Predicting the Lifespan of World Leaders

Cathy Lee, Alice Liao, Ashley Murray, and Matty Pahren

8-30-20

I. Introduction

World leaders are influential people that have the power to impact millions of people's lives. Some leaders are elected for terms while others serve for life, so their expected survival is of great interest. In this paper, we seek to examine Popes, US Presidents, Dalai Lamas, Chinese Emperors, and Japanese Emperors to see how their lifespans compare. Additionally, we want to analyze the impact of a leader's birth year on survival, and whether or not this impact might vary depending on the type of leadership. After we build a model to analyze the effects, we also wish to obtain predictions for the lifespans of leaders that are currently alive, as well as make comparative statements about the survival times of these leaders. We will accomplish this by using survival analysis. We use Bayesian Inference to estimate the model parameters with the help of the JAGS program.

The rest of the paper is structured as follows. First, we will describe the data used to carry out our analysis. Then, we will discuss the Weibull model and Bayesian framework used for analysis to ground our analysis plan. Then, we will show how we carried out our analysis plan and recount our results. After that, we discuss our conclusions and areas for further research. Finally, our code and some additional information can be found in the Appendix.

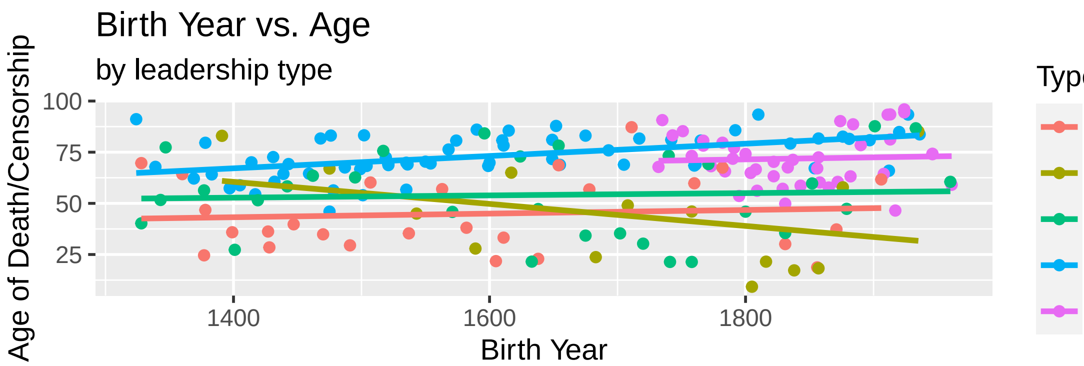
II. Data

II a. Description of the Data

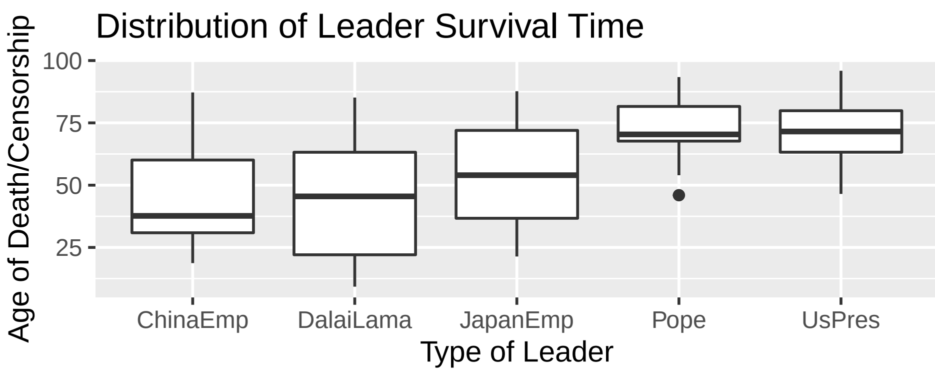
The data used in this analysis contains entries for 177 different world leaders. The types of world leaders present in the data include Popes, US Presidents, Dalai Lamas, Chinese Emperors, and Japanese Emperors. Each individuals's birth date and their leadership position is recorded. We calculated each leader's birth year from the birth date column. For some of these groups, we have data dating all the way back to the 14th century. Additionally, leaders who have passed away also have their death date and age of death recorded. For leaders that are still living, these columns instead contain the date the dataset was created (July 31, 2020) and their current age on that date. We updated that date to the due date of the report, August 31, 2020. In order to further clarify who is dead or alive, there is a column titled "Censored," which takes a value of 0 if the person is dead and a value of 1 if that person is still alive. Related to this, there is another column called "Fail," which takes on a value of 1 if the person is dead and a value of 0 if the person is alive. The problem with censored data is that we don't know exactly when a person will die. Thus, we have to come up with a way to model a censored person's death date if we use them in building our model. In total, there are 10 living leaders in our dataset, 4 of whose age of death we are trying to predict.

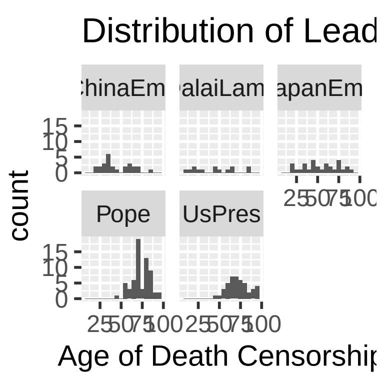
II b. Exploratory Data Analysis

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



Based on this scatterplot, it appears that birth year's association with lifespan does depends on type of leadership.





From the boxplot, the distribution of the lifespans of Popes and US Presidents is centered higher, which makes intuitive sense given that these leaders are elected later in life, while Dalai Lamas and Emperors can be elected at much younger ages. For instance, if a person is a toddler and dies at age five, that person can never be president (and therefore would not have their age be part of the dataset). However, that person can become a Dalai Lama as a toddler and have their age at death influence the distribution plotted. These same patterns can also be observed in the histograms. Also, in the histograms, there is no extreme skewing, so it does not appear tha log transformations are needed.

III. Methods

III a. Motivating the Model

We use survival analysis as a way to model T_i, the lifespan of a given leader i depending on their year of birth and type of leadership. Survival analysis is useful in that it allows the consideration of "censored" data. This means that we do not always observe the outcome for each data point. For example, we do not know the death date of leaders that are still alive. Thus, we do not know their lifespan, we just know that their survival time T_i will be greater than their current age.

We model the lifespan T_i of an individual i after the Weibull distribution specified below. We choose to use the Weibull distribution because it is often utilized in survival analysis and allows the user to specify a flexible shape parameter for the distribution. The first parameter r, also known as the shape parameter, is a positive scalar, and the second parameter μ, the scale parameter, is a linear function of the covariates (in this case, the century of the birth year and types of leadership as well as their interactions).

$T_i \sim Weibull(r, \mu_i)$

$\log(\mu_i) = \beta_0 + \sum_{j=1,\cdots,6} \beta_j I(Birth\ Century_i = j) + \sum_{k=7,\cdots,10} \beta_k I(Leadership_i = k)$
$+ \sum_{j=4,\cdots,6,\ k=7} \beta_{j,k} I(Birth\ Century_i = j) * I(Leadership_i = k)$
$+ \sum_{j=1,\cdots,6,\ k=8,\cdots,10} \beta_{j,k} I(Birth\ Century_i = j) * I(Leadership_i = k)$

where the indicator functions $Birth\ Century_i = 1, \cdots, 6$ corresponds to a leader i born in the 15th, 16th, 17th, 18th, 19th and 20th century respectively. The 14th century is the baseline for comparison. $Leadership_i = 7, \cdots, 10$ corresponds to a leader i being a U.S. President, a Chinese Emperor, a Dalai Lamai and a Japanese Emperor respectively. $Leadership_i$ uses Pope the baseline for comparison. $\beta_{j,k}$ is the parameter for the interaction effect between brith year century j and leadership type k.

For example, for Pope Francis who was born in 1963, his birth year century is the 20th century and his leadership type is Pope. $I(Birth\ Century_{Francis} = j)$ will all be 0 except for $j = 6$ (which indicates he was born in the 20th century) and $I(Leadership_i = k)$ will all be 0 since Pope is the baseline for comparison for leadership types. We can write $\log(\mu_{Francis}) = \beta_0 + \beta_6$ for Pope Francis.

Note that the first U.S. President, George Washington, was born in 1732 (18th century, $j = 4$) and hence the interactions between $Birth\ Century_i = 1, 2, 3$ (born in the 15th century, 16th century or 17th century) and $Leadership_7$ (being a U.S. President) are meaningless. As such, we didn't include those interaction terms in our model.

Prior to using this model, we considered a model that used leaders' birth years as a continuous variable instead of binning them into centuries. However, that model resulted in a less desirable model predictive ability as we ran model diagnostic. The residual plot showed a clear trend between residuals and predicted lifespans. Using a quadratic polynomial improved the residuals, but our current model is still superior. Model diagnostic of the current model can be found in section V.c and residual plots of previous models with continuous birth year variable can be found in the Appendix. *make sure we include them in the Appendix*

III b. Addressing Censored Data

To handle censored observations, we specify their contribution to the likelihood function using the Poisson "zeros trick." Details on this method and how to implement it can be found in Appendix A.1.

III c. Prior Choice

We assume that our priors are independent because intuitively a given observation could not be two types of leaders. Additionally, we do not have sufficiently strong prior knowledge about the relationship between birth year and type of leadership to specify an informative prior. Therefore, we use uninformative priors for the betas in our model. For our prior for r, we chose a prior of $exp(1)$, as we believe the hazard of death increases with time.

```
params_to_output = c("beta_0", "beta_1", "beta_2", "beta_3", "beta_4", "beta_5", "beta_6", "beta_7", "be
```

```
priors = "
  beta_0 ~ dnorm(0.0, 1.0E-3) # priors on betas are all normal w/ low precision
  beta_1 ~ dnorm(0.0, 1.0E-3)
  beta_2 ~ dnorm(0.0, 1.0E-3)
```

```
  beta_3 ~ dnorm(0.0, 1.0E-3)
  beta_4 ~ dnorm(0.0, 1.0E-3)
  beta_5 ~ dnorm(0.0, 1.0E-3)
  beta_6 ~ dnorm(0.0, 1.0E-3)
  beta_7 ~ dnorm(0.0, 1.0E-3)
  beta_8 ~ dnorm(0.0, 1.0E-3)
  beta_9 ~ dnorm(0.0, 1.0E-3)
  beta_10 ~ dnorm(0.0, 1.0E-3)
  beta_1_8 ~ dnorm(0.0, 1.0E-3)
  beta_1_9 ~ dnorm(0.0, 1.0E-3)
  beta_1_10 ~ dnorm(0.0, 1.0E-3)

  beta_2_8 ~ dnorm(0.0, 1.0E-3)
  beta_2_9 ~ dnorm(0.0, 1.0E-3)
  beta_2_10 ~ dnorm(0.0, 1.0E-3)

  beta_3_8 ~ dnorm(0.0, 1.0E-3)
  beta_3_9 ~ dnorm(0.0, 1.0E-3)
  beta_3_10 ~ dnorm(0.0, 1.0E-3)
  beta_4_7 ~ dnorm(0.0, 1.0E-3)
  beta_4_8 ~ dnorm(0.0, 1.0E-3)
  beta_4_9 ~ dnorm(0.0, 1.0E-3)
  beta_4_10 ~ dnorm(0.0, 1.0E-3)
  beta_5_7 ~ dnorm(0.0, 1.0E-3)
  beta_5_8 ~ dnorm(0.0, 1.0E-3)
  beta_5_9 ~ dnorm(0.0, 1.0E-3)
  beta_5_10 ~ dnorm(0.0, 1.0E-3)
  beta_6_7 ~ dnorm(0.0, 1.0E-3)
  beta_6_8 ~ dnorm(0.0, 1.0E-3)
  beta_6_9 ~ dnorm(0.0, 1.0E-3)
  beta_6_10 ~ dnorm(0.0, 1.0E-3)

  r ~ dexp(0.1) # Prior on r
  "
```

```
model_text = function(priors=""){
  file <- tempfile()

  likelihood = "#set up likelihood
  for(i in 1:n_censored) {
    z_censored[i] ~ dpois(phi_censored[i])
    phi_censored[i] <- mu_censored[i] * pow(t_censored[i], r)
    mu_censored[i] <- exp(beta_censored[i])
    beta_censored[i] <- beta_0 + beta_1*x_1_censored[i] +
      beta_2*x_2_censored[i] + beta_3*x_3_censored[i] + beta_4*x_4_censored[i] + beta_5*x_5_censored[i]

     beta_1_8*x_1_8_censored[i] +  beta_1_9*x_1_9_censored[i] + beta_1_10*x_1_10_censored[i] +


          beta_2_8*x_2_8_censored[i] +
          beta_2_9*x_2_9_censored[i] +
          beta_2_10*x_2_10_censored[i] +
```

```
        beta_3_8*x_3_8_censored[i] +
        beta_3_9*x_3_9_censored[i] +
        beta_3_10*x_3_10_censored[i] +

        beta_4_7*x_4_7_censored[i] +
        beta_4_8*x_4_8_censored[i] +
        beta_4_9*x_4_9_censored[i] +
        beta_4_10*x_4_10_censored[i] +

        beta_5_7*x_5_7_censored[i] +
        beta_5_8*x_5_8_censored[i] +
        beta_5_9*x_5_9_censored[i] +
        beta_5_10*x_5_10_censored[i] +

        beta_6_7*x_6_7_censored[i] +
        beta_6_8*x_6_8_censored[i] +
        beta_6_9*x_6_9_censored[i] +
        beta_6_10*x_6_10_censored[i]

  }

  for(j in 1:n_non_censored) { #total rows - 6 ** 6 are censored in leaders_nopred
    survival_non_censored[j] ~ dweib(r, mu[j])
    mu[j] <- exp(beta[j])
    beta[j] <- beta_0 + beta_1*x_1_non_censored[j] +
      beta_2*x_2_non_censored[j] + beta_3*x_3_non_censored[j] + beta_4*x_4_non_censored[j] + beta_5*x_5

     + beta_1_8*x_1_8_non_censored[j] +  beta_1_9*x_1_9_non_censored[j] + beta_1_10*x_1_10_non_censored


        beta_2_8*x_2_8_non_censored[j] +
        beta_2_9*x_2_9_non_censored[j] +
        beta_2_10*x_2_10_non_censored[j] +


        beta_3_8*x_3_8_non_censored[j] +
        beta_3_9*x_3_9_non_censored[j] +
        beta_3_10*x_3_10_non_censored[j] +

        beta_4_7*x_4_7_non_censored[j] +
        beta_4_8*x_4_8_non_censored[j] +
        beta_4_9*x_4_9_non_censored[j] +
        beta_4_10*x_4_10_non_censored[j] +

        beta_5_7*x_5_7_non_censored[j] +
        beta_5_8*x_5_8_non_censored[j] +
        beta_5_9*x_5_9_non_censored[j] +
        beta_5_10*x_5_10_non_censored[j] +

        beta_6_7*x_6_7_non_censored[j] +
        beta_6_8*x_6_8_non_censored[j] +
        beta_6_9*x_6_9_non_censored[j] +
        beta_6_10*x_6_10_non_censored[j]
```

```
}
"

alphas_percentage_pred = "# set up alphas and percentages for interpret
# define alphas
alpha_0 <- - beta_0 / r
alpha_1 <- - beta_1 / r
alpha_2 <- - beta_2 / r
alpha_3 <- - beta_3 / r
alpha_4 <- - beta_4 / r
alpha_5 <- - beta_5 / r
alpha_6 <- - beta_6 / r
alpha_7 <- - beta_7 / r
alpha_8 <- - beta_8 / r
alpha_9 <- - beta_9 / r
alpha_10 <- - beta_10 / r


alpha_1_8 <- - beta_1_8 / r
alpha_1_9 <- - beta_1_9 / r
alpha_1_10 <- - beta_1_10 / r


alpha_2_8 <- - beta_2_8 / r
alpha_2_9 <- - beta_2_9 / r
alpha_2_10 <- - beta_2_10 / r


alpha_3_8 <- - beta_3_8 / r
alpha_3_9 <- - beta_3_9 / r
alpha_3_10 <- - beta_3_10 / r

alpha_4_7 <- - beta_4_7 / r
alpha_4_8 <- - beta_4_8 / r
alpha_4_9 <- - beta_4_9 / r
alpha_4_10 <- - beta_4_10 / r

alpha_5_7 <- - beta_5_7 / r
alpha_5_8 <- - beta_5_8 / r
alpha_5_9 <- - beta_5_9 / r
alpha_5_10 <- - beta_5_10 / r

alpha_6_7 <- - beta_6_7 / r
alpha_6_8 <- - beta_6_8 / r
alpha_6_9 <- - beta_6_9 / r
alpha_6_10 <- - beta_6_10 / r


# Percentage increases
p_i_Yr15 <- 100*(exp(alpha_1) - 1)
p_i_Yr16 <- 100*(exp(alpha_2) - 1)
p_i_Yr17 <- 100*(exp(alpha_3) - 1)
p_i_Yr18 <- 100*(exp(alpha_4) - 1)
```

```
p_i_Yr19 <- 100*(exp(alpha_5) - 1)
p_i_Yr20 <- 100*(exp(alpha_6) - 1)
p_i_UsPres <- 100*(exp(alpha_7) - 1)
p_i_ChinaEmp <- 100*(exp(alpha_8) - 1)
p_i_DalaiLama <- 100*(exp(alpha_9) - 1)
p_i_JapanEmp <- 100*(exp(alpha_10) - 1)

p_i_Yr15ChinaEmp <- 100*(exp(alpha_1_8) - 1)
p_i_Yr15DalaiLama <- 100*(exp(alpha_1_9) - 1)
p_i_Yr15JapanEmp <- 100*(exp(alpha_1_10) - 1)


p_i_Yr16ChinaEmp <- 100*(exp(alpha_2_8) - 1)
p_i_Yr16DalaiLama <- 100*(exp(alpha_2_9) - 1)
p_i_Yr16JapanEmp <- 100*(exp(alpha_2_10) - 1)


p_i_Yr17ChinaEmp <- 100*(exp(alpha_3_8) - 1)
p_i_Yr17DalaiLama <- 100*(exp(alpha_3_9) - 1)
p_i_Yr17JapanEmp <- 100*(exp(alpha_3_10) - 1)

p_i_Yr18UsPres <- 100*(exp(alpha_4_7) - 1)
p_i_Yr18ChinaEmp <- 100*(exp(alpha_4_8) - 1)
p_i_Yr18DalaiLama <- 100*(exp(alpha_4_9) - 1)
p_i_Yr18JapanEmp <- 100*(exp(alpha_4_10) - 1)

p_i_Yr19UsPres <- 100*(exp(alpha_5_7) - 1)
p_i_Yr19ChinaEmp <- 100*(exp(alpha_5_8) - 1)
p_i_Yr19DalaiLama <- 100*(exp(alpha_5_9) - 1)
p_i_Yr19JapanEmp <- 100*(exp(alpha_5_10) - 1)

p_i_Yr20UsPres <- 100*(exp(alpha_6_7) - 1)
p_i_Yr20ChinaEmp <- 100*(exp(alpha_6_8) - 1)
p_i_Yr20DalaiLama <- 100*(exp(alpha_6_9) - 1)
p_i_Yr20JapanEmp <- 100*(exp(alpha_6_10) - 1)

# Predictive distribution of age at the new values
beta_Francis <- beta_0 + beta_6
mu_Francis <- exp(beta_Francis)
survival_Francis ~ dweib(r, mu_Francis)T(present_length_Francis, upper_length)
age_Francis_predictive <- survival_Francis

beta_Obama <- beta_0 + beta_6 + beta_7 + beta_6_7
mu_Obama <- exp(beta_Obama)
survival_Obama ~ dweib(r, mu_Obama)T(present_length_Obama, upper_length)
age_Obama_predictive <- survival_Obama

beta_Dalai <- beta_0 + beta_6 + beta_10 + beta_6_10
mu_Dalai <- exp(beta_Dalai)
survival_Dalai ~ dweib(r, mu_Dalai)T(present_length_Dalai, upper_length)
age_Dalai_predictive <- survival_Dalai

beta_Naruhito <- beta_0 + beta_6 + beta_9 + beta_6_9
```

```
  mu_Naruhito <- exp(beta_Naruhito)
  survival_Naruhito ~ dweib(r, mu_Naruhito)T(present_length_Naruhito, upper_length)
  age_Naruhito_predictive <- survival_Naruhito

  beta_Benedict <- beta_0 + beta_6
  mu_Benedict <- exp(beta_Benedict)
  survival_Benedict ~ dweib(r, mu_Benedict)T(present_length_Benedict, upper_length)
  age_Benedict_predictive <- survival_Benedict

  beta_Carter <- beta_0 + beta_6 + beta_7 + beta_6_7
  mu_Carter <- exp(beta_Carter)
  survival_Carter ~ dweib(r, mu_Carter)T(present_length_Carter, upper_length)
  age_Carter_predictive <- survival_Carter

  beta_Clinton <- beta_0 + beta_6 + beta_7 + beta_6_7
  mu_Clinton <- exp(beta_Clinton)
  survival_Clinton ~ dweib(r, mu_Clinton)T(present_length_Clinton, upper_length)
  age_Clinton_predictive <- survival_Clinton

  beta_Bush <- beta_0 + beta_6 + beta_7 + beta_6_7
  mu_Bush <- exp(beta_Bush)
  survival_Bush ~ dweib(r, mu_Bush)T(present_length_Bush, upper_length)
  age_Bush_predictive <- survival_Bush

  beta_Trump <- beta_0 + beta_6 + beta_7 + beta_6_7
  mu_Trump <- exp(beta_Trump)
  survival_Trump ~ dweib(r, mu_Trump)T(present_length_Trump, upper_length)
  age_Trump_predictive <- survival_Trump

  beta_Akihito <- beta_0 + beta_6 + beta_9 + beta_6_9
  mu_Akihito <- exp(beta_Naruhito)
  survival_Akihito ~ dweib(r, mu_Akihito)T(present_length_Akihito, upper_length)
  age_Akihito_predictive <- survival_Akihito
  "

  model_text <- paste("model{ ", likelihood, priors, alphas_percentage_pred, "}")
  writeLines(model_text,con=file)
  return(file)
}
```

```
run_model = function(priors, name) {
  file <- model_text(priors)
  # if (file.exists(name)) {
  #   load(file = name)
  # } else {
    model_output <- jags(data = data_build_model,
                         parameters.to.save =  params_to_output,
                         n.iter = 50000,
                         n.chains = 3,
                         model.file = file)
    save(model_output, file = name)
  #}
  return(model_output)
}
```

```r
model_output = run_model(priors, "model_used_in_paper")
```

```
## module glm loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6044
##
## Initializing model
```

```r
model_output
```

```
## Inference for Bugs model at "/var/folders/hs/d7t9jp1x3t7cwq2xz5rf7f8h0000gn/T//RtmpnV1gX5/file166d83
##  3 chains, each with 50000 iterations (first 25000 discarded), n.thin = 25
##  n.sims = 3000 iterations saved
##                               mu.vect      sd.vect     2.5%      25%      50%
## age_Akihito_predictive   9.407200e+01 3.804000e+00   87.137   90.958   94.340
## age_Benedict_predictive  9.664000e+01 1.914000e+00   93.580   94.969   96.589
## age_Bush_predictive      8.734900e+01 7.308000e+00   74.935   81.194   87.305
## age_Carter_predictive    9.793400e+01 1.172000e+00   96.013   96.926   97.929
## age_Clinton_predictive   8.744100e+01 7.309000e+00   74.809   81.308   87.736
## age_Dalai_predictive     9.266500e+01 4.309000e+00   85.544   88.927   92.776
## age_Francis_predictive   9.191000e+01 4.646000e+00   84.135   87.973   91.874
## age_Naruhito_predictive  8.641000e+01 1.035700e+01   63.258   79.345   88.768
## age_Obama_predictive     8.189700e+01 1.099600e+01   61.262   73.260   82.831
## age_Trump_predictive     8.731100e+01 7.289000e+00   74.883   81.258   87.317
## alpha_0                  4.319000e+00 7.900000e-02    4.181    4.264    4.313
## alpha_1                 -8.700000e-02 9.600000e-02   -0.284   -0.149   -0.085
## alpha_10                -1.530000e-01 1.300000e-01   -0.402   -0.240   -0.155
## alpha_1_10              -1.200000e-02 1.680000e-01   -0.348   -0.121   -0.009
## alpha_1_8               -4.150000e-01 1.610000e-01   -0.740   -0.524   -0.415
## alpha_1_9               -1.410000e-01 3.610000e-01   -0.847   -0.357   -0.141
## alpha_2                 -2.900000e-02 9.200000e-02   -0.219   -0.087   -0.026
## alpha_2_10               2.110000e-01 1.800000e-01   -0.139    0.088    0.211
## alpha_2_8               -6.300000e-02 1.690000e-01   -0.390   -0.177   -0.062
## alpha_2_9               -7.680000e-01 3.000000e-01   -1.419   -0.939   -0.751
## alpha_3                  7.600000e-02 1.020000e-01   -0.130    0.007    0.077
## alpha_3_10              -6.100000e-02 1.690000e-01   -0.400   -0.173   -0.063
## alpha_3_8               -1.430000e-01 1.660000e-01   -0.469   -0.250   -0.147
## alpha_3_9               -5.260000e-01 3.080000e-01   -1.226   -0.706   -0.500
## alpha_4                  4.900000e-02 1.090000e-01   -0.166   -0.019    0.050
## alpha_4_10              -1.610000e-01 1.690000e-01   -0.495   -0.268   -0.155
## alpha_4_7                3.540000e-01 3.597000e+00   -5.901   -2.421   -0.629
## alpha_4_8                2.360000e-01 1.870000e-01   -0.125    0.108    0.232
## alpha_4_9               -6.770000e-01 3.090000e-01   -1.354   -0.846   -0.651
## alpha_5                  1.030000e-01 1.090000e-01   -0.121    0.032    0.104
## alpha_5_10              -2.950000e-01 1.910000e-01   -0.668   -0.424   -0.298
## alpha_5_7                2.030000e-01 3.597000e+00   -6.011   -2.570   -0.769
## alpha_5_8               -6.770000e-01 1.860000e-01   -1.029   -0.800   -0.680
## alpha_5_9               -8.730000e-01 2.820000e-01   -1.539   -1.031   -0.837
```

```
## alpha_6                 3.050000e-01 1.760000e-01    0.008    0.182    0.289
## alpha_6_10              2.540000e-01 3.160000e-01   -0.338    0.052    0.232
## alpha_6_7               2.940000e-01 3.595000e+00   -5.994   -2.482   -0.660
## alpha_6_8              -1.410000e-01 3.170000e-01   -0.729   -0.349   -0.156
## alpha_6_9               4.585000e+00 3.799000e+00   -0.148    1.653    3.726
## alpha_7                -3.720000e-01 3.597000e+00   -5.944   -4.075    0.627
## alpha_8                -2.460000e-01 1.220000e-01   -0.483   -0.327   -0.250
## alpha_9                 2.210000e-01 2.570000e-01   -0.181    0.041    0.181
## beta_0                 -2.261300e+01 1.392000e+00  -25.583  -23.461  -22.603
## beta_1                  4.580000e-01 5.010000e-01   -0.498    0.119    0.444
## beta_10                 7.990000e-01 6.780000e-01   -0.594    0.368    0.811
## beta_1_10               6.300000e-02 8.760000e-01   -1.705   -0.516    0.045
## beta_1_8                2.175000e+00 8.500000e-01    0.528    1.606    2.166
## beta_1_9                7.400000e-01 1.880000e+00   -3.110   -0.393    0.738
## beta_2                  1.510000e-01 4.780000e-01   -0.757   -0.166    0.135
## beta_2_10              -1.103000e+00 9.390000e-01   -3.029   -1.706   -1.092
## beta_2_8                3.290000e-01 8.780000e-01   -1.388   -0.257    0.324
## beta_2_9                4.017000e+00 1.577000e+00    1.166    2.956    3.926
## beta_3                 -3.940000e-01 5.340000e-01   -1.414   -0.764   -0.397
## beta_3_10               3.160000e-01 8.810000e-01   -1.433   -0.268    0.324
## beta_3_8                7.450000e-01 8.640000e-01   -0.984    0.186    0.766
## beta_3_9                2.748000e+00 1.609000e+00   -0.120    1.688    2.596
## beta_4                 -2.560000e-01 5.660000e-01   -1.361   -0.648   -0.262
## beta_4_10               8.380000e-01 8.740000e-01   -0.818    0.243    0.805
## beta_4_7               -1.942000e+00 1.887600e+01  -30.829  -20.830    3.251
## beta_4_8               -1.233000e+00 9.780000e-01   -3.181   -1.875   -1.216
## beta_4_9                3.543000e+00 1.621000e+00    0.600    2.485    3.410
## beta_5                 -5.410000e-01 5.670000e-01   -1.644   -0.927   -0.539
## beta_5_10               1.544000e+00 1.004000e+00   -0.494    0.895    1.560
## beta_5_7               -1.153000e+00 1.886600e+01  -30.238  -19.919    3.943
## beta_5_8                3.542000e+00 9.910000e-01    1.604    2.867    3.555
## beta_5_9                4.563000e+00 1.475000e+00    2.117    3.552    4.373
## beta_6                 -1.593000e+00 9.160000e-01   -3.602   -2.164   -1.505
## beta_6_10              -1.328000e+00 1.654000e+00   -4.909   -2.326   -1.202
## beta_6_7               -1.629000e+00 1.886400e+01  -30.839  -20.436    3.387
## beta_6_8                7.400000e-01 1.652000e+00   -2.871   -0.227    0.816
## beta_6_9               -2.392900e+01 1.973100e+01  -71.421  -34.778  -19.404
## beta_7                  2.036000e+00 1.887700e+01  -30.826  -12.437   -3.192
## beta_8                  1.289000e+00 6.370000e-01   -0.001    0.876    1.306
## beta_9                 -1.156000e+00 1.344000e+00   -4.380   -1.858   -0.943
## p_i_ChinaEmp           -2.125900e+01 9.697000e+00  -38.307  -27.920  -22.113
## p_i_DalaiLama           2.934000e+01 3.877400e+01  -16.594    4.161   19.854
## p_i_JapanEmp           -1.343000e+01 1.145400e+01  -33.070  -21.357  -14.350
## p_i_UsPres              3.553291e+03 1.437921e+04  -99.738  -98.301   87.137
## p_i_Yr15               -7.942000e+00 8.808000e+00  -24.720  -13.878   -8.134
## p_i_Yr15ChinaEmp       -3.314000e+01 1.076100e+01  -52.267  -40.803  -33.986
## p_i_Yr15DalaiLama      -7.171000e+00 3.665400e+01  -57.124  -30.053  -13.192
## p_i_Yr15JapanEmp        2.040000e-01 1.691700e+01  -29.356  -11.374   -0.853
## p_i_Yr16               -2.413000e+00 8.907000e+00  -19.678   -8.306   -2.528
## p_i_Yr16ChinaEmp       -4.741000e+00 1.627600e+01  -32.276  -16.183   -6.019
## p_i_Yr16DalaiLama      -5.153800e+01 1.408600e+01  -75.800  -60.894  -52.831
## p_i_Yr16JapanEmp        2.549700e+01 2.321100e+01  -12.995    9.170   23.542
## p_i_Yr17                8.406000e+00 1.106600e+01  -12.234    0.654    8.025
## p_i_Yr17ChinaEmp       -1.208500e+01 1.471400e+01  -37.407  -22.128  -13.689
```

```
## p_i_Yr17DalaiLama       -3.813300e+01 1.836200e+01  -70.665  -50.639  -39.331
## p_i_Yr17JapanEmp        -4.523000e+00 1.627300e+01  -32.971  -15.927   -6.103
## p_i_Yr18                 5.668000e+00 1.154900e+01  -15.293   -1.930    5.088
## p_i_Yr18ChinaEmp         2.882700e+01 2.458700e+01  -11.748   11.424   26.118
## p_i_Yr18DalaiLama       -4.681800e+01 1.580700e+01  -74.172  -57.069  -47.867
## p_i_Yr18JapanEmp        -1.362300e+01 1.452900e+01  -39.017  -23.519  -14.318
## p_i_Yr18UsPres           4.844190e+03 1.038808e+04  -99.726  -91.119  -46.680
## p_i_Yr19                 1.155900e+01 1.213500e+01  -11.398    3.275   10.922
## p_i_Yr19ChinaEmp        -4.828000e+01 9.761000e+00  -64.253  -55.086  -49.364
## p_i_Yr19DalaiLama       -5.663900e+01 1.132800e+01  -78.540  -64.330  -56.699
## p_i_Yr19JapanEmp        -2.416500e+01 1.477400e+01  -48.720  -34.573  -25.786
## p_i_Yr19UsPres           4.173833e+03 9.028347e+03  -99.755  -92.344  -53.650
## p_i_Yr20                 3.786200e+01 2.603600e+01    0.827   19.982   33.514
## p_i_Yr20ChinaEmp        -8.505000e+00 3.226000e+01  -51.764  -29.442  -14.436
## p_i_Yr20DalaiLama        1.728730e+09 7.219759e+10  -13.795  422.199 4051.756
## p_i_Yr20JapanEmp         3.585000e+01 4.806900e+01  -28.674    5.351   26.148
## p_i_Yr20UsPres           4.622872e+03 1.007858e+04  -99.751  -91.644  -48.291
## r                        5.235000e+00 3.040000e-01    4.627    5.044    5.234
## deviance                 1.391077e+03 8.210000e+00 1377.018 1385.329 1390.294
##                               75%         97.5%  Rhat n.eff
## age_Akihito_predictive     97.503        99.780 1.001  2900
## age_Benedict_predictive    98.303        99.846 1.001  2800
## age_Bush_predictive        93.749        99.320 1.002  1900
## age_Carter_predictive      98.929        99.888 1.001  3000
## age_Clinton_predictive     93.705        99.227 1.001  3000
## age_Dalai_predictive       96.498        99.610 1.001  3000
## age_Francis_predictive     96.017        99.517 1.002  1800
## age_Naruhito_predictive    95.114        99.569 1.001  3000
## age_Obama_predictive       91.062        99.113 1.003   780
## age_Trump_predictive       93.559        99.390 1.001  2100
## alpha_0                     4.367         4.490 1.003   960
## alpha_1                    -0.023         0.095 1.002  1000
## alpha_10                   -0.070         0.117 1.002  2400
## alpha_1_10                  0.098         0.323 1.001  3000
## alpha_1_8                  -0.307        -0.102 1.003   950
## alpha_1_9                   0.075         0.604 1.003   760
## alpha_2                     0.032         0.150 1.003   730
## alpha_2_10                  0.325         0.583 1.002  1700
## alpha_2_8                   0.049         0.274 1.002  1200
## alpha_2_9                  -0.566        -0.225 1.003  1800
## alpha_3                     0.147         0.271 1.002  1300
## alpha_3_10                  0.052         0.275 1.001  3000
## alpha_3_8                  -0.036         0.186 1.002  1700
## alpha_3_9                  -0.320         0.023 1.003  1200
## alpha_4                     0.123         0.266 1.001  3000
## alpha_4_10                 -0.047         0.155 1.002  2600
## alpha_4_7                   4.056         5.907 2.503     4
## alpha_4_8                   0.360         0.603 1.001  3000
## alpha_4_9                  -0.476        -0.120 1.002  1800
## alpha_5                     0.177         0.315 1.003   670
## alpha_5_10                 -0.171         0.093 1.001  3000
## alpha_5_7                   3.905         5.758 2.496     4
## alpha_5_8                  -0.549        -0.312 1.002  1300
## alpha_5_9                  -0.679        -0.408 1.004   570
```

```
## alpha_6                   0.411          0.699 1.004   550
## alpha_6_10                0.446          0.935 1.001  2400
## alpha_6_7                 3.993          5.876 2.487     4
## alpha_6_8                 0.043          0.561 1.002  2100
## alpha_6_9                 6.654         13.877 1.001  3000
## alpha_7                   2.410          5.829 2.499     4
## alpha_8                  -0.168          0.000 1.003   820
## alpha_9                   0.353          0.836 1.005   580
## beta_0                  -21.736        -19.761 1.011   530
## beta_1                    0.786          1.470 1.002  1100
## beta_10                   1.254          2.083 1.001  2900
## beta_1_10                 0.627          1.823 1.001  2900
## beta_1_8                  2.742          3.881 1.004   640
## beta_1_9                  1.874          4.458 1.004   600
## beta_2                    0.458          1.138 1.003   740
## beta_2_10                -0.456          0.712 1.002  2000
## beta_2_8                  0.918          2.063 1.002  1100
## beta_2_9                  4.906          7.512 1.005   870
## beta_3                   -0.034          0.682 1.002  1100
## beta_3_10                 0.902          2.048 1.001  3000
## beta_3_8                  1.315          2.446 1.002  1300
## beta_3_9                  3.684          6.388 1.005   730
## beta_4                    0.105          0.875 1.001  3000
## beta_4_10                 1.405          2.550 1.002  3000
## beta_4_7                 12.538         30.941 2.498     4
## beta_4_8                 -0.567          0.642 1.001  3000
## beta_4_9                  4.444          7.254 1.004   880
## beta_5                   -0.167          0.622 1.004   580
## beta_5_10                 2.225          3.479 1.001  3000
## beta_5_7                 13.286         31.363 2.491     4
## beta_5_8                  4.214          5.505 1.004   610
## beta_5_9                  5.385          8.026 1.005   470
## beta_6                   -0.962         -0.044 1.005   450
## beta_6_10                -0.278          1.734 1.001  2900
## beta_6_7                 12.769         31.285 2.482     4
## beta_6_8                  1.805          3.772 1.002  1800
## beta_6_9                 -8.695          0.780 1.001  2800
## beta_7                   20.942         30.994 2.494     4
## beta_8                    1.717          2.543 1.002  1100
## beta_9                   -0.218          0.954 1.007   450
## p_i_ChinaEmp            -15.446          0.014 1.003   800
## p_i_DalaiLama            42.347        130.597 1.010   520
## p_i_JapanEmp             -6.750         12.386 1.002  2300
## p_i_UsPres             1013.173      33894.317 1.221    49
## p_i_Yr15                 -2.275         10.013 1.003   960
## p_i_Yr15ChinaEmp        -26.404         -9.728 1.002  1000
## p_i_Yr15DalaiLama         7.808         82.901 1.002  1100
## p_i_Yr15JapanEmp         10.282         38.092 1.001  3000
## p_i_Yr16                  3.218         16.161 1.003   710
## p_i_Yr16ChinaEmp          5.027         31.581 1.002  1100
## p_i_Yr16DalaiLama       -43.244        -20.118 1.001  3000
## p_i_Yr16JapanEmp         38.385         79.072 1.003  1600
## p_i_Yr17                 15.840         31.104 1.002  1400
## p_i_Yr17ChinaEmp         -3.564         20.385 1.002  1600
```

```
## p_i_Yr17DalaiLama         -27.408         2.288 1.002  1800
## p_i_Yr17JapanEmp            5.312        31.697 1.001  3000
## p_i_Yr18                   13.058        30.526 1.001  3000
## p_i_Yr18ChinaEmp           43.343        82.735 1.001  3000
## p_i_Yr18DalaiLama         -37.869       -11.319 1.001  2900
## p_i_Yr18JapanEmp           -4.600        16.757 1.002  3000
## p_i_Yr18UsPres           5672.726     36669.006 1.617     8
## p_i_Yr19                   19.404        37.039 1.003   670
## p_i_Yr19ChinaEmp          -42.243       -26.803 1.002  1300
## p_i_Yr19DalaiLama         -49.296       -33.513 1.004   660
## p_i_Yr19JapanEmp          -15.757         9.713 1.001  3000
## p_i_Yr19UsPres           4866.999     31562.512 1.607     8
## p_i_Yr20                   50.792       101.192 1.004   560
## p_i_Yr20ChinaEmp            4.426        75.316 1.003  3000
## p_i_Yr20DalaiLama       77515.758 106362753.560 1.277  3000
## p_i_Yr20JapanEmp           56.157       154.817 1.004  1700
## p_i_Yr20UsPres           5324.166     35520.357 1.591     8
## r                           5.414         5.891 1.014   400
## deviance                 1396.281      1409.331 1.001  3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 33.7 and DIC = 1424.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

IV. Analysis

As Stander et al (2018) pointed out, following this Weibull distribution, $log(T_i)$ is equal in distribution to $\frac{1}{r}(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_9 x_{i1} x_{i5}) + \frac{1}{r}\log(\epsilon)$ where $\epsilon \sim exp(1)$. $\alpha_j = -\beta_j/r,\ j = 1, 2, ..., 9$. The interpretation of coefficients depends on the interaction terms (i.e. both year of birth year century and the types of leadership).

For example, while keeping all others constant, if a Pope is born one year later, he is expected to live longer by a multiplicative factor of $exp(\alpha_1)$, or his lifespan is expected to increase by a percentage of $100 * exp(\alpha_1 - 1)$. If a U.S. President is born one year later, he is expected to live longer by a multiplicative factor of $exp(\alpha_1 + \alpha_6)$. If a U.S. President and a Chinese Emperor are born in the same year, y after 1677, the Chinese emperor is expected to live longer by a multiplicative factor of $exp(\alpha_3 - \alpha_2 + (\alpha_7 - \alpha_6) * y)$ and so on.

V. Results

V a. Output and Interpretation of the Model

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
beta_0	-22.613	1.392	-25.583	-22.603	-19.761
beta_1	0.458	0.501	-0.498	0.444	1.470
beta_2	0.799	0.678	-0.594	0.811	2.083
beta_3	0.063	0.876	-1.705	0.045	1.823
beta_4	2.175	0.850	0.528	2.166	3.881
beta_5	0.740	1.880	-3.110	0.738	4.458
beta_6	0.151	0.478	-0.757	0.135	1.138
beta_7	-1.103	0.939	-3.029	-1.092	0.712

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
beta_8	0.329	0.878	-1.388	0.324	2.063
beta_9	4.017	1.577	1.166	3.926	7.512
beta_10	-0.394	0.534	-1.414	-0.397	0.682
beta_1_8	0.316	0.881	-1.433	0.324	2.048
beta_1_9	0.745	0.864	-0.984	0.766	2.446
beta_1_10	2.748	1.609	-0.120	2.596	6.388
beta_2_8	-0.256	0.566	-1.361	-0.262	0.875
beta_2_9	0.838	0.874	-0.818	0.805	2.550
beta_2_10	-1.942	18.876	-30.829	3.251	30.941
beta_3_8	-1.233	0.978	-3.181	-1.216	0.642
beta_3_9	3.543	1.621	0.600	3.410	7.254
beta_3_10	-0.541	0.567	-1.644	-0.539	0.622
beta_4_7	1.544	1.004	-0.494	1.560	3.479
beta_4_8	-1.153	18.866	-30.238	3.943	31.363
beta_4_9	3.542	0.991	1.604	3.555	5.505
beta_4_10	4.563	1.475	2.117	4.373	8.026
beta_5_7	-1.593	0.916	-3.602	-1.505	-0.044
beta_5_8	-1.328	1.654	-4.909	-1.202	1.734
beta_5_9	-1.629	18.864	-30.839	3.387	31.285
beta_5_10	0.740	1.652	-2.871	0.816	3.772
beta_6_7	-23.929	19.731	-71.421	-19.404	0.780
beta_6_8	2.036	18.877	-30.826	-3.192	30.994
beta_6_9	1.289	0.637	-0.001	1.306	2.543
beta_6_10	-1.156	1.344	-4.380	-0.943	0.954
r	5.235	0.304	4.627	5.234	5.891

While we may expect leaders who were born earlier to live a shorter life, $\hat{\beta}_1$ which correpsond to the leaders' year of birth appears to be an insignificant parameter as it is very close to 0 (-0.0016) although statistically significant. $\hat{\beta}_6$ to $\hat{\beta}_9$ which correspond to the interaction effects are very weak and mostly insignificant as well. Types of leadership ($\hat{\beta}_2$ to $\hat{\beta}_5$) do seem to be significant. As explained in Section IV, the model is easier to understand if we interpret the output in terms of α_j, defined as $\alpha_j = -\beta_j/r$, $j = 1, 2, ..., 9$. Some examples of the interpretation are provided below.

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
alpha_0	4.319	0.079	4.181	4.313	4.490
alpha_1	-0.087	0.096	-0.284	-0.085	0.095
alpha_2	-0.153	0.130	-0.402	-0.155	0.117
alpha_3	-0.012	0.168	-0.348	-0.009	0.323
alpha_4	-0.415	0.161	-0.740	-0.415	-0.102
alpha_5	-0.141	0.361	-0.847	-0.141	0.604
alpha_6	-0.029	0.092	-0.219	-0.026	0.150
alpha_7	0.211	0.180	-0.139	0.211	0.583
alpha_8	-0.063	0.169	-0.390	-0.062	0.274
alpha_9	-0.768	0.300	-1.419	-0.751	-0.225
alpha_10	0.076	0.102	-0.130	0.077	0.271
alpha_1_8	-0.061	0.169	-0.400	-0.063	0.275
alpha_1_9	-0.143	0.166	-0.469	-0.147	0.186
alpha_1_10	-0.526	0.308	-1.226	-0.500	0.023
alpha_2_8	0.049	0.109	-0.166	0.050	0.266
alpha_2_9	-0.161	0.169	-0.495	-0.155	0.155
alpha_2_10	0.354	3.597	-5.901	-0.629	5.907

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
alpha_3_8	0.236	0.187	-0.125	0.232	0.603
alpha_3_9	-0.677	0.309	-1.354	-0.651	-0.120
alpha_3_10	0.103	0.109	-0.121	0.104	0.315
alpha_4_7	-0.295	0.191	-0.668	-0.298	0.093
alpha_4_8	0.203	3.597	-6.011	-0.769	5.758
alpha_4_9	-0.677	0.186	-1.029	-0.680	-0.312
alpha_4_10	-0.873	0.282	-1.539	-0.837	-0.408
alpha_5_7	0.305	0.176	0.008	0.289	0.699
alpha_5_8	0.254	0.316	-0.338	0.232	0.935
alpha_5_9	0.294	3.595	-5.994	-0.660	5.876
alpha_5_10	-0.141	0.317	-0.729	-0.156	0.561
alpha_6_7	4.585	3.799	-0.148	3.726	13.877
alpha_6_8	-0.372	3.597	-5.944	0.627	5.829
alpha_6_9	-0.246	0.122	-0.483	-0.250	0.000
alpha_6_10	0.221	0.257	-0.181	0.181	0.836

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
p_i_Yr15	9.407200e+01	3.804000e+00	87.137	94.340	99.780
p_i_Yr16	9.664000e+01	1.914000e+00	93.580	96.589	99.846
p_i_Yr17	8.734900e+01	7.308000e+00	74.935	87.305	99.320
p_i_Yr18	9.793400e+01	1.172000e+00	96.013	97.929	99.888
p_i_Yr19	8.744100e+01	7.309000e+00	74.809	87.736	99.227
p_i_Yr20	9.266500e+01	4.309000e+00	85.544	92.776	99.610
p_i_UsPres	9.191000e+01	4.646000e+00	84.135	91.874	99.517
p_i_ChinaEmp	8.641000e+01	1.035700e+01	63.258	88.768	99.569
p_i_DalaiLama	8.189700e+01	1.099600e+01	61.262	82.831	99.113
p_i_JapanEmp	8.731100e+01	7.289000e+00	74.883	87.317	99.390
p_i_Yr15ChinaEmp	-2.125900e+01	9.697000e+00	-38.307	-22.113	0.014
p_i_Yr15DalaiLama	2.934000e+01	3.877400e+01	-16.594	19.854	130.597
p_i_Yr15JapanEmp	-1.343000e+01	1.145400e+01	-33.070	-14.350	12.386
p_i_Yr16ChinaEmp	3.553291e+03	1.437921e+04	-99.738	87.137	33894.317
p_i_Yr16DalaiLama	-7.942000e+00	8.808000e+00	-24.720	-8.134	10.013
p_i_Yr16JapanEmp	-3.314000e+01	1.076100e+01	-52.267	-33.986	-9.728
p_i_Yr17ChinaEmp	-7.171000e+00	3.665400e+01	-57.124	-13.192	82.901
p_i_Yr17DalaiLama	2.040000e-01	1.691700e+01	-29.356	-0.853	38.092
p_i_Yr17JapanEmp	-2.413000e+00	8.907000e+00	-19.678	-2.528	16.161
p_i_Yr18UsPres	-4.741000e+00	1.627600e+01	-32.276	-6.019	31.581
p_i_Yr18ChinaEmp	-5.153800e+01	1.408600e+01	-75.800	-52.831	-20.118
p_i_Yr18DalaiLama	2.549700e+01	2.321100e+01	-12.995	23.542	79.072
p_i_Yr18JapanEmp	8.406000e+00	1.106600e+01	-12.234	8.025	31.104
p_i_Yr19UsPres	-1.208500e+01	1.471400e+01	-37.407	-13.689	20.385
p_i_Yr19ChinaEmp	-3.813300e+01	1.836200e+01	-70.665	-39.331	2.288
p_i_Yr19DalaiLama	-4.523000e+00	1.627300e+01	-32.971	-6.103	31.697
p_i_Yr19JapanEmp	5.668000e+00	1.154900e+01	-15.293	5.088	30.526
p_i_Yr20UsPres	2.882700e+01	2.458700e+01	-11.748	26.118	82.735
p_i_Yr20ChinaEmp	-4.681800e+01	1.580700e+01	-74.172	-47.867	-11.319
p_i_Yr20DalaiLama	-1.362300e+01	1.452900e+01	-39.017	-14.318	16.757
p_i_Yr20JapanEmp	4.844190e+03	1.038808e+04	-99.726	-46.680	36669.006
age_Francis_predictive	1.155900e+01	1.213500e+01	-11.398	10.922	37.039
age_Obama_predictive	-4.828000e+01	9.761000e+00	-64.253	-49.364	-26.803
age_Dalai_predictive	-5.663900e+01	1.132800e+01	-78.540	-56.699	-33.513

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
age_Naruhito_predictive	-2.416500e+01	1.477400e+01	-48.720	-25.786	9.713
age_Benedict_predictive	4.173833e+03	9.028347e+03	-99.755	-53.650	31562.512
age_Carter_predictive	3.786200e+01	2.603600e+01	0.827	33.514	101.192
age_Clinton_predictive	-8.505000e+00	3.226000e+01	-51.764	-14.436	75.316
age_Bush_predictive	1.728730e+09	7.219759e+10	-13.795	4051.756	106362753.560
age_Trump_predictive	3.585000e+01	4.806900e+01	-28.674	26.148	154.817
age_Akihito_predictive	4.622872e+03	1.007858e+04	-99.751	-48.291	35520.357

- If a Pope is born one year later, he is expected to live longer by a multiplicative factor of $exp(\alpha_1) = exp(0.0003931) = 1$. This basically means the expected lifespan would hardly change.
- If a U.S. President is born one year later, he is expected to live longer by a multiplicative factor of $exp(\alpha_1 + \alpha_6) = exp(0.0003931 + 0.0004852) = 1$.
- If a U.S. President and a Chinese Emperor were born in the same year 1700, 33 years after the average birth year 1667, the Chinese emperor is expected to live longer by a multiplicative factor of $exp(\alpha_3 - \alpha_2 + (\alpha_7 - \alpha_6) * y)$ = exp(-0.31925+0.1449877 + 33*(-0.0001094-0.0004852) = 0.82. That's to say, the Chinese emperor's life expectancy is 18% shorter.
- If a Japanese Emperor is born 100 years later, he is expected to live longer by a multiplicative factor of $exp(\alpha_9) = exp(100 * -0.0000595) = 0.994$, which means the lifespan decreases by 0.6%, whereas for Dalai Lama, the lifespan is expected to increase by a factor of exp(100*-0.0016203) = 0.85, or decrease by 15%.

```r
vars3 = c("age_Francis_predictive", "age_Obama_predictive","age_Dalai_predictive", "age_Naruhito_predic
output <- data.frame(model_output$BUGSoutput$summary)
output <- output[row.names(output) %in% vars3, ]
coef <- output %>%
  mutate("Variable" = vars3,
         "Mean" = output$mean,
         "Standard Deviation" = output$sd,
         "2.5% Quantile" = output$X2.5.,
         "Median" = output$X50.,
         "97.5% Quantile" = output$X97.5.
         ) %>%
  select("Variable", "Mean", "Standard Deviation", "2.5% Quantile","Median", "97.5% Quantile")
kable(coef, digits = 3)
```

Variable	Mean	Standard Deviation	2.5% Quantile	Median	97.5% Quantile
age_Francis_predictive	94.072	3.804	87.137	94.340	99.780
age_Obama_predictive	96.640	1.914	93.580	96.589	99.846
age_Dalai_predictive	87.349	7.308	74.935	87.305	99.320
age_Naruhito_predictive	97.934	1.172	96.013	97.929	99.888
age_Benedict_predictive	87.441	7.309	74.809	87.736	99.227
age_Carter_predictive	92.665	4.309	85.544	92.776	99.610
age_Clinton_predictive	91.910	4.646	84.135	91.874	99.517
age_Bush_predictive	86.410	10.357	63.258	88.768	99.569
age_Trump_predictive	81.897	10.996	61.262	82.831	99.113
age_Akihito_predictive	87.311	7.289	74.883	87.317	99.390

The table above shows the estimate and 95% credible interval (the bounds are the 2.5% quantile and 97.5% quantile) for all 10 living leaders in the dataset.

this part below is now extraneous

V b. Posterior Inference for the 10 Alive Leaders

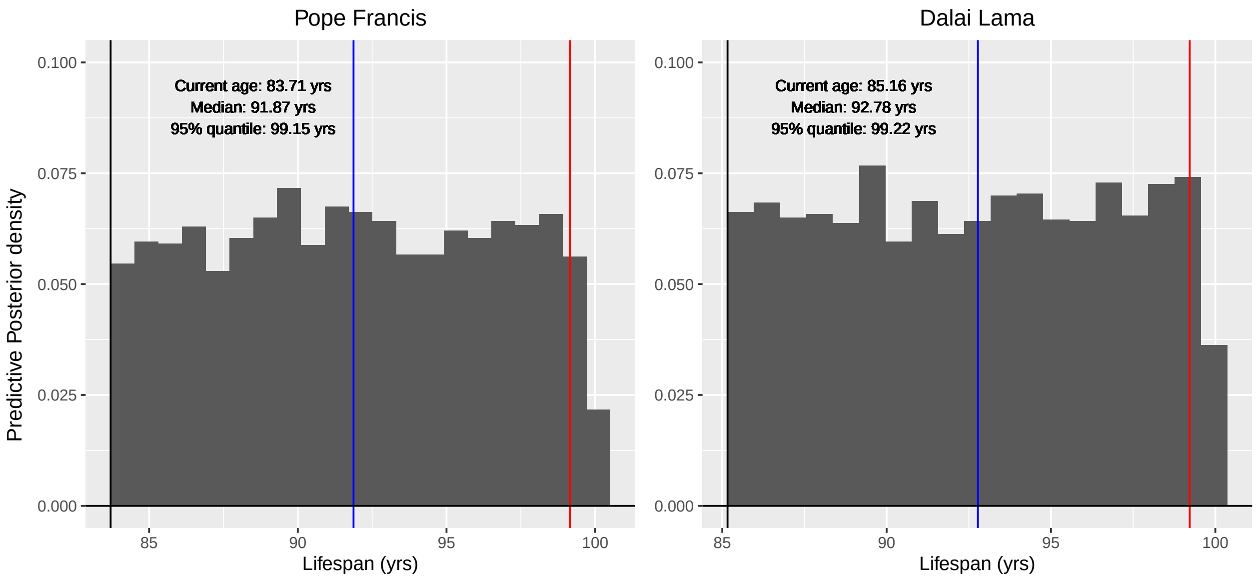


Fig. 3a The posterior predictive probability density function of the lifespan for hypothetical leaders with the same attributes as Pope Francis (right) and the 14th Dalai Lama (left). The black vertical line marks current age the blue marks posterior median, and the red marks the posterior 95% quantile.

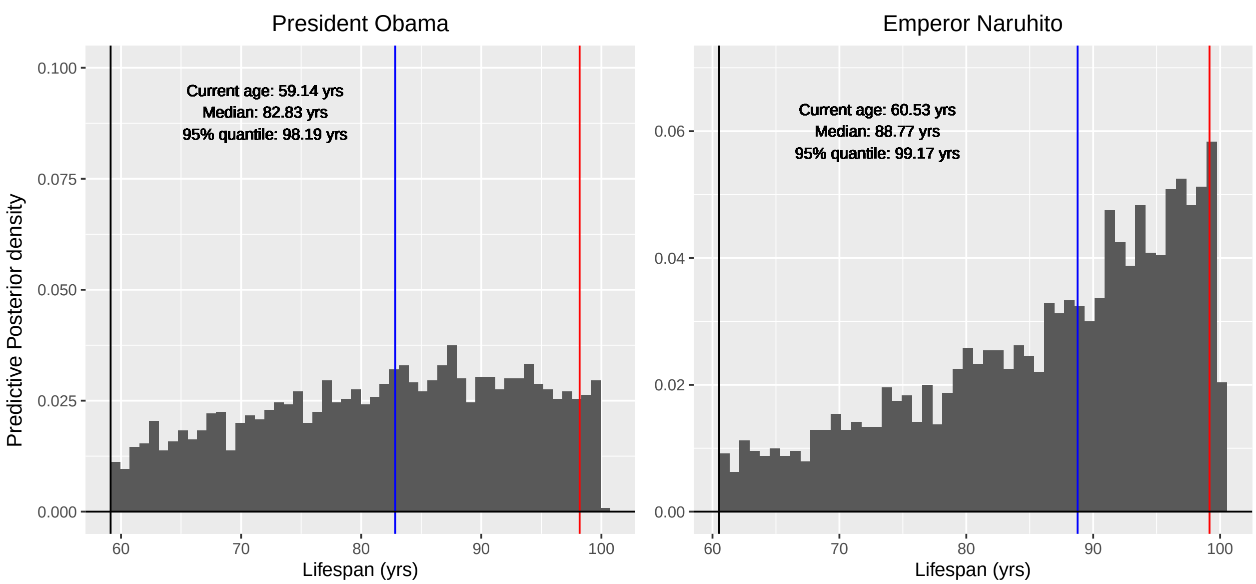


Fig. 3b The posterior predictive probability density function of the lifespan for hypothetical leaders with the same attributes as Obama (right) and Emperor Naruhito (left). The black vertical line marks current age, the blue marks posterior median, and the red marks the posterior 95% quantile.

The histogram of the posterior predictive distributions of the lifespans for a leader with the same birth year and leadership type as Pope Francis and that of a leader with the same above attributes as the 14th Dalai Lama are more uniform. The histogram for a leader with the same birth year and leadership type as President Obama and that of a leader with the same above attributes as Emperor Naruhito are both left skewed, with the modes in the late 90s.

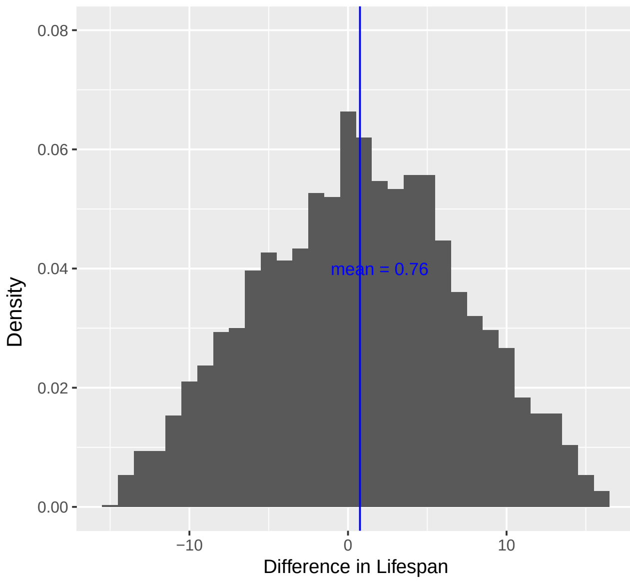


Figure 4a. This histogram shows the posterior predictive distribution of the difference in lifespan (14th Dalai Lama − Pope Francis).

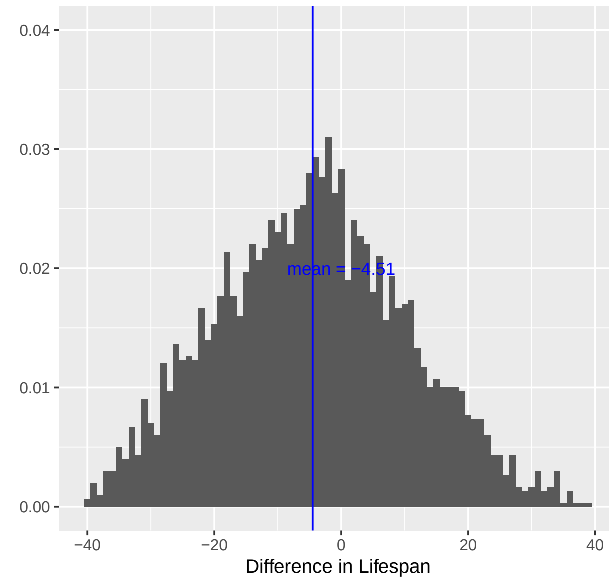
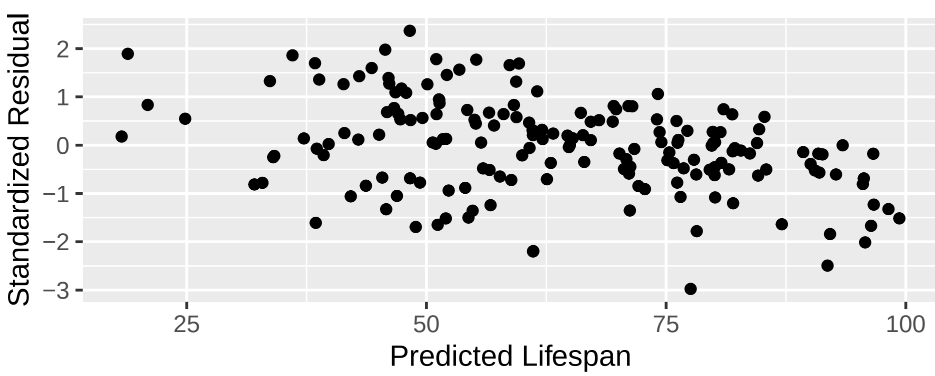


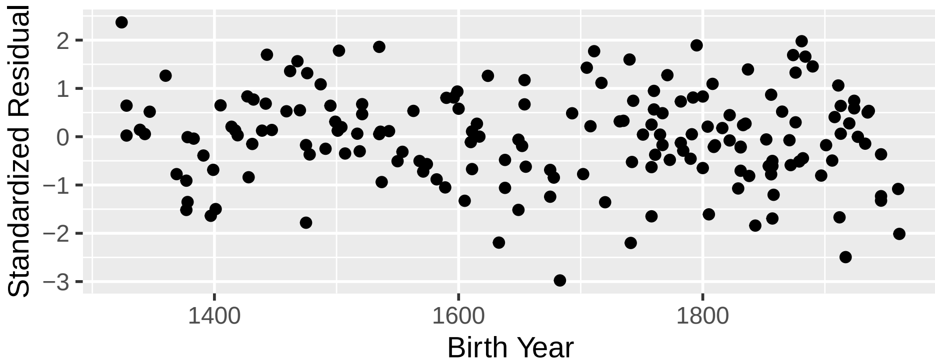
Figure 4b. This histogram shows the posterior predictive distribution of the difference in lifespan (Obama − Emperor Naruhito).

Figure 4 depicts the distribution of the difference in posterior predictive lifespans between the two sets of leaders this paper focuses on. The probability that the 14th Dalai Lama will have a longer lifespan than Pope Francis is 0.552. The probability that President Obama will have a longer lifespan than Emperor Naruhito is 0.3717.

V c. MCMC and Model Diagnostics

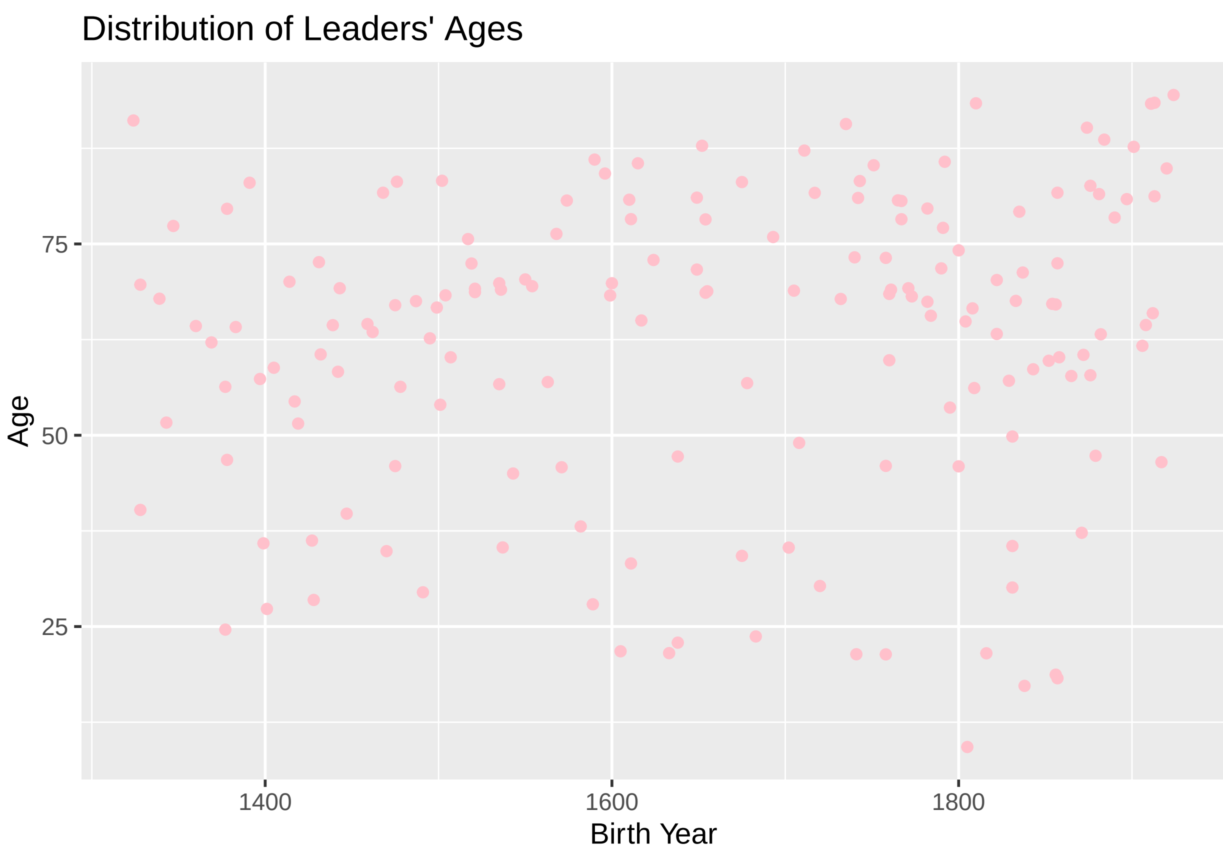
The combination of traceplots, lag-1 scatterplots, and acf plots suggest the chain for each parameters converges. Code for those plots are included in appendix A.2. The Rhat's are all close to 1, which is another indicator of converge. Most of the effective sample sizes are greater than 1000 (the effective sample size of 170 is notably low, so more iterations may be needed.)

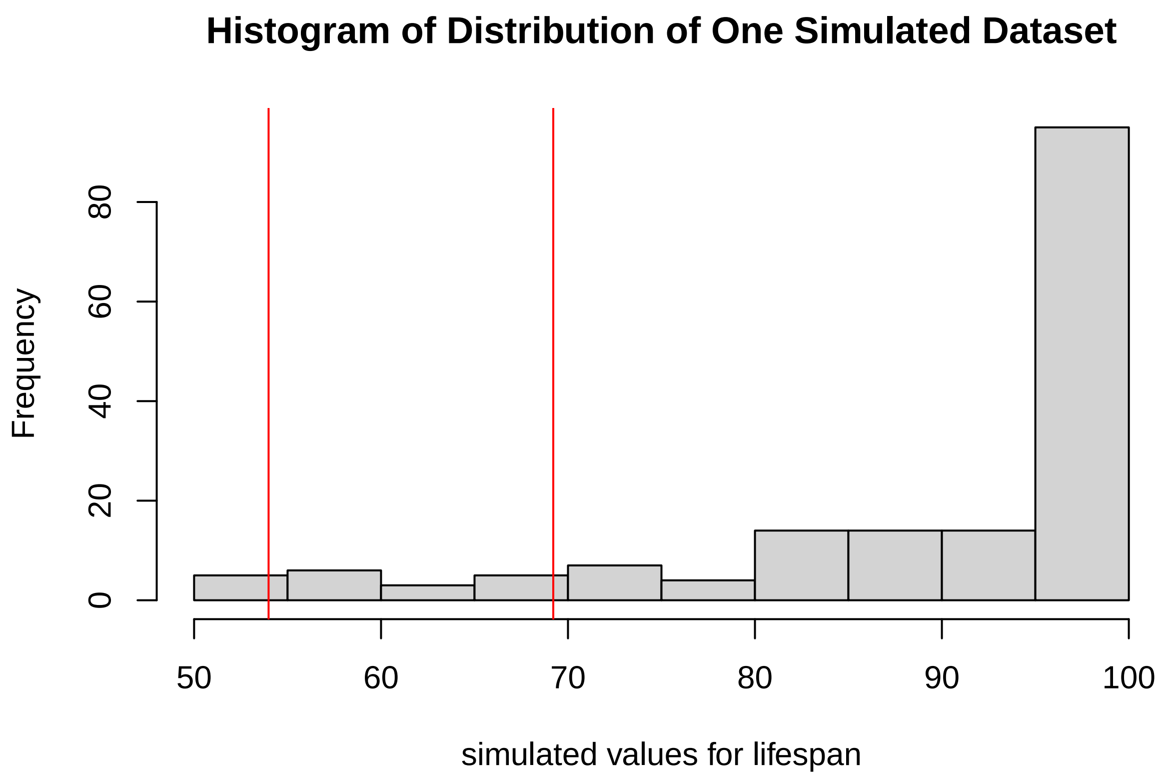




The standardized residual plot for predicted lifespan shows patterning. It appears that our model tends to overpredict lifespan compared to what was actually observed in many cases. This might be due to the fact that we only have two predictors, Birth Year and Type of Leader. Therefore, we are missing information that might help predict whether or not a leader will die earlier, for example if they have some underlying health condition, etc.

The standardized residuals when plotted against Birth Year show random scattering, however, and this means that our model is not over or underpredicting lifespan according to the year a leader was born.





For the posterior predictive checks, it appears that our model returns values that are more concentrated around the upper truncation limit, which is why the dataframe's Inf values are encoded as 100 (the upper truncation limit defined in our function). This follows what the standardized residuals found, where most of the values from the model are left-skewed. Additionally, the red lines on our model are defined as the ages from two randomly sampled leaders that are non-censored.

VI. Sensitivity Analysis

```
priors1 = "
  beta_0 ~ dnorm(0.0, 1) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
```

```
  beta_3_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_3_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  r ~ dexp(0.001) # Prior on r, Stander et al
  "

sensitivity_1 <- run_model(priors1, "sensitivity1")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6043
##
## Initializing model
```

```
priors2 = "
  beta_0 ~ dnorm(0.0, 1) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_3_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
```

```
  beta_3_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  r ~ dgamma(10,10) # Prior on r, constant hazard
  "

sensitivity_2 <- run_model(priors2, "sensitivity2")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6043
##
## Initializing model
```

```
priors3 = "
  beta_0 ~ dnorm(0.0, 1) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  beta_3_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_3_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
```

```
  beta_4_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_4_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_5_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1) # Prior on beta_1 is normal with low precision

  r ~ dgamma(10,100) # Prior on r, decreasing hazard
  "

sensitivity_3 <- run_model(priors3, "sensitivity3")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6043
##
## Initializing model
```

```
priors4 = "
  beta_0 ~ dnorm(0.0, 1/10000) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1/10000) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1/10000) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1/10000) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1/10000) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1/10000) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1/10000) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1/10000) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1/10000) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_3_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
```

```
  beta_4_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  r ~ dexp(0.001) # Prior on r, Stander et al
  "

sensitivity_4 <- run_model(priors4, "sensitivity4")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6045
##
## Initializing model
```

```
priors5 = "
  beta_0 ~ dnorm(0.0, 1/10000) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1/10000) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1/10000) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1/10000) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1/10000) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1/10000) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1/10000) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1/10000) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1/10000) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_3_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
```

```
  beta_5_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  r ~ dgamma(10,10) # Prior on r, constant hazard
  "

sensitivity_5 <- run_model(priors5, "sensitivity5")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6045
##
## Initializing model
```

```
priors6 = "
  beta_0 ~ dnorm(0.0, 1/10000) # Prior on beta_0 is normal with low precision
  beta_1 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2 ~ dnorm(0.0, 1/10000) # Prior on beta_2 is normal with low precision
  beta_3 ~ dnorm(0.0, 1/10000) # Prior on beta_3 is normal with low precision
  beta_4 ~ dnorm(0.0, 1/10000) # Prior on beta_4 is normal with low precision
  beta_5 ~ dnorm(0.0, 1/10000) # Prior on beta_5 is normal with low precision
  beta_6 ~ dnorm(0.0, 1/10000) # Prior on beta_6 is normal with low precision
  beta_7 ~ dnorm(0.0, 1/10000) # Prior on beta_7 is normal with low precision
  beta_8 ~ dnorm(0.0, 1/10000) # Prior on beta_8 is normal with low precision
  beta_9 ~ dnorm(0.0, 1/10000) # Prior on beta_9 is normal with low precision
  beta_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_1_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_1_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_2_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_2_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  beta_3_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_3_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_4_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_5_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
```

```
  beta_5_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_7 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_8 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_9 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision
  beta_6_10 ~ dnorm(0.0, 1/10000) # Prior on beta_1 is normal with low precision

  r ~ dgamma(10,100) # Prior on r, decreasing hazard
  "

sensitivity_6 <- run_model(priors6, "sensitivity6")
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 177
##    Unobserved stochastic nodes: 43
##    Total graph size: 6045
##
## Initializing model
```

```
parameter = "beta_6"
sa1 = data.frame(sensitivity_1$BUGSoutput$sims.matrix) %>%
  select(parameter)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(parameter)` instead of `parameter` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
sa2 = data.frame(sensitivity_2$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa3 = data.frame(sensitivity_3$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa4 = data.frame(sensitivity_4$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa5 = data.frame(sensitivity_5$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa6 = data.frame(sensitivity_6$BUGSoutput$sims.matrix) %>%
  select(parameter)

sa_df = data.frame(vals = rbind(sa1, sa2, sa3, sa4, sa5, sa6),
                   type = c(rep("r~Exp(0.001), B~N(0,1)", nrow(sa1)),
                            rep("r~Ga(10,10), B~N(0,1)", nrow(sa2)),
                            rep("r~Ga(10,1000), B~N(0,1)", nrow(sa3)),
                            rep("r~Exp(0.001), B~N(0,10000)", nrow(sa4)),
                            rep("r~Ga(10,10), B~N(0,10000)", nrow(sa5)),
                            rep("r~Ga(10,1000), B~N(0,10000)", nrow(sa6))))

neworder <- c("r~Exp(0.001), B~N(0,1)", "r~Ga(10,10), B~N(0,1)", "r~Ga(10,1000), B~N(0,1)",
              "r~Exp(0.001), B~N(0,10000)", "r~Ga(10,10), B~N(0,10000)", "r~Ga(10,1000), B~N(0,10000)")

sa_df_plot <- arrange(mutate(sa_df, type=factor(type,levels=neworder)), type)

sampled_vals = model_output$BUGSoutput$summary["beta_6", ]
```

```r
sampled_beta = sampled_vals["mean"]

ggplot(sa_df_plot) + geom_density(aes(x = beta_6)) +
  facet_wrap(~type,  scales = "free_y", nrow = 2) +
  geom_vline(xintercept = sampled_beta, color = "blue") +
  labs(x = expression(beta[6]), caption = paste("Fig. 8 The above plots show sensitivity analysis for di
  theme(plot.caption = element_text(hjust = 0.5, vjust = -0.5, size = 4))
```

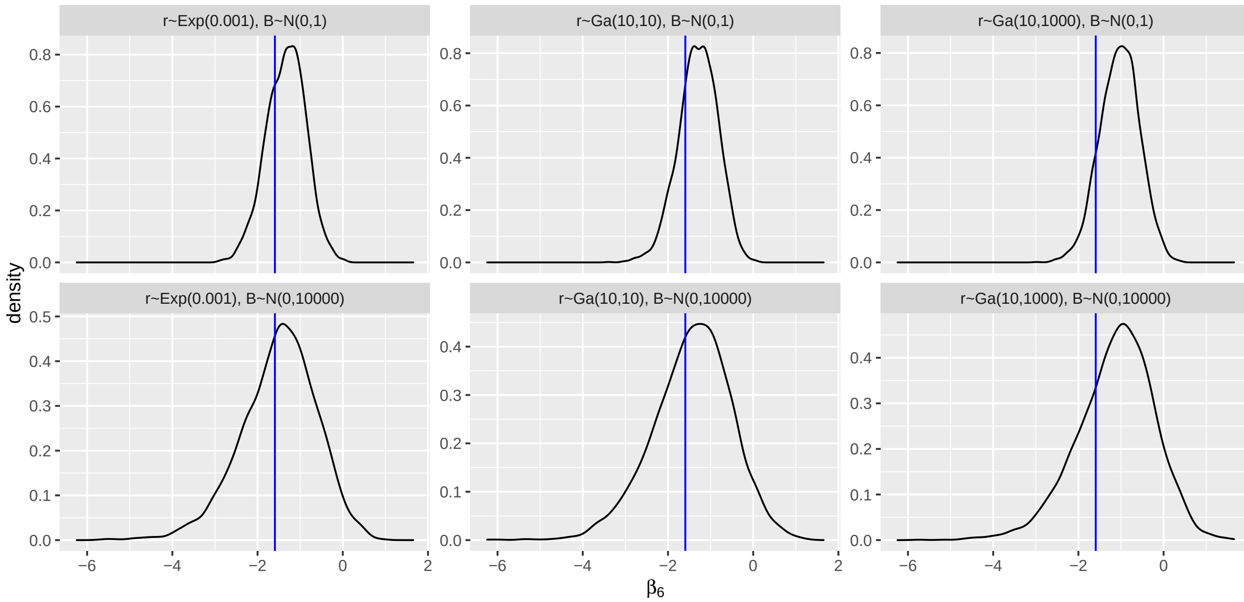


Fig. 8 The above plots show sensitivity analysis for different priors on r and the coefficients. The blue line indicates the estimate of the coefficient from the propsed model. It appears that the analysis is generally not sensitive to different priors.

We tested $N(0,1) and N(0,10000)$ priors for the β's and $Exp(0.001)$ (Stander et al prior), $Gamma(10,10)$ (mean = 1), and $Gamma(10,100)$ (mean = 0.1) priors for r. Distribution of β_6 was approximated using those different priors and plotted above.The blue line, which is estimate from our model crosses, or is close to the mode of all but one of the distributions, so our analysis is mostly robust against changes in prior. However, it is somewhat sensitive to the prior combination $r \sim Gamma(10,1000)$ and β_6 $N(0,1)$ Code to conduct ensitivity analysis for the remaining betas can be found in appendix A.3.

VII. Cross Validation

```r
#read data
monarchs <- read.csv("english_monarchs.csv", header = FALSE)
monarchs$Age <- as.numeric(monarchs$V3) - as.numeric(monarchs$V2)
monarchs <- monarchs %>%
  filter(V2 >= 1300)
```

```r
monarchs$Age[monarchs$V1 == "Elizabeth II"] <- 94
```

```r
monarchs$Censored <- 0
monarchs$Censored[monarchs$V1 == "Elizabeth II"] <- 1
```

```r
monarchs$survival <- monarchs$Age
monarchs$survival[monarchs$V1 == "Elizabeth II"] <- NA
```

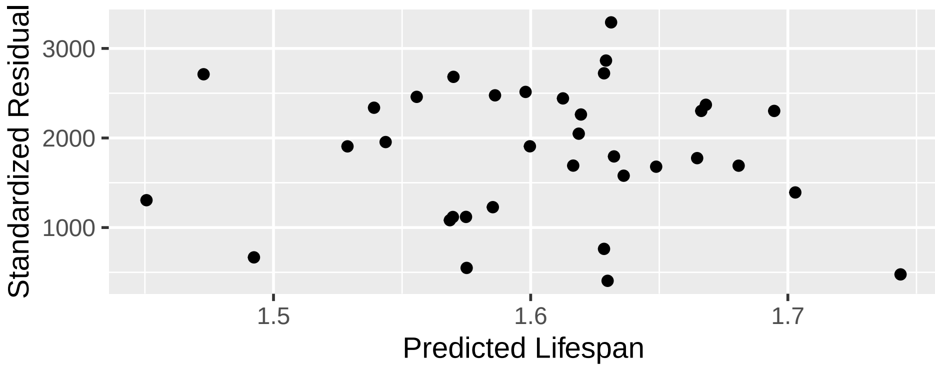
```r
monarchs$TypeJapanEmp <- 1
monarchs <- monarchs %>%
  mutate(Yr14 = as.factor(case_when(V2 <= 1400 ~1,
```

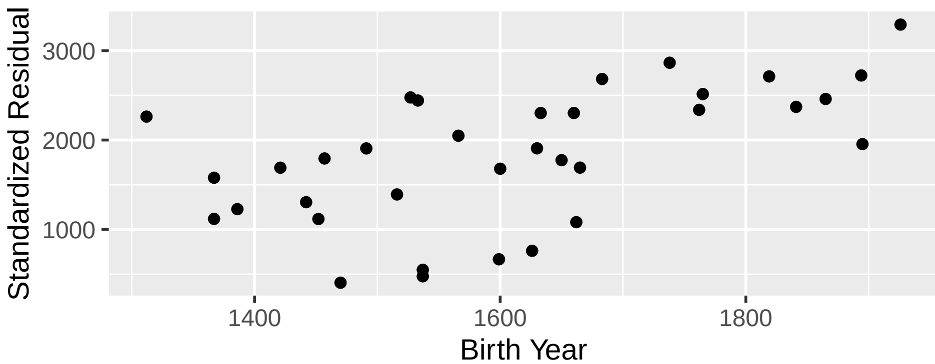
```
                                        TRUE ~0)),
         Yr15 = as.factor(case_when(V2 >= 1401 & V2 <= 1500 ~ 1,
                                         TRUE ~0)),
         Yr16 = as.factor(case_when(V2 >= 1501 & V2 <= 1600 ~ 1,
                                         TRUE ~0)),
         Yr17 = as.factor(case_when(V2 >= 1601 & V2 <= 1700 ~ 1,
                                         TRUE ~0)),
         Yr18 = as.factor(case_when(V2 >= 1701 & V2 <= 1800 ~ 1,
                                         TRUE ~0)),
         Yr19 = as.factor(case_when(V2 >= 1801 & V2 <= 1900 ~ 1,
                                         TRUE ~0)),
         Yr20 = as.factor(case_when(V2 >= 1901 & V2 <= 2000 ~ 1,
                                         TRUE ~0))
         )

z_1 <- as.numeric(as.character(monarchs$Yr15))
z_2 <- as.numeric(as.character(monarchs$Yr16))
z_3 <- as.numeric(as.character(monarchs$Yr17))
z_4 <-  as.numeric(as.character(monarchs$Yr18))
z_5 <-  as.numeric(as.character(monarchs$Yr19))
z_6 <-  as.numeric(as.character(monarchs$Yr20))
z_7 <- 0
z_8 <- 0
z_9 <- 0
z_10 <- as.numeric(as.character(monarchs$TypeJapanEmp))

# add interaction b/w Yr__ and Type___
z_1_8 <- z_1*z_8; z_1_9 <- z_1*z_9; z_1_10 <- z_1*z_10


z_2_8 <- z_2*z_8
z_2_9 <- z_2*z_9
z_2_10 <- z_2*z_10


z_3_8 <- z_3*z_8
z_3_9 <- z_3*z_9
z_3_10 <- z_3*z_10

z_4_7 <- z_4*z_7
z_4_8 <- z_4*z_8
z_4_9 <- z_4*z_9
z_4_10 <- z_4*z_10

z_5_7 <- z_5*z_7
z_5_8 <- z_5*z_8
z_5_9 <- z_5*z_9
z_5_10 <- z_5*z_10

z_6_7 <- z_6*z_7
z_6_8 <- z_6*z_8
z_6_9 <- z_6*z_9
z_6_10 <- z_6*z_10
```

```
cv_data.mat = as.matrix(cbind(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_10,

                           z_1_8,
                           z_1_9,
                           z_1_10,

                           z_2_8,
                           z_2_9,
                           z_2_10,

                           z_3_8,
                           z_3_9,
                           z_3_10,
                             z_4_7,
                           z_4_8,
                           z_4_9,
                           z_4_10,
                             z_5_7,
                           z_5_8,
                           z_5_9,
                           z_5_10,
                             z_6_7,
                           z_6_8,
                           z_6_9,
                           z_6_10))

beta.mat = as.matrix(betas[2:length(betas)], nrow = p, byrow = TRUE)
temp = cv_data.mat %*% t(beta.mat)
logmus = as.numeric(betas["beta_0"]) + temp

# draw predicted values based on mus calculated above
set.seed(123)
cv_n = nrow(monarchs)
Ts = rtweibull(cv_n, r, (1/exp(logmus))^(1/r), 0, 100)
```

```
## Warning in (1/exp(logmus))^(1/r): longer object length is not a multiple of
## shorter object length
```

```
#calculate residuals
resid = monarchs$Age - Ts

#calculate standard deviations
meanTs = mean(Ts)
diff = Ts - meanTs
sd = sqrt((1/n)*(sum((diff)^2)) * (1+ (1/n) + (diff)^2/sum((diff)^2)))
```





We sought to perform external cross-validation on our model. The data that we used for the cross-validation was from all of England's monarchs, accessed from Wikipedia. This dataset only included one person that was still alive. As shown in the standardized residuals plot, there is an upward trend, as the birth year of the monarchs increases. Additionally, in the plot showing Predicted Lifespan and the standardized residuals, there is a downward trend, underpredicting a lot of the leaders' lifespans.

VI. Conclusion and Further Discussion

We sought to compare Popes, US Presidents, Dalai Lamas, Chinese Emperors, and Japanese Emperors to see how their lifespans compare. Our model has found that the impact of birth year on lifespan does not change based on leadership type. Additionally, our model has found that lifespan does depend on year of birth, but the association is not strong. One limitation worth noting is that our data does not include election year, so we have no way to control for the fact that Presidents and Popes must have a minimum lifespan. On the other hand, Dalai Lamas can be chosen close to birth and thus have a younger death date.

This model could be improved by including more predictors; for example, health varies widely among people, and could lead to a better model for prediction. For further implementations of this research question, it would be valuable to interpret economic or quality of life data as it concerns the different countries that the leaders grew up in, as the conditions that a person live in lead to changes in a person's lifespan.

VII. References

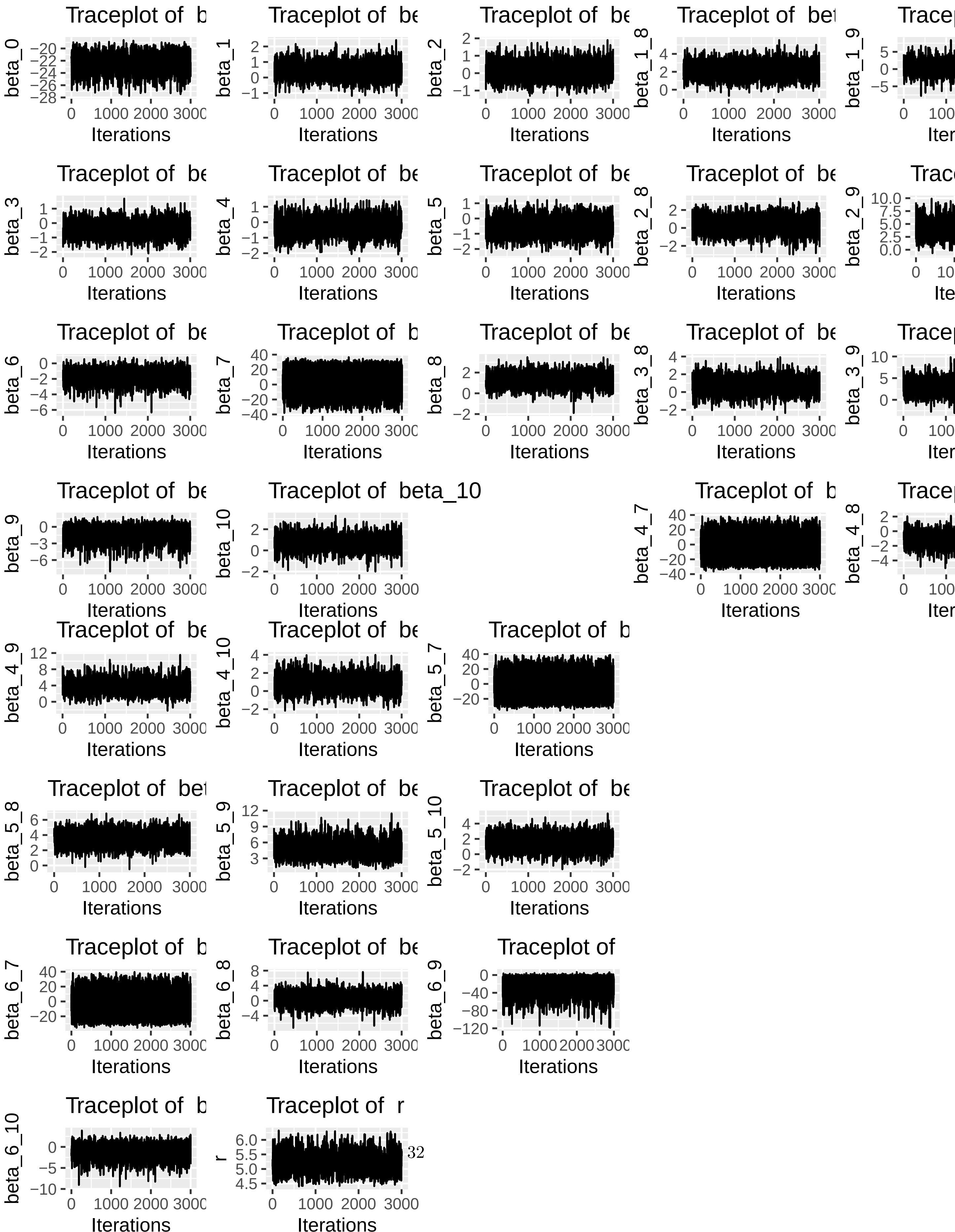
Stander, J., Dalla Valle, L., and Cortina-Borja, M. (2018). A Bayesian Survival Analysis of a Historical Dataset: How Long Do Popes Live? The American Statistician 72(4):368-375.

VIII. Appendix

A.1 Details on the Poisson "Zero-Trick" Method

We use the Poisson "zeros trick," where a set of 0's are created and the likelihood of observing an 0 follows a $Poisson(\phi_i)$ distribution. The likelihood L_i is $Pr(z_i = 0) = exp(-\phi)$. In other words, ϕ_i equals $-log(L_i)$. This mechanism is necessary because we don't know the lifespan of currently living people, so we need a way to model their expected survival times. The censored vector has 0 for dead leaders and 1 for alive leaders. The vector censoring_limits contains 100 (chosen as an upper bound on age) for dead leaders and current age for alive leaders. The non-100 observations in censoring_limits are saved in t_censored, and t_censored is taken to the power of r then multiplied by mu_censored to calculate phi_censored, which is the mean parameter of the Poisson likelihood described above.

A.2 Code for Traceplots, Lag-1 Scatterplots, and Acf Plots



All of the traceplots and code for plotting is above.

Display of lag-1 scatterplot and acf plot for β_1 is shown above, code provided allows generations of plots for all parameters.

add old model and resid code here

A.3 Code for Sensitivity Analysis

```
parameter = "beta_6" # change parameter here to run for different betas

sa1 = data.frame(sensitivity_1$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa2 = data.frame(sensitivity_2$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa3 = data.frame(sensitivity_3$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa4 = data.frame(sensitivity_4$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa5 = data.frame(sensitivity_5$BUGSoutput$sims.matrix) %>%
  select(parameter)
sa6 = data.frame(sensitivity_6$BUGSoutput$sims.matrix) %>%
  select(parameter)

sa_df = data.frame(vals = rbind(sa1, sa2, sa3, sa4, sa5, sa6),
                   type = c(rep("r~Exp(0.001), B~N(0,1)", nrow(sa1)),
                            rep("r~Ga(10,10), B~N(0,1)", nrow(sa2)),
                            rep("r~Ga(10,1000), B~N(0,1)", nrow(sa3)),
                            rep("r~Exp(0.001), B~N(0,100)", nrow(sa4)),
                            rep("r~Ga(10,10), B~N(0,100)", nrow(sa5)),
                            rep("r~Ga(10,1000), B~N(0,100)", nrow(sa6))))

neworder <- c("r~Exp(0.001), B~N(0,1)", "r~Ga(10,10), B~N(0,1)", "r~Ga(10,1000), B~N(0,1)",
              "r~Exp(0.001), B~N(0,100)", "r~Ga(10,10), B~N(0,100)", "r~Ga(10,1000), B~N(0,100)")

sa_df_plot <- arrange(mutate(sa_df, type=factor(type,levels=neworder)), type)

sampled_vals = model_output$BUGSoutput$summary[parameter, ]
sampled_beta = sampled_vals["mean"]

ggplot(sa_df_plot) + geom_density(aes(x = sa_df_plot[,1])) +
  facet_wrap(~type,  scales = "free_y", nrow = 2) +
  geom_vline(xintercept = sampled_beta, color = "blue") +
  labs(x = expression(beta[6]), caption = paste("Fig. 8 The above plots show sensitivity analysis for di
  theme(plot.caption = element_text(hjust = 0.5, vjust = -0.5, size = 4))
```

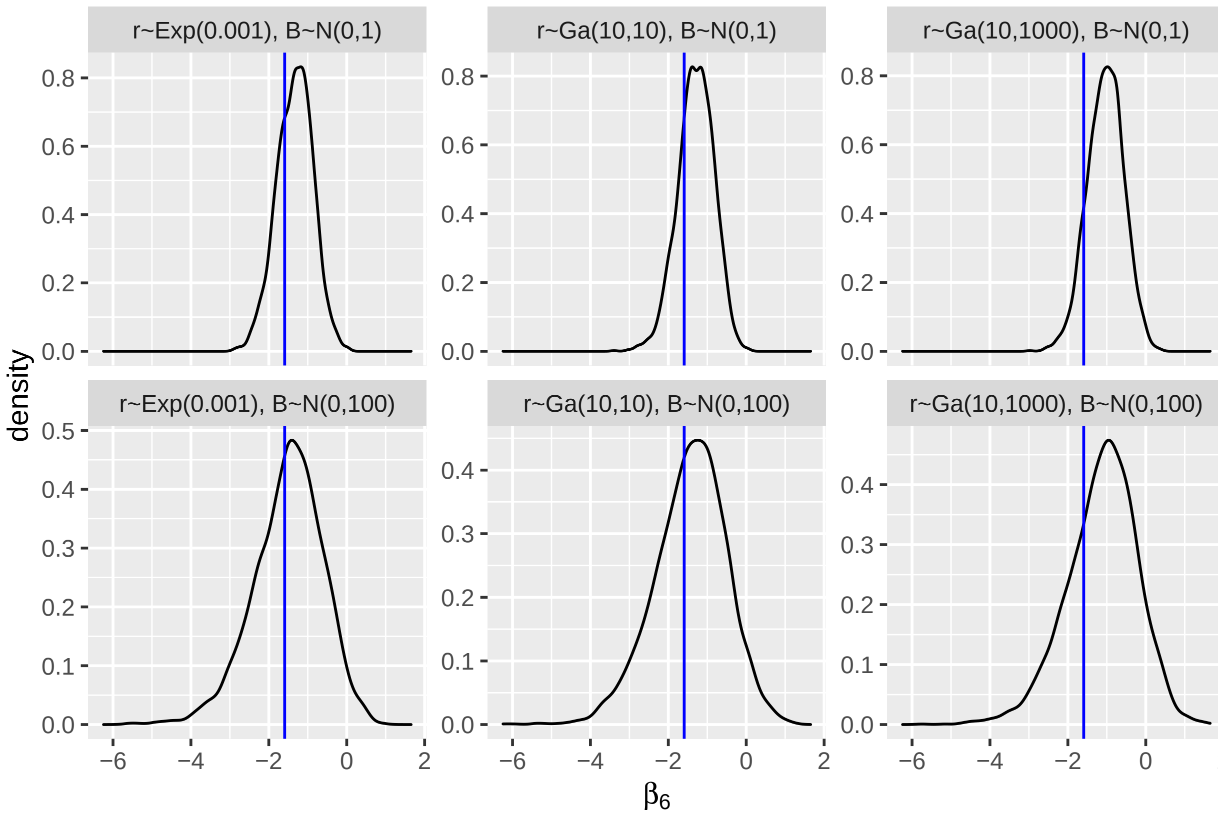


Fig. 8 The above plots show sensitivity analysis for different priors on r and the coefficients. The blue line indicates the estimate of the coefficient from the propsed model. It appears that the analysis is generally not sensitive to different priors.