

Bayesian Project

Thanwi Lalu, Joshua Mann, Ailis Muldoon, Havelah Queen, Anne Sartori

2025-03-10

Introduction

This report addresses the question of whether or not offspring of centenarians different from controls in the growth of their cognitive function later in life, using Bayesian hierarchical methods. We analyze a dataset that contains up to 5 longitudinally collected measures of “Telephone Interview for Cognitive Status” (TICS) in offspring of centenarians and controls enrolled in two studies of longevity conducted at Boston University. While we find evidence that TICS scores grow nonlinearly with time, first decreasing and then increasing, we do not find evidence that TICS scores differ for offspring of centenarians versus those of controls, adjusting for both the subject’s age and their TICS score at baseline.

Exploratory analysis of confounders

(Primary author: Joshua Mann)

Numerical covariates

To assess whether baseline numerical covariates differ between centenarian offspring and control subjects, we fit four separate univariate Bayesian models comparing each continuous variable across the two groups. Specifically, we compared baseline age (Age01), body mass index (BMI), TICS score at baseline (TICS01), and years of education. Each model used an appropriate likelihood for the response and specified uninformative priors. For variables Age01 and BMI, log-normal models were used to reflect skewness; for TICS01, a normal model was used; and for Years of Education, a Poisson likelihood was assumed. Posterior samples were drawn using rjags, and we report group-specific posterior means as well as the posterior distribution of the difference between groups (offspring minus control).

Age at Baseline

The posterior mean age was 74.19 for controls and 74.78 for centenarian offspring. The estimated difference (offspring minus control) was -0.59 years (95% CrI: [-1.70, 0.54]), suggesting that offspring were slightly older on average, though this difference was not statistically significant, since the 95% credible interval includes zero.

Body Mass Index (BMI)

Posterior means were 27.41 for controls and 26.56 for offspring, with a posterior mean difference of 0.85 units (95% CrI: [0.16, 1.53]). This indicates that centenarian offspring had significantly lower BMI at baseline than controls and should be accounted for as a confounder.

Baseline TICS Score

The posterior mean for TICS01 was 13.56 in controls and 14.09 in offspring, with a mean difference of -0.53 (95% CrI: [-1.18, 0.10]). While the offspring appear to score slightly higher, this difference is not significant, as the credible interval includes zero.

Years of Education

The posterior difference in expected years of education between groups was estimated at -0.23 years (95% CrI: [-0.82, 0.36]), indicating no statistically significant difference in educational attainment at baseline.

Binomial covariates

To examine differences in binomial covariates (stroke, diabetes, smoking history, hypertension) between offspring of centenarians and controls, we conducted a Bayesian logistic regression using group status as the predictor and covariate value as the binary outcome. The model assumes a binomial likelihood with a logit link and includes uninformative Normal(0, 0.0001) priors for both the intercept and the group coefficient. The simulation used 5,000 MCMC samples following a 2,000 iteration burn-in. We derived the odds ratio (OR) for covariates in offspring relative to controls by exponentiating the coefficient for group status.

Summary of 95% Credible Intervals and Significance by Variable

Variable	Odds.ratio	X95..Credible.Interval..OR.	Significant
Stroke	0.39	[0.13, 0.87]	Yes
Diabetes	0.40	[0.22, 0.66]	Yes
Smoking	0.63	[0.45, 0.84]	Yes
Hypertension	0.53	[0.39, 0.72]	Yes
Coronary Artery Disease	0.36	[0.18, 0.63]	Yes
Cancer	0.88	[0.62, 1.23]	No
Heart Attack	0.71	[0.35, 1.32]	No
Sex (Male)	1.88	[1.37, 2.51]	Yes
Aspirin Use	0.75	[0.55, 1.00]	No

We conclude that

Distributions of TICS score at baseline

(Primary author: Havelah Queen)

Methods

In order to ascertain whether the distributions of TICS score at baseline differs between offspring type, a linear regression model assuming normal distribution with mean ($\mu[i]$) and precision (τ) was used containing possible confounders as covariates. BMI was centered around the mean to reduce multicollinearity. Uninformative Normal priors with $\mu = 0$ and $\tau = 0.0001$ were used for all regression coefficients while a Gamma prior with shape=1 and rate=1 was used for τ . 1500 adaptation, 10000 burn-in, and 10000 posterior sampling iterations were used to estimate β_{ptype} and probabilities of β_{ptype} being greater or lesser than 0.

Results

Relevant Posterior Estimates

Variable	Estimate	SD
P_beta.ptype_greater	0.7168	0.45060
P_beta.ptype_lesser	0.2832	0.45060
beta.ptype	0.1897	0.33920
mean.BMI	27.2432	0.00002

Conclusions

β_{ptype} is estimated to be 0.1897 such that a child of centenarians has a baseline TICS score greater by 0.1897, all else held constant. However, this estimate is not significant as the 95% credible interval of (-0.4813, 0.8599) contains 0. Additionally, the $P(\beta_{ptype} < 0)$ is estimated to be 0.7168 while $P(\beta_{ptype} > 0)$ is estimated to be 0.2832, which does not provide overwhelming evidence that β_{ptype} is not 0. The trace and autocorrelation plots of β_{ptype} and its probabilities support convergence.

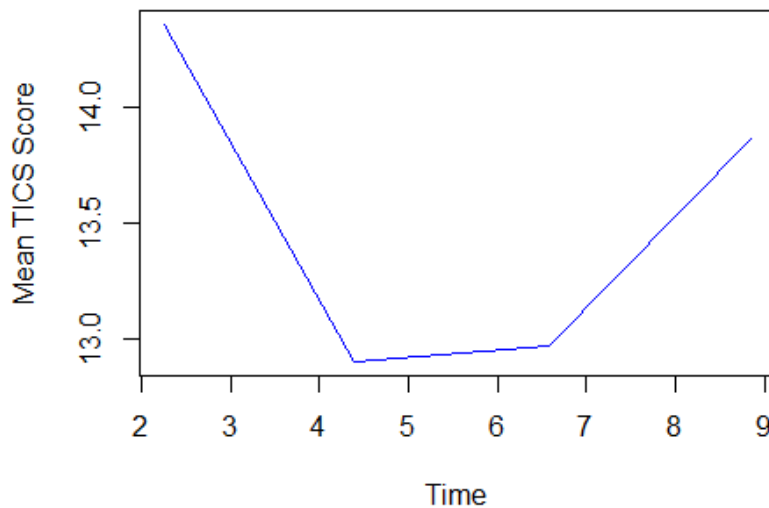
Rate of change of cognitive scores (TICS) among offspring of centenarians and controls

(Primary author: Anne Sartori)

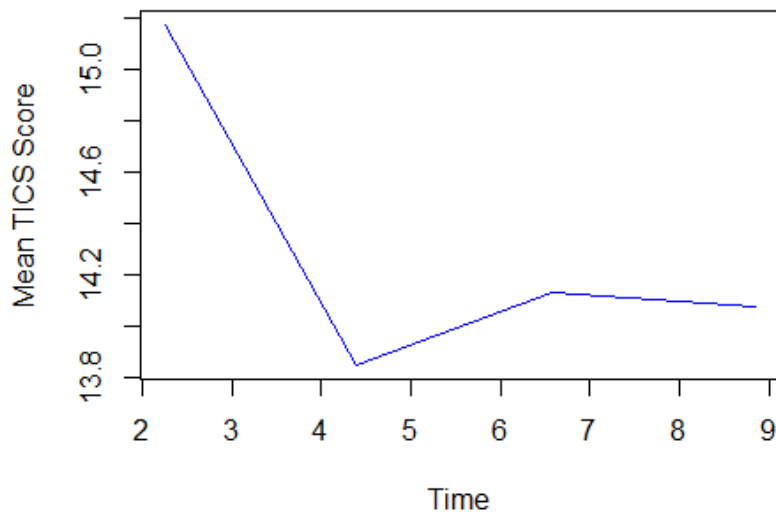
We now investigate the change of TICS scores in offspring of centenarians and controls. Plots, shown in the appendix, suggest considerable variation in the baseline levels of TICS among both groups. For neither the offspring of centenarians nor the controls is there a clear relationship between TICS scores and time. (Note that $t = 2$ in this paper denotes the

time at the second survey; in the statistical analyses, the corresponding value of the variable is the average amount of time passed since the first survey, measured in years.) However, in the following plots, both groups appear to have *average* TICS scores that decrease somewhat dramatically and then increase more slowly over time.

Mean TICS Score vs Time, Not Offspring of Centenar



Mean TICS Score vs Time, Offspring of Centenaria



We estimate three models. In the interest of brevity, we describe them all verbally and provide equations for the second model, our central one.

We represent the TICS scores as normally distributed and model the expected TICS score for a respondent as a function of time, of the treatment variable (centenarians vs. controls), and of the age and TICS score at baseline. While the TICS scale is always positive, we believe the Normal distribution to be a reasonable approximation for the scores. All of the models interact the treatment variable with time, and all predictors, including time, are de-meaned unless dichotomous. All models also include baseline age and TICS at baseline as covariates.

We use uninformative priors because we are not subject experts and are unfamiliar with the prior literature. In particular, every non-random effect is assumed to have an uninformative normally distributed prior mean and uninformative gamma-distributed prior variance.

Model 1: Change of TICS scores as a linear function of time

Our first model represents the change of TICS scores as a linear function of time. It includes random intercepts, and also random slopes for both the time variable and the interaction of treatment and time. The means of these random parameters are modeled as normally distributed, with uninformative normally distributed prior means and uninformative gamma-distributed prior variances. This first model allows us to investigate more carefully our inference from the plots that the random effects for slopes are unnecessary.

Results from the first model are shown in the following table. The predicted TICS score at baseline is essentially identical for offspring of centenarians and controls. Since the baseline time is 0, which is less than the mean time, the estimated baselines take into account the effects of time. Estimated in rjags, the predicted average score for offspring of centenarians at baseline is 0.258 lower than that of controls, but the 95% credible interval for the difference is (-0.992, 0.487) so we cannot reject the null hypothesis that the difference is 0 at the $\alpha = 0.05$ level.

The estimates suggest that TICS scores decline, on average, by about two tenths of a percentage point per year for controls (estimate -0.225, 95% CI (-0.340, -0.113)) and decline, on average, a twentieth of a percentage point less than that per year for offspring of centenarians. The difference between offspring of centenarians and the controls is both substantively small and statistically insignificant (0.0468, 95% CI: (-0.0884, 0.180)). Only about half the samples had a positive difference in the growth rate between offspring of centenarians and controls. Thus, this model would not lead us to reject the hypothesis of no difference between offspring of centenarians in the growth of cognitive scores.

Estimates from Model 1: Linear Model with Random Intercept and Slopes

Parameter	Estimate	SE
difference in expected baseline, treatment-control	-0.2580	0.3840
mean intercept	13.2000	0.2050

Parameter	Estimate	SE
treatment	0.9770	0.2540
mean slope for Time-mean(Time)	-0.2250	0.0568
Mean slope for treatment*(Time-mean(Time))	0.0468	0.0698
variance of intercept random effects parameters	5.1300	0.4240
variance of slope random effects parameters	0.1100	0.0226
variance of treatment-time interaction random effects parameters	0.1010	0.0227
TICS01-mean of TICS018	0.5100	0.0311
Age01-mean(Age01)	-0.1690	0.0182

Also of interest in the first model is the variation of the random effects. The variance of the intercept is 5.13, which is fairly substantial, since the mean of all TICS scores in the dataset is 14.2. The variance of the two random slopes is much smaller, supporting the idea that random slopes are unlikely to be important for understanding the growth of TICS scores.

Model 2: Change of TICS scores as a quadratic function of time

To better model the possible nonlinear relationship suggested by the plots, our second model represents the growth of TICS as a quadratic function of time. This model retains random intercepts (specified as in the first model), but moves to fixed effects for the slopes of the time variable and the interaction of the treatment with time.

The second model assumes:

$$\begin{aligned}
 y_{i,t} &\sim N(\mu_{i,t}, \sigma^2) \text{ where } i \text{ refers to subject } i, t \text{ to time period } t \\
 \mu_{i,t} &= \alpha_i + \beta_1 * ptype_i + \beta_2 * (Time_t - mean(Time)) \\
 &+ \beta_3 * ptype_i * (Time_t - mean(Time)) + \beta_4 * (Time_t^2 - mean(Time^2)) \\
 &+ \beta_5 * ptype_i * (Time_t^2 - mean(Time^2)) + \beta_6 * TICS01_i - mean(TICS01)) \\
 &+ \beta_6 * AGE01_i - mean(AGE01)) \\
 \alpha_i &\sim N(abar, \tau_\alpha) \text{ (distribution of random effect for intercept)}
 \end{aligned}$$

The prior distributions of the parameters are assumed to be:

$$\begin{aligned}
 \beta_1 &\sim N(0.0, 1.0E - 6) \\
 \beta_2 &\sim N(0.0, 1.0E - 6) \\
 \beta_3 &\sim N(0.0, 1.0E - 6) \\
 \beta_4 &\sim N(0.0, 1.0E - 6) \\
 \beta_5 &\sim N(0.0, 1.0E - 6) \\
 \beta_6 &\sim N(0.0, 1.0E - 6) \\
 abar &\sim N(0.0, 1.0E - 6) \\
 \tau &= 1/\sigma^2 \sim \Gamma[1, 1] \\
 \tau_\alpha &\sim \Gamma[1, 1]
 \end{aligned}$$

The following table shows estimates from Model 2, which includes a quadratic time variable, both separately and interacted with the treatment. Again, we see that the predicted TICS scores at baseline are virtually indistinguishable for offspring of centenarians and controls; the estimated difference is positive, but again with considerable variation.

Estimates from Model 2: Quadratic Model with Random Intercept and Fixed Slopes

Parameter	Estimate	SE
difference in expected baseline, treatment-control	0.4390	0.8050
effect of time for controls, t=2	-0.5220	0.1690
effect of time for controls, t=3	-0.3000	0.0670
effect of time for controls, t=4	-0.0762	0.0940
effect of time for controls, t=5	0.1560	0.2090
interaction, treatment and time, t=2	-0.1020	0.2060
interaction, treatment and time, t=3	-0.0265	0.0852
interaction, treatment and time, t=4	0.0498	0.1120
interaction, treatment and time, t=5	0.1290	0.2480
mean intercept	13.3000	0.2160
interaction, treatment and time (offspring of centenarian)	0.9740	0.2550
Time-mean(Time)	-0.7580	0.2920
treatment*(Time-mean(Time))	-0.2600	0.3560
Time squared - mean(Time squared)	0.0512	0.0275
treatment*(Time squared-mean(Time squared))	0.0273	0.0331
TICS01-mean of TICS018	0.5090	0.0322
Age01-mean(Age01)	-0.1670	0.0175
variance of intercept random effect parameters	4.9500	0.4270

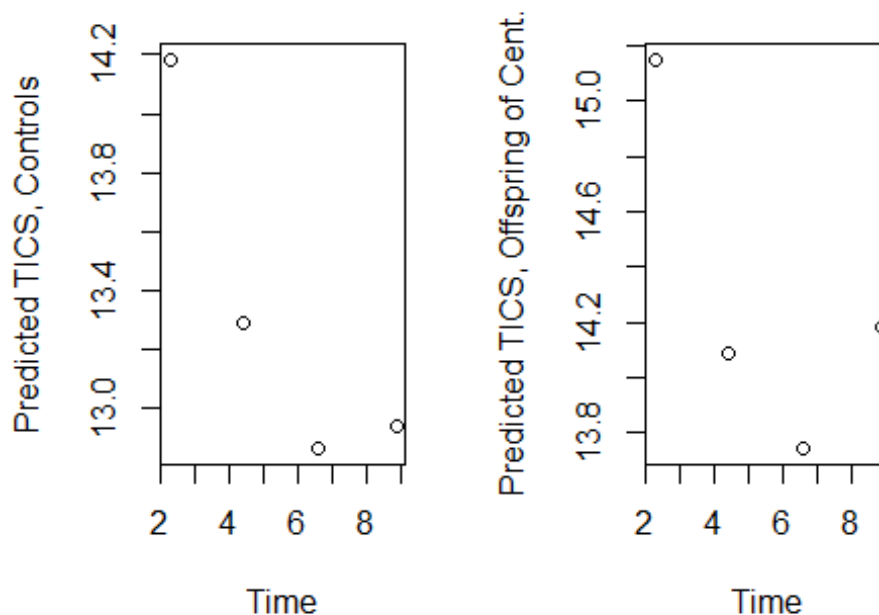
The estimates from Model 2 show the nonlinearity of the trend in TICS scores. For both offspring of centenarians and controls, TICS scores decline and then increase over time. Because of the nonlinearity, investigation of the growth rate of TICS, and of the difference in the growth rate of TICS for offspring of centenarians and controls, requires calculating a function of multiple coefficients. These effects are calculated as part of the MCMC estimation. (See appendix.)

For the control group, the results indicate a growth pattern in which TICS scores decline over time and then rise. However, the growth rate is statistically significant only in the first two periods, in which it is negative. (At time 2, the coefficient is -0.522 (95% CI: (-0.577, 0.258)), and at time 3, it is -0.300, 95% CI: (-0.180, 0.147); after that it is positive but the standard error is greater than the coefficient.

Model 2 again shows no statistically significant difference between offspring of centenarians and controls in the growth rate of TICS scores over time. As the table shows, the estimated effect of being offspring of a centenarian versus a control is negative for the first two time periods and then positive for the last two. That is, the estimates suggest that TICS scores of offspring of centenarians decline and then increase more rapidly than those of controls. However, this overall treatment effect, sampled in rjags, is statistically insignificant at all levels of time in the sample; the standard deviation for all time periods is approximately twice the treatment effect or more. Thus, the analyses do not provide sufficient evidence to conclude that there is a difference.

Model 2 appears to be a reasonable match for the data, which we can see by comparing the following plots, showing the predicted average TICS scores versus time, to the plots of the data presented earlier in the Methods section of this paper. The two sets of plots show the same general pattern of nonlinearity, though the inflection point occurs at earlier in the plots of the data than in those of the predictions of the model.

Predicted Average TICS, Model 2



Model 3: Change of TICS scores as a quadratic function of time including confounders

Our third and final model adds possible confounders identified in our initial exploratory analysis as covariates in the second model, to assess whether the effects we identify persist when we adjust for ways in which the offspring of centenarians differ from the controls.

Our third model includes all terms from Model 2, and adds subjects' reported BMI (de-meaned), sex, presence or absence of diabetes, smoking status (yes or no), hypertension (presence or absence) and hypertension (yes or no). The conclusions are identical to those from Model 2. Again, there is no statistically significant difference between the predicted TICS at baseline for offspring of centenarians and controls (estimate 0.31, 95% CI: (-1.16, 1.79)). Again, the estimated effect of being an offspring of a centenarian on the growth rate of TICS scores declines and then rises with time, but the 95% credible intervals are very wide. For example, for the second time period, the estimated effect is -0.098, and the 95% credible interval is (-0.483, 0.273).

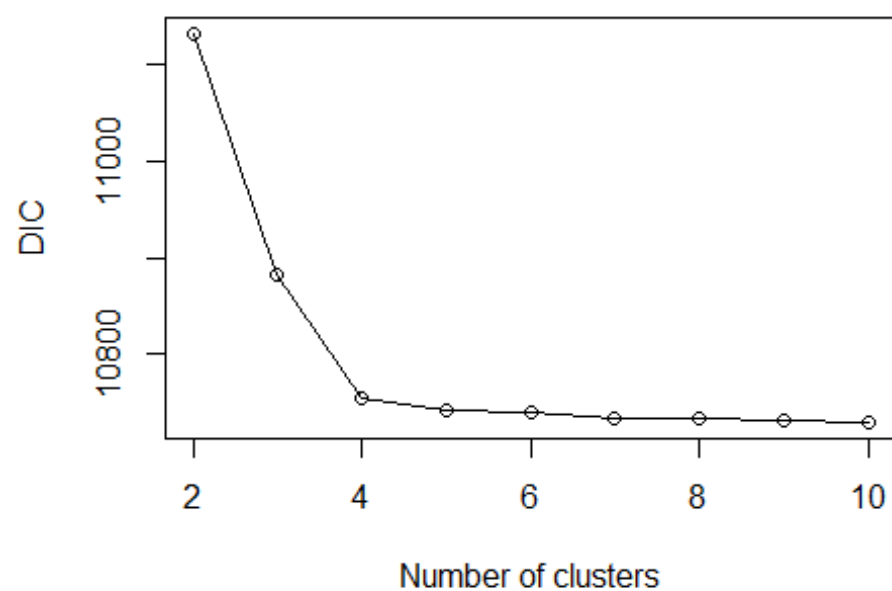
Conclusion

In keeping with a simple visual investigation, our estimates indicate that the growth rate of TICS scores is nonlinear, with TICS scores falling and then rising over time. However, they do not indicate a difference between offspring of centenarians and controls, either in the baseline predicted TICS scores or in the growth of TICS over time.

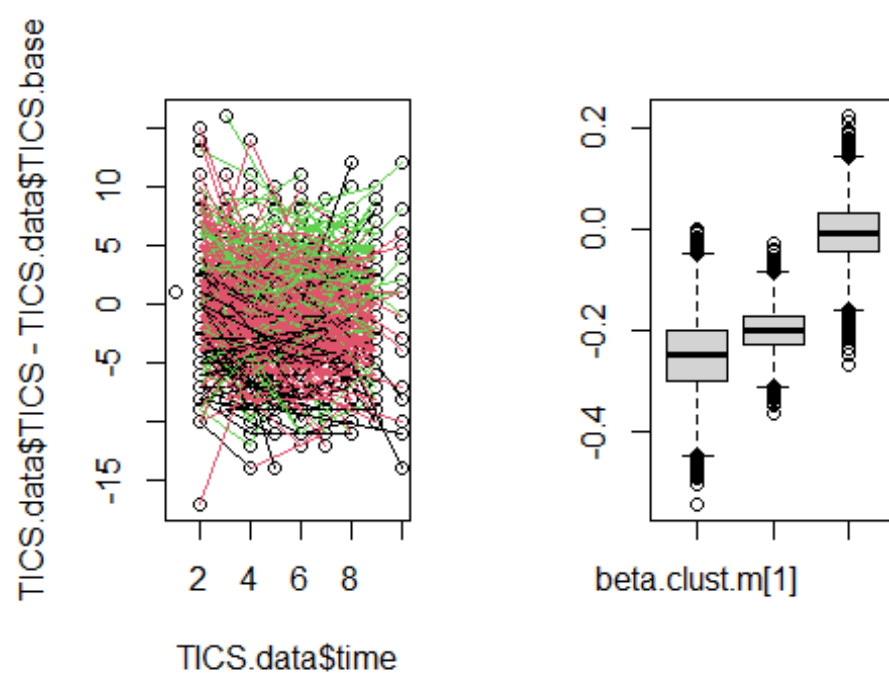
Presence of significant groups in rates of change in TICS score

(Primary author: Ailis Muldoon)

In light of the lack of association in rates of change of TICS score with centenarian offspring vs. controls, we investigated whether there were any clusters within the data that were significantly associated with different rates of change in TICS score. To do so, we constructed a model randomly assigning each individual subject to a category, with each category having an individual slope and intercept. The best number of clusters was determined by running models with numbers of clusters ranging from 2 to 10, then comparing the DIC values. The best number of clusters was picked by using the "kink" on a scree plot of DIC. In all cases, uninformative priors using a normal distribution of mean 0 and precision 0.001 were used for each cluster's intercept and slope; an uninformative Dirichlet prior was used for probabilities of each category; and the TICS score was modelled as a normal distribution based on the linear combination of intercept and slope multiplied by time, with a precision modelled by an uninformative gamma distribution. To minimize the impact of varying intercepts on subjects' cluster assignment, changes in TICS score from baseline were used in place of raw TICS score. The scree plot of DIC is shown below.



Based on the plot of DIC, the number of clusters chosen to simulate was 3.



As shown in the plots above, clusters are still more separated by intercept than by slope, and there is still significant overlap in slopes between groups 1 and 2; however, cluster 3 is somewhat more separated. The table below summarizes the differences in covariates earlier identified as possible confounders between the three clusters.

Recovered proportions for binomial variables in by cluster

	Mean cluster 1	CI lower bound cluster 1	CI upper bound cluster 1	Mean cluster 2	CI lower bound cluster 2	CI upper bound cluster 2	Mean cluster 3	CI lower bound cluster 3	CI upper bound cluster 3
s e x	0.423	0.343	0.502	0.547	0.495	0.599	0.799	0.734	0.856
c e n t	0.555	0.472	0.635	0.670	0.622	0.717	0.769	0.700	0.831
s t r o k e	0.049	0.019	0.089	0.022	0.010	0.039	0.012	0.001	0.034
d i a b e t e s	0.110	0.064	0.165	0.078	0.053	0.107	0.036	0.014	0.069
s m o k i n g	0.590	0.510	0.670	0.556	0.504	0.607	0.539	0.464	0.616
h t n	0.403	0.323	0.487	0.352	0.303	0.403	0.260	0.196	0.329

	Mean cluster 1	CI lower bound cluster 1	CI upper bound cluster 1	Mean cluster 2	CI lower bound cluster 2	CI upper bound cluster 2	Mean cluster 3	CI lower bound cluster 3	CI upper bound cluster 3
c a d	0.104	0.060	0.157	0.061	0.039	0.088	0.024	0.006	0.051

Based on the above table, we can see that cluster 1, which had the greatest decrease in TICS scores, has the lowest proportion of centenarian offspring (significant compared to 3) and the highest proportions of women (significant compared to 3) and those with a history of stroke (significant compared to 3), diabetes (significant compared to 3), hypertension (not significant), and coronary artery disease (significant compared to 3). Cluster 3, which had mostly flat TICS scores, has the highest proportion of centenarians (significant compared to 1) and the lowest proportion of women (significant compared to 1 and 2) and those with a history of stroke, diabetes, hypertension, and coronary artery disease. Smoking rates were similar across clusters.

Recovered proportions for binomial variables in by cluster

	Mean cluster 1	CI lower bound cluster 1	CI upper bound cluster 1	Mean cluster 2	CI lower bound cluster 2	CI upper bound cluster 2	Mean cluster 3	CI lower bound cluster 3	CI upper bound cluster 3
B M I	28.700	27.905	29.501	27.048	26.556	27.550	26.456	25.744	27.200
a g e. b a s e	79.736	78.713	80.765	75.144	74.512	75.790	69.704	68.789	70.661
Tl C S. b a s e	10.621	10.052	11.194	13.977	13.625	14.337	17.494	16.984	18.026

Similarly, we can see that cluster 1 has on average the highest baseline BMI (significant compared to cluster 3) and age (significant compared to clusters 2 and 3) and the lowest

baseline TICS scores (significant compared to clusters 2 and 3), while cluster 3 has the lowest baseline BMI and age and highest baseline TICS scores.

From this analysis, it seems likely that rates of change in TICS scores, while somewhat influenced by status as centenarian offspring, are more strongly associated with predictors such as age and baseline cognitive functioning.

Robustness to imputation

(Primary author: Thanwi Anna Lalu)

Several participants had missing values in the Age variables (Age02–Age05), which were used to construct the time variable in our main models. Initially, individuals with missing values were excluded from the complete-case analyses. To assess the robustness of our findings to this exclusion, we performed mean imputation by replacing each missing Age value with the average Age at that visit across all participants.

We re-estimated both Model 1 (random intercept and slope) and Model 2 (random intercept only with quadratic time) using the imputed dataset.

Results

Model 1 results remained stable after imputation. The estimated baseline difference in TICS scores between centenarian offspring and controls slightly decreased in magnitude but remained statistically insignificant. Fixed effects and variance components also showed minimal changes, indicating that the exclusion of participants with missing Age values did not meaningfully affect the model's conclusions.

Comparison of Model 1 Estimates: Complete Case vs. Imputed Age Variables

Variable	Model.1..Complete. Case.	Model.1..Imputed
difference in expected baseline, treatment-control	-0.2580	-0.2240
mean intercept	13.2000	13.2000
treatment	0.9770	0.9800
mean slope for (Time - mean(Time))	-0.2250	-0.2370
mean slope for treatment * (Time - mean(Time))	0.0468	0.0463
variance of intercept	5.1300	5.1900
variance of slope	0.1100	0.1310
variance of treatment-time interaction	0.1010	0.1220
TICS01 - mean(TICS01)	0.5100	0.5100

Variable	Model.1..Complete. Case.	Model.1..Imputed .
Age01 - mean(Age01)	-0.1690	-0.1750

In Model 2, both control and treatment groups had slightly higher baseline predictions with the imputed dataset, but the estimated difference between them remained small (0.418 vs. 0.439) and statistically non-significant. The same non-linear trend in TICS scores—initial decline followed by improvement—was preserved. The time and treatment-time interaction effects showed comparable patterns across both datasets.

Key Parameter Estimates: Model 2 (Complete Case vs Imputed Data)

Variable	Complete_Case	Imputed
difference in expected baseline (a0diff)	0.4390	0.418
baseline control (a0control)	13.7500	14.950
baseline treatment (a0treat)	14.1900	15.370
Time effect at t=2 (control1)	-0.5220	-0.597
Time effect at t=3 (control2)	-0.3000	-0.329
Time effect at t=4 (control3)	-0.0762	-0.062
Time effect at t=5 (control4)	0.1560	0.211
Interaction at t=2 (treatment1)	-0.1020	-0.145
Interaction at t=3 (treatment2)	-0.0265	-0.019
Interaction at t=4 (treatment3)	0.0498	0.107
Interaction at t=5 (treatment4)	0.1290	0.236
mean intercept (abar)	13.3000	12.660
variance of intercept (sigma.alpha)	4.9500	5.150

Across both Model 1 and Model 2, our findings remained consistent whether missing Age values were imputed or excluded. While mean imputation allowed for the inclusion of more participants, it did not materially alter the parameter estimates or overall conclusions. These results provide strong evidence that our analyses of cognitive trajectories are robust to the handling of missing Age data, supporting the validity of the original findings based on complete cases.

Conclusions

After thoroughly examining the dataset, we conclude that the association between being the offspring of a centenarian and cognitive function as measured by TICS score is not significant at a 95% confidence level as observed through Bayesian modelling. We further conclude that there is no significant association between rates of change of TICS score

over time with being the offspring of a centenarian. There is more evidence for an association between cognitive function and its change over time and some of the covariates included in the data, such as history of stroke, age at baseline, or BMI.

Appendix 1: Code

```
knitr::opts_chunk$set(echo = FALSE)

library(rjags)
library(coda)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(coda)
library(rjags)
library(GGally)
library(MASS)
library(knitr)
library(snowfall)
library(kableExtra)
data_wide <- read.csv("tics.data.2025.csv")

headers <- colnames(data_wide)
headers <- c(headers[1:(length(headers)-10)], "TICS", "Age")

TICS.long <- matrix(nrow = nrow(data_wide)*5, ncol = length(headers))
for (i in 1:nrow(TICS.long)) {
  for (p in 1:(length(headers)-2)) {
    TICS.long[i,p] <- data_wide[floor((i-1)/5)+1,p]
  }
  TICS.long[i,(length(headers)-1)] <- data_wide[floor(((i-1)/5)+1),
    paste("TICS0", as.character(((i-1) %% 5) + 1), sep = "")]
  TICS.long[i,(length(headers))] <- data_wide[floor(((i-1)/5)+1),
    paste("Age0", as.character(((i-1) %% 5) + 1), sep = "")]
}

TICS.long <- as.data.frame(TICS.long)
colnames(TICS.long) <- headers

df <- data_wide %>%
  filter(ptype %in% c(0,1))

#Make a factor for groups
df$Group <- factor(df$ptype, levels = c(0,1),
  labels = c("Control", "Offspring"))
Confounders_Age <- "
model {
  for(i in 1:N) {
```

```

    Age01[i] ~ dlnorm(mu_log.Age[ptype[i]+1], tau)
  }

  # Priors (log-scale for Age)
  mu_log.Age[1] ~ dnorm(0, 0.001) # Control (log-scale mean)
  mu_log.Age[2] ~ dnorm(0, 0.001) # Offspring (log-scale mean)
  tau ~ dgamma(0.001, 0.001)

  # Differences (exponentiate to interpret in real scale)
  mu.Age[1] <- exp(mu_log.Age[1])
  mu.Age[2] <- exp(mu_log.Age[2])
  diff.Age <- mu.Age[1] - mu.Age[2]
}
"

Confounders_BMI <- "
model {
  for(i in 1:N) {
    BMI[i] ~ dlnorm(mu_log.BMI[ptype[i]+1], tau)
  }

  # Priors (log-scale for BMI)
  mu_log.BMI[1] ~ dnorm(0, 0.001) # Control (log-scale mean)
  mu_log.BMI[2] ~ dnorm(0, 0.001) # Offspring (log-scale mean)
  tau ~ dgamma(0.001, 0.001)

  # Differences (exponentiate to interpret in real scale)
  mu.BMI[1] <- exp(mu_log.BMI[1])
  mu.BMI[2] <- exp(mu_log.BMI[2])
  diff.BMI <- mu.BMI[1] - mu.BMI[2]
}
"

Confounders_TICS <- "
model {
  for(i in 1:N) {
    TICS01[i] ~ dnorm(mu.TICS[ptype[i]+1], tau)
  }

  # Priors
  mu.TICS[1] ~ dnorm(0, 0.001) # Control
  mu.TICS[2] ~ dnorm(0, 0.001) # Offspring
  tau ~ dgamma(0.001, 0.001)

  # Differences
  diff.TICS <- mu.TICS[1] - mu.TICS[2]
}
"

```



```

Confounders_Education <- "
model {
  for(i in 1:N) {
    Years.of.Education[i] ~ dpois(lambda.Edu[ptype[i]+1])
  }

  # Priors (Poisson mean rate)
  lambda.Edu[1] ~ dgamma(0.001, 0.001) # Control
  lambda.Edu[2] ~ dgamma(0.001, 0.001) # Offspring

  # Differences
  diff.Edu <- lambda.Edu[1] - lambda.Edu[2]
}
"

# Set initial values
initsQ1 <- list(.RNG.name="base::Super-Duper", .RNG.seed=10)

# Prepare dataset for JAGS
tics.list <- list(
  N = nrow(df),
  ptype = df$ptype,
  Age01 = df$Age01,
  BMI = df$BMI,
  Years.of.Education = df$Years.of.Education,
  TICS01 = df$TICS01
)

# Run Models
model.Age <- jags.model(textConnection(Confounders_Age),
  data=tics.list, n.adapt=1000, inits=initsQ1)
model.BMI <- jags.model(textConnection(Confounders_BMI),
  data=tics.list, n.adapt=1000, inits=initsQ1)
model.TICS <- jags.model(textConnection(Confounders_TICS),
  data=tics.list, n.adapt=1000, inits=initsQ1)
model.Edu <- jags.model(textConnection(Confounders_Education),
  data=tics.list, n.adapt=1000, inits=initsQ1)

# Burn-in
update(model.Age, n.iter=10000)
update(model.BMI, n.iter=10000)
update(model.TICS, n.iter=10000)
update(model.Edu, n.iter=10000)

# MCMC Sampling
test.Age <- coda.samples(model.Age, c("mu.Age[1]", "mu.Age[2]", "diff.Age"),
  n.iter=10000)
test.BMI <- coda.samples(model.BMI, c("mu.BMI[1]", "mu.BMI[2]", "diff.BMI"),
  n.iter=10000)

```

```

test.TICS <- coda.samples(model.TICS, c("mu.TICS[1]", "mu.TICS[2]",
"diff.TICS"),
                        n.iter=10000)
test.Edu <- coda.samples(model.Edu, c("mu.Edu[1]", "mu.Edu[2]", "diff.Edu"),
                        n.iter=10000)

#Summary
summary(test.Age)
summ.age <- summary(test.Age)
summary(test.BMI)
summ.BMI <- summary(test.BMI)
summary(test.TICS)
summ.TICS <- summary(test.TICS)
summary(test.Edu)
summ.edu <- summary(test.Edu)

# stroke
# 1. Create a 0/1 outcome for Stroke
df$StrokeYN <- ifelse(df$MC.Stroke == "Yes", 1, 0)
# 2. Create a 0/1 indicator for Group (Offspring = 1, Control = 0)
df$GroupBinary <- ifelse(df$Group == "Offspring", 1, 0)
# 3. Subset out any missing values in these two columns
df_stroke <- subset(df, !is.na(StrokeYN) & !is.na(GroupBinary))
# 4. Build data list for JAGS
stroke_data <- list(
  N = nrow(df_stroke),      # total observations
  Y = df_stroke$StrokeYN,   # outcome 0/1
  X = df_stroke$GroupBinary # group 0/1
)
# 5. Write the JAGS Logistic model as a string
stroke_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    # Priors
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)

    # Derived quantity: odds ratio for Offspring vs. Control
    OR <- exp(beta1)
  }
"
# 6. Run model
jags_stroke <- jags.model(
  textConnection(stroke_model),
  data = stroke_data,
  n.chains = 2,
  n.adapt = 1000
)

```

```

# Burn-in
update(jags_stroke, n.iter = 2000)
# Draw posterior samples
stroke_samples <- coda.samples(
  jags_stroke,
  variable.names = c("beta0", "beta1", "OR"),
  n.iter = 5000
)
# 7. Posterior summary & diagnostics
summary(stroke_samples)
plot(stroke_samples)

# Diabetes
# 1. Create a 0/1 outcome
df$DiabetesYN <- ifelse(df$MC.Diabetes.Mellitus == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_diabetes <- subset(df, !is.na(DiabetesYN) & !is.na(GroupBinary))
# 3. Build data list
diabetes_data <- list(
  N = nrow(df_diabetes),
  Y = df_diabetes$DiabetesYN,
  X = df_diabetes$GroupBinary
)
# 4. Model string
diabetes_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"
# 5. Compile + burn-in + sample
jags_diabetes <- jags.model(
  textConnection(diabetes_model),
  data = diabetes_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_diabetes, 2000)
diabetes_samples <- coda.samples(
  jags_diabetes,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(diabetes_samples)
plot(diabetes_samples)

```

```

# Smoking
# 1. Create a 0/1 outcome
df$SmokeYN <- ifelse(df$SH.Ever.Smoked. == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_smoking <- subset(df, !is.na(SmokeYN) & !is.na(GroupBinary))
# 3. Build data list
smoking_data <- list(
  N = nrow(df_smoking),
  Y = df_smoking$SmokeYN,
  X = df_smoking$GroupBinary
)

# 4. Model string
smoking_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"

# 5. Compile + burn-in + sample
jags_smoking <- jags.model(
  textConnection(smoking_model),
  data = smoking_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_smoking, 2000)
smoking_samples <- coda.samples(
  jags_smoking,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(smoking_samples)
plot(smoking_samples)

# Hypertension
# 1. Create a 0/1 outcome
df$HTNYN <- ifelse(df$MC.HTN == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_htn <- subset(df, !is.na(HTNYN) & !is.na(GroupBinary))
# 3. Build data list
htn_data <- list(
  N = nrow(df_htn),

```

```

Y = df_htn$HTNYN,
X = df_htn$GroupBinary
)
# 4. Model string
htn_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"
# 5. Compile + burn-in + sample
jags_htn <- jags.model(
  textConnection(htn_model),
  data = htn_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_htn, 2000)
htn_samples <- coda.samples(
  jags_htn,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(htn_samples)
plot(htn_samples)

# Coronary artery disease
# 1. Create a 0/1 outcome
df$CADYN <- ifelse(df$MC.Coronary.Artery.Disease == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_cad <- subset(df, !is.na(CADYN) & !is.na(GroupBinary))
# 3. Build data list
cad_data <- list(
  N = nrow(df_cad),
  Y = df_cad$CADYN,
  X = df_cad$GroupBinary
)
# 4. Model string
cad_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)

```

```

    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"
# 5. Compile + burn-in + sample
jags_cad <- jags.model(
  textConnection(cad_model),
  data = cad_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_cad, 2000)
cad_samples <- coda.samples(
  jags_cad,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(cad_samples)
plot(cad_samples)

# Cancer
# 1. Create a 0/1 outcome
df$CancerYN <- ifelse(df$MC.Cancer == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_cancer <- subset(df, !is.na(CancerYN) & !is.na(GroupBinary))
# 3. Build data list
cancer_data <- list(
  N = nrow(df_cancer),
  Y = df_cancer$CancerYN,
  X = df_cancer$GroupBinary
)
# 4. Model string
cancer_model <- "
model {
  for (i in 1:N) {
    Y[i] ~ dbin(p[i], 1)
    logit(p[i]) <- beta0 + beta1 * X[i]
  }
  beta0 ~ dnorm(0, 0.0001)
  beta1 ~ dnorm(0, 0.0001)
  OR <- exp(beta1)
}
"
# 5. Compile + burn-in + sample
jags_cancer <- jags.model(
  textConnection(cancer_model),
  data = cancer_data,
  n.chains = 2,
  n.adapt = 1000
)

```

```

update(jags_cancer, 2000)
cancer_samples <- coda.samples(
  jags_cancer,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(cancer_samples)
plot(cancer_samples)

# Heart attack
# 1. Create a 0/1 outcome
df$HeartAttackYN <- ifelse(df$MC.Heart.Attack == "Yes", 1, 0)
# 2. We already have df$GroupBinary
df_heartattack <- subset(df, !is.na(HeartAttackYN) & !is.na(GroupBinary))
# 3. Build data list
heartattack_data <- list(
  N = nrow(df_heartattack),
  Y = df_heartattack$HeartAttackYN,
  X = df_heartattack$GroupBinary
)
# 4. Model string
heartattack_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"
# 5. Compile + burn-in + sample
jags_heartattack <- jags.model(
  textConnection(heartattack_model),
  data = heartattack_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_heartattack, 2000)
heartattack_samples <- coda.samples(
  jags_heartattack,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(heartattack_samples)
plot(heartattack_samples)

# Sex
# 1. Create a 0/1 outcome

```

```

# In this dataset: sex = 0 for Females, 1 for Males
df$SexYN <- ifelse(df$sex == 1, 1, 0)
# 2. We already have df$GroupBinary
df_sex <- subset(df, !is.na(SexYN) & !is.na(GroupBinary))
# 3. Build data list
sex_data <- list(
  N = nrow(df_sex),
  Y = df_sex$SexYN,
  X = df_sex$GroupBinary
)
# 4. Model string
sex_model <- "
  model {
    for (i in 1:N) {
      Y[i] ~ dbin(p[i], 1)
      logit(p[i]) <- beta0 + beta1 * X[i]
    }
    beta0 ~ dnorm(0, 0.0001)
    beta1 ~ dnorm(0, 0.0001)
    OR <- exp(beta1)
  }
"
# 5. Compile + burn-in + sample
jags_sex <- jags.model(
  textConnection(sex_model),
  data = sex_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_sex, 2000)
sex_samples <- coda.samples(
  jags_sex,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)
summary(sex_samples)
plot(sex_samples)

# Aspirin
# 1. Create a 0/1 outcome
df$AspirinYN <- ifelse(df$MC.Aspirin == "Yes", 1, 0)

# 2. We already have df$GroupBinary
df_aspirin <- subset(df, !is.na(AspirinYN) & !is.na(GroupBinary))

# 3. Build data list
aspirin_data <- list(
  N = nrow(df_aspirin),
  Y = df_aspirin$AspirinYN,

```



```

X = df_aspirin$GroupBinary
)

# 4. Model string
aspirin_model <- "
model {
  for (i in 1:N) {
    Y[i] ~ dbin(p[i], 1)
    logit(p[i]) <- beta0 + beta1 * X[i]
  }
  beta0 ~ dnorm(0, 0.0001)
  beta1 ~ dnorm(0, 0.0001)
  OR <- exp(beta1)
}
"

# 5. Compile + burn-in + sample
jags_aspirin <- jags.model(
  textConnection(aspirin_model),
  data = aspirin_data,
  n.chains = 2,
  n.adapt = 1000
)
update(jags_aspirin, 2000)

aspirin_samples <- coda.samples(
  jags_aspirin,
  c("beta0", "beta1", "OR"),
  n.iter = 5000
)

# 6. Summary and plot
summary(aspirin_samples)
plot(aspirin_samples)
results <- data.frame(
  Variable = c(
    "Stroke",
    "Diabetes",
    "Smoking",
    "Hypertension",
    "Coronary Artery Disease",
    "Cancer",
    "Heart Attack",
    "Sex (Male)",
    "Aspirin Use"
  ),
  `Odds ratio` = c(
    "0.39",
    "0.40",

```

```

      "0.63",
      "0.53",
      "0.36",
      "0.88",
      "0.71",
      "1.88",
      "0.75"
    ),
    `95% Credible Interval (OR)` = c(
      "[0.13, 0.87]",
      "[0.22, 0.66]",
      "[0.45, 0.84]",
      "[0.39, 0.72]",
      "[0.18, 0.63]",
      "[0.62, 1.23]",
      "[0.35, 1.32]",
      "[1.37, 2.51]",
      "[0.55, 1.00]"
    ),
    Significant = c(
      "Yes",
      "Yes",
      "Yes",
      "Yes",
      "Yes",
      "No",
      "No",
      "Yes",
      "No"
    )
  )
)

kable(results, caption = "Summary of 95% Credible Intervals and
Significance by Variable")
table(data_wide$type)
summary(data_wide)
tics<-data_wide %>% drop_na(BMI,
sex,MC.Stroke,MC.Diabetes.Mellitus,MC.HTN,SH.Ever.Smoked.,MC.Coronary.Artery.
Disease)

#Change yes/no to 1,0
tics <- tics %>%
  mutate(across(c(MC.Stroke, MC.Diabetes.Mellitus,
MC.HTN,SH.Ever.Smoked.,MC.Coronary.Artery.Disease), ~ ifelse(. == "No", 0,
1)))
tics.list <- as.list(tics)

```

```

#Double check missing removed
vars_to_check <- c("BMI",
"sex", "MC.Stroke", "MC.Diabetes.Mellitus", "MC.HTN", "SH.Ever.Smoked.", "MC.Coronary.Artery.Disease")
sapply(tics.list[vars_to_check], function(x) any(is.na(x)))

summary(tics.list)
tics.list$N <- length(tics.list$ID)
head(tics.list)
colnames(tics)
TICS.dist<-
  "model{
    for (i in 1:N){
      TICS01[i] ~ dnorm(mu[i], tau)
      mu[i] <-beta.0+beta.BMI*(BMI[i]-
mean.BMI)+beta.sex*sex[i]+beta.MC.Stroke*MC.Stroke[i]+beta.MC.Diabetes.Mellitus*MC.Diabetes.Mellitus[i]+beta.MC.HTN*MC.HTN[i]+beta.SH.Ever.Smoked.*SH.Ever.Smoked.[i]+beta.MC.Coronary.Artery.Disease*MC.Coronary.Artery.Disease[i]+beta.ptype*ptype[i]}

##prior
tau ~ dgamma(1,1);
beta.0 ~ dnorm(0, 0.0001);
beta.BMI ~ dnorm(0, 0.0001);
beta.sex ~ dnorm(0, 0.0001);
beta.MC.Stroke ~ dnorm(0, 0.0001);
beta.MC.Diabetes.Mellitus ~ dnorm(0, 0.0001);
beta.MC.HTN ~ dnorm(0, 0.0001);
beta.SH.Ever.Smoked. ~ dnorm(0, 0.0001);
beta.MC.Coronary.Artery.Disease ~ dnorm(0, 0.0001);
beta.ptype ~ dnorm(0, 0.0001);

#Inference
parameter[1]<-beta.0
parameter[2]<-beta.BMI
parameter[3]<-beta.sex
parameter[4]<-beta.MC.Stroke
parameter[5]<-beta.MC.Diabetes.Mellitus
parameter[6]<-beta.MC.HTN
parameter[7]<-beta.SH.Ever.Smoked.
parameter[8]<-beta.MC.Coronary.Artery.Disease
parameter[9]<-beta.ptype

P_beta.ptype_greater<-step(beta.ptype-0)
P_beta.ptype_lesser<-step(0-beta.ptype)

mean.BMI<-mean(BMI[])

```

```

}"
inits.TIC.Dist <- list(.RNG.name="base::Super-Duper", .RNG.seed=11)
model.TICS.dist <- jags.model(textConnection(TICS.dist),data=tics.list,
n.adapt=1500, inits = inits.TIC.Dist)
update(model.TICS.dist, n.iter=10000)
test.TICS.dist <- coda.samples(model.TICS.dist,
c('beta.ptype', 'P_beta.ptype_greater', 'P_beta.ptype_lesser', 'mean.BMI'),
n.iter=10000)
summary(test.TICS.dist)

par(mfrow=c(3,2))
traceplot(test.TICS.dist)
plot(test.TICS.dist)
autocorr.plot(test.TICS.dist)
Q2_table <- data.frame(
  Variable = c("P_beta.ptype_greater", "P_beta.ptype_lesser", "beta.ptype",
"mean.BMI"),
  Estimate = c( 0.7168, 0.2832, 0.1897, 27.2432),
  SD = c(0.4506, 0.4506, 0.3392, 0.00002)
)
kable(Q2_table, caption = "Relevant Posterior Estimates")
summary_test <- summary(test.TICS.dist)
summary_data <- summary_test$statistics[, c("Mean", "SD")]

summary_df <- data.frame(
  Variable = rownames(summary_data),
  Estimate = summary_data[, "Mean"],
  SD = summary_data[, "SD"]
)
kable(summary_df, caption = "Full Posterior Estimates")
#Create a time variable that is average of visit time
tics.data <- data_wide[complete.cases(data_wide[, c("ptype", "Age01",
"TICS01",
"Age02", "Age03",
"Age04")]), ]

Time<- rep(0,4)
Time[1]<-mean(tics.data$Age02 - tics.data$Age01,na.rm=TRUE)
Time[2]<-Time[1] + mean(tics.data$Age03 - tics.data$Age02,na.rm=TRUE)
Time[3]<-Time[2] + mean(tics.data$Age04 - tics.data$Age03,na.rm=TRUE)
Time[4]<-Time[3] + mean(tics.data$Age05 - tics.data$Age04,na.rm=TRUE)
#Time
tbar <- mean(Time)
Timesq <- Time^2
t2bar <- mean(Timesq)
tics_data_p_0 <- tics.data[data_wide$ptype == 0, ]
tics_data_p_1 <- tics.data[data_wide$ptype == 1, ]
y0 <-as.matrix(tics_data_p_0[,c('TICS02', 'TICS03', 'TICS04', 'TICS05')])
y1 <-as.matrix(tics_data_p_1[,c('TICS02', 'TICS03', 'TICS04', 'TICS05')])

```

```

plot(c(2,10),range(y0,na.rm=TRUE),type='n',ylab='TICS Score',xlab='Visit',
     main='Controls')
for (i in 1:nrow(y0)){
  lines(Time,y0[i,])
}
plot(c(2,10),range(y1,na.rm=TRUE),type='n',ylab='TICS Score',xlab='Visit',
     main='Offspring of Centenarians')
for (i in 1:nrow(y1)){
  lines(Time,y1[i,])
}
mean_tics_0 <- colMeans(tics_data_p_0[, c("TICS02", "TICS03", "TICS04",
"TICS05")], na.rm = TRUE)
plot(Time, mean_tics_0, type = "l", col = "blue", xlab = "Time", ylab = "Mean
TICS Score", main = "Mean TICS Score vs Time, Not Offspring of Centenarians")
mean_tics_1 <- colMeans(tics_data_p_1[, c("TICS02", "TICS03", "TICS04",
"TICS05")], na.rm = TRUE)
plot(Time, mean_tics_1, type = "l", col = "blue", xlab = "Time", ylab = "Mean
TICS Score", main = "Mean TICS Score vs Time, Offspring of Centenarians")
#Model 1

#Define the data that the model will use
y <- as.matrix(tics.data[,c('TICS02','TICS03','TICS04','TICS05')]); dim(y)
ptype <- as.numeric(tics.data$ptype)
Age01 <- as.numeric(tics.data$Age01)
Age01bar <- mean(Age01)
tics1 <- as.numeric(tics.data$TICS01)
TICS01bar <- mean(tics1)

cognitive.data <- list(Time = Time, N = nrow(tics.data), Y=y, ptype=ptype,
                      tics1=tics1, tbar=tbar, Age01bar=Age01bar,
Age01=Age01,
                      TICS01bar=TICS01bar)

model.1 <- "
  model
  {
    for( i in 1 : N ) {
      for( j in 1 : 4 ) {
        Y[i , j] ~dnorm(mu[i , j],tau)
        mu[i , j] <- alpha[i]+ beta1*ptype[i] + beta[i]*(Time[j]-tbar)
        + gamma[i]*ptype[i]*(Time[j]-tbar) + beta3*(tics1[i]-TICS01bar) +
        beta4*(Age01[i]-Age01bar)
      }
      alpha[i]~dnorm(abar,tau.a)
      beta[i]~dnorm(bbar,tau.b)
      gamma[i]~dnorm(gbar,tau.g)
    }
  }

```

```

#Priors

abbar~dnorm(0,1.0E-6)
bbar~dnorm(0,1.0E-6)
gbar~dnorm(0,1.0E-6)
tau ~dgamma(1,1)
tau.a ~ dgamma(1,1)
tau.b ~ dgamma(1,1)
tau.g ~ dgamma(1,1)
beta1 ~dnorm(0.0,1.0E-6)
beta3 ~dnorm(0.0,1.0E-6)
beta4 ~dnorm(0.0,1.0E-6)
sigma <- 1/tau
sigma.alpha <- 1/tau.a #var of random intercepts
sigma.beta <- 1/tau.b #var of random slopes on time
sigma.gamma <- 1/tau.g #var of random slopes on time*ptype
a0control <- abar + bbar*(-tbar) #baseline prediction for control group
at t=0
a0treat <- abar + bbar*(-tbar) +gbar*(-tbar)#baseline prediction,
treatment at t=0
a0diff <- a0treat - a0control
gbar_positive <- step(gbar)
}"

jags.1 <- jags.model(textConnection(model.1), data = cognitive.data,
  n.adapt = 1500, n.chains = 3,, inits = inits.TIC.Dist)
update(jags.1, 5000)
test.tics <- coda.samples(jags.1, c("a0control", "a0treat", "a0diff", "abar",
  "bbar", "gbar", "gbar_positive", "beta1",
  "beta3", "beta4", "sigma", "sigma.alpha",
  "sigma.beta", "sigma.gamma"),
  n.iter = 30000, thin=40)

mcmc_summary <- summary(test.tics)
mcmc_summary

# Extract samples to summarize
selected_params <- test.tics[, c('a0control', 'a0treat', 'a0diff', 'abar',
  'bbar', 'gbar', 'sigma.alpha', 'sigma.beta',
  'sigma.gamma', 'beta1', 'beta3', 'beta4')]

# Summary statistics
summary_stats <- summary(selected_params)
summary_stats
#save(test.tics, file = "Model1samples.RData")
# geweke.diag(test.tics, frac1 = 0.1, frac2 = 0.5)
# #Diagnostics
# plot(test.tics)

```

```

# autocorr.plot(test.tics)
# gelman.plot(test.tics, ylim = c(1, 4))

table_data <- data.frame(
  Parameter = c("difference in expected baseline, treatment-control",
    "mean intercept", "treatment", "mean slope for Time-
mean(Time)",
    "Mean slope for treatment*(Time-mean(Time))",
    "variance of intercept random effects parameters",
    "variance of slope random effects parameters",
    "variance of treatment-time interaction random effects
parameters",
    "TICS01-mean of TICS018", "Age01-mean(Age01)"),
  Estimate = c(-0.258, 13.2, 0.977, -0.225, 0.0468, 5.13, 0.110, 0.101,
0.510, -0.169),
  SE = c(0.384, 0.205, 0.254, 0.0568, 0.0698, 0.424, 0.0226, 0.0227, 0.0311,
0.0182)
)

# Print the table:
kable(table_data, caption = "Estimates from Model 1: Linear Model with Random
Intercept and Slopes")

##Model 2 : Model with only intercept random, quadratic Time variable
#(variables defined earlier)

#New data list because now we have square of Time
cognitive.data1b <- list(Time = Time, tbar=tbar, Timesq=Timesq, t2bar=t2bar,
N = nrow(tics.data), Y=y, ptype=ptype, tics1=tics1, Age01bar=Age01bar,
Age01=Age01, TICS01bar=TICS01bar)

model.2 <- "
  model
  {
    for( i in 1 : N ) {
      for( j in 1 : 4 ) {
        Y[i , j] ~dnorm(mu[i , j],tau)
        mu[i , j] <- alpha[i]
          + beta1*ptype[i]
          + beta2*(Time[j]-tbar)
          + beta3*ptype[i]*(Time[j]-tbar)
          + beta4*(Timesq[j]-t2bar)
          + beta5*ptype[i]*(Timesq[j]-t2bar)
          + beta6*(tics1[i]-TICS01bar)
          + beta7*(Age01[i]-Age01bar)
      }
      alpha[i]~dnorm(abar,tau.a)
    }
  }

```

```

#Priors

abbar~dnorm(0,1.0E-6)
tau ~dgamma(1,1)
tau.a ~ dgamma(1,1)
beta1 ~dnorm(0.0,1.0E-6)
beta2 ~dnorm(0.0,1.0E-6)
beta3 ~dnorm(0.0,1.0E-6)
beta4 ~dnorm(0.0,1.0E-6)
beta5 ~dnorm(0.0,1.0E-6)
beta6 ~dnorm(0.0,1.0E-6)
  beta7 ~dnorm(0.0,1.0E-6)
sigma <- 1/tau
sigma.alpha <- 1/tau.a #var of random intercepts
a0control <- abar + beta2*(-tbar) + beta4*(-t2bar)
#baseline prediction for control group at t=0
a0treat <- abar + beta2*(-tbar) + beta3*(-tbar) + beta4*(-t2bar) +
  beta5*(-t2bar)
  #baseline prediction, treatment at t=0
a0diff <- a0treat - a0control
control1 <- beta2 + 2*beta4*2.306092
control2 <- beta2 + 2*beta4*4.469031
control3 <- beta2 + 2*beta4*6.652050
control4 <- beta2 + 2*beta4*8.917675
treatment1 <- beta3 + 2*beta5*2.306092
#treatment 1 is derivative of
#beta3*(Time-tbar)+beta3*(Timesq-t2bar) eval at 2nd t, etc.
treatment2 <- beta3 + 2*beta5*4.469031
treatment3 <- beta3 + 2*beta5*6.652050
treatment4 <- beta3 + 2*beta5*8.917675

}"

jags.1b <- jags.model(textConnection(model.2), data = cognitive.data1b,
  n.adapt = 2500, n.chains = 4)
update(jags.1b, 2500)
test1b.tics <- coda.samples(jags.1b, c("abar", "a0control", "a0treat",
"a0diff",
  "control1", "control2", "control3", "control4",
  "treatment1", "treatment2", "treatment3", "treatment4",
  "beta1", "beta2", "beta3", "beta4", "beta5",
  "beta6", "beta7", "sigma", "sigma.alpha"),
  n.iter = 40000, thin=35)

mcmc_summary1b <- summary(test1b.tics)
mcmc_summary1b
# Extract samples
selected_params <- test1b.tics[, c('abar', "a0control", "a0treat", "a0diff",

```



```

      "treatment1", "treatment2", "treatment3", "treatment4",
      "control1", "control2", "control3", "control4",
      'sigma.alpha', 'beta1', 'beta2', 'beta3', 'beta4',
'beta5', 'beta6', 'beta7')])

# Summary statistics
summary_stats <- summary(selected_params)
print(summary_stats)

#Diagnostics
# geweke.diag(test1b.tics, frac1 = 0.1, frac2 = 0.5)
# plot(test1b.tics)
# autocorr.plot(test1b.tics)
# gelman.plot(test1b.tics, ylim = c(1, 4))
library(knitr)

# Create a data frame
table_data <- data.frame(
  Parameter = c("difference in expected baseline, treatment-control",
    "effect of time for controls, t=2",
    "effect of time for controls, t=3",
    "effect of time for controls, t=4",
    "effect of time for controls, t=5",
    "interaction, treatment and time, t=2",
    "interaction, treatment and time, t=3",
    "interaction, treatment and time, t=4",
    "interaction, treatment and time, t=5",
    "mean intercept",
    "interaction, treatment and time (offspring of centenarian)",
    "Time-mean(Time)",
    "treatment*(Time-mean(Time))",
    "Time squared - mean(Time squared)",
    "treatment*(Time squared-mean(Time squared))",
    "TICS01-mean of TICS018",
    "Age01-mean(Age01)",
    "variance of intercept random effect parameters"),
  Estimate = c(0.439, -0.522, -0.300, -0.0762, 0.156, -0.102, -0.0265, 0.0498, 0.129,
    13.3, 0.974, -0.758, -0.260, 0.0512, 0.0273, 0.509, -0.167,
4.95),
  SE = c(0.805, 0.169, 0.0670, 0.0940, 0.209, 0.206, 0.0852, 0.112, 0.248, 0.216,
    0.255, 0.292, 0.356, 0.0275, 0.0331, 0.0322, 0.0175, 0.427)
)

# Print the table
kable(table_data, caption = "Estimates from Model 2: Quadratic Model with
Random Intercept and Fixed Slopes")
#Plots of Predicted Average TICS vs. Time, Offspring of Centenarians and
#Controls
Time<- rep(0,4)

```

```

Time[1]<-mean(tics.data$Age02 - tics.data$Age01,na.rm=TRUE)
Time[2]<-Time[1] + mean(tics.data$Age03 - tics.data$Age02,na.rm=TRUE)
Time[3]<-Time[2] + mean(tics.data$Age04 - tics.data$Age03,na.rm=TRUE)
Time[4]<-Time[3] + mean(tics.data$Age05 - tics.data$Age04,na.rm=TRUE)
tbar <- mean(Time)
predicted_tics_avg_0 <- 13.31685 - 0.75794 * (Time - tbar) + 0.05124*
(Timesq - t2bar)
predicted_tics_avg_1 <- 13.31685 + 0.97405 - (0.75794+0.26017 ) * (Time -
tbar) +
(0.05124+0.02734) * (Timesq - t2bar)
par(mfrow=c(1,2))
plot(Time, predicted_tics_avg_0, xlab = "Time",
      ylab = "Predicted TICS, Controls")
plot(Time, predicted_tics_avg_1, xlab = "Time",
      ylab = "Predicted TICS, Offspring of Cent.")
mtext("Predicted Average TICS, Model 2", outer = TRUE, cex = 1.5, line = -
1.5)

par(mfrow=c(1,1))
#tics.c is tics.data with only complete cases on possible confounders
tics.c <- tics.data[complete.cases(tics.data[, c("ptype","Age01", "BMI",
"sex", "MC.Stroke", "MC.Diabetes.Mellitus", "SH.Ever.Smoked.", "MC.HTN",
"MC.Coronary.Artery.Disease"))], ]

BMI <- as.numeric(tics.c$BMI)
sex <- as.numeric(tics.c$sex)
#Change yes/no to 1,0
stroke <- ifelse(tics.c$MC.Stroke == "Yes", 1, 0)
diabetes <- ifelse(tics.c$MC.Diabetes.Mellitus == "Yes", 1, 0)
smoke <- ifelse(tics.c$SH.Ever.Smoked. == "Yes", 1, 0)
HTN <- ifelse(tics.c$MC.HTN == "Yes", 1, 0)
CAD <- ifelse(tics.c$MC.Coronary.Artery.Disease == "Yes", 1, 0)

#Create a time variable that is average of visit time
Time<- rep(0,4)
Time[1]<-mean(tics.c$Age02 - tics.c$Age01,na.rm=TRUE)
Time[2]<-Time[1] + mean(tics.c$Age03 - tics.c$Age02,na.rm=TRUE)
Time[3]<-Time[2] + mean(tics.c$Age04 - tics.c$Age03,na.rm=TRUE)
Time[4]<-Time[3] + mean(tics.c$Age05 - tics.c$Age04,na.rm=TRUE)
Time

#The first TICS will be a covariate so create y as the other TICS
y <- as.matrix(tics.c[,c('TICS02','TICS03','TICS04','TICS05')])

#Put data in the form we want for rjags hierarchical model
ptype <- as.numeric(tics.c$ptype)
tbar <- mean(Time)
Timesq <- Time^2
t2bar <- mean(Timesq)

```

```

BMIbar <- mean(tics.c$BMI)
Age01 <- as.numeric(tics.c$Age01)
cognitive.data.c <- list(Time = Time, tbar=tbar, Timesq=Timesq, t2bar=t2bar,
                        N = nrow(tics.c), Y=y, ptype=ptype, tics1=tics1,
                        BMI=BMI, sex = sex, stroke = stroke,

diabetes=diabetes,

                        smoke=smoke, HTN=HTN, CAD=CAD,
                        TICS01bar=TICS01bar,
                        Age01= Age01, Age01bar=Age01bar)

model.3 <- "
  model
  {
    for( i in 1 : N ) {
      for( j in 1 : 4 ) {
        Y[i , j] ~dnorm(mu[i , j],tau)
        mu[i , j] <-  alpha[i]
          + beta1*(Time[j]-tbar)
          + beta2*ptype[i]
          + beta3*ptype[i]*(Time[j]-tbar)
          + beta4*(Timesq[j]-t2bar)
          + beta5*ptype[i]*(Timesq[j]-t2bar)
          + beta6*(tics1[i]-TICS01bar)
          + beta7*(Age01[i]-Age01bar)
          + beta8*(BMI[i]-27.24)
          + beta9*sex[i]
          + beta10*stroke[i]
          + beta11*diabetes[i]
          + beta12*smoke[i]
          + beta13*HTN[i]
          + beta14*CAD[i]

      }
      alpha[i]~dnorm(alpha.mu,tau.a)
    }

    #Priors

    alpha.mu~dnorm(0,0.001)
    tau ~dgamma(1,1)
    tau.a ~ dgamma(1,1)
    beta1~dnorm(0.0,1.0E-6)
    beta2~dnorm(0.0,1.0E-6)
    beta3 ~dnorm(0.0,1.0E-6)
    beta4 ~dnorm(0.0,1.0E-6)
    beta5 ~dnorm(0.0,1.0E-6)
    beta6 ~dnorm(0.0,1.0E-6)
    beta7 ~dnorm(0.0,1.0E-6)
    beta8 ~dnorm(0.0,1.0E-6)

```

```

    beta9 ~dnorm(0.0,1.0E-6)
    beta10 ~dnorm(0.0,1.0E-6)
    beta11 ~dnorm(0.0,1.0E-6)
    beta12 ~dnorm(0.0,1.0E-6)
    beta13 ~dnorm(0.0,1.0E-6)
    beta14 ~dnorm(0.0,1.0E-6)
    alpha.0n <- alpha.mu + beta1*(-tbar) + beta4*(-t2bar)
    #baseline tics for ptype=0, children of non-centenarians, BMI at its
#mean, all dummies=0
    alpha.0c <- alpha.mu + beta1*(-tbar) + beta4*(-t2bar) +
      beta3*(-tbar) + beta5*(-t2bar)
    #baseline tics for ptype=1, children of centenarians, BMI at its
#mean, all dummies=0
    alphadiff <- alpha.0c - alpha.0n
    sigma <- 1/tau
    sigma.alpha <- 1/tau.a #var of random intercepts
    treatment1 <- beta3 + 2*beta5*2.306092
    treatment2 <- beta3 + 2*beta5*4.469031
    treatment3 <- beta3 + 2*beta5*6.652050
    treatment4 <- beta3 + 2*beta5*8.917675

  }"

parameters <- c('alphadiff', 'beta1', 'beta2',
  'beta3', 'beta4', 'beta5',
  "treatment1", "treatment2", "treatment3", "treatment4",
  'sigma', 'sigma.alpha')

jags.3 <- jags.model(textConnection(model.3), data = cognitive.data.c,
  n.adapt = 2000, n.chains = 4)

update(jags.3, 25000)
test.tics.c <- coda.samples(jags.3, parameters, n.adapt=3000, n.iter = 40000,
thin=30)

summary(test.tics.c)

#Diagnostics
# geweke.diag(test.tics.c, frac1 = 0.1, frac2 = 0.5)
# # plot to a PDF
# plot(test.tics.c)
# autocorr.plot(test.tics.c)
# gelman.plot(test.tics.c, ylim = c(1, 4))
TICS.data.all <- data_wide[order(data_wide$ID),]
TICS.data <- TICS.data.all[,c("ID", "fam.num", "sex", "BMI")]
TICS.data$cent <- TICS.data.all$ptype
TICS.data$stroke <- as.numeric(TICS.data.all$MC.Stroke == "Yes")
TICS.data$diabetes <- as.numeric(TICS.data.all$MC.Diabetes.Mellitus == "Yes")
TICS.data$smoking <- as.numeric(TICS.data.all$SH.Ever.Smoked. == "Yes")

```

```

TICS.data$htn <- as.numeric(TICS.data.all$MC.HTN == "Yes")
TICS.data$cad <- as.numeric(TICS.data.all$MC.Coronary.Artery.Disease ==
"Yes")
TICS.data$age.base <- TICS.data.all$Age01
TICS.data$TICS.base <- TICS.data.all$TICS01

headers <- colnames(TICS.data)
TICS.long <- matrix(nrow = nrow(TICS.data)*4, ncol = length(headers) + 2)
for (i in 1:nrow(TICS.long)) {
  for (p in 1:(length(headers))) {
    TICS.long[i,p] <- TICS.data[(floor((i-1)/4)+1),p]
  }
  TICS.long[i,(length(headers)+1)] <- TICS.data.all[floor(((i-1)/4)+1),
paste("Age0", as.character(((i-1) % 4) + 2), sep = "")] -
TICS.data.all$Age01[floor(((i-1)/4)+1)]
  TICS.long[i,(length(headers)+2)] <- TICS.data.all[floor(((i-1)/4)+1),
paste("TICS0", as.character(((i-1) % 4) + 2), sep = "")]
}

headers <- c(headers, "time", "TICS")
TICS.long <- as.data.frame(TICS.long)
colnames(TICS.long) <- headers

TICS.data <- na.omit(TICS.long)
TICS.data$idx[1] <- 1
for (i in 2:nrow(TICS.data)) {
  if (TICS.data$ID[i] == TICS.data$ID[i-1]) {
    TICS.data$idx[i] <- TICS.data$idx[i-1]
  } else {
    TICS.data$idx[i] <- TICS.data$idx[i-1] + 1
  }
}

model.clusters <- "model{
  for (i in 1:N) {
    score[i] ~ dnorm(mu[i], tau.0)
    mu[i] <- beta.clust.c[clust.id[idx[i]]] +
beta.clust.m[clust.id[idx[i]]]*(time[i])
  }

  for (i in 1:nidx) {
    clust.id[i] ~ dcat(theta)
  }

  for (i in 1:n.clust) {
    beta.clust.c[i] ~ dnorm(0.0, 0.001);
  }

  for (i in 1:n.clust) {

```

```

    beta.clust.m[i] ~ dnorm(0.0, 0.001);
  }

  theta ~ ddirch(clust.prior);
  tau.0 ~ dgamma(1, 1);
}"
# test model to make sure it converges
n.clust <- 3
dir.prior <- rep(1, n.clust)
data <- list(N = nrow(TICS.data), idx = TICS.data$idx, nidx =
length(unique(TICS.data$idx)),
            time = TICS.data$time, dscore = TICS.data$TICS -
TICS.data$TICS.base, n.clust = n.clust, clust.prior = dir.prior)

jags.cluster <- jags.model(textConnection(model.clusters), data=data,
n.adapt=2000, n.chains = 2)
update(jags.cluster, 2000)
test.cluster <- coda.samples(jags.cluster, c('beta.clust.c', 'beta.clust.m',
'theta'), n.adapt=2000, n.iter=5000)
summary(test.cluster)
plot(test.cluster)
autocorr.plot(test.cluster)
geweke.plot(test.cluster)
gelman.plot(test.cluster)
dic.samples(jags.cluster, n.iter = 5000, n.chain = 2)
Jf <- function(j,...) {
  DIC.clust <- c()
  n.clust <- j
  dir.prior <- rep(1, n.clust)

  data <- list(N = nrow(TICS.data), idx = TICS.data$idx, nidx =
length(unique(TICS.data$idx)),
              time = TICS.data$time, score = TICS.data$TICS -
TICS.data$TICS.base, n.clust = n.clust, clust.prior = dir.prior)

  jags.cluster <- jags.model(textConnection(model.clusters), data=data,
n.adapt=2000, n.chains = 5)
  update(jags.cluster, 2000)
  DIC <- dic.samples(jags.cluster, n.iter = 5000)
  DIC.clust$dev <- sum(DIC$deviance)
  DIC.clust$pen <- sum(DIC$penalty)
  DIC.clust
  # Write out
}

start.time <- Sys.time()
sfInit(parallel=TRUE, cpus=9) # Start Cluster
sfLibrary(coda) # Ask for R-Package on cluster
sfLibrary(rjags)
sfExport('TICS.data', 'model.clusters') # Export all the data that will be

```

used by the function.

```
DIC.samp <- do.call('cbind', sfLapply(2:10, Jf)) # Distribute calculations
sfStop() # Stop the cluster
end.time <- Sys.time()
t <- as.numeric(difftime(end.time, start.time, units='secs'))

DIC.clust <- c()
for (i in 1:9) {
  DIC.clust[i] <- DIC.samp[,i]$dev + DIC.samp[,i]$pen
}
plot(2:10, DIC.clust, xlab="Number of clusters", ylab="DIC")
lines(2:10, DIC.clust)
n.clust <- 3
dir.prior <- rep(1, n.clust)
data <- list(N = nrow(TICS.data), idx = TICS.data$idx, nidx =
length(unique(TICS.data$idx)),
            time = TICS.data$time, score = TICS.data$TICS, n.clust = n.clust,
clust.prior = dir.prior)
inits <- list(.RNG.name="base::Super-Duper", .RNG.seed=13)

jags.cluster <- jags.model(textConnection(model.clusters), data=data,
n.adapt=2000, n.chains = 1, inits = inits)
update(jags.cluster, 2000)
test.cluster <- coda.samples(jags.cluster, c('beta.clust.c', 'beta.clust.m',
'theta', 'clust.id'),
                           n.adapt=2000, n.iter=10000, n.thin = 50)
summ.cluster <- summary(test.cluster)
# plot(test.cluster[[1]][,c(1:6, 668:670)])
# autocorr.plot(test.cluster[[1]][,c(1:6, 668:670)])
# geweke.plot(test.cluster)
# summary(test.cluster[[1]][,c(1:6, 668:670)])

for (i in 1:nrow(TICS.data)) {
  TICS.data$clust <- summ.cluster$quantiles[paste('clust.id',
as.character(TICS.data$idx), ']', sep=''), '50%']
}
par(mfrow = c(1,2))
plot(TICS.data$time, TICS.data$TICS - TICS.data$TICS.base)
for (i in 1:length(unique(TICS.data$idx))) {
  this.sub <- TICS.data[TICS.data$idx == i,]
  lines(this.sub$time, this.sub$TICS - this.sub$TICS.base, col =
this.sub$clust[1])
}

slopes <- as.matrix(test.cluster[[1]][,4:6])
boxplot(slopes)
addtl.rows <- duplicated(TICS.data$idx)
TICS.first <- TICS.data[!addtl.rows,]
```

```

model.bin <- "model {
  for (i in 1:N) {
    x[i] ~ dbin(theta[clust.id[i]], 1);
  }

  for (i in 1:n.clust) {
    theta[i] ~ dbeta(1,1);
  }
}"

bin.col <- c('sex', 'cent', 'stroke', 'diabetes', 'smoking', 'htn', 'cad')

Jf.bin <- function(j,...) {
  data <- list(N = nrow(TICS.first), n.clust =
length(unique(TICS.first$clust)), clust.id = TICS.first$clust, x =
TICS.first[,bin.col[j]])

  jags.bin <- jags.model(textConnection(model.bin), data=data, n.adapt=1000,
n.chains = 1)
  update(jags.bin, 1000)
  test.cluster <- coda.samples(jags.bin, c('theta'), n.adapt=2000,
n.iter=5000, n.thin = 1)
  test.cluster
}

start.time <- Sys.time()
sfInit(parallel=TRUE, cpus=length(bin.col)) # Start Cluster
sfLibrary(coda) # Ask for R-Package on cluster
sfLibrary(rjags)
sfExport('TICS.first', 'bin.col', 'model.bin') # Export all the data that
will be used by the function.
thetas <- do.call('cbind', sfLapply(1:length(bin.col), Jf.bin)) # Distribute
calculations
sfStop() # Stop the cluster
end.time <- Sys.time()
end.time-start.time
t <- as.numeric(difftime(end.time, start.time, units='secs'))
lb.1 <- c(); ub.1 <- c(); avg.1 <- c();
lb.2 <- c(); ub.2 <- c(); avg.2 <- c();
lb.3 <- c(); ub.3 <- c(); avg.3 <- c();
for (i in 1:length(bin.col)) {
  lb.1 <- rbind(lb.1, summary(thetas[[i]])$quantiles["theta[1]", "2.5%"])
  ub.1 <- rbind(ub.1, summary(thetas[[i]])$quantiles["theta[1]", "97.5%"])
  avg.1 <- rbind(avg.1, summary(thetas[[i]])$statistics["theta[1]", "Mean"])

  lb.2 <- rbind(lb.2, summary(thetas[[i]])$quantiles["theta[2]", "2.5%"])
  ub.2 <- rbind(ub.2, summary(thetas[[i]])$quantiles["theta[2]", "97.5%"])
  avg.2 <- rbind(avg.2, summary(thetas[[i]])$statistics["theta[2]", "Mean"])

```



```

lb.3 <- rbind(lb.3, summary(thetas[[i]])$quantiles["theta[3]", "2.5%"])
ub.3 <- rbind(ub.3, summary(thetas[[i]])$quantiles["theta[3]", "97.5%"])
avg.3 <- rbind(avg.3, summary(thetas[[i]])$statistics["theta[3]", "Mean"])

  boxplot(as.matrix(thetas[[i]]), main=bin.col[i])
}

binomial.var <- data.frame(avg.1, lb.1, ub.1, avg.2, lb.2, ub.2, avg.3, lb.3,
ub.3)
colnames(binomial.var) <- c('Mean cluster 1', 'CI lower bound cluster 1', 'CI
upper bound cluster 1',
                           'Mean cluster 2', 'CI lower bound cluster 2', 'CI
upper bound cluster 2',
                           'Mean cluster 3', 'CI lower bound cluster 3', 'CI
upper bound cluster 3')
rownames(binomial.var) <- bin.col
kable(binomial.var, caption = "Recovered proportions for binomial variables
in by cluster", digits=3)
model.norm <- "model {
  for (i in 1:N) {
    x[i] ~ dnorm(mu[clust.id[i]], tau.0);
  }

  for (i in 1:n.clust) {
    mu[i] ~ dnorm(0.0, 0.001);
  }
  tau.0 ~ dgamma(1,1)
}"

norm.col <- c('BMI', 'age.base', 'TICS.base')

Jf.norm <- function(j,...) {
  data <- list(N = nrow(TICS.first), n.clust =
length(unique(TICS.first$clust)), clust.id = TICS.first$clust, x =
TICS.first[,norm.col[j]])

  jags.norm <- jags.model(textConnection(model.norm), data=data,
n.adapt=1000, n.chains = 1)
  update(jags.norm, 1000)
  test.cluster <- coda.samples(jags.norm, c('mu'), n.adapt=2000, n.iter=5000,
n.thin = 1)
  test.cluster
}

start.time <- Sys.time()
sfInit(parallel=TRUE, cpus=length(norm.col)) # Start Cluster
sfLibrary(coda) # Ask for R-Package on cluster
sfLibrary(rjags)
sfExport('TICS.first', 'norm.col', 'model.norm') # Export all the data that

```

will be used by the function.

```
mus <- do.call('cbind', sfLapply(1:length(norm.col), Jf.norm)) # Distribute
calculations
sfStop() # Stop the cluster
end.time <- Sys.time()
end.time-start.time
t <- as.numeric(difftime(end.time, start.time, units='secs'))

lb.1 <- c(); ub.1 <- c(); avg.1 <- c();
lb.2 <- c(); ub.2 <- c(); avg.2 <- c();
lb.3 <- c(); ub.3 <- c(); avg.3 <- c();
for (i in 1:length(norm.col)) {
  lb.1 <- rbind(lb.1, summary(mus[[i]])$quantiles["mu[1]", "2.5%"])
  ub.1 <- rbind(ub.1, summary(mus[[i]])$quantiles["mu[1]", "97.5%"])
  avg.1 <- rbind(avg.1, summary(mus[[i]])$statistics["mu[1]", "Mean"])

  lb.2 <- rbind(lb.2, summary(mus[[i]])$quantiles["mu[2]", "2.5%"])
  ub.2 <- rbind(ub.2, summary(mus[[i]])$quantiles["mu[2]", "97.5%"])
  avg.2 <- rbind(avg.2, summary(mus[[i]])$statistics["mu[2]", "Mean"])

  lb.3 <- rbind(lb.3, summary(mus[[i]])$quantiles["mu[3]", "2.5%"])
  ub.3 <- rbind(ub.3, summary(mus[[i]])$quantiles["mu[3]", "97.5%"])
  avg.3 <- rbind(avg.3, summary(mus[[i]])$statistics["mu[3]", "Mean"])

  boxplot(as.matrix(mus[[i]]), main=norm.col[i])
}

normal.var <- data.frame(avg.1, lb.1, ub.1, avg.2, lb.2, ub.2, avg.3, lb.3,
ub.3)
colnames(normal.var) <- c('Mean cluster 1', 'CI lower bound cluster 1', 'CI
upper bound cluster 1',
                          'Mean cluster 2', 'CI lower bound cluster 2', 'CI
upper bound cluster 2',
                          'Mean cluster 3', 'CI lower bound cluster 3', 'CI
upper bound cluster 3')
rownames(normal.var) <- norm.col
kable(normal.var, caption = "Recovered proportions for binomial variables in
by cluster", digits=3)
tics.data.m <- data_wide
# Check for missing values
#colSums(is.na(tics.data.m))

for (i in 1:5) {
  age_col <- paste0("Age0", i)
  mean_age <- mean(tics.data.m[[age_col]], na.rm=TRUE) # Compute mean of
non-missing values
  tics.data.m[[age_col]][is.na(tics.data.m[[age_col]])] <- mean_age #
Replace missing values
}
```

```

#Create a time variable that is average of visit time
Time<- rep(0,4)
Time[1]<-mean(tics.data.m$Age02 - tics.data.m$Age01,na.rm=TRUE)
Time[2]<-Time[1] + mean(tics.data.m$Age03 - tics.data.m$Age02,na.rm=TRUE)
Time[3]<-Time[2] + mean(tics.data.m$Age04 - tics.data.m$Age03,na.rm=TRUE)
Time[4]<-Time[3] + mean(tics.data.m$Age05 - tics.data.m$Age04,na.rm=TRUE)
tbar <- mean(Time)
Timesq <- Time^2
t2bar <- mean(Timesq)

#Define the data that the model will use
tics.data.m <- tics.data.m[complete.cases(tics.data.m[, c("ptype", "Age01",
"TICS01", "Age02","Age03", "Age04"))], ]
y <- as.matrix(tics.data.m[,c('TICS02','TICS03','TICS04','TICS05')]); dim(y)
ptype <- as.numeric(tics.data.m$ptype)
Age01 <- as.numeric(tics.data.m$Age01)
Age01bar <- mean(Age01)
tics1 <- as.numeric(tics.data.m$TICS01)
TICS01bar <- mean(tics1)
cognitive.data.m <- list(Time = Time, N = nrow(tics.data.m), Y=y,
ptype=ptype, tics1=tics1, tbar=tbar, Age01bar=Age01bar,
Age01=Age01, TICS01bar=TICS01bar)

jags.1.m <- jags.model(textConnection(model.1), data = cognitive.data.m,
n.adapt = 1500, n.chains = 3)
update(jags.1.m, 5000)
test.tics.m <- coda.samples(jags.1.m, c("a0control", "a0treat", "a0diff",
"abar", "bbar", "gbar", "gbar_positive", "beta1","beta3", "beta4", "sigma",
"sigma.alpha", "sigma.beta", "sigma.gamma"), n.iter = 30000, thin=40)

mcmc_summary <- summary(test.tics.m)
# Extract samples for the specified parameters
selected_params <- test.tics.m[, c('a0control', 'a0treat', 'a0diff', 'abar',
'bbar', 'gbar', 'sigma.alpha', 'sigma.beta', 'sigma.gamma', 'beta1', 'beta3',
'beta4')]

# Obtain summary statistics
summary_stats <- summary(selected_params)
summary_stats

##Model 2 : Model with only intercept random, quadratic Time variable
#(variables defined earlier)

#New data list because now we have square of Time
cognitive.data1b.m <- list(Time = Time, tbar=tbar, Timesq=Timesq,
t2bar=t2bar, N = nrow(tics.data.m), Y=y, ptype=ptype, tics1=tics1,
Age01bar=Age01bar, Age01=Age01, TICS01bar=TICS01bar)

```

```

model.2.m <- "
model
{
  for( i in 1 : N ) {
    for( j in 1 : 4 ) {
      Y[i , j] ~dnorm(mu[i , j],tau)
      mu[i , j] <- alpha[i]
      + beta1*ptype[i]
      + beta2*(Time[j]-tbar)
      + beta3*ptype[i]*(Time[j]-tbar)
      + beta4*(Timesq[j]-t2bar)
      + beta5*ptype[i]*(Timesq[j]-t2bar)
      + beta6*(tics1[i]-TICS01bar)
      + beta7*(Age01[i]-Age01bar)
    }
    alpha[i]~dnorm(abar,tau.a)
  }

  #Priors

  abar~dnorm(0,1.0E-6)
  tau ~dgamma(1,1)
  tau.a ~ dgamma(1,1)
  beta1 ~dnorm(0.0,1.0E-6)
  beta2 ~dnorm(0.0,1.0E-6)
  beta3 ~dnorm(0.0,1.0E-6)
  beta4 ~dnorm(0.0,1.0E-6)
  beta5 ~dnorm(0.0,1.0E-6)
  beta6 ~dnorm(0.0,1.0E-6)
  beta7 ~dnorm(0.0,1.0E-6)
  sigma <- 1/tau
  sigma.alpha <- 1/tau.a #var of random intercepts
  a0control <- abar + beta2*(-tbar) + beta4*(-t2bar)
  #baseline prediction for control group at t=0
  a0treat <- abar + beta2*(-tbar) + beta3*(-tbar) + beta4*(-t2bar) +
    beta5*(-t2bar)
    #baseline prediction, treatment at t=0
  a0diff <- a0treat - a0control
  control1 <- beta2 + 2*beta4*2.007773
  control2 <- beta2 + 2*beta4*3.897765
  control3 <- beta2 + 2*beta4*5.775691
  control4 <- beta2 + 2*beta4*7.700159
  treatment1 <- beta3 + 2*beta5*2.007773
  #treatment 1 is derivative of
  #beta3*(Time-tbar)+beta3*(Timesq-t2bar) eval at 2nd t, etc.
  treatment2 <- beta3 + 2*beta5*3.897765
  treatment3 <- beta3 + 2*beta5*5.775691
  treatment4 <- beta3 + 2*beta5*7.700159

```

```

    }"
jags.1b.m <- jags.model(textConnection(model.2.m), data = cognitive.data1b.m,
                        n.adapt = 2500, n.chains = 4)
update(jags.1b.m, 2500)
test1b.tics.m <- coda.samples(jags.1b.m, c("abar", "a0control", "a0treat",
"a0diff",
                                "control1", "control2", "control3",
"control4",
                                "treatment1", "treatment2",
"treatment3", "treatment4",
                                "beta1", "beta2", "beta3", "beta4",
"beta5",
                                "beta6", "beta7",
                                "sigma", "sigma.alpha"), n.iter =
40000, thin=35)

mcmc_summary1b <- summary(test1b.tics.m)
# Extract samples for the specified parameters
selected_params <- test1b.tics.m[, c('abar', "a0control", "a0treat",
"a0diff",
                                "treatment1", "treatment2", "treatment3",
"treatment4",
                                "control1", "control2", "control3",
"control4",
                                'sigma.alpha', 'beta1', 'beta2', 'beta3',
'beta4', 'beta5', 'beta6', 'beta7')]]

# Obtain summary statistics
summary_stats <- summary(selected_params)
print(summary_stats)
comparison_table <- data.frame(
  Variable = c(
    "difference in expected baseline, treatment-control",
    "mean intercept",
    "treatment",
    "mean slope for (Time - mean(Time))",
    "mean slope for treatment * (Time - mean(Time))",
    "variance of intercept",
    "variance of slope",
    "variance of treatment-time interaction",
    "TICS01 - mean(TICS01)",
    "Age01 - mean(Age01)"
  ),
  `Model 1 (Complete Case)` = c(
    -0.258, 13.2, 0.977, -0.225, 0.0468, 5.13, 0.110, 0.101, 0.510, -0.169
  ),
  `Model 1 (Imputed)` = c(
    -0.224, 13.2, 0.980, -0.237, 0.0463, 5.19, 0.131, 0.122, 0.510, -0.175
  )
)

```

```
kable(comparison_table, caption = "Comparison of Model 1 Estimates: Complete Case vs. Imputed Age Variables")
```

```
comparison_table <- data.frame(
  Variable = c("difference in expected baseline (a0diff)",
    "baseline control (a0control)",
    "baseline treatment (a0treat)",
    "Time effect at t=2 (control1)",
    "Time effect at t=3 (control2)",
    "Time effect at t=4 (control3)",
    "Time effect at t=5 (control4)",
    "Interaction at t=2 (treatment1)",
    "Interaction at t=3 (treatment2)",
    "Interaction at t=4 (treatment3)",
    "Interaction at t=5 (treatment4)",
    "mean intercept (abar)",
    "variance of intercept (sigma.alpha)"),
  Complete_Case = c(0.439, 13.75, 14.19, -0.522, -0.300, -0.0762, 0.156,
    -0.102, -0.0265, 0.0498, 0.129, 13.3, 4.95),
  Imputed = c(0.418, 14.95, 15.37, -0.597, -0.329, -0.062, 0.211,
    -0.145, -0.019, 0.107, 0.236, 12.66, 5.15)
)
```

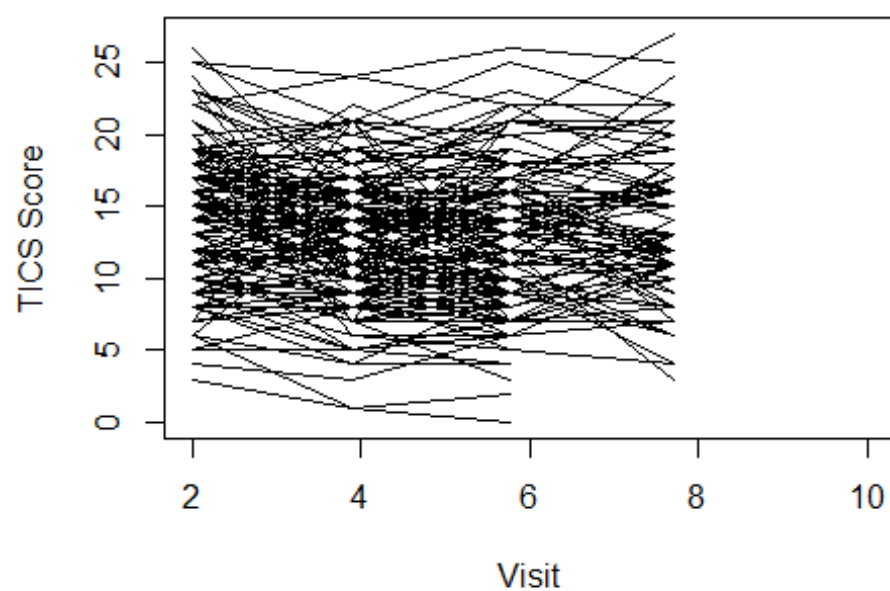
```
kable(comparison_table, caption = "Key Parameter Estimates: Model 2 (Complete Case vs Imputed Data)")
```

Appendix 2: Additional figures and tables

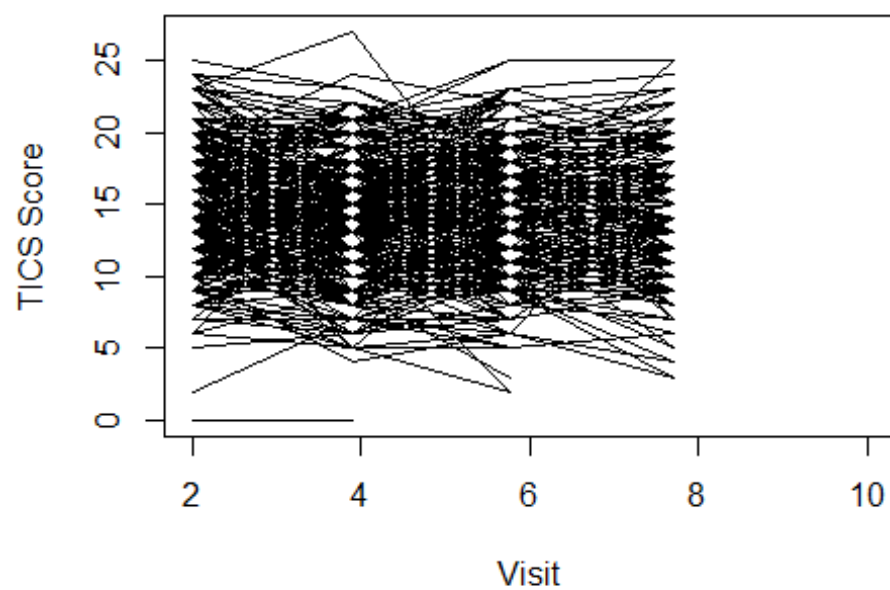
Full Posterior Estimates

	Variable	Estimate	SD
P_beta.ptype_greater	P_beta.ptype_greater	0.7168000	0.4505753
P_beta.ptype_lesser	P_beta.ptype_lesser	0.2832000	0.4505753
beta.ptype	beta.ptype	0.1897308	0.3392002
mean.BMI	mean.BMI	27.2431861	0.0000000

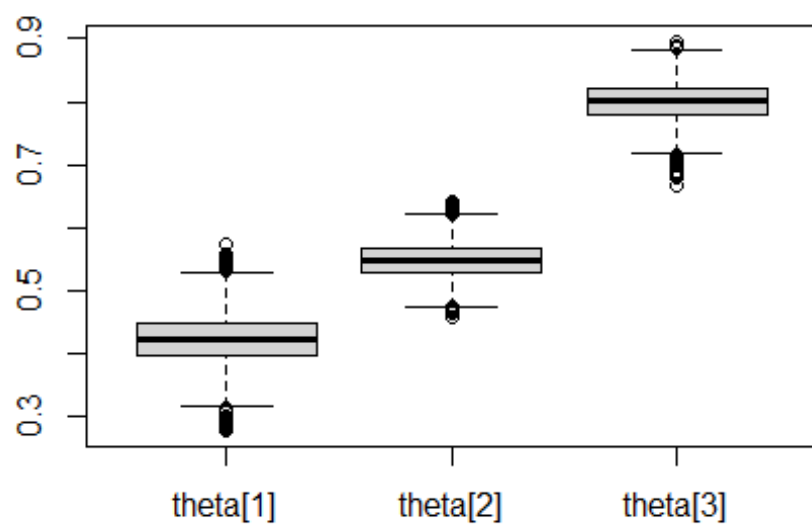
Controls



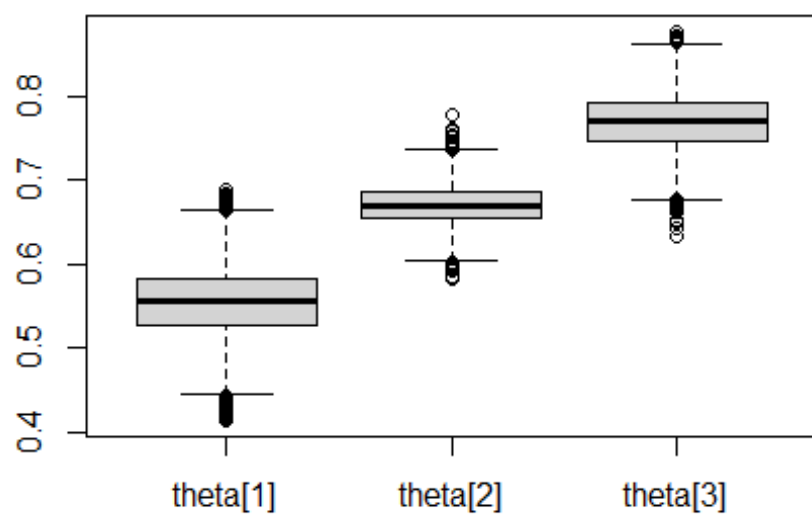
Offspring of Centenarians



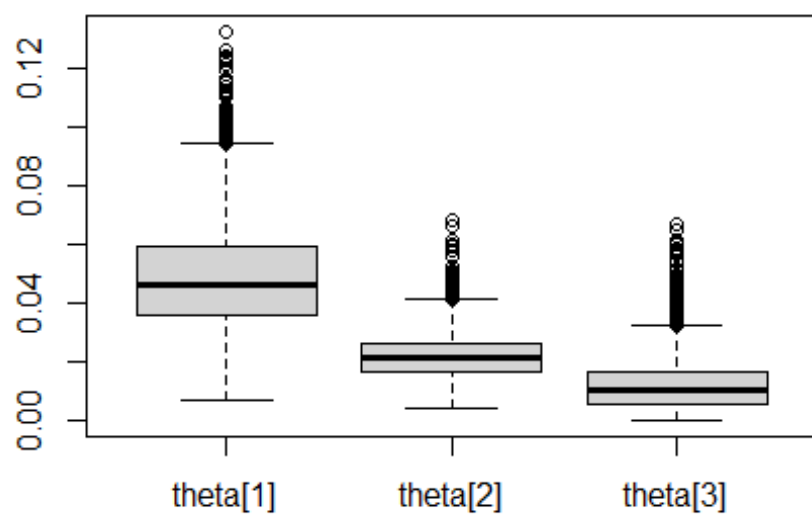
sex



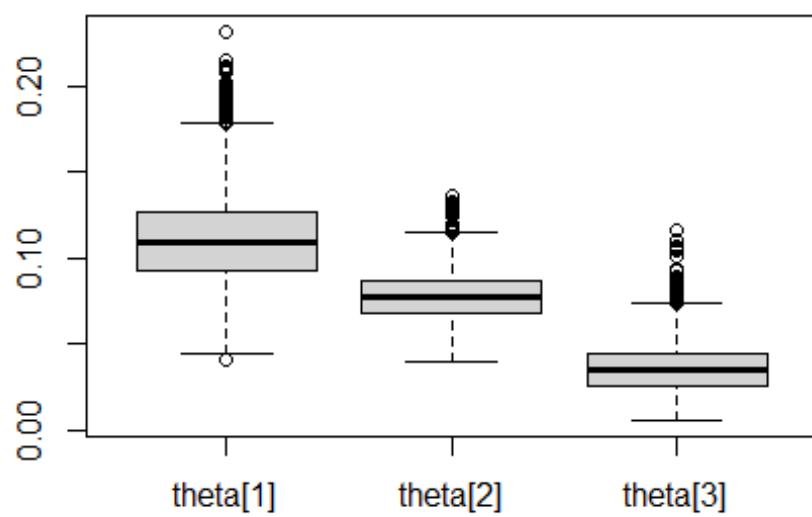
cent



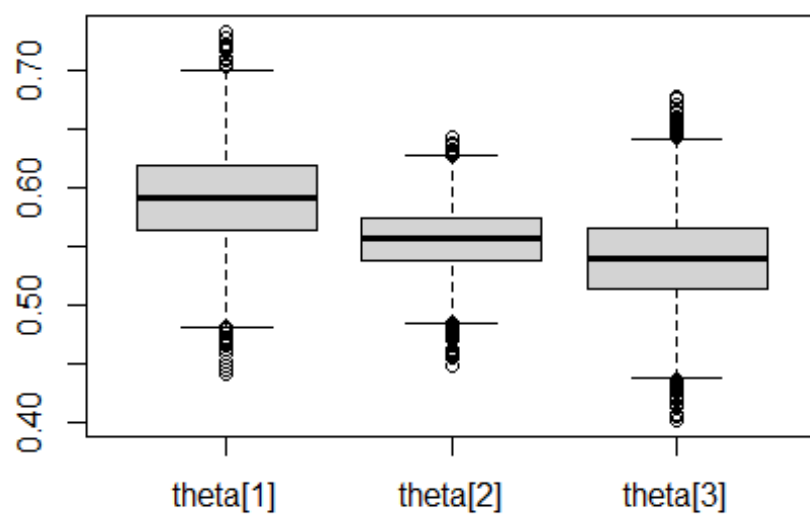
stroke



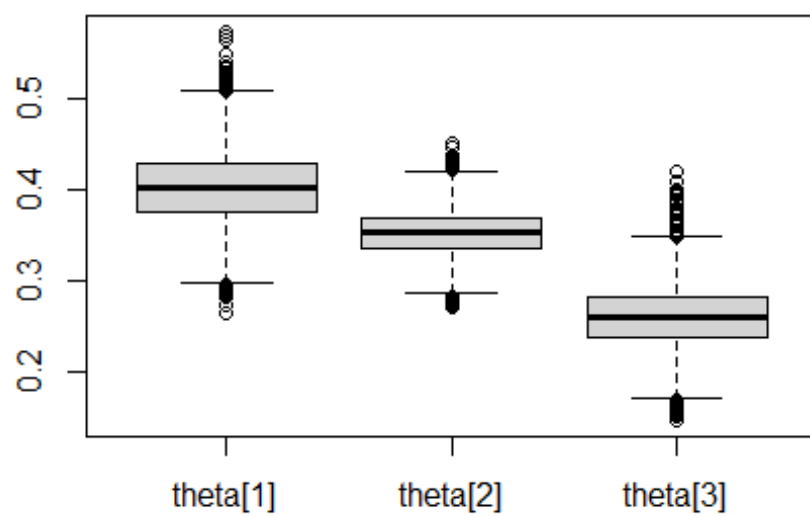
diabetes

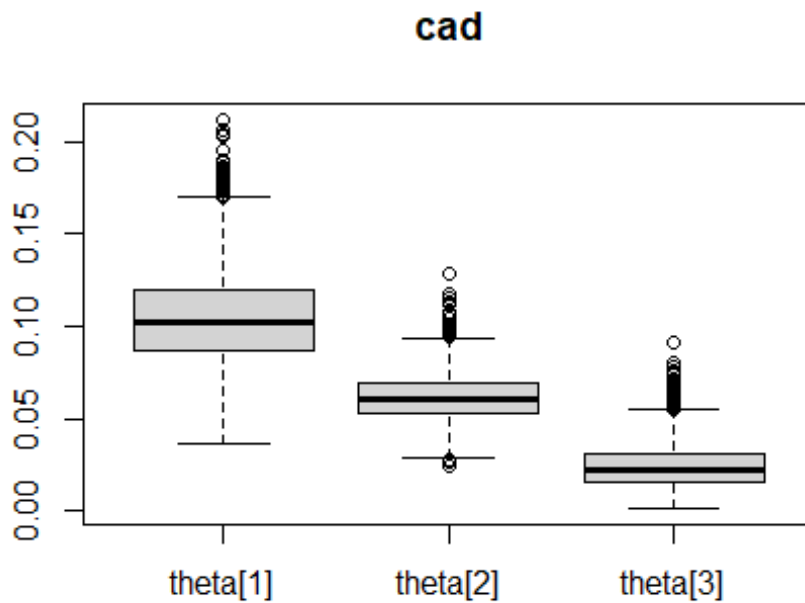


smoking



htn





```
## [1] 760 4

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 2165
##   Unobserved stochastic nodes: 3165
##   Total graph size: 16890
##
## Initializing model

##
## Iterations = 5040:35000
## Thinning interval = 40
## Number of chains = 3
## Sample size per chain = 750
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## a0control  13.75154 0.31210 0.0065797    0.0082709
## a0treat    13.52228 0.27749 0.0058499    0.0067096
## a0diff     -0.22926 0.35713 0.0075290    0.0118181
## abar       12.59972 0.19042 0.0040143    0.0043940
## bbar       -0.23772 0.06200 0.0013071    0.0019848
```

```

## gbar          0.04731 0.07371 0.0015539      0.0024391
## sigma.alpha   5.20517 0.41997 0.0088537      0.0091383
## sigma.beta    0.13023 0.02815 0.0005935      0.0006529
## sigma.gamma   0.12318 0.02855 0.0006018      0.0006920
## beta1         0.98046 0.23309 0.0049139      0.0054293
## beta3         0.51057 0.02787 0.0005875      0.0005872
## beta4        -0.17510 0.01635 0.0003446      0.0003447
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%     97.5%
## a0control    13.12143 13.541214 13.75465 13.962331 14.3427
## a0treat      12.98271 13.326968 13.53406 13.706394 14.0592
## a0diff       -0.94642 -0.465542 -0.22029  0.005953  0.4673
## abar         12.21603 12.469162 12.60300 12.726204 12.9672
## bbar         -0.36049 -0.280690 -0.23709 -0.195281 -0.1178
## gbar         -0.09644 -0.001229  0.04546  0.096080  0.1953
## sigma.alpha   4.43004  4.913157  5.20822  5.478305  6.0415
## sigma.beta    0.08213  0.110239  0.12733  0.147869  0.1911
## sigma.gamma   0.07741  0.102614  0.12010  0.140150  0.1883
## beta1         0.53017  0.823727  0.97977  1.136601  1.4344
## beta3         0.45575  0.491126  0.51131  0.530522  0.5624
## beta4        -0.20742 -0.186026 -0.17470 -0.163984 -0.1448
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 2165
##   Unobserved stochastic nodes: 1645
##   Total graph size: 9344
##
## Initializing model
##
## Iterations = 2535:42470
## Thinning interval = 35
## Number of chains = 4
## Sample size per chain = 1142
##
## 1. Empirical mean and standard deviation for each variable,
##   plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## abar          12.65660 0.19467 0.0028803      0.0031550
## a0control     14.90007 0.58740 0.0086910      0.0239376
## a0treat       15.37202 0.43379 0.0064182      0.0081175
## a0diff         0.47196 0.67761 0.0100258      0.0281537
## treatment1    -0.16192 0.19864 0.0029390      0.0087925
## treatment2    -0.02310 0.08370 0.0012384      0.0024860

```

```
## treatment3    0.11485 0.11459 0.0016954    0.0040557
## treatment4    0.25621 0.24308 0.0035965    0.0104838
## control1      -0.58361 0.16499 0.0024411    0.0072677
## control2      -0.32412 0.06952 0.0010286    0.0020142
## control3      -0.06629 0.09751 0.0014427    0.0032796
## control4       0.19794 0.20528 0.0030372    0.0085805
## sigma.alpha   5.15457 0.43671 0.0064614    0.0064898
## beta1         0.97774 0.23394 0.0034613    0.0038736
## beta2        -0.85927 0.28421 0.0042052    0.0130324
## beta3        -0.30940 0.34113 0.0050472    0.0155268
## beta4         0.06865 0.03084 0.0004563    0.0013760
## beta5         0.03673 0.03682 0.0005448    0.0016657
## beta6         0.51070 0.02714 0.0004015    0.0004016
## beta7        -0.17407 0.01612 0.0002384    0.0002384
```

```
##
```

```
## 2. Quantiles for each variable:
```

```
##
```

```
##           2.5%      25%      50%      75%     97.5%
## abar      12.278173 12.52550 12.65418 12.786888 13.0375
## a0control 13.763516 14.49841 14.90954 15.303852 16.0049
## a0treat   14.504708 15.07996 15.37818 15.666684 16.2115
## a0diff    -0.864040  0.03251  0.46299  0.928380  1.8327
## treatment1 -0.563074 -0.29493 -0.15449 -0.032289  0.2333
## treatment2 -0.188420 -0.07905 -0.02287  0.032081  0.1419
## treatment3 -0.107980  0.03927  0.11506  0.189834  0.3418
## treatment4 -0.207599  0.09032  0.25117  0.418087  0.7333
## control1   -0.899583 -0.69638 -0.58467 -0.472150 -0.2634
## control2   -0.457378 -0.37143 -0.32391 -0.277084 -0.1866
## control3   -0.260806 -0.12861 -0.06499 -0.001601  0.1254
## control4   -0.210919  0.06174  0.20241  0.337633  0.5909
## sigma.alpha 4.341922  4.85429  5.13644  5.444563  6.0359
## beta1       0.518169  0.81942  0.98080  1.133518  1.4424
## beta2      -1.404576 -1.05238 -0.86392 -0.664559 -0.3019
## beta3      -1.003671 -0.53649 -0.29995 -0.083455  0.3684
## beta4       0.008161  0.04768  0.06932  0.089342  0.1278
## beta5      -0.034748  0.01212  0.03586  0.061106  0.1100
## beta6       0.457473  0.49249  0.51079  0.529022  0.5638
## beta7      -0.205645 -0.18495 -0.17436 -0.162772 -0.1426
```