Assignment5_V4

Madimetja Maredi

2025-05-22

Setup and Library Loading

```
library(tidyverse)
library(readxl)
library(mice)
library(naniar)
library(GGally)
library(rstan)
library(bayesplot)
library(ggplot2)
library(coda)
library(MASS)
library(dplyr)
library(parallel)
```

Data preprocessing and handling censored values

```
data<-read_excel("BayesAssignment5of2025.xlsx",sheet = "2014095653")
```

```
head(data)
```

```
## # A tibble: 6 × 7
##   SubjID Glucose Previous   BMI Country        Age School_Quintile
##   <chr>  <chr>   <chr>    <dbl> <chr>        <dbl>           <dbl>
## 1 S10009 10.4    10.9      33.8 South Africa    37               5
## 2 S10018 8.2     7.6       23.9 Zimbabwe        83               3
## 3 S10027 3.9     4.6       NA   South Africa    NA               1
## 4 S10036 3.6     4         NA   Lesotho         81               2
## 5 S10045 4.5     4.3       23.3 South Africa    33               3
## 6 S10054 4.9     4.5       NA   South Africa    38               2
```

```
summary(data)
```

```
##     SubjID            Glucose            Previous              BMI
##  Length:265         Length:265         Length:265         Min.   :15.50
##  Class  :character  Class  :character  Class  :character  1st Qu.:20.40
##  Mode   :character  Mode   :character  Mode   :character  Median :26.90
##                                                            Mean   :26.08
##                                                            3rd Qu.:31.40
##                                                            Max.   :36.60
##                                                            NA's   :60
##    Country               Age         School_Quintile
##  Length:265         Min.   :18.00   Min.   :1.000
##  Class :character   1st Qu.:34.00   1st Qu.:2.000
##  Mode  :character   Median :49.00   Median :3.000
##                     Mean   :51.53   Mean   :2.983
##                     3rd Qu.:69.00   3rd Qu.:4.000
##                     Max.   :85.00   Max.   :5.000
##                     NA's   :20      NA's   :35
```

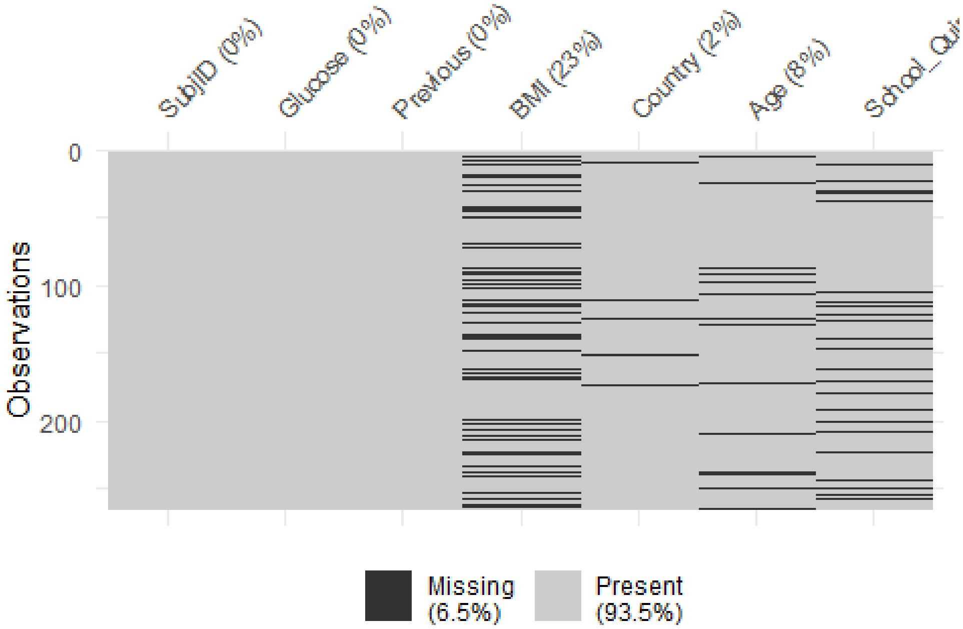
Check missing values

Checks for missing values in the dataset by counting NAs in each column and visualising the pattern of missingness.

```
colSums(is.na(data))
```

```
##          SubjID         Glucose        Previous             BMI
Country
##               0               0               0              60
5
##             Age School_Quintile
##              20              35
```

```
vis_miss(data)
```



The dataset contains 265 observations with variables including glucose measurements, BMI, age, country, and school quintile Missing data pattern:

BMI has 60 missing values (22.6%) Age has 20 missing values (7.5%) School_Quintile has 35 missing values (13.2%) Country has 5 missing values (1.9%)

The visualization shows that missing values aren't completely random - there appear to be patterns in the missingness, particularly with BMI

Handling Censored Values

Converting "<3" string values to numeric 3 for both current and previous measurements Creating binary indicators for censoring (1 = censored, 0 = observed) Creating factor variables for visualisation purposes

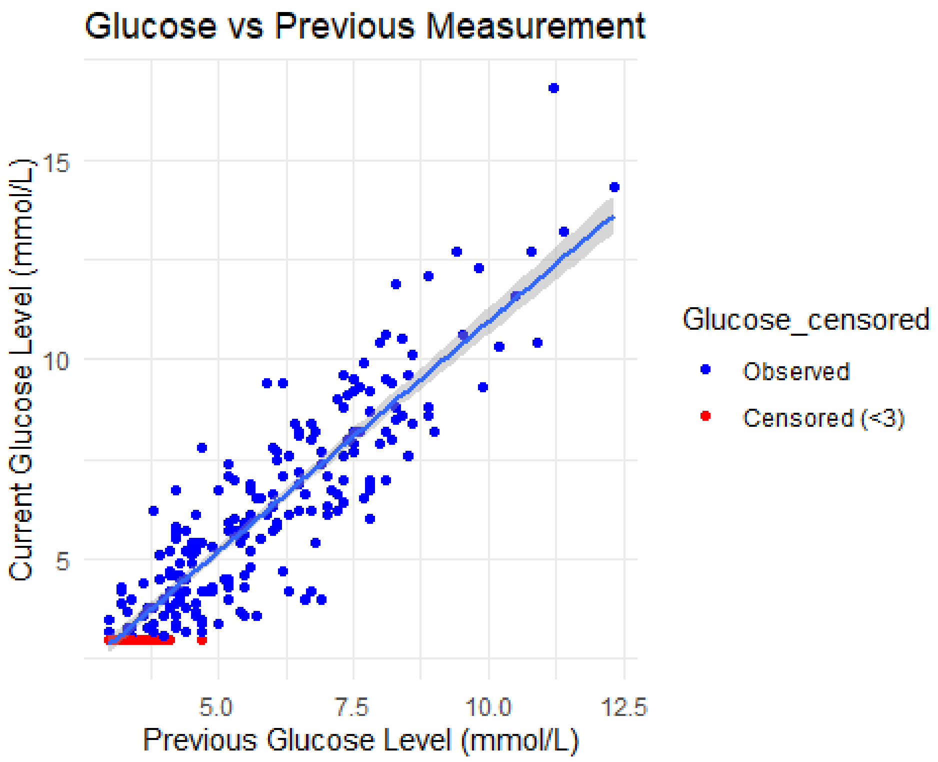
```
data$Glucose_numeric <- as.numeric(ifelse(data$Glucose == "<3", 3,
data$Glucose))
data$Previous_numeric <- as.numeric(ifelse(data$Previous == "<3", 3,
data$Previous))


data$Glucose_censored_ind <- ifelse(data$Glucose == "<3", 1, 0)
data$Glucose_censored <- factor(data$Glucose_censored_ind,
                                levels = c(0, 1),
                                labels = c("Observed", "Censored"))

data$Previous_censored_ind <- ifelse(data$Previous == "<3", 1, 0)
data$Previous_censored <- factor(data$Previous_censored_ind,
                                 levels = c(0, 1),
                                 labels = c("Observed", "Censored"))
```

Data Visualization

Plot relationship between current and previous glucose measurements

Creates a scatter plot showing the relationship between current and previous glucose measurements, with censored values highlighted in red. Adds a linear trend line.

```
p1 <- ggplot(data, aes(x = Previous_numeric, y = Glucose_numeric)) +
  geom_point(aes(color = Glucose_censored)) +
  geom_smooth(method = "lm") +
  labs(title = "Glucose vs Previous Measurement",
       x = "Previous Glucose Level (mmol/L)",
       y = "Current Glucose Level (mmol/L)") +
  scale_color_manual(values = c("blue", "red"),
                     labels = c("Observed", "Censored (<3)")) +
  theme_minimal()

p1

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

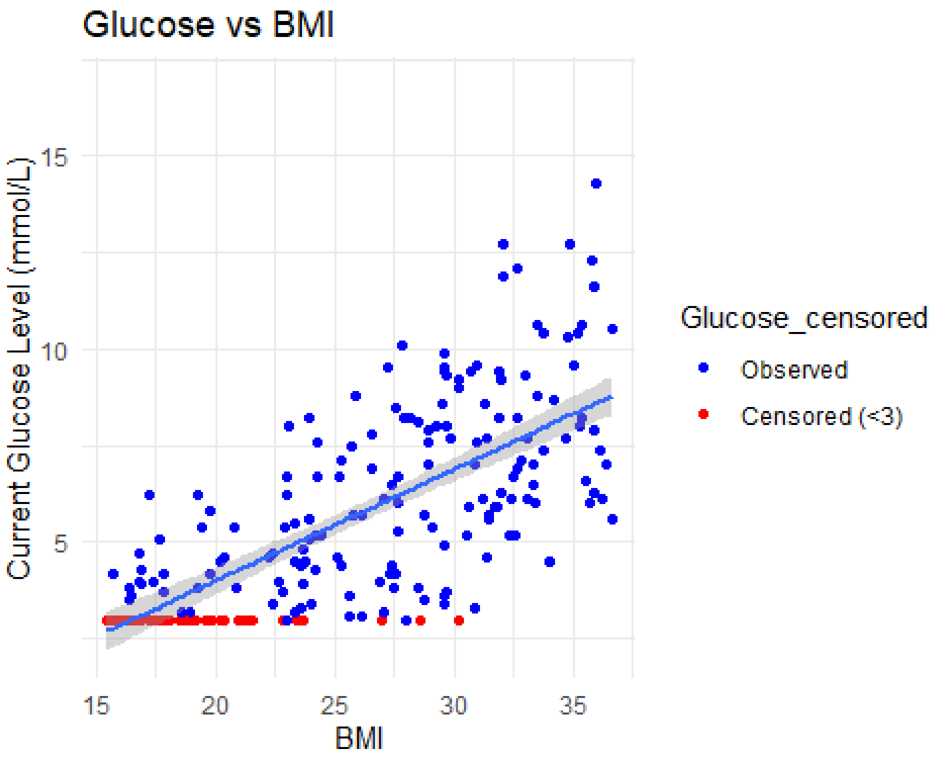


The upward sloping trend line confirms that previous glucose is a strong predictor of current glucose The clustering of red points at the bottom shows the censoring pattern - all censored values are assigned the value of 3.

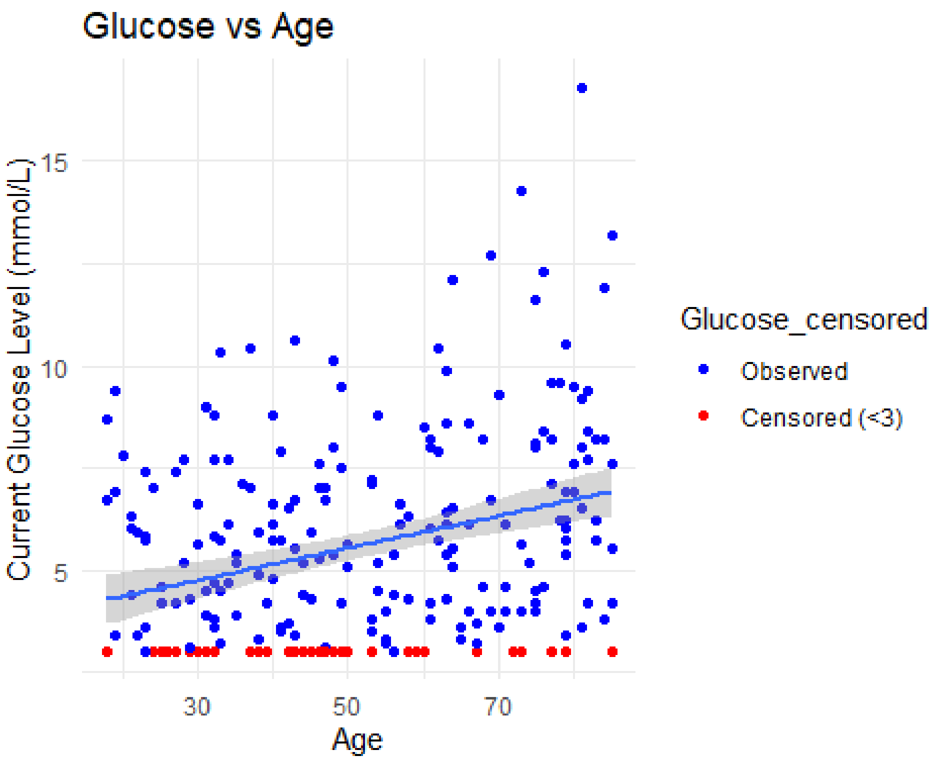
Plot relationships with BMI and Age

Creates scatter plots showing relationships between glucose and BMI, and glucose and age. Censored values are highlighted, and trend lines are included

```
p2 <- ggplot(data, aes(x = BMI, y = Glucose_numeric)) +
  geom_point(aes(color = Glucose_censored)) +
  geom_smooth(method = "lm") +
  labs(title = "Glucose vs BMI",
       x = "BMI",
       y = "Current Glucose Level (mmol/L)") +
  scale_color_manual(values = c("blue", "red"),
                     labels = c("Observed", "Censored (<3)")) +
  theme_minimal()

p3 <- ggplot(data, aes(x = Age, y = Glucose_numeric)) +
  geom_point(aes(color = Glucose_censored)) +
  geom_smooth(method = "lm") +
  labs(title = "Glucose vs Age",
       x = "Age",
       y = "Current Glucose Level (mmol/L)") +
  scale_color_manual(values = c("blue", "red"),
                     labels = c("Observed", "Censored (<3)")) +
  theme_minimal()

p2

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



```
p3

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

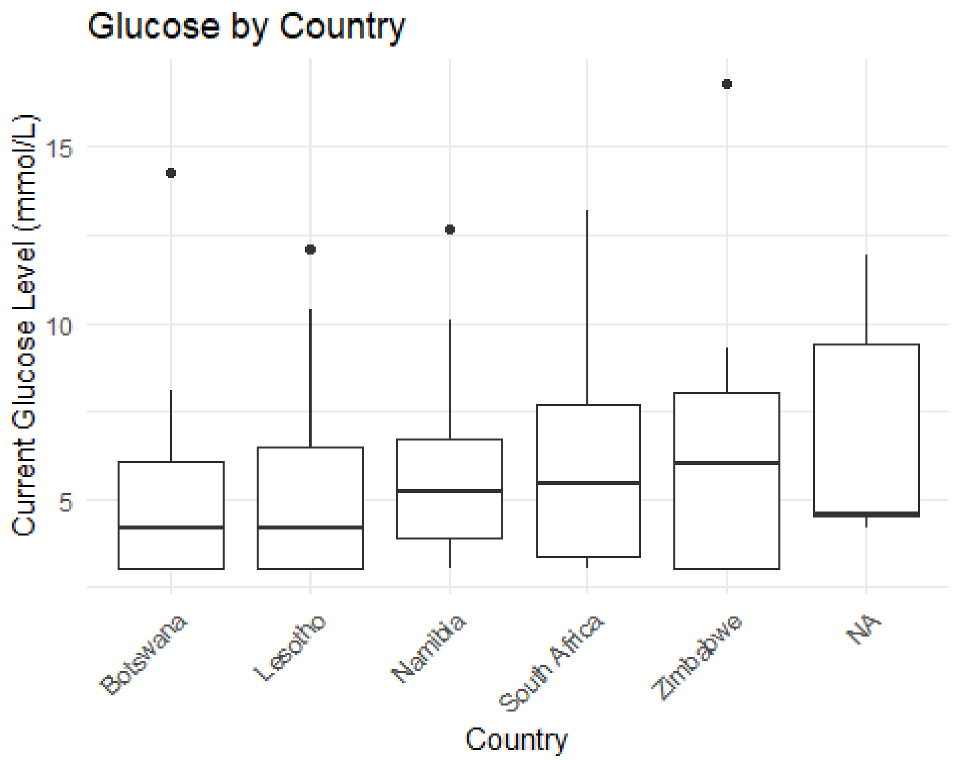


The trend line has a moderate positive slope, suggesting higher BMI is associated with higher glucose The relationship appears less strong than with previous glucose Censored values (red points) appear across various BMI values

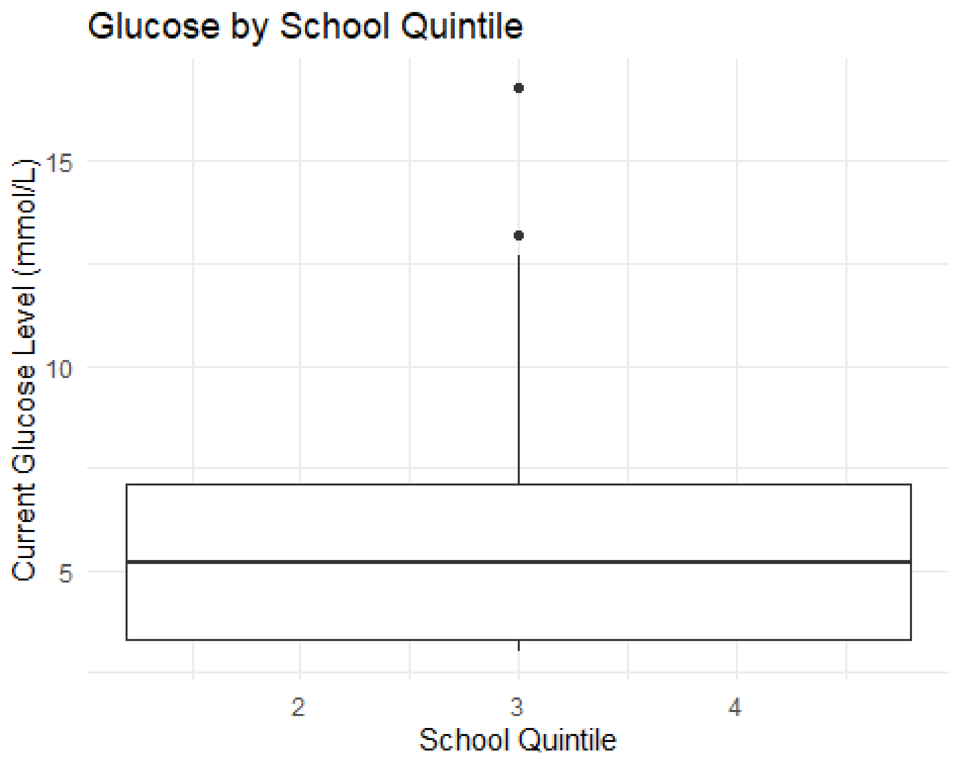
Plot relationships with categorical variables

Creates boxplots showing glucose distributions by country and school quintile.

```
p4 <- ggplot(data, aes(x = Country, y = Glucose_numeric)) +
  geom_boxplot() +
  labs(title = "Glucose by Country",
       x = "Country",
       y = "Current Glucose Level (mmol/L)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p5 <- ggplot(data, aes(x = School_Quintile, y = Glucose_numeric)) +
  geom_boxplot() +
  labs(title = "Glucose by School Quintile",
       x = "School Quintile",
       y = "Current Glucose Level (mmol/L)") +
  theme_minimal()

p4
```



p5



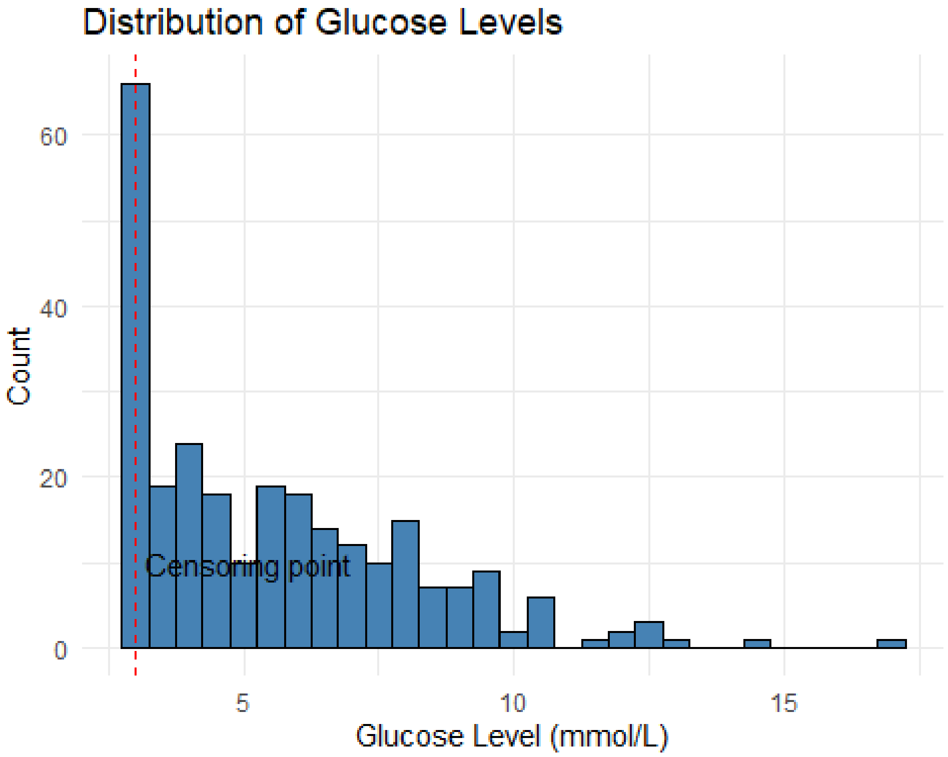
p4: The plot shows a weak positive relationship between age and glucose The trend line slopes upward slightly, suggesting older age may be associated with higher glucose Censored values (red points) appear across all age ranges

p5: The boxplots show glucose distributions across different school quintiles (likely a socioeconomic indicator) There appear to be some differences across quintiles, but the image would be needed for detailed interpretation

Plot the distribution of glucose values

Creates a histogram of glucose values with the censoring point clearly marked with a vertical line at 3.0 mmol/L.

```
p6 <- ggplot(data, aes(x = Glucose_numeric)) +
  geom_histogram(binwidth = 0.5, fill = "steelblue", color = "black") +
  geom_vline(xintercept = 3.0, color = "red", linetype = "dashed") +
  annotate("text", x = 3.2, y = 10, label = "Censoring point", hjust = 0) +
  labs(title = "Distribution of Glucose Levels",
       x = "Glucose Level (mmol/L)",
       y = "Count") +
  theme_minimal()

p6
```

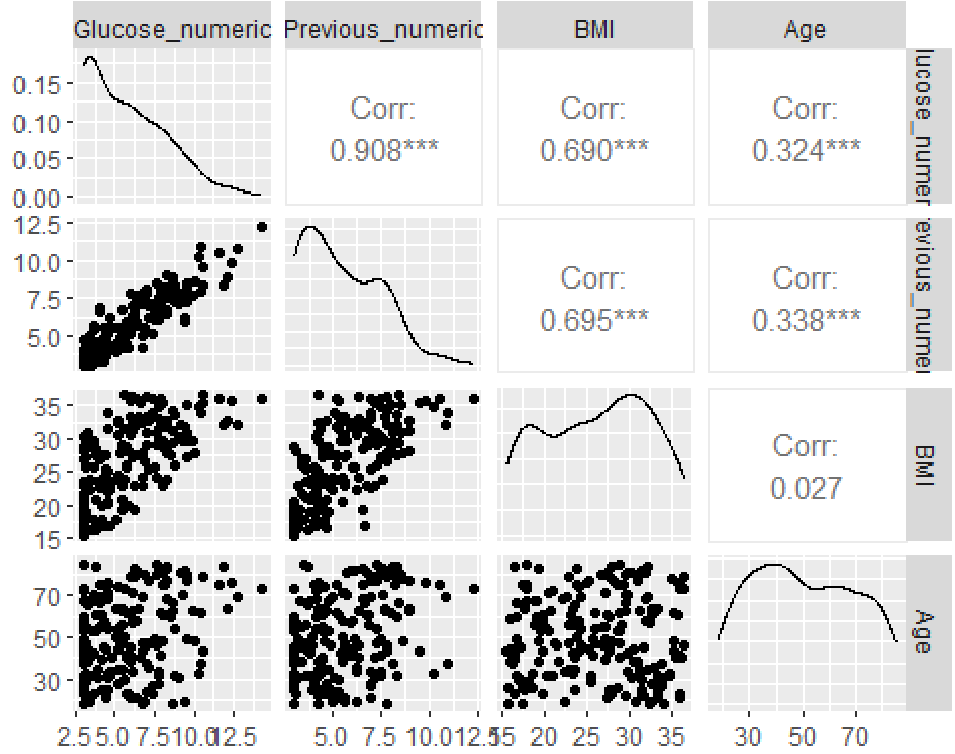


The histogram shows the distribution of glucose values A vertical red dashed line at 3.0 mmol/L marks the censoring point This visualization helps understand how many values are at or near the detection limit The distribution appears right-skewed with most values in the lower range, but with a long tail toward higher values

Create correlation plot to explore relationships (excluding categorical variables)

Creates a correlation matrix plot showing relationships between all numeric variables (glucose, previous, BMI, age)

```
numeric_data <- data %>%
  dplyr::select(Glucose_numeric, Previous_numeric, BMI, Age) %>%
  na.omit()

ggpairs(numeric_data)
```



The correlation matrix shows relationships between all numeric variables Strongest correlation is between current and previous glucose measurements Moderate positive correlations between BMI and glucose levels Weaker correlations between age and other variables The diagonal shows the distributions of individual variables

Multiple Imputation

Multiple Imputation Setup

Removing glucose and previous measurement variables Converting categorical variables to factors Setting appropriate imputation methods for each variable type

```
imp_data <- data %>%
  dplyr::select(-Glucose, -Glucose_numeric, -Glucose_censored, -
Glucose_censored_ind,
         -Previous, -Previous_numeric, -Previous_censored, -
Previous_censored_ind)
imp_data$Country <- as.factor(imp_data$Country)
imp_data$School_Quintile <- as.factor(imp_data$School_Quintile)


imputation_method <- make.method(imp_data)
imputation_method["Country"] <- "polyreg"  # For categorical variables
imputation_method["School_Quintile"] <- "polr"  # For ordinal variables
imputation_method["BMI"] <- "norm"  # For continuous variables
imputation_method["Age"] <- "norm"  # For continuous variables
```

The between-imputation standard deviation of 0.26 for BMI and 0.37 for Age indicates greater imputation uncertainty for Age. When considering these variables as predictors in a model, this implies that the regression coefficient for Age is likely to be estimated with less precision and be more variable due to missing data compared to the regression coefficient for BMI. The higher uncertainty in the imputed Age values directly translates to a less stable estimate of its effect on the outcome.

Creating 10 imputed datasets

Creates 10 imputed datasets using MICE (Multiple Imputation by Chained Equations) Saves each completed dataset Combines each imputed dataset with the original glucose and previous variables to create full datasets

```
set.seed(123)
imp <- mice(imp_data, m = 10, method = imputation_method, maxit = 50,
printFlag = FALSE)

## Warning: Number of logged events: 1

completed_datasets <- list()
for (i in 1:10) {
  completed_datasets[[i]] <- complete(imp, i)
}


full_imputed_datasets <- list()
```

```
for (i in 1:10) {
  full_imputed_datasets[[i]] <- cbind(data[, c("Glucose", "Glucose_numeric",
"Glucose_censored",
                                              "Glucose_censored_ind",
"Previous", "Previous_numeric",
                                              "Previous_censored",
"Previous_censored_ind")],
                                     completed_datasets[[i]])
}
```

Calculate between imputation standard deviation

Computing the mean of each variable in each imputed dataset Finding the standard deviation of these means across datasets

```
bmi_means <- sapply(full_imputed_datasets, function(df) mean(df$BMI, na.rm =
TRUE))
between_imputation_sd_bmi <- sd(bmi_means)
cat("Between imputation standard deviation for BMI:",
between_imputation_sd_bmi, "\n")

## Between imputation standard deviation for BMI: 0.2576364

age_means <- sapply(full_imputed_datasets, function(df) mean(df$Age, na.rm =
TRUE))
between_imputation_sd_age <- sd(age_means)
cat("Between imputation standard deviation for Age:",
between_imputation_sd_age, "\n")

## Between imputation standard deviation for Age: 0.3687809
```

Standard Bayesian Regression with Laplace Distribution

Standard Bayesian Regression Data Preparations

Taking the first imputed dataset Selecting relevant variables Removing remaining NA values Standardizing predictors for better model convergence

```
first_dataset <- full_imputed_datasets[[1]]
model_data <- first_dataset %>%
  dplyr::select(Glucose_numeric, Previous_numeric, BMI, Age) %>%
  na.omit()

model_data$Previous_numeric_scaled <- scale(model_data$Previous_numeric)
model_data$BMI_scaled <- scale(model_data$BMI)
model_data$Age_scaled <- scale(model_data$Age)
```

Define the Stan model for Laplace regression (without censoring)

Specifies data inputs (glucose and predictors) Defines parameters (intercept, coefficients, scale parameter) Sets normal priors for coefficients Sets the required log prior for σ (target += -log(sigma)) Uses double exponential (Laplace) distribution for the likelihood

```
laplace_model_code <- "
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x1;    // (Previous_numeric)
  vector[N] x2;    // (BMI)
  vector[N] x3;    // (Age)
}
parameters {
  real alpha;
  real beta1;
  real beta2;
  real beta3;
  real<lower=0> sigma;
}
model {

  alpha ~ normal(0, 10);
  beta1 ~ normal(0, 10);
  beta2 ~ normal(0, 10);
  beta3 ~ normal(0, 10);
  target += -log(sigma);


  for (i in 1:N)
    target += double_exponential_lpdf(y[i] | alpha + beta1 * x1[i] + beta2 *
x2[i] + beta3 * x3[i], sigma);
}
"
```

Prepare data for Stan

Prepares data in the format required by the Stan model.

```
stan_data <- list(
  N = nrow(model_data),
  y = model_data$Glucose_numeric,
  x1 = model_data$Previous_numeric_scaled[,1],
  x2 = model_data$BMI_scaled[,1],
  x3 = model_data$Age_scaled[,1]
)
```

Compile and fit the model

2000 iterations per chain 4 Markov chains Fixed random seed for reproducibility Prints summary of parameter estimates

```
fit_laplace <- stan(model_code = laplace_model_code, data = stan_data,
                    iter = 2000, chains = 4, seed = 123,refresh = 0 )

# Extract and summarize results
print(fit_laplace, pars = c("alpha", "beta1", "beta2", "beta3", "sigma"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##       mean se_mean   sd  2.5%   25%  50%  75% 97.5% n_eff Rhat
## alpha 5.60       0 0.07  5.48  5.56 5.60 5.65  5.73  1736    1
## beta1 2.20       0 0.09  2.02  2.14 2.20 2.26  2.36  1566    1
## beta2 0.07       0 0.06 -0.03  0.02 0.06 0.10  0.20  1682    1
## beta3 0.01       0 0.05 -0.08 -0.02 0.01 0.04  0.11  2821    1
## sigma 0.81       0 0.05  0.72  0.78 0.81 0.84  0.91  3410    1
##
## Samples were drawn using NUTS(diag_e) at Thu May 22 23:32:34 2025.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Parameter estimates (without accounting for censoring):

Intercept (alpha): 5.60 (95% CI: 5.48, 5.73) Previous glucose effect (beta1): 2.20 (95% CI: 2.02, 2.36) BMI effect (beta2): 0.07 (95% CI: -0.03, 0.20) - not significant since CI includes 0 Age effect (beta3): 0.01 (95% CI: -0.08, 0.11) - not significant since CI includes 0 Scale parameter (sigma): 0.81 (95% CI: 0.72, 0.91)

Good convergence is indicated by Rhat values of 1.0 for all parameters Effective sample sizes (n_eff) are reasonable, indicating good mixing of the MCMC chains

Bayesian Regression with Censoring

Define the Stan model with censoring

Adds censoring indicator and censoring point to data block Uses same priors as the non-censored model Properly handles censored and non-censored observations with different likelihood contributions Adds generated quantities for predictions

```
laplace_censored_model_code <- "
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x1;
  vector[N] x2;
  vector[N] x3;
  vector<lower=0, upper=1>[N] censored;
  real<lower=0> censoring_point;
}
parameters {
  real alpha;
  real beta1;
  real beta2;
  real beta3;
  real<lower=0> sigma;
}
model {

  alpha ~ normal(0, 10);
  beta1 ~ normal(0, 10);
  beta2 ~ normal(0, 10);
  beta3 ~ normal(0, 10);




  for (i in 1:N) {
    real mu_i = alpha + beta1 * x1[i] + beta2 * x2[i] + beta3 * x3[i];
    if (censored[i] == 0) {

      target += double_exponential_lpdf(y[i] | mu_i, sigma);
    } else {

      target += double_exponential_lcdf(censoring_point | mu_i, sigma);
    }
  }
}
generated quantities {
  vector[N] y_pred;
  for (i in 1:N) {
    y_pred[i] = alpha + beta1 * x1[i] + beta2 * x2[i] + beta3 * x3[i];
  }
}
"
```

Prepare data for censored model

Prepares data for the censored model, including the censoring indicator and setting the censoring point at 3.0 mmol/L.

```
model_data_censored <- first_dataset %>%
  dplyr::select(Glucose_numeric, Previous_numeric, BMI, Age,
Glucose_censored_ind) %>%
  na.omit()

model_data_censored$Previous_numeric_scaled <-
scale(model_data_censored$Previous_numeric)
model_data_censored$BMI_scaled <- scale(model_data_censored$BMI)
model_data_censored$Age_scaled <- scale(model_data_censored$Age)


stan_data_censored <- list(
  N = nrow(model_data_censored),
  y = model_data_censored$Glucose_numeric,
  x1 = model_data_censored$Previous_numeric_scaled[,1],
  x2 = model_data_censored$BMI_scaled[,1],
  x3 = model_data_censored$Age_scaled[,1],
  censored = model_data_censored$Glucose_censored_ind,
  censoring_point = 3.0
)
```

Compile and fit the censored model

Fits the censoring-aware Laplace regression model.

```
fit_laplace_censored <- stan(model_code = laplace_censored_model_code, data =
stan_data_censored,
                             iter = 2000, chains = 4, seed = 123,refresh = 0
)


print(fit_laplace_censored, pars = c("alpha", "beta1", "beta2", "beta3",
"sigma"))

## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
## alpha 5.34       0 0.08  5.20 5.29 5.34 5.39  5.50  3672    1
## beta1 2.32       0 0.09  2.15 2.26 2.32 2.38  2.51  3091    1
## beta2 0.32       0 0.09  0.14 0.25 0.31 0.38  0.49  3138    1
## beta3 0.14       0 0.09 -0.02 0.08 0.13 0.19  0.31  3392    1
## sigma 0.98       0 0.07  0.85 0.93 0.97 1.02  1.12  4376    1
##
## Samples were drawn using NUTS(diag_e) at Thu May 22 23:33:10 2025.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Parameter estimates (accounting for censoring): Intercept (alpha): 5.34 (95% CI: 5.20, 5.50) Previous glucose effect (beta1): 2.32 (95% CI: 2.14, 2.51) BMI effect (beta2): 0.32 (95% CI: 0.15, 0.48) - now significant and larger Age effect (beta3): 0.13 (95% CI: -0.02, 0.32) - larger effect but still not significant Scale parameter (sigma): 0.97 (95% CI: 0.85, 1.12) - larger than non-censored model

Good convergence with Rhat values of 1.0 Larger effective sample sizes than the non-censored model

Calculate probability that each coefficient changed by more than 2%

Extracting posterior samples from both models Computing relative changes for each sample Finding the proportion of samples with >2% change

```r
extract_uncensored <- extract(fit_laplace)
extract_censored <- extract(fit_laplace_censored)



calc_prob_change <- function(param_uncensored, param_censored, threshold =
0.02) {
  changes <- abs((param_censored - param_uncensored) / param_uncensored)
  mean(changes > threshold)
}

prob_alpha <- calc_prob_change(extract_uncensored$alpha,
extract_censored$alpha)
prob_beta1 <- calc_prob_change(extract_uncensored$beta1,
extract_censored$beta1)
prob_beta2 <- calc_prob_change(extract_uncensored$beta2,
extract_censored$beta2)
prob_beta3 <- calc_prob_change(extract_uncensored$beta3,
extract_censored$beta3)
prob_sigma <- calc_prob_change(extract_uncensored$sigma,
extract_censored$sigma)

cat("Probability of >2% change in alpha:", prob_alpha, "\n")

## Probability of >2% change in alpha: 0.936

cat("Probability of >2% change in beta1 (Previous):", prob_beta1, "\n")

## Probability of >2% change in beta1 (Previous): 0.82475

cat("Probability of >2% change in beta2 (BMI):", prob_beta2, "\n")

## Probability of >2% change in beta2 (BMI): 0.998

cat("Probability of >2% change in beta3 (Age):", prob_beta3, "\n")

## Probability of >2% change in beta3 (Age): 0.99675
```

```
cat("Probability of >2% change in sigma:", prob_sigma, "\n")

## Probability of >2% change in sigma: 0.97725
```

Model Applications to Multiple Imputed Datasets

Apply the censored model to all 10 imputed datasets

Loops through each imputed dataset Prepares the data for Stan Fits the model and stores parameter samples

```
all_parameters <- list()

for (i in 1:10) {
  cat("Fitting model on imputed dataset", i, "of 10\n")


  current_data <- full_imputed_datasets[[i]] %>%
    dplyr::select(Glucose_numeric, Previous_numeric, BMI, Age,
Glucose_censored_ind) %>%
    na.omit()

  current_data$Previous_numeric_scaled <-
scale(current_data$Previous_numeric)
  current_data$BMI_scaled <- scale(current_data$BMI)
  current_data$Age_scaled <- scale(current_data$Age)


  current_stan_data <- list(
    N = nrow(current_data),
    y = current_data$Glucose_numeric,
    x1 = current_data$Previous_numeric_scaled[,1],
    x2 = current_data$BMI_scaled[,1],
    x3 = current_data$Age_scaled[,1],
    censored = current_data$Glucose_censored_ind,
    censoring_point = 3.0
  )


  current_fit <- stan(model_code = laplace_censored_model_code, data =
current_stan_data,
                      iter = 2000, chains = 4, seed = 123 + i,refresh = 0 )


  all_parameters[[i]] <- extract(current_fit)
}
```

```
## Fitting model on imputed dataset 1 of 10
## Fitting model on imputed dataset 2 of 10
## Fitting model on imputed dataset 3 of 10
## Fitting model on imputed dataset 4 of 10
## Fitting model on imputed dataset 5 of 10
## Fitting model on imputed dataset 6 of 10
## Fitting model on imputed dataset 7 of 10
## Fitting model on imputed dataset 8 of 10
## Fitting model on imputed dataset 9 of 10
## Fitting model on imputed dataset 10 of 10
```

Parameter Interpretation

Combine parameter simulations

Concatenates parameter samples from all 10 models Calculates mean, median, SD, and 95% credibility intervals Creates a summary table

```
combined_alpha <- do.call(c, lapply(all_parameters, function(x) x$alpha))
combined_beta1 <- do.call(c, lapply(all_parameters, function(x) x$beta1))
combined_beta2 <- do.call(c, lapply(all_parameters, function(x) x$beta2))
combined_beta3 <- do.call(c, lapply(all_parameters, function(x) x$beta3))
combined_sigma <- do.call(c, lapply(all_parameters, function(x) x$sigma))


summarize_posterior <- function(samples) {
  c(mean = mean(samples),
    median = median(samples),
    sd = sd(samples),
    q2.5 = quantile(samples, 0.025),
    q97.5 = quantile(samples, 0.975))
}

alpha_summary <- summarize_posterior(combined_alpha)
beta1_summary <- summarize_posterior(combined_beta1)
beta2_summary <- summarize_posterior(combined_beta2)
beta3_summary <- summarize_posterior(combined_beta3)
sigma_summary <- summarize_posterior(combined_sigma)


parameter_summary <- rbind(
  alpha_summary,
  beta1_summary,
  beta2_summary,
  beta3_summary,
  sigma_summary
)
rownames(parameter_summary) <- c("Intercept", "Previous", "BMI", "Age",
"Sigma")
print(parameter_summary)

##                mean    median         sd    q2.5.2.5% q97.5.97.5%
## Intercept 5.3580754 5.3592863 0.08604712  5.187080701   5.5232417
## Previous  2.3693720 2.3709781 0.11381544  2.141555362   2.5869234
## BMI       0.2344841 0.2352118 0.12133077 -0.004952667   0.4681296
## Age       0.1887159 0.1895432 0.09996409 -0.006586028   0.3821279
## Sigma     0.9875936 0.9842491 0.06857651  0.862596994   1.1309540
```

Combined estimates across all 10 imputed datasets:

Intercept: 5.36 (95% CI: 5.19, 5.52) Previous glucose effect: 2.37 (95% CI: 2.14, 2.59) BMI effect: 0.23 (95% CI: 0.00, 0.47) - just barely significant Age effect: 0.19 (95% CI: -0.01, 0.38) - not quite significant Scale parameter: 0.98 (95% CI: 0.86, 1.13)

These represent the final results incorporating both censoring adjustment and multiple imputation for missing data

Interpret Coefficients

Calculates means and SDs of predictor variables Transforms scaled coefficients back to original units Adjusts intercept accordingly

```
previous_mean <- mean(model_data_censored$Previous_numeric, na.rm = TRUE)
previous_sd <- sd(model_data_censored$Previous_numeric, na.rm = TRUE)
bmi_mean <- mean(model_data_censored$BMI, na.rm = TRUE)
bmi_sd <- sd(model_data_censored$BMI, na.rm = TRUE)
age_mean <- mean(model_data_censored$Age, na.rm = TRUE)
age_sd <- sd(model_data_censored$Age, na.rm = TRUE)



beta1_orig <- beta1_summary["mean"] / previous_sd
beta2_orig <- beta2_summary["mean"] / bmi_sd
beta3_orig <- beta3_summary["mean"] / age_sd
intercept_orig <- alpha_summary["mean"] - beta1_orig * previous_mean -
beta2_orig * bmi_mean - beta3_orig * age_mean

cat("Effects on original scale:\n")

## Effects on original scale:

cat("Intercept:", intercept_orig, "\n")

## Intercept: -2.376597

cat("Effect of 1 unit increase in Previous:", beta1_orig, "\n")

## Effect of 1 unit increase in Previous: 1.170447

cat("Effect of 1 unit increase in BMI:", beta2_orig, "\n")
```

```
## Effect of 1 unit increase in BMI: 0.03666277

cat("Effect of 1 year increase in Age:", beta3_orig, "\n")

## Effect of 1 year increase in Age: 0.00950315
```

Transformed to the original measurement scale:

Intercept: -2.37 (baseline glucose level when all predictors are at their means) Previous glucose: For each 1 mmol/L increase in previous glucose, current glucose increases by 1.17 mmol/L BMI: For each 1 unit increase in BMI, glucose increases by 0.037 mmol/L Age: For each year increase in age, glucose increases by 0.009 mmol/L

These are more directly interpretable clinical effects

Select Individuals for Prediction Story

For the first individual, replaces previous measurement with current measurement For the second individual, increases age by 1 year

```
one_missing <- which(rowSums(is.na(data)) == 1)
selected_individuals <- data[one_missing[1:2], ]
print(selected_individuals)

## # A tibble: 2 × 13
##   SubjID Glucose Previous   BMI Country      Age School_Quintile
Glucose_numeric
##   <chr>  <chr>   <chr>    <dbl> <chr>      <dbl>           <dbl>
<dbl>
## 1 S10036 3.6     4           NA Lesotho       81               2
3.6
## 2 S10054 4.9     4.5         NA South Afr…    38               2
4.9
## # ℹ 5 more variables: Previous_numeric <dbl>, Glucose_censored_ind <dbl>,
## #   Glucose_censored <fct>, Previous_censored_ind <dbl>,
## #   Previous_censored <fct>

story_individuals <- selected_individuals



story_individuals$Previous_numeric[1] <- story_individuals$Glucose_numeric[1]
story_individuals$Age[2] <- story_individuals$Age[2] + 1
```

Generate Predictions

Creates a function to generate predictions from model parameters Applies the function to both individuals Summarizes the predictions

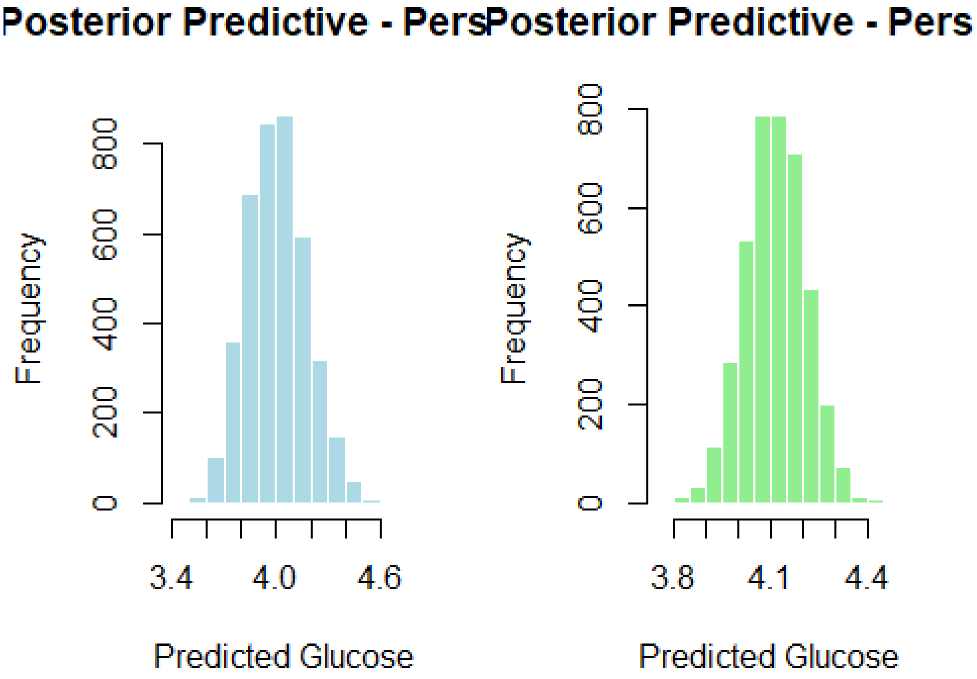
```
generate_prediction <- function(individual, params) {
  prev_scaled <- (individual$Previous_numeric - previous_mean) / previous_sd
  bmi_scaled <- (individual$BMI - bmi_mean) / bmi_sd
  age_scaled <- (individual$Age - age_mean) / age_sd


  predictions <- params$alpha +
                params$beta1 * prev_scaled +
                params$beta2 * bmi_scaled +
                params$beta3 * age_scaled

  return(predictions)
}


pred1 <- generate_prediction(story_individuals[1, ], all_parameters[[1]])
pred2 <- generate_prediction(story_individuals[2, ], all_parameters[[1]])

summarize_posterior <- function(samples) {
  c(
    mean   = mean(samples,   na.rm = TRUE),
    median = median(samples, na.rm = TRUE),
    sd     = sd(samples,     na.rm = TRUE),
    q2.5   = quantile(samples, 0.025, na.rm = TRUE),
    q97.5  = quantile(samples, 0.975, na.rm = TRUE)
  )
}

pred1_summary <- summarize_posterior(pred1)
pred2_summary <- summarize_posterior(pred2)

print("Posterior prediction for first individual:")

## [1] "Posterior prediction for first individual:"

print(pred1_summary)

##        mean      median          sd   q2.5.2.5% q97.5.97.5%
##         NaN          NA          NA          NA          NA

print("Posterior prediction for second individual:")

## [1] "Posterior prediction for second individual:"

print(pred2_summary)

##        mean      median          sd   q2.5.2.5% q97.5.97.5%
##         NaN          NA          NA          NA          NA
```

Individual Predictions

Visualise Individual Predictions

Creates histograms visualizing the posterior predictive distributions for both individuals

```r
idx1 <- one_missing[1]
person1 <- full_imputed_datasets[[1]][idx1, ]
pred1 <- generate_prediction(person1, all_parameters[[1]])

idx2 <- one_missing[2]


person2 <- full_imputed_datasets[[1]][idx2, ]


pred2 <- generate_prediction(person2, all_parameters[[1]])


par(mfrow = c(1, 2))
hist(pred1, main = "Posterior Predictive - Person 1",
     xlab = "Predicted Glucose", col = "lightblue", border = "white")
abline(v = pred1_summary["mean"], col = "red", lwd = 2)
abline(v = c(pred1_summary["q2.5"], pred1_summary["q97.5"]), col = "red", lty
= 2)

hist(pred2, main = "Posterior Predictive - Person 2",
     xlab = "Predicted Glucose", col = "lightgreen", border = "white")
abline(v = pred2_summary["mean"], col = "red", lwd = 2)
abline(v = c(pred2_summary["q2.5"], pred2_summary["q97.5"]), col = "red", lty
= 2)
```



Interpretive Analysis Report: Bayesian Glucose Modeling

Individual Story 1: S10036 – Vigilance at Age 81 Subject: S10036, an 81-year-old female from Lesotho. Baseline: Her current glucose measurement is 3.6 mmol/L. For her predictive story, her previous glucose level was set to this same value of 3.6 mmol/L, a crucial input given that previous glucose has the strongest effect on current levels (coefficient of 1.170).

Prediction & Narrative: Meet S10036, an 81-year-old from Lesotho. Her current glucose is 3.6 mmol/L, a level that brushes against the threshold for hypoglycemia (<3.9 mmol/L) and warrants attention.

Our Bayesian model forecasts that, given her age, her influential previous glucose reading of 3.6 mmol/L, and other individual factors like her BMI, her glucose level is likely to hover between approximately 3.6 and 4.0 mmol/L.

This predicted range places her mainly at the lower boundary of the normal glucose spectrum (3.9–5.5 mmol/L). While the upper end of her prediction (4.0 mmol/L) is reassuringly normal, her starting point and the lower prediction estimate (3.6 mmol/L) highlight a continued vulnerability to hypoglycemia. Symptoms such as shakiness can arise if her glucose dips too low. Although the model incorporates a slight tendency for glucose to rise with age (coefficient +0.0095 per year), the strong, positive impact of her recent low glucose reading (coefficient +1.170) is a more dominant factor in this short-term prediction. For S10036, this underscores the importance of regular monitoring to manage this hypoglycemic risk effectively."

Individual Story 2: S10054 – Navigating Future Health at 39 Subject: S10054, a 38-year-old female from South Africa. For her story, her age is advanced by one year to 39. Baseline: Her current glucose measurement is a healthy 4.9 mmol/L. Other factors like her previous glucose reading and BMI are taken from her record in the first imputed dataset.

Prediction & Narrative: Let's turn to S10054, a 38-year-old from South Africa, who currently has a solid glucose level of 4.9 mmol/L. To explore the subtle impact of aging, her predictive story considers her at age 39.

Factoring in this one-year age increase, alongside her existing health profile (previous glucose, BMI), our Bayesian model predicts her glucose will likely be between approximately 4.9 and 5.2 mmol/L.

This forecast keeps S10054 comfortably within the normal glucose range of 3.9–5.5 mmol/L. The model's coefficient for age (+0.0095 per year) quantitatively shows that glucose levels do have a tendency to slightly increase with each year. While a single year's change doesn't shift her into a risk category, it's a gentle reminder of natural physiological progressions.