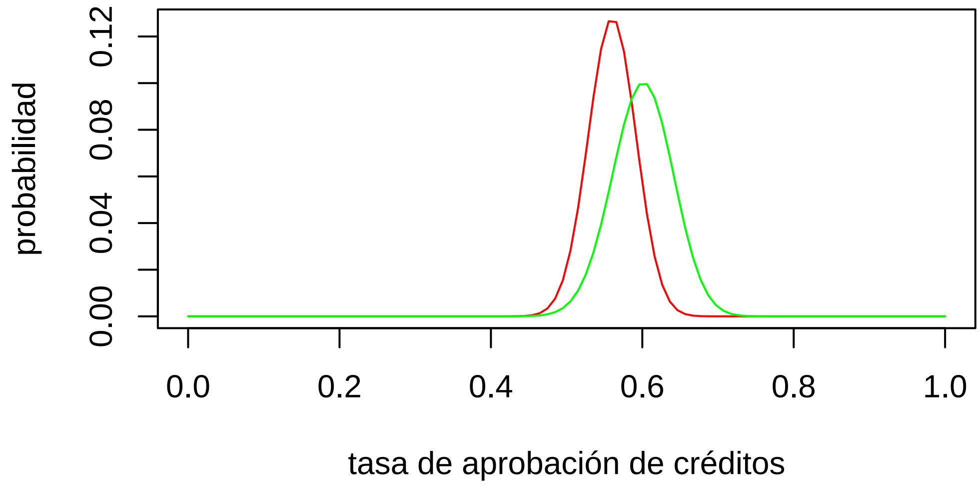
Sabemos que en el 2024 de 100 créditos, se otrorgaron únicamente 50, por lo que la verosimilitud es:

$$\mathcal{L} = \theta^{50}(1-\theta)^{50}$$
$$\log \mathcal{L} = 50 \log \theta + 50 \log(1-\theta)$$

Para a) Beta(89.4, 59.6)

Sabemos que para el conjugado beta-binomial, la posterior es una $Beta(\alpha + n, \beta + N - n)$

```
alpha_1_pos <- alpha_1 + 50
beta_1_pos <- beta_1 + 50

prior <- prior_beta_1
posterior <- dbeta(p, alpha_1_pos, beta_1_pos)/sum(dbeta(p, alpha_1_pos, beta_1_pos))

plot(p, posterior, type = "l", col = "red",
     xlab="tasa de aprobación de créditos", ylab="probabilidad")
lines(p, prior, type = "l", col = "green")
```



Haciendolo con STAN

```
modelo_1a <- cmdstan_model("./parcial1ej1a.stan")
print(modelo_1a)
```

```
data {
  int<lower=0> N;
  array[N] int<lower=0,upper=1> y;
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(89.4, 59.6);  // uniform prior on interval 0,1
  y ~ bernoulli(theta);
}
```

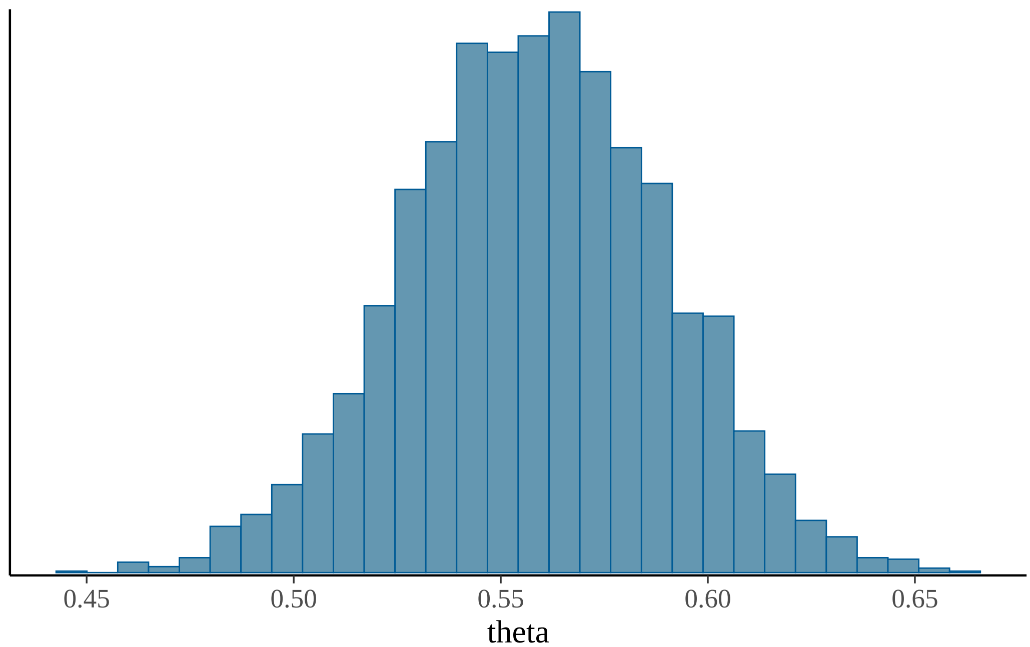
```
data_list <- list(N = 100, y = c(rep(1, 50),rep(0, 50)))

fit1a <- modelo_1a$sample(
  data = data_list,
```

```
  seed = 123,
  chains = 4,
  parallel_chains = 4,
  refresh = 500 # print update every 500 iters
)
```

```
Running MCMC with 4 parallel chains...

Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 2 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 3 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 3 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 4 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 4 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1 finished in 0.0 seconds.
Chain 2 finished in 0.0 seconds.
Chain 3 finished in 0.0 seconds.
Chain 4 finished in 0.0 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.0 seconds.
Total execution time: 0.3 seconds.
```

```r
mcmc_hist(fit1a$draws("theta"))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Para b) Normal(0.6,0.04)

$$\propto \left[\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2} \Big(50 \log\theta + 50\log(1-\theta)\Big) \right] d\theta$$

```r
modelo_1b <- cmdstan_model("./parcial1ej1b.stan")
print(modelo_1b)
```

```
data {
  int<lower=0> N;
  array[N] int<lower=0,upper=1> y;
}
parameters {
  real<lower=0,upper=1> theta;
}
```

```
model {
  theta ~ normal(0.6, 0.04);  // uniform prior on interval 0,1
  y ~ bernoulli(theta);
}
```

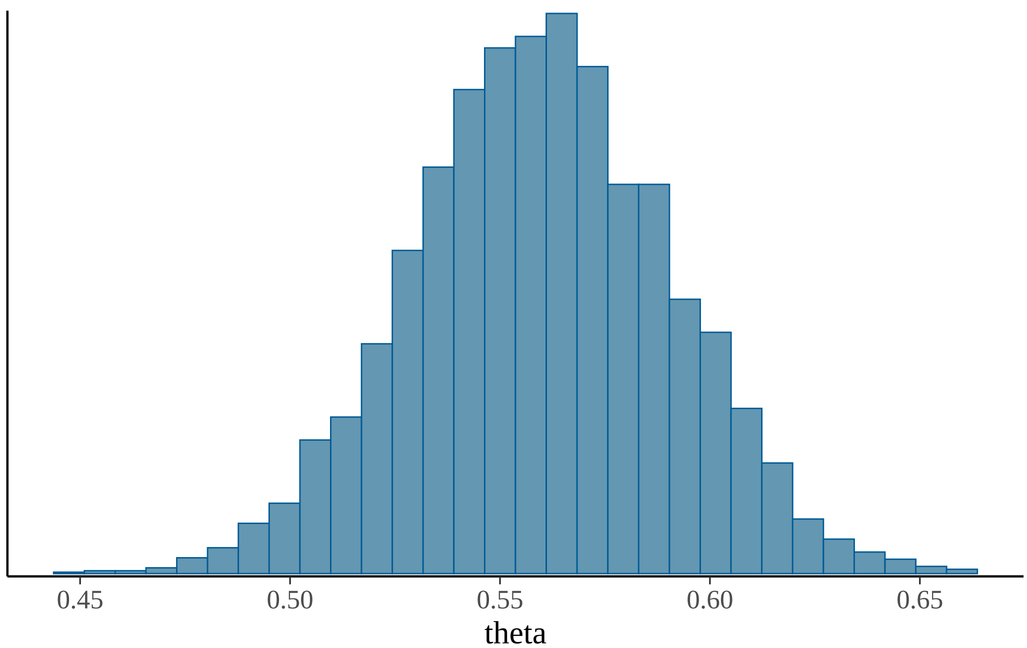
```r
data_list <- list(N = 100, y = c(rep(1, 50),rep(0, 50)))

fit1b <- modelo_1b$sample(
  data = data_list,
  seed = 123,
  chains = 4,
  parallel_chains = 4,
  refresh = 500 # print update every 500 iters
)
```

```
Running MCMC with 4 parallel chains...

Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 2 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 3 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 3 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 4 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 4 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1 finished in 0.0 seconds.
```

```
Chain 2 finished in 0.0 seconds.
Chain 3 finished in 0.0 seconds.
Chain 4 finished in 0.0 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.0 seconds.
Total execution time: 0.2 seconds.
```

```r
mcmc_hist(fit1b$draws("theta"))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Para c) Uniforme en [0,1]

```r
modelo_1c <- cmdstan_model("./parcial1ej1c.stan")
modelo_1c$print()
```

```
data {
  int<lower=0> N;
  array[N] int<lower=0,upper=1> y;
```

```
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(1,1);  // uniform prior on interval 0,1
  y ~ bernoulli(theta);
}
```

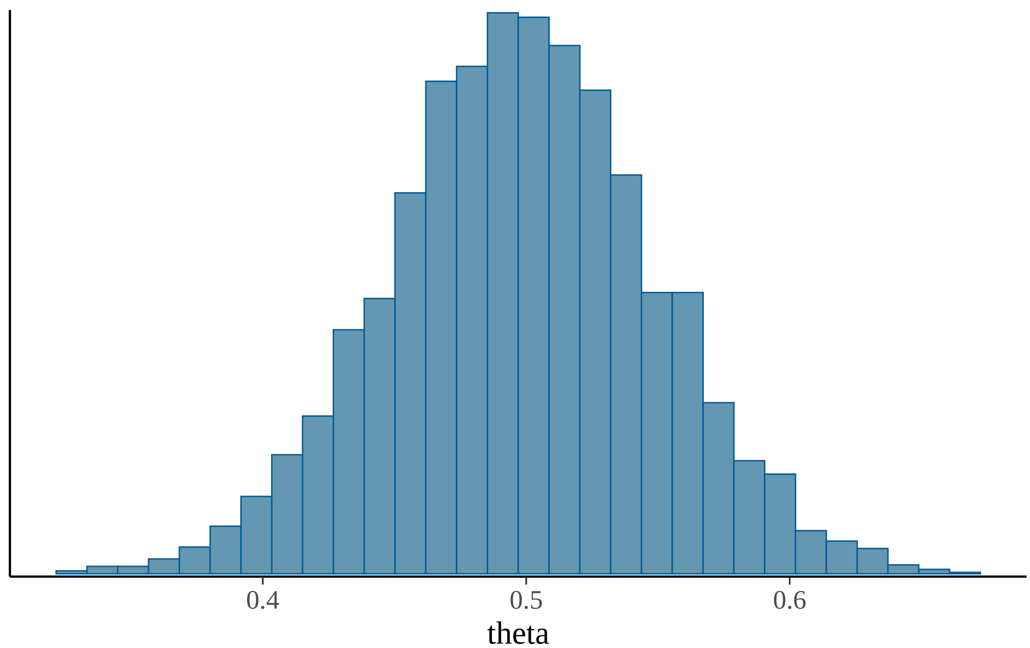
```r
data_list <- list(N = 100, y = c(rep(1, 50),rep(0, 50)))

fit1c <- modelo_1c$sample(
  data = data_list,
  seed = 123,
  chains = 4,
  parallel_chains = 4,
  refresh = 500 # print update every 500 iters
)
```

```
Running MCMC with 4 parallel chains...

Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 2 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 3 Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 3 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4 Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4 Iteration:  500 / 2000 [ 25%]  (Warmup)
Chain 4 Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
Chain 4 Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4 Iteration: 1500 / 2000 [ 75%]  (Sampling)
Chain 4 Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1 finished in 0.0 seconds.
Chain 2 finished in 0.0 seconds.
Chain 3 finished in 0.0 seconds.
Chain 4 finished in 0.0 seconds.

All 4 chains finished successfully.
Mean chain execution time: 0.0 seconds.
Total execution time: 0.2 seconds.
```

```r
mcmc_hist(fit1c$draws("theta"))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



e. Estima la tasa de créditos otorgados, usando las 3 distribuciones finales del inciso (d).

Para a) Beta(89.4, 59.6)

```r
fit1a$summary()
```

```
# A tibble: 2 x 10
  variable     mean   median     sd    mad       q5      q95  rhat ess_bulk
  <chr>       <dbl>    <dbl>  <dbl>  <dbl>    <dbl>    <dbl> <dbl>    <dbl>
1 lp__     -171.    -171.     0.700  0.314  -173.    -171.     1.00    1838.
2 theta       0.558    0.559 0.0314 0.0316    0.505    0.610  1.00    1487.
# i 1 more variable: ess_tail <dbl>
```

Para b) Normal(0.6,0.04)

```r
fit1b$summary()
```

```
# A tibble: 2 x 10
  variable    mean  median     sd    mad      q5     q95  rhat ess_bulk ess_tail
  <chr>      <dbl>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl> <dbl>    <dbl>    <dbl>
1 lp__     -72.4   -72.2    0.703  0.308  -73.8   -71.9    1.00    2123.    2750.
2 theta      0.560   0.560 0.0307 0.0306   0.509   0.610  1.00    1510.    1874.
```

Para c) Uniforme en [0,1]

```r
fit1c$summary()
```

```
# A tibble: 2 x 10
  variable    mean  median     sd    mad      q5     q95  rhat ess_bulk ess_tail
  <chr>      <dbl>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl> <dbl>    <dbl>    <dbl>
1 lp__     -71.2   -70.9    0.760  0.314  -72.7   -70.7    1.00    1858.    1783.
2 theta      0.498   0.498 0.0502 0.0488   0.414   0.581  1.00    1584.    1589.
```

f. Estima el momio de otorgar un crédito, i.e., $\phi = \frac{\theta}{1-\theta'}$, usando las 3 distribuciones finales del inciso (d).

```r
phi <- function(theta){
  theta/(1-theta)
}
```

Para a) Beta(89.4, 59.6)

```r
fit1a$draws('theta', format = "df") %>%
  mutate(phi=phi(theta)) %>%
  summarise(mean(phi))
```

```
# A tibble: 1 x 1
  `mean(phi)`
        <dbl>
1        1.28
```

Para b) Normal(0.6,0.04)

```
fit1b$draws('theta', format = "df") %>%
  mutate(phi=phi(theta)) %>%
  summarise(mean(phi))
```

```
# A tibble: 1 x 1
  `mean(phi)`
        <dbl>
1        1.28
```

Para c) Uniforme en [0,1]

```
fit1c$draws('theta', format = "df") %>%
  mutate(phi=phi(theta)) %>%
  summarise(mean(phi))
```

```
# A tibble: 1 x 1
  `mean(phi)`
        <dbl>
1        1.01
```

Pregunta 2

Las utilidades mensuales de una compañía tienen una distribución $\mathcal{N}(\mu, \sigma^2)$ (aquí se da la varianza, no la precisión). Suponer que una muestra de 10 meses de esta compañía dio como resultado las siguientes utilidades: (212, 207, 210, 196, 223, 193, 196, 210, 202, 221).

```
datos_2 <- c(212, 207, 210, 196, 223, 193, 196, 210, 202, 221)
```

a. La incertidumbre sobre la utilidad promedio anual μ se puede representar por una distribución $\mathcal{N}(200, 40)$, y la incertidumbre de la desviación estándar de las utilidades mensuales se puede representar mediante una distribución $\mathcal{G}(10, 1)$. Mediante la distribución posterior estima μ y σ^2.

```r
data_list <- list(
  N = length(datos_2),
  datos_2 = datos_2
)

modelo_2 <- cmdstan_model("./parcial1ej2.stan")
```

```r
print(modelo_2)
```

```
data {
  int<lower=0> N;  // Number of observations
  vector[N] datos_2;  // Monthly utility data
}

parameters {
  real mu;  // Mean parameter
  real<lower=0> sigma;  // Standard deviation parameter
}

model {
  datos_2 ~ normal(mu, sigma);  // Likelihood
  mu ~ normal(200, sqrt(40));  // Prior for the mean
  sigma ~ gamma(10, 1);  // Prior for the standard deviation
}
```

```r
fit2 <- modelo_2$sample(
  data = data_list,
  chains = 4,
  iter_sampling = 2000,
  iter_warmup = 500,
  show_messages = FALSE,
  show_exceptions = FALSE)
```

```r
fit2
```

```
 variable   mean median   sd  mad     q5    q95 rhat ess_bulk ess_tail
    lp__  -16.30 -15.99 1.01 0.72 -18.32 -15.33 1.00     3517     4285
    mu    205.46 205.52 2.95 2.90 200.50 210.23 1.00     5289     4265
    sigma  10.56  10.36 2.02 1.95   7.67  14.13 1.00     4593     4855
```

b. Utilizando una distribución inicial no informativa, estima mediante la correspondiente distribución inicial μ y σ^2.

```r
mean(datos_2)
```

```
[1] 207
```

```r
sd(datos_2)
```

```
[1] 10.31719
```

Como distribuciones poco informativas se utilizó:

- una normal N(150,300)
- una normal N(10,20)

```r
modelo_2b <- cmdstan_model("./parcial1ej2b.stan")
print(modelo_2b)
```

```
data {
  int<lower=0> N;  // Number of observations
  vector[N] datos_2;  // Monthly utility data
}

parameters {
  real mu;  // Mean parameter
  real<lower=0> sigma;  // Standard deviation parameter
}

model {
  datos_2 ~ normal(mu, sigma);  // Likelihood
  mu ~ normal(150,300);  // Prior for the mean
  sigma ~ normal(10,20);  // Prior for the standard deviation
}
```

```r
fit2b <- modelo_2b$sample(
  data = data_list,
  chains = 4,
```

```
    iter_sampling = 2000,
    iter_warmup = 500,
    show_messages = FALSE,
    show_exceptions = FALSE)
```

```
fit2b
```

```
 variable   mean median   sd  mad     q5    q95 rhat ess_bulk ess_tail
    lp__  -26.59 -26.26 1.05 0.77 -28.69 -25.58 1.00     3212     3792
    mu    206.86 206.87 3.86 3.60 200.51 213.21 1.00     4367     4075
    sigma  11.90  11.31 3.23 2.74   7.91  17.85 1.00     4731     4222
```

Pregunta 3

A continuación se presenta una base de datos de calificaciones de 20 empresas financieras hechas por las dos compañías calificadores más importantes S&P y Moody's (ver el archivo calificaciones.txt. Realiza un análisis Bayesiano completo de los datos, ajustando un modelo de regresión lineal, tomando como variable respuesta las calificaciones de S&P y como variable explicativa las calificaciones de Moody's.

Nuestro Modelo es:

$$Y_{S\&P} = \beta_0 + \beta_1 X_{Moody's}$$

donde:

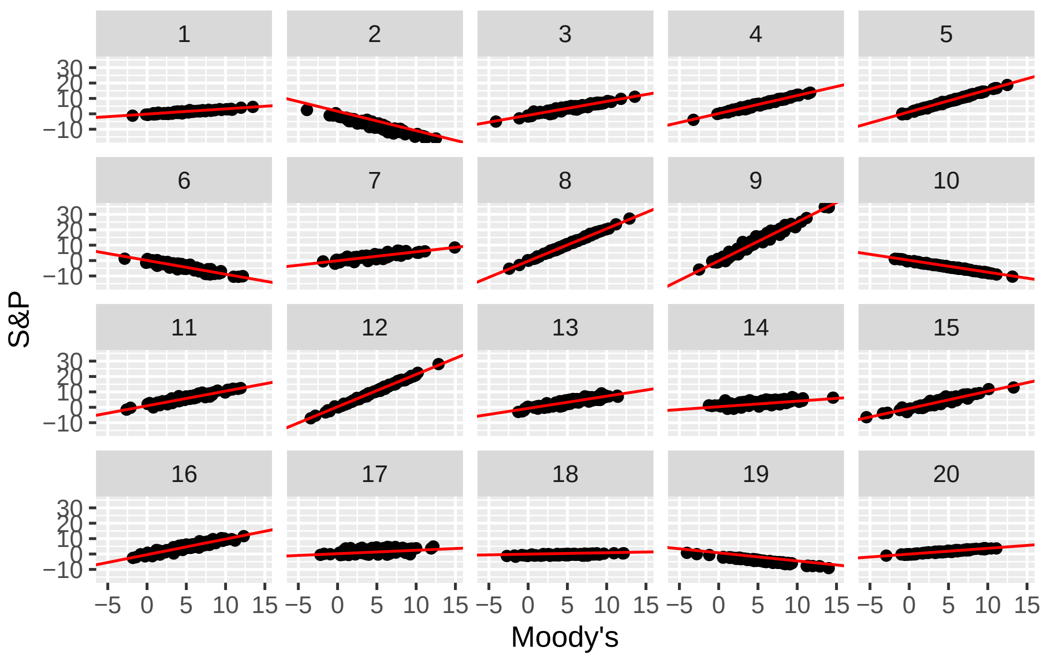
$$Y_{S\&P} \sim \mathcal{N}(\mu, \sigma)$$
$$\beta_0 \sim \mathcal{N}(0, 1)$$
$$\beta_1 \sim \mathcal{N}(1, 1)$$
$$\sigma \sim \mathcal{N}(0, 1)$$

```
sim_calificaciones<- function(n= 10){
  beta0 <- rnorm(1, 0, 1)
  beta1 <- rnorm(1, 1, 1)
  sigma <- abs(rnorm(1, 0, 1))
  # simular Moody's
  MO <- rnorm(n, 5, 3)
  mu_MO = beta0 + beta1 * MO
  # simular perturbación de calificacion
  U <- rnorm(n, 0, sigma)
```

```
  # regresión lineal de SP dado MO
  SP <- mu_MO + U
  tibble(beta0, beta1, sigma, MO, SP)
}
```

```
sims_tbl <- map_df(1:20, function(rep) {
  sim_calificaciones(100) |> mutate(rep = rep)
})
```

```
sims_tbl |>
  ggplot(aes(x = MO, y = SP)) +
  geom_point() +
  geom_abline(aes(intercept = beta0 * beta1, slope = beta1), data = sims_tbl, color = "red
  labs(x = "Moody's", y = "S&P") +
  facet_wrap(~rep)
```



Lucen como suposiciones razonables, por lo que nos movemos al siguiente paso, el modelo

```
# Read a txt file, named "mtcars.txt"
calificaciones <- read.table("./datos/calificaciones.txt", header = TRUE, sep = "", dec =
```

```r
glimpse(calificaciones)
```

```
Rows: 20
Columns: 2
$ SP <dbl> 3.1, 2.3, 3.0, 1.9, 2.5, 3.7, 3.4, 2.6, 2.8, 1.6, 2.0, 2.9, 2.3, 3.~
$ MO <dbl> 5.5, 4.8, 4.7, 3.9, 4.5, 6.2, 6.0, 5.2, 4.7, 4.3, 4.9, 5.4, 5.0, 6.~
```

```r
modelo_3 <- cmdstan_model("./parcial1ej3.stan")
print(modelo_3)
```

```
data {
  int<lower=0> N;  // Number of observations
  vector[N] calificaciones_SP;  // S&P ratings
  vector[N] calificaciones_MO;  // Moody's ratings
}

parameters {
  real beta_0;
  real beta_1;
    real<lower=0> sigma;
}

transformed parameters{
  vector[N] Y_SP;
  Y_SP = beta_0 + beta_1 * calificaciones_MO;
}

model {
  calificaciones_SP ~ normal(Y_SP, sigma);  // Predicted S&P ratings
  beta_0 ~ normal(0, 1);
  beta_1 ~ normal(1, 1);
  sigma ~ normal(0, 1);
}
```

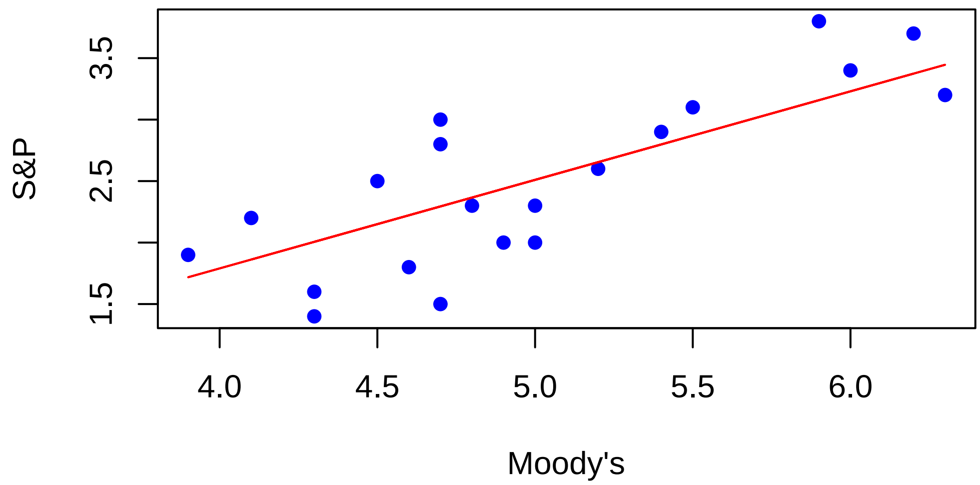
```r
data_list <- list(
  N = nrow(calificaciones),
  calificaciones_SP = calificaciones$SP,
  calificaciones_MO = calificaciones$MO
)
```

```r
fit3 <- modelo_3$sample(
  data = data_list,
  chains = 4,
  iter_sampling = 2000,
  iter_warmup = 500,
  show_messages = FALSE,
  show_exceptions = FALSE)
```

```r
fit3
```

```
variable  mean median   sd  mad    q5   q95 rhat ess_bulk ess_tail
 lp__      4.20   4.55 1.28 1.01  1.65  5.56 1.00     2741     2900
 beta_0   -1.08  -1.10 0.62 0.59 -2.07 -0.05 1.00     2415     2682
 beta_1    0.72   0.72 0.12 0.12  0.51  0.92 1.00     2377     2555
 sigma     0.47   0.46 0.08 0.08  0.36  0.62 1.00     2999     3185
 Y_SP[1]  2.87   2.87 0.12 0.12  2.67  3.07 1.00     4998     5478
 Y_SP[2]  2.37   2.37 0.10 0.10  2.20  2.54 1.00     6820     5392
 Y_SP[3]  2.30   2.30 0.11 0.10  2.12  2.47 1.00     6053     5343
 Y_SP[4]  1.72   1.72 0.17 0.16  1.45  2.00 1.00     3059     3379
 Y_SP[5]  2.15   2.15 0.12 0.11  1.96  2.35 1.00     4696     5089
 Y_SP[6]  3.38   3.38 0.18 0.18  3.07  3.67 1.00     3143     3492

# showing 10 of 24 rows (change via 'max_rows' argument or 'cmdstanr_max_rows' option)
```

```r
calificaciones <- calificaciones %>% mutate(SP_line = -1.09 + 0.72 * MO)

plot(calificaciones$MO, calificaciones$SP, pch = 16, col = "blue", xlab = "Moody's", ylab
lines(calificaciones$MO, calificaciones$SP_line, col = "red", type = "l")
```



La recta ajusta bien a nuestras observaciones.

Pregunta 4

Un investigador desea evaluar la relación entre el salario anual de trabajadores de una compañía de nivel medio y alto (Y, en miles de dólares) y el índice de calidad de trabajo (X_1), número de años de experiencia (X_2) y el índice de éxito en publicaciones (X_3). La muestra consiste de 24 trabajadores. Realiza un análisis Bayesiano completo de los datos y obtén las predicciones de salarios para 3 nuevos empleados con variables explicativas:

$$x'_{1F} = (5, 4, 17, 6, 0),$$
$$x'_{2F} = (6, 2, 12, 5, 8),$$
$$x'_{3F} = (6, 4, 21, 6, 1)$$

Los datos se encuentran en el archivo salarios.txt.

```
# Read a txt file, named "mtcars.txt"
salarios <- read.table("./datos/salarios.txt", header = TRUE, sep = "", dec = ".")
glimpse(salarios)
```

```
Rows: 24
Columns: 4
$ Y  <dbl> 33.2, 40.3, 38.7, 46.8, 41.4, 37.5, 39.0, 40.7, 30.1, 52.9, 38.2, 3~
$ X1 <dbl> 3.5, 5.3, 5.1, 5.8, 4.2, 6.0, 6.8, 5.5, 3.1, 7.2, 4.5, 4.9, 8.0, 6.~
$ X2 <dbl> 9, 20, 18, 33, 31, 13, 25, 30, 5, 47, 25, 11, 23, 35, 39, 21, 7, 40~
$ X3 <dbl> 6.1, 6.4, 7.4, 6.7, 7.5, 5.9, 6.0, 4.0, 5.8, 8.3, 5.0, 6.4, 7.6, 7.~
```

```r
x1_p <- c(5, 4, 17, 6, 0)
x2_p <- c(6, 2, 12, 5, 8)
x3_p <- c(6, 4, 21, 6, 1)
datos_4 <- tibble(x1_p, x2_p, x3_p)
glimpse(datos_4)
```

```
Rows: 5
Columns: 3
$ x1_p <dbl> 5, 4, 17, 6, 0
$ x2_p <dbl> 6, 2, 12, 5, 8
$ x3_p <dbl> 6, 4, 21, 6, 1
```

Nuestro modelo es:

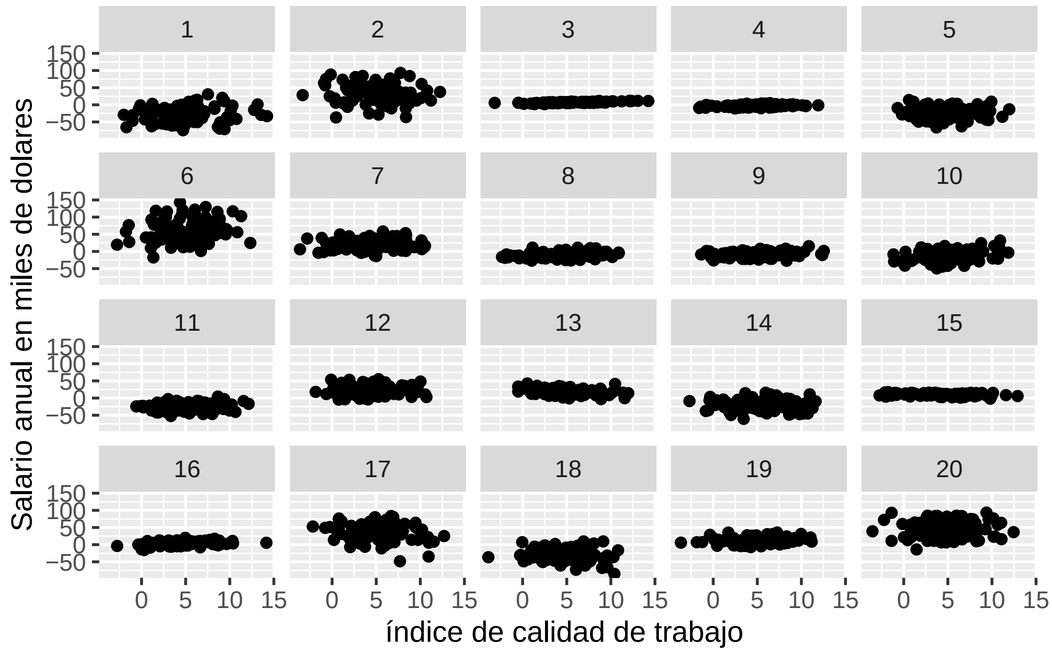
$$Y_{media} = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \beta_3 * X_3;$$
$$Y \sim \mathcal{N}(Y_{media}, \sigma)$$
$$\beta_0 \sim \mathcal{N}(0, 1)$$
$$\beta_1 \sim \mathcal{N}(0, 1)$$
$$\beta_2 \sim \mathcal{N}(0, 1)$$
$$\beta_3 \sim \mathcal{N}(0, 1)$$
$$\sigma \sim \mathcal{N}(0, 1)$$

```r
sim_salarios<- function(n= 10){
  beta0 <- rnorm(1, 0, 1)
  beta1 <- rnorm(1, 0, 1)
  beta2 <- rnorm(1, 0, 1)
  beta3 <- rnorm(1, 0, 1)
  sigma <- abs(rnorm(1, 0, 1))
  X1 <-rnorm(n, 5, 3)
  X2 <-rnorm(n, 25, 15)
  X3 <-rnorm(n, 6, 3)
```

```
  mu_y = beta0 + beta1 * X1 + beta2 * X2 + beta3 * X3
  Y <- rnorm(n, mu_y, sigma)
  tibble(beta0, beta1, beta2, beta3, sigma, X1, X2, X3, Y)
}
```

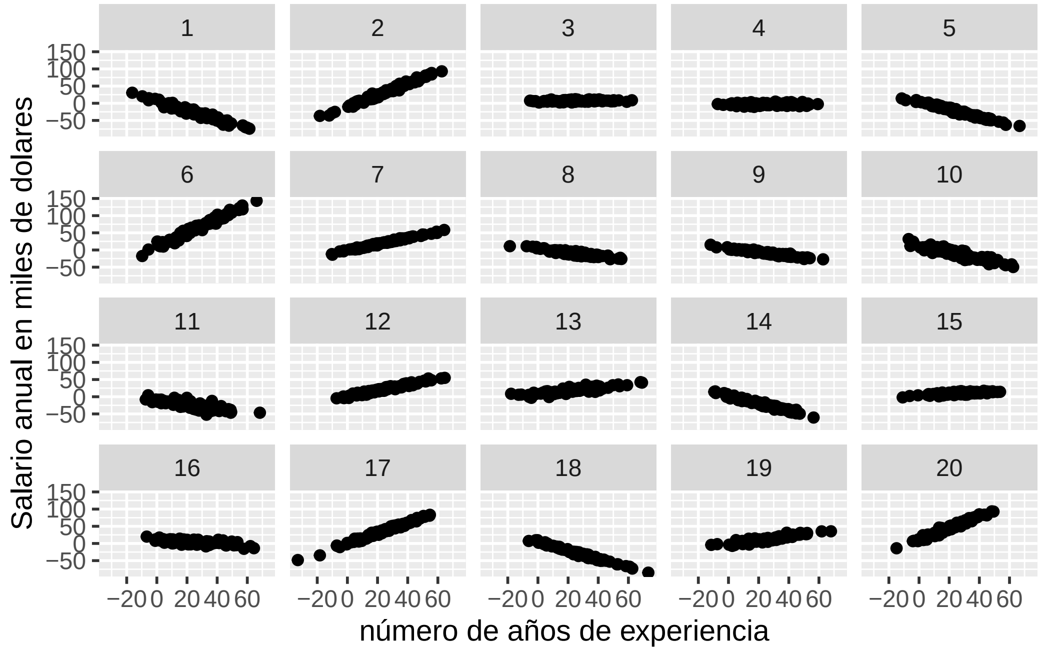
```
sims_tbl <- map_df(1:20, function(rep) {
  sim_salarios(100) |> mutate(rep = rep)
})
```

```
sims_tbl |>
  ggplot(aes(x = X1, y = Y)) +
  geom_point() +
  # geom_abline(aes(intercept = beta0 - 160 * beta1, slope = beta1), data = sims_tbl, colo
  labs(x = "índice de calidad de trabajo ", y = "Salario anual en miles de dolares") +
  facet_wrap(~rep)
```

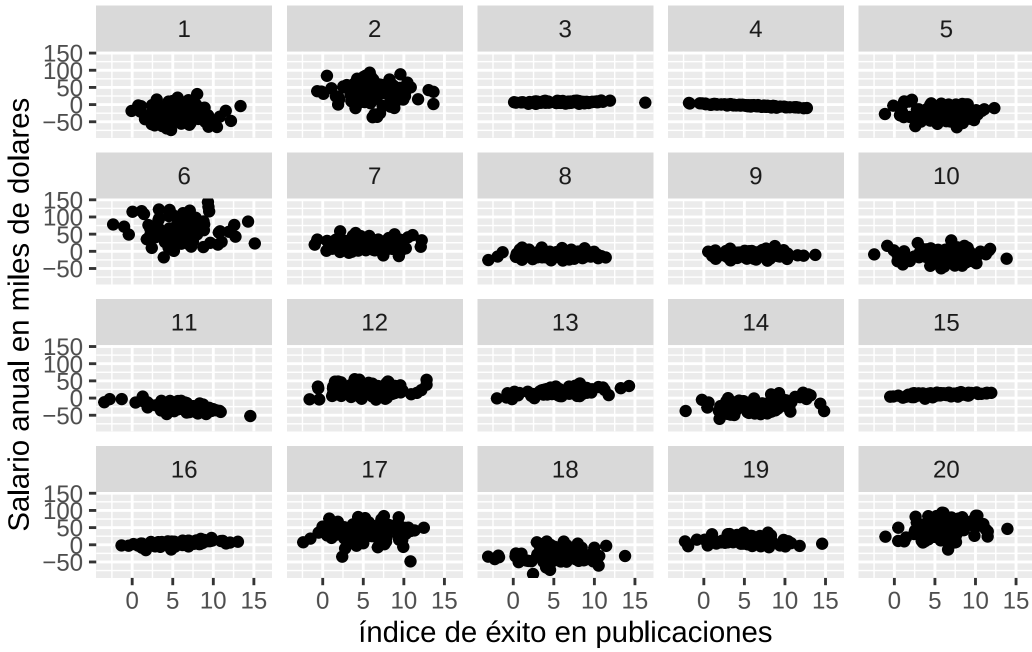



```
sims_tbl |>
  ggplot(aes(x = X2, y = Y)) +
  geom_point() +
  # geom_abline(aes(intercept = beta0 - 160 * beta1, slope = beta1), data = sims_tbl, colo
```

```
  labs(x = "número de años de experiencia ", y = "Salario anual en miles de dolares") +
  facet_wrap(~rep)
```




```
sims_tbl |>
  ggplot(aes(x = X3, y = Y)) +
  geom_point() +
  # geom_abline(aes(intercept = beta0 - 160 * beta1, slope = beta1), data = sims_tbl, colo
  labs(x = "índice de éxito en publicaciones", y = "Salario anual en miles de dolares") +
  facet_wrap(~rep)
```



Las simulaciones lucen razonables, por lo que procederemos a correr el modelo

```
modelo_4 <- cmdstan_model("./parcial1ej4.stan")
print(modelo_4)
```

```
data {
  int<lower=0> N;
  vector[N] Y;
  vector[N] X1;
  vector[N] X2;
  vector[N] X3;
  real X1_P;
  real X2_P;
  real X3_P;
}

parameters {
  real beta_0;
  real beta_1;
  real beta_2;
  real beta_3;
  real<lower=0> sigma;
```

```
}

transformed parameters {
  vector[N] Y_media;
  Y_media = beta_0 + beta_1 * X1 + beta_2 * X2 + beta_3 * X3;
}

model {
  Y ~ normal(Y_media, sigma);
  beta_0 ~ normal(0, 1);
  beta_1 ~ normal(0, 1);
  beta_2 ~ normal(0, 1);
  beta_3 ~ normal(0, 1);
  sigma ~ normal(0, 1);
}

generated quantities {
  real pred;
  {
    array[2000] real sim_pred;

    for(k in 1:2000){
      sim_pred[k] = beta_0 + beta_1 * X1_P + beta_2 * X2_P + beta_3 * X3_P;

    }
    pred = mean(sim_pred);
  }
}
```

```
# Para el dato 1 predicción:
data_list_1 <- list(
  N = nrow(salarios),
  Y = salarios$Y,
  X1 = salarios$X1,
  X2 = salarios$X2,
  X3 = salarios$X3,
  # M = nrow(datos_4),
  X1_P = x1_p[1],
  X2_P = x2_p[1],
  X3_P = x3_p[1]
)
fit4_1 <- modelo_4$sample(
```

```
  data = data_list_1,
  chains = 4,
  iter_sampling = 2000,
  iter_warmup = 500,
  show_messages = FALSE,
  show_exceptions = FALSE)
fit4_1$summary(c( "pred"))
```

```
# A tibble: 1 x 10
  variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 pred      31.2   31.3  1.29  1.29  29.1  33.4  1.00    5723.    5327.
```

```
# Para el dato 2 predicción:
data_list_1 <- list(
  N = nrow(salarios),
  Y = salarios$Y,
  X1 = salarios$X1,
  X2 = salarios$X2,
  X3 = salarios$X3,
  # M = nrow(datos_4),
  X1_P = x1_p[2],
  X2_P = x2_p[2],
  X3_P = x3_p[2]
)
fit4_1 <- modelo_4$sample(
  data = data_list_1,
  chains = 4,
  iter_sampling = 2000,
  iter_warmup = 500,
  show_messages = FALSE,
  show_exceptions = FALSE)
fit4_1$summary(c( "pred"))
```

```
# A tibble: 1 x 10
  variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 pred      22.2   22.2  1.19  1.17  20.2  24.1  1.00    5609.    4893.
```

```r
# Para el dato 3 predicción:
data_list_1 <- list(
  N = nrow(salarios),
  Y = salarios$Y,
  X1 = salarios$X1,
  X2 = salarios$X2,
  X3 = salarios$X3,
  # M = nrow(datos_4),
  X1_P = x1_p[3],
  X2_P = x2_p[3],
  X3_P = x3_p[3]
)
fit4_1 <- modelo_4$sample(
  data = data_list_1,
  chains = 4,
  iter_sampling = 2000,
  iter_warmup = 500,
  show_messages = FALSE,
  show_exceptions = FALSE)
fit4_1$summary(c( "pred"))
```

```
# A tibble: 1 x 10
  variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 pred      101.   101.  5.29  5.22  91.9  109.  1.00    5835.    5039.
```

```r
# Para el dato 4 predicción:
data_list_1 <- list(
  N = nrow(salarios),
  Y = salarios$Y,
  X1 = salarios$X1,
  X2 = salarios$X2,
  X3 = salarios$X3,
  # M = nrow(datos_4),
  X1_P = x1_p[4],
  X2_P = x2_p[4],
  X3_P = x3_p[4]
)
fit4_1 <- modelo_4$sample(
  data = data_list_1,
  chains = 4,
```

```r
    iter_sampling = 2000,
    iter_warmup = 500,
    show_messages = FALSE,
    show_exceptions = FALSE)
  fit4_1$summary(c( "pred"))
```

```
# A tibble: 1 x 10
  variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 pred      33.1   33.2  1.57  1.52  30.5  35.7  1.00    5289.    5064.
```

```r
  # Para el dato 5 predicción:
  data_list_1 <- list(
    N = nrow(salarios),
    Y = salarios$Y,
    X1 = salarios$X1,
    X2 = salarios$X2,
    X3 = salarios$X3,
    # M = nrow(datos_4),
    X1_P = x1_p[5],
    X2_P = x2_p[5],
    X3_P = x3_p[5]
  )
  fit4_1 <- modelo_4$sample(
    data = data_list_1,
    chains = 4,
    iter_sampling = 2000,
    iter_warmup = 500,
    show_messages = FALSE,
    show_exceptions = FALSE)
  fit4_1$summary(c( "pred"))
```

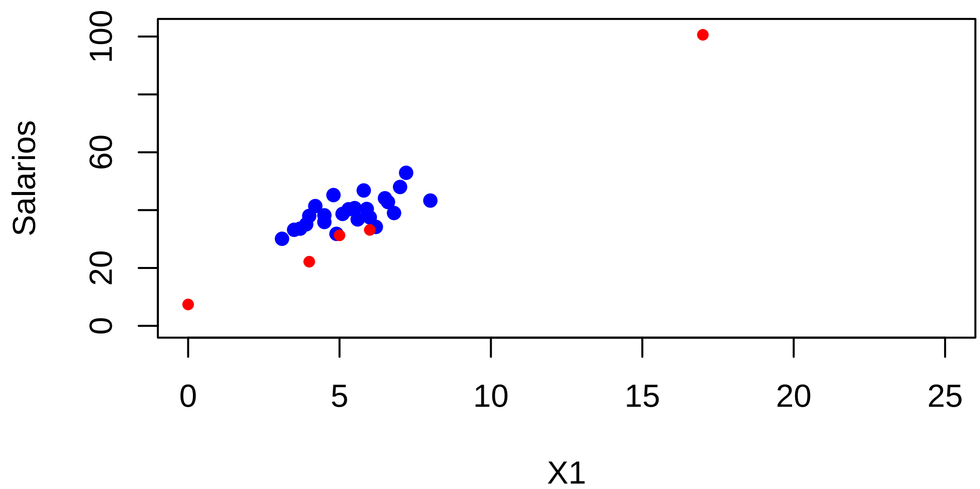
```
# A tibble: 1 x 10
  variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
  <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 pred      7.35   7.37  1.05  1.03  5.64  9.08  1.00    5729.    5105.
```

```r
  fit4_1$summary()
```

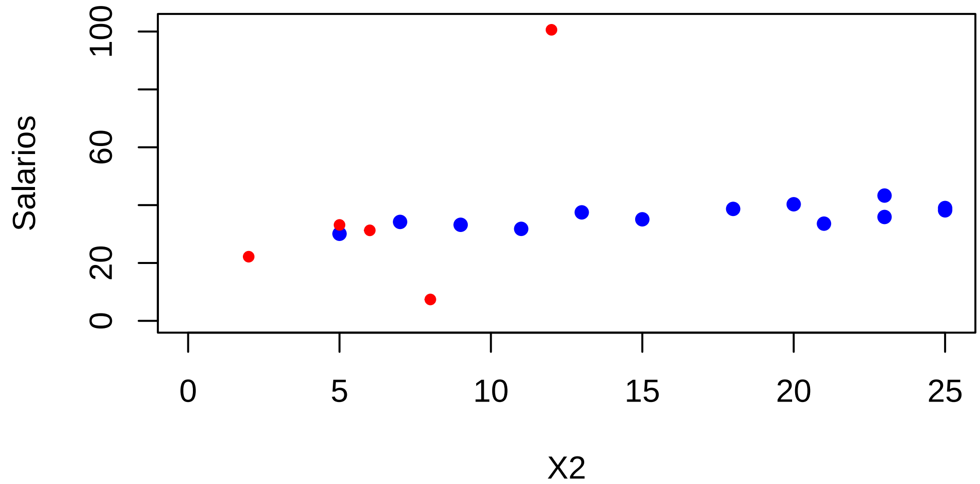
```
# A tibble: 31 x 10
   variable      mean  median     sd    mad      q5     q95  rhat ess_bulk
   <chr>        <dbl>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl> <dbl>    <dbl>
 1 lp__       -54.7   -54.4   1.65   1.45   -57.9   -52.8    1.00    3221.
 2 beta_0       1.78    1.78  1.00   1.00     0.108   3.43   1.00    5821.
 3 beta_1       2.22    2.22  0.450  0.438    1.49    2.96   1.00    4901.
 4 beta_2       0.359   0.358 0.0641 0.0638   0.257   0.467  1.00    5794.
 5 beta_3       2.70    2.70  0.365  0.367    2.10    3.30   1.00    4971.
 6 sigma        3.04    3.02  0.374  0.362    2.48    3.69   1.00    5771.
 7 Y_media[1]  29.3    29.3   1.13   1.12    27.4    31.1    1.00    5986.
 8 Y_media[2]  38.0    38.0   0.734  0.702   36.8    39.2    1.00    7185.
 9 Y_media[3]  39.6    39.6   1.02   0.988   37.9    41.3    1.00    6196.
10 Y_media[4]  44.6    44.6   0.786  0.769   43.3    45.9    1.00    7511.
# i 21 more rows
# i 1 more variable: ess_tail <dbl>
```

Revisamos que los valores predichos hagan sentido con los observados

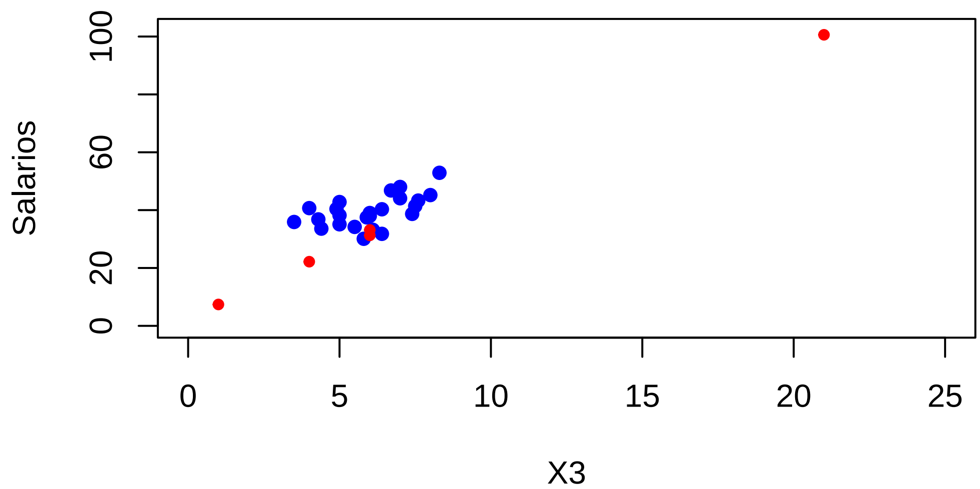
```
xlim <- range(c(0,25))
ylim <- range(c(0,102))
Y_p <- c(31.28429, 22.18005, 100.5972, 33.1545, 7.392581)
predic_4 <- tibble(Y=Y_p, X1=x1_p, X2=x2_p, X3=x3_p)
plot(salarios$X1, salarios$Y, pch = 16, col = "blue", xlab = "X1", ylab = "Salarios",xlim=
points(predic_4$X1, predic_4$Y, pch = 20, col = "red")
```




```
Y_p <- c(31.28429, 22.18005, 100.5972, 33.1545, 7.392581)
predic_4 <- tibble(Y=Y_p, X1=x1_p, X2=x2_p, X3=x3_p)
plot(salarios$X2, salarios$Y, pch = 16, col = "blue", xlab = "X2", ylab = "Salarios",xlim=
points(predic_4$X2, predic_4$Y, pch = 20, col = "red")
```




```
Y_p <- c(31.28429, 22.18005, 100.5972, 33.1545, 7.392581)
predic_4 <- tibble(Y=Y_p, X1=x1_p, X2=x2_p, X3=x3_p)
plot(salarios$X3, salarios$Y, pch = 16, col = "blue", xlab = "X3", ylab = "Salarios",xlim=
points(predic_4$X3, predic_4$Y, pch = 20, col = "red")
```



Podemos ver que ajusta bien salvo por un dato atípico.

Pregunta 5

Una compañía de seguros quiere lanzar un nuevo seguro médico para mineros. Para ello desea estimar la probabilidad de muerte (π_i), con base en el tiempo de exposición al mineral (x_i en horas). Se cuenta con información de las muertes registradas entre 1950 y 1959, junto con el tiempo de exposición al mineral y el número de mineros expuestos.

Realiza un análisis Bayesiano de los datos y obtén la distribución predictiva del número de muertes suponiendo que hay 100 mineros con un tiempo de exposición de 200 horas. Los datos se encuentran en el archivo mortality.txt.

```r
mortality <- read.table("./datos/mortality.txt", header = TRUE, sep = "", dec = ".")
glimpse(mortality)
```

```
Rows: 6
Columns: 3
$ x <int> 0, 5, 30, 75, 150, 250
$ y <int> 13, 5, 5, 3, 4, 18
$ n <int> 391, 205, 156, 50, 35, 51
```

El modelo es el siguiente: Para $i = 1, ..., N$

$$Y_i|\pi_i \sim Bin(n_i, \pi_i)$$

$$logit(\pi_i) = \beta_0 + \beta_1 x_i$$

con $\beta_0 \sim \mathcal{N}(0, 0.001)$ y $\beta 1 \sim N(0, 0.001)$.

```r
modelo_5 <- cmdstan_model("./parcial1ej5.stan")
print(modelo_5)
```

```
data {
  int<lower=0> N;         // Number of observations
  vector<lower=0>[N] x;                // Time of exposure
  array[N] int y;      // Number of deaths
  array[N] int n;      // Number of miners
  array[1] int new_n;     // New number of miners for prediction
  real<lower=0> new_x;    // New time of exposure for prediction
}

parameters {
  real beta0;               // Intercept
  real beta1;               // Slope
}

transformed parameters {
  vector[N] p_logit_res;
  real pf;
 // for (i in 1:N) {
    p_logit_res= inv_logit(beta0 + beta1 * x); //Liga logistica
 // }
   pf = inv_logit(beta0 + beta1 * new_x);
}

model {
  // Likelihood
  for (i in 1:N) {
    y[i] ~ binomial(n[i], p_logit_res[i]);
  }

  // Priors
  beta0 ~ normal(0, 0.0001);
  beta1 ~ normal(0, 0.0001);
```

```
}

generated quantities {
  array[N] int new_y;       // Predicted number of deaths for new data;
  real pred;
    new_y = binomial_rng(new_n, pf);;
    pred = mean(new_y);

}
```

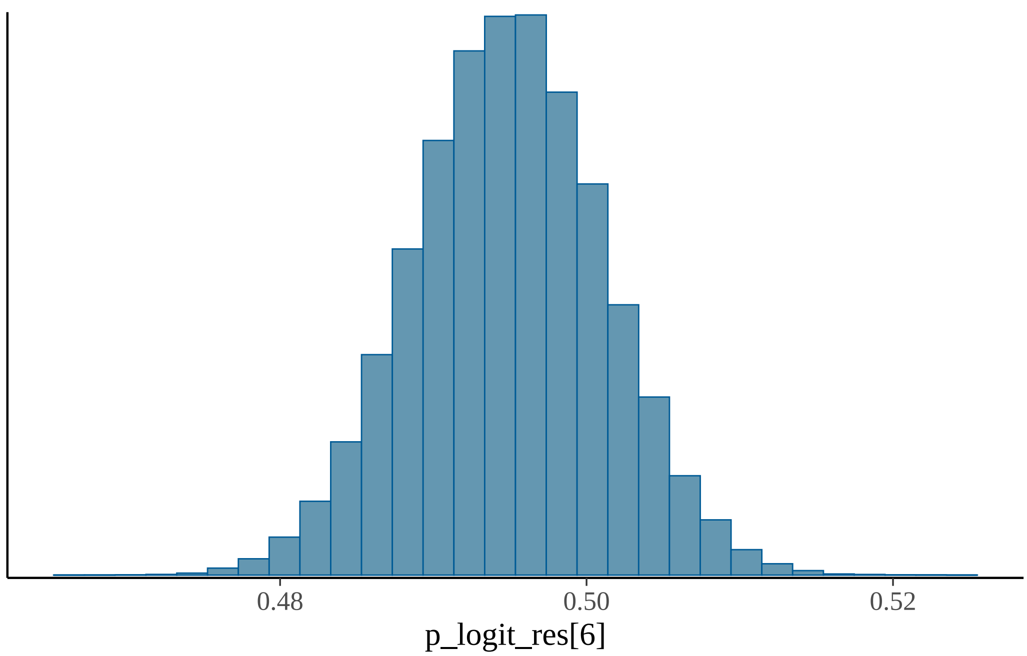
```r
data_list_5 <- list(
  N = nrow(mortality),
  x = mortality$x,
  y = mortality$y,
  n = mortality$n,
  new_n = 100,
  new_x = 200
)
fit5 <- modelo_5$sample(
  data = data_list_5,
  chains = 2,
  iter_sampling = 50000,
  iter_warmup = 5000,
show_messages = FALSE,
show_exceptions = FALSE)
fit5$summary()
```

```
# A tibble: 17 x 10
   variable              mean    median        sd       mad        q5       q95  rhat
   <chr>                <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl> <dbl>
 1 lp__              -6.16e+2 -6.16e+2  1.00e+0  7.13e-1 -6.18e+2 -6.15e+2  1.00
 2 beta0             -4.23e-6 -4.15e-6  9.99e-5  1.00e-4 -1.68e-4  1.60e-4  1.00
 3 beta1             -8.17e-5 -8.14e-5  9.94e-5  9.93e-5 -2.45e-4  8.14e-5  1.00
 4 p_logit_res[1]     5.00e-1  5.00e-1  2.50e-5  2.52e-5  5.00e-1  5.00e-1  1.00
 5 p_logit_res[2]     5.00e-1  5.00e-1  1.27e-4  1.26e-4  5.00e-1  5.00e-1  1.00
 6 p_logit_res[3]     4.99e-1  4.99e-1  7.46e-4  7.46e-4  4.98e-1  5.01e-1  1.00
 7 p_logit_res[4]     4.98e-1  4.98e-1  1.86e-3  1.86e-3  4.95e-1  5.02e-1  1.00
 8 p_logit_res[5]     4.97e-1  4.97e-1  3.73e-3  3.73e-3  4.91e-1  5.03e-1  1.00
 9 p_logit_res[6]     4.95e-1  4.95e-1  6.21e-3  6.21e-3  4.85e-1  5.05e-1  1.00
10 pf                 4.96e-1  4.96e-1  4.97e-3  4.97e-3  4.88e-1  5.04e-1  1.00
11 new_y[1]         NaN       NA        NA        NA        NA        NA        NA
```

```
12 new_y[2]          NaN       NA        NA        NA        NA        NA        NA
13 new_y[3]          NaN       NA        NA        NA        NA        NA        NA
14 new_y[4]          NaN       NA        NA        NA        NA        NA        NA
15 new_y[5]          NaN       NA        NA        NA        NA        NA        NA
16 new_y[6]          NaN       NA        NA        NA        NA        NA        NA
17 pred              NaN       NA        NA        NA        NA        NA        NA
# i 2 more variables: ess_bulk <dbl>, ess_tail <dbl>
```

```r
mcmc_hist(fit5$draws("p_logit_res[6]"))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
sims <- fit5$draws(c("pf"), format = "df")
pred <- 0
for (i in 1:length(sims)){
  pred = pred + sum(rbinom(100, 1, sims[[1]][i]))
}
pred/length(sims)
```

```
[1] 48.5
```

Se calculan alrededor de 51 muertes de los 100 mineros expuestos 200 horas al mineral

```
library(boot)
data(coal)
coal <- coal %>% mutate(year=as.integer(substr(date,1,4)),
                accidents=as.integer(substr(date, 6,9))) %>%
  group_by(year) %>%
  summarise(accidents=sum(accidents))
year_all <- seq(min(coal$year), max(coal$year))
coal2<-data.frame(year=year_all)
coal<-merge(x=coal2, y=coal, by='year', all.x = TRUE) %>% mutate()
coal[is.na(coal)] <- 0
```

En el mismo contexto del problema enunciado (que hicimos en la última clase), supongamos ahora que la compañía de seguros está interesada en modelar el número total de desastres (Yt) que ocurren en la mina. Se cuenta con N = 112 observaciones durante los años 1851 a 1962. Se proponen tres modelos:

i. Modelo con tasa variable en función del tiempo:

$$Y_t|\mu_t \sim Poi(\mu_t)$$

$$log(\mu_t) = \beta_0 + \beta_1 x_t$$

con $\beta_0 \sim N(0, 0.001)$ y $\beta_1 \sim N(0, 0.001)$.

```
modelo_5i <- cmdstan_model("./parcial1ej5i.stan")
print(modelo_5i)
```

```
data {
  int<lower=0> N;           // Number of observations
  array[N] int y;       // Observed values
  vector<lower=0>[N] x;               // Predictor variable
}

parameters {
  real beta0;               // Intercept
  real beta1;               // Slope
}

transformed parameters {
  vector[N] mu;              // Expected values
  for (i in 1:N) {
```

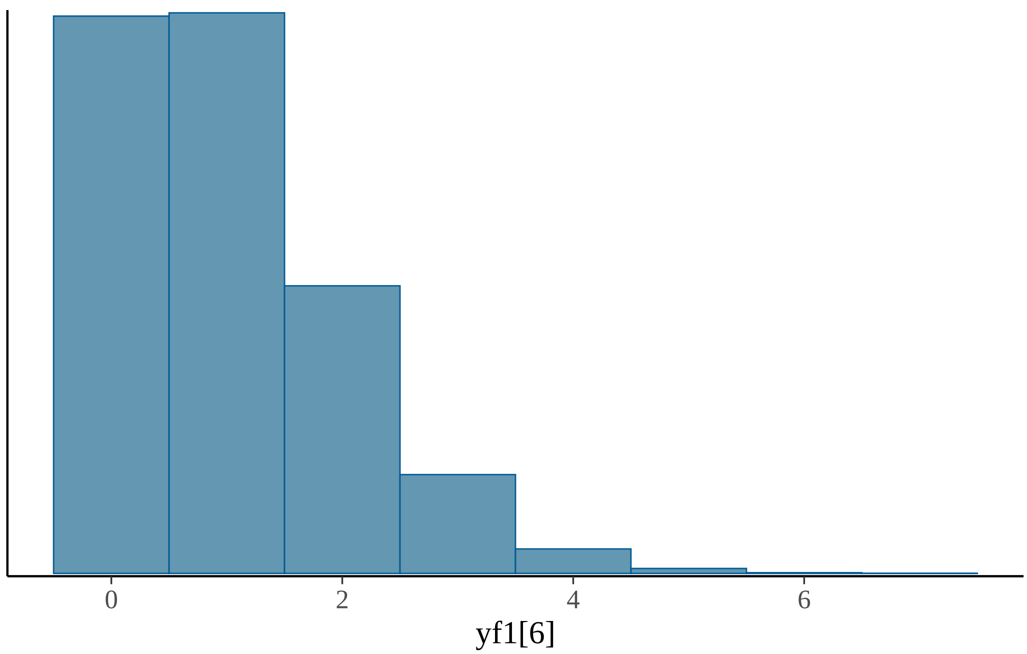
```
    mu[i] = exp(beta0 + beta1 * x[i]);  // Link function: Poisson(mu)
  }
}

model {
  // Likelihood
  for (i in 1:N) {
    y[i] ~ poisson(mu[i]);
  }

  // Priors
  beta0 ~ normal(0, 0.0001);
  beta1 ~ normal(0, 0.0001);
}

generated quantities {
  array[N] int yf1;              // Generated predictions for yf1
  for (i in 1:N) {
    yf1[i] = poisson_rng(mu[i]);
  }
}
```

```r
data_list_5i <- list(
  N = nrow(coal),
  x = coal$year,
  y = coal$accidents
)
fit5i <- modelo_5i$sample(
  data = data_list_5i,
  chains = 2,
  iter_sampling = 50000,
  iter_warmup = 5000,
show_messages = FALSE,
show_exceptions = FALSE)
fit5i$summary()
```

```
# A tibble: 227 x 10
   variable     mean   median        sd      mad        q5      q95  rhat ess_bulk
   <chr>       <dbl>    <dbl>     <dbl>    <dbl>     <dbl>    <dbl> <dbl>    <dbl>
 1 lp__      7.74e+6 7.74e+6 0.852          0      7.74e+6 7.74e+6  1.00   57060.
 2 beta0     1.23e-4 1.23e-4 0.000101     1.01e-4 -4.28e-5 2.89e-4  1.00   34969.
```

```
 3 beta1    4.74e-3 4.74e-3 0.000000537  5.34e-7  4.74e-3 4.74e-3  1.00   82128.
 4 mu[1]    6.50e+3 6.50e+3 6.43         6.42e+0  6.49e+3 6.52e+3  1.00   84192.
 5 mu[2]    6.54e+3 6.54e+3 6.46         6.46e+0  6.52e+3 6.55e+3  1.00   84192.
 6 mu[3]    6.57e+3 6.57e+3 6.50         6.49e+0  6.56e+3 6.58e+3  1.00   84190.
 7 mu[4]    6.60e+3 6.60e+3 6.53         6.52e+0  6.59e+3 6.61e+3  1.00   84190.
 8 mu[5]    6.63e+3 6.63e+3 6.57         6.57e+0  6.62e+3 6.64e+3  1.00   84189.
 9 mu[6]    6.66e+3 6.66e+3 6.60         6.60e+0  6.65e+3 6.67e+3  1.00   84190.
10 mu[7]    6.69e+3 6.69e+3 6.64         6.63e+0  6.68e+3 6.70e+3  1.00   84188.
# i 217 more rows
# i 1 more variable: ess_tail <dbl>
```

```r
mcmc_hist(fit5i$draws("yf1[6]"), binwidth = 1)
```



ii. Modelo con tasa constante en dos períodos: Se cree que la tasa promedio de desastres es constante, pero que en el siglo XX la tasa ha disminuido.

Esto se traduce en el siguiente modelo:

$$Y_t|\mu_t \sim Poi(\mu_t)$$

$$log(\mu_t) = \beta_0 + \beta_1 I(t \geq \tau)$$

con $\beta_0 \sim N(0, 0.001)$ y $\beta_1 \sim N(0, 0.001)$ y $\tau \sim U\{1, ..., N\}$.

```r
modelo_5ii <- cmdstan_model("./parcial1ej5ii.stan")
print(modelo_5ii)
```

```
data {
  real<lower=0> r_e;
  real<lower=0> r_l;

  int<lower=1> T;
  array[T] int<lower=0> D;
}
transformed data {
  real log_unif;
  log_unif = -log(T);
}
parameters {
  real<lower=0> e;
  real<lower=0> l;
}
transformed parameters {
  vector[T] lp;
  lp = rep_vector(log_unif, T);
  for (s in 1:T) {
    for (t in 1:T) {
      lp[s] = lp[s] + poisson_lpmf(D[t] | t < s ? e : l);
    }
  }
}
model {
  e ~ exponential(r_e);
  l ~ exponential(r_l);
  target += log_sum_exp(lp);
}

generated quantities {
  real target_log_prob;
  real log_prob_e = exponential_lpdf(e | r_e);
  real log_prob_l = exponential_lpdf(l | r_l);
  target_log_prob = log_sum_exp(lp) + log_prob_e + log_prob_l;
}
```

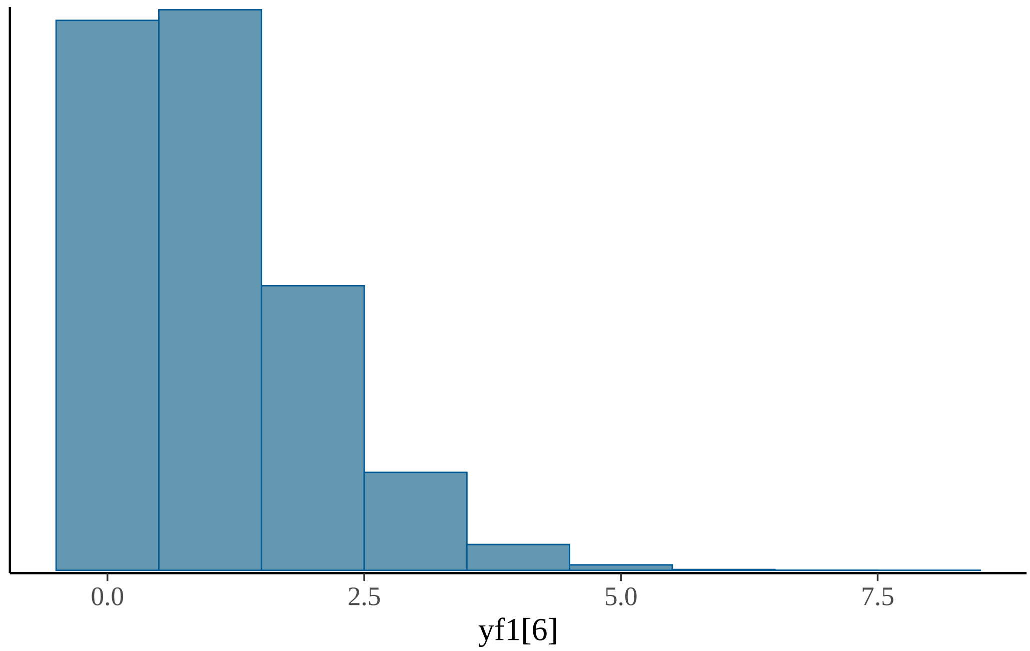
```r
fit5ii <- modelo_5ii$sample(
  data = data_list_5i,
  chains = 2,
  iter_sampling = 50000,
  iter_warmup = 5000,
show_messages = FALSE,
show_exceptions = FALSE)
```

```
Warning: 12482 of 100000 (12.0%) transitions ended with a divergence.
See https://mc-stan.org/misc/warnings for details.
```

```r
fit5ii$summary()
```

```
# A tibble: 228 x 10
   variable       mean     median      sd     mad      q5     q95  rhat ess_bulk
   <chr>         <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>    <dbl>
 1 lp__     9301.      9301.      1.42e+0 1.20e+0 9.30e+3 9.30e+3  1.00   12936.
 2 beta0       0.00971    0.00971 1.00e-4 9.98e-5 9.55e-3 9.88e-3  1.00   97053.
 3 beta1       0.00969    0.00969 1.00e-4 1.00e-4 9.53e-3 9.86e-3  1.00   94086.
 4 tau       111.       111.      5.78e-1 7.43e-1 1.10e+2 1.12e+2  1.00    7833.
 5 mu[1]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
 6 mu[2]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
 7 mu[3]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
 8 mu[4]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
 9 mu[5]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
10 mu[6]       1.02       1.02    1.44e-4 1.48e-4 1.02e+0 1.02e+0  1.00   95087.
# i 218 more rows
# i 1 more variable: ess_tail <dbl>
```

```r
mcmc_hist(fit5ii$draws("yf1[6]"), binwidth = 1)
```


0.0
2.5
5.0
7.5
yf1[6]