Comparing Methods for missing data

Adam Hunt

May 2025

Contents

```r
set.seed(123)
```

```r
R <- read.csv("CHAIN.csv")
```

1 Introduction

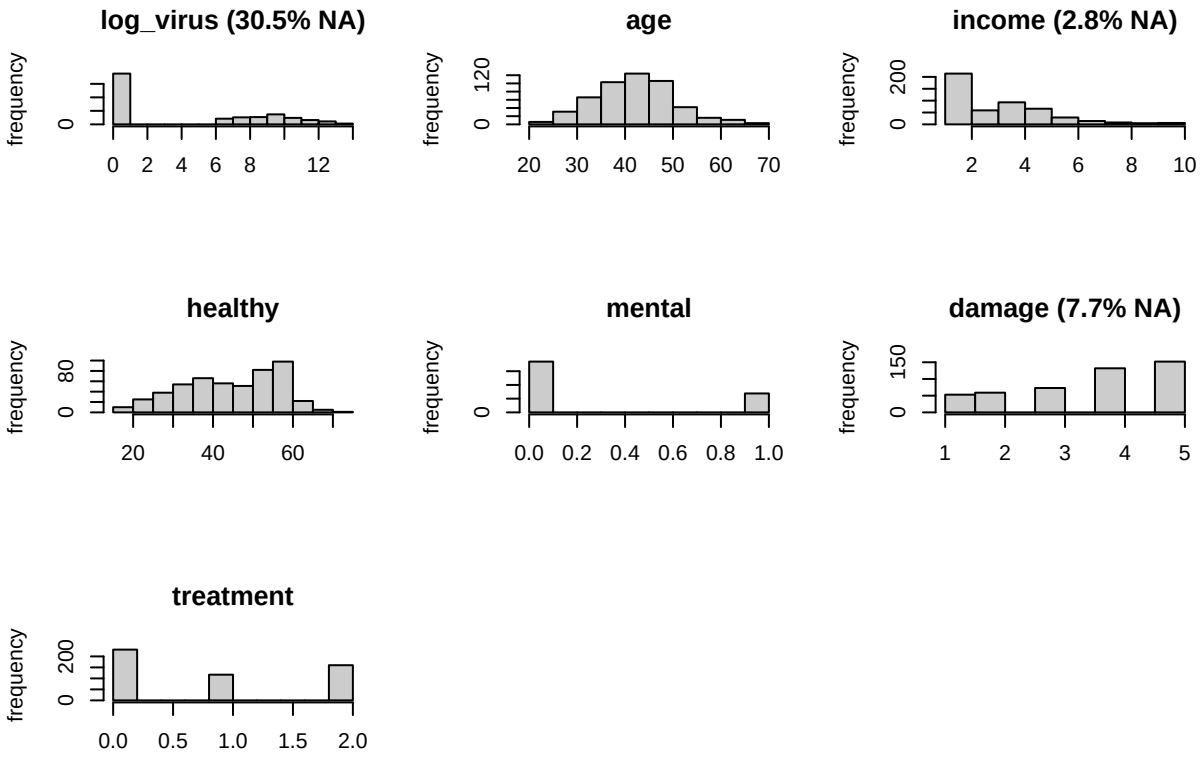
```r
str(R)
```

```
## 'data.frame':    508 obs. of  7 variables:
##  $ log_virus: num  10.7 0 NA NA 0 ...
##  $ age      : int  21 22 23 24 25 25 26 26 26 27 ...
##  $ income   : int  2 5 1 1 10 1 3 1 1 5 ...
##  $ healthy  : num  56.3 41.6 60.7 57.5 51.8 ...
##  $ mental   : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ damage   : int  1 5 2 3 2 5 1 5 5 NA ...
##  $ treatment: int  1 1 0 0 2 1 0 0 1 0 ...
```

2 Exploratory data analysis

2.1 Univariate EDA

```r
Plot = plot_all(R)
```

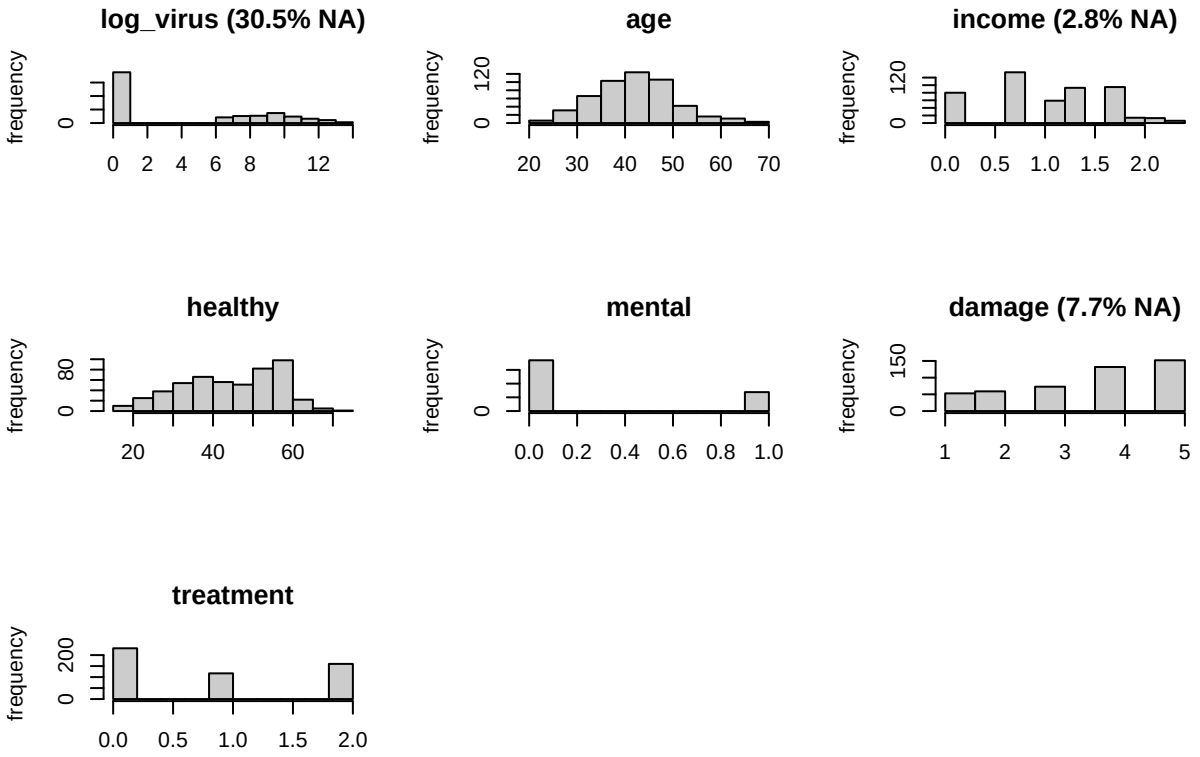



```r
print(Plot)
```

```
## NULL
```

```r
X = R
X$income = log(X$income)
```

```r
Plot = plot_all(X)
```




```r
print(Plot)
```

```
## NULL
```

2.2 Bivariate or pariwise EDA

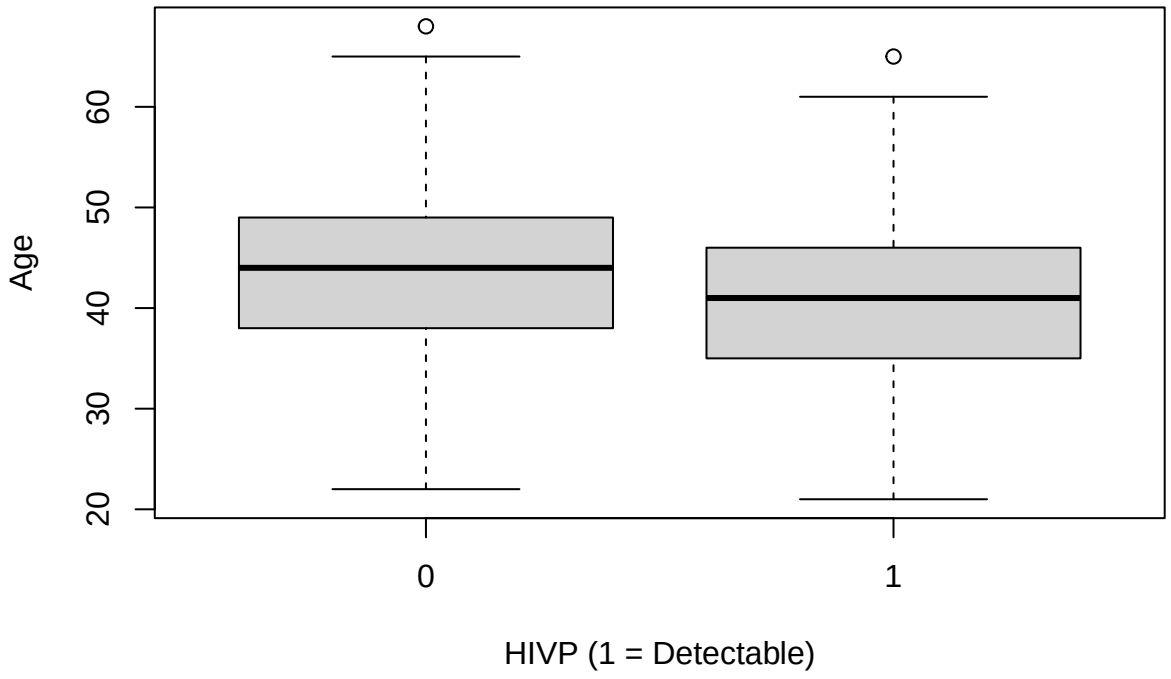
2.2.1 Target Variable Definition and Bivariate Boxplots

```r
# Create binary target variable
HIVP <- as.numeric(R$log_virus > 0)
```

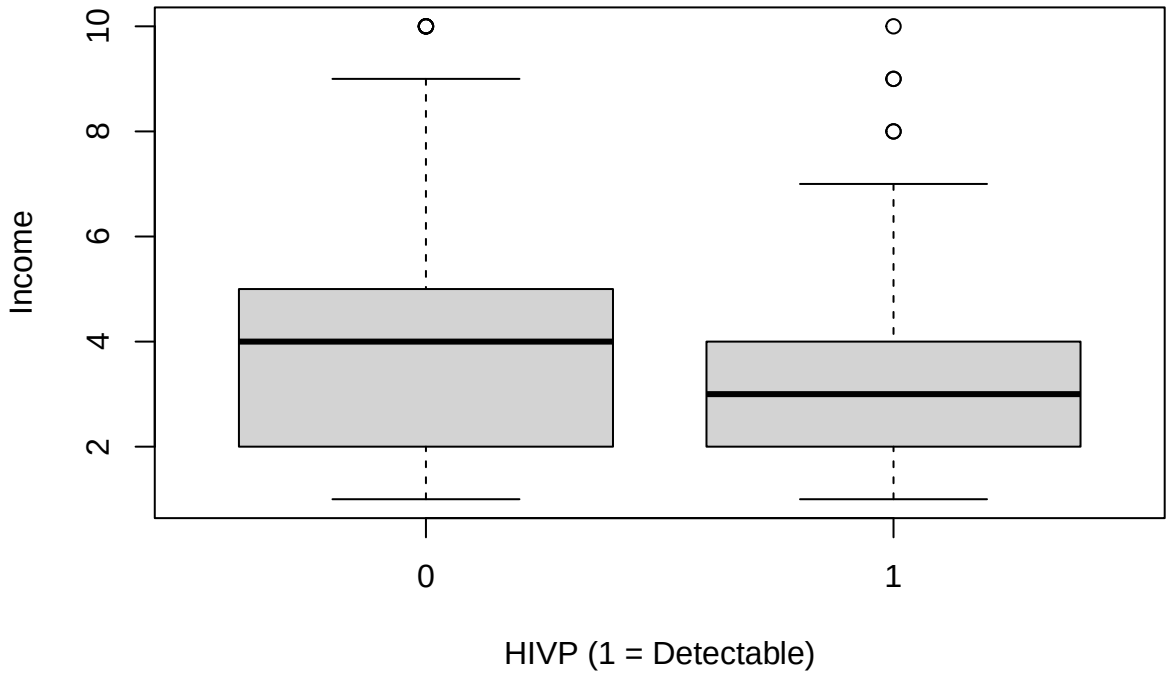
```r
# Add HIVP to dataset
R$HIVP <- HIVP
```

```r
# Remove log_virus
R$log_virus <- NULL
```

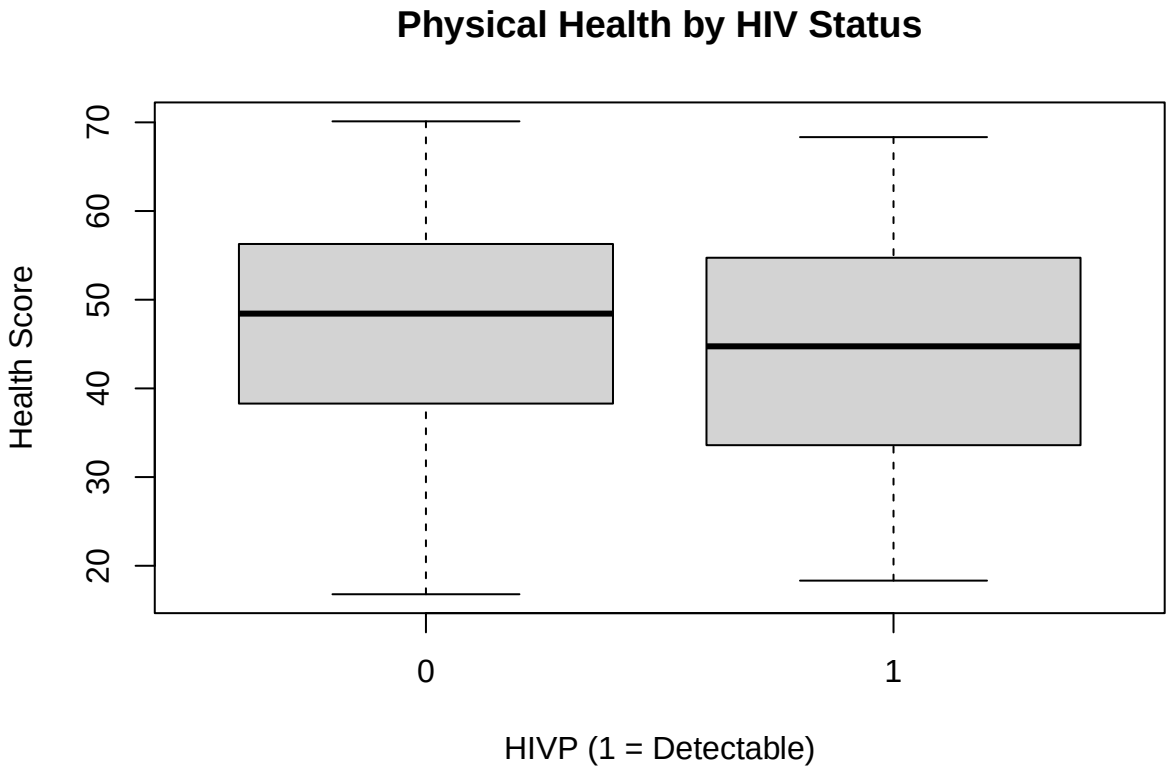
```r
# Age vs HIVP
boxplot(age ~ HIVP, data = R, main = "Age by HIV Status", xlab = "HIVP (1 = Detectable)", ylab = "Age")
```



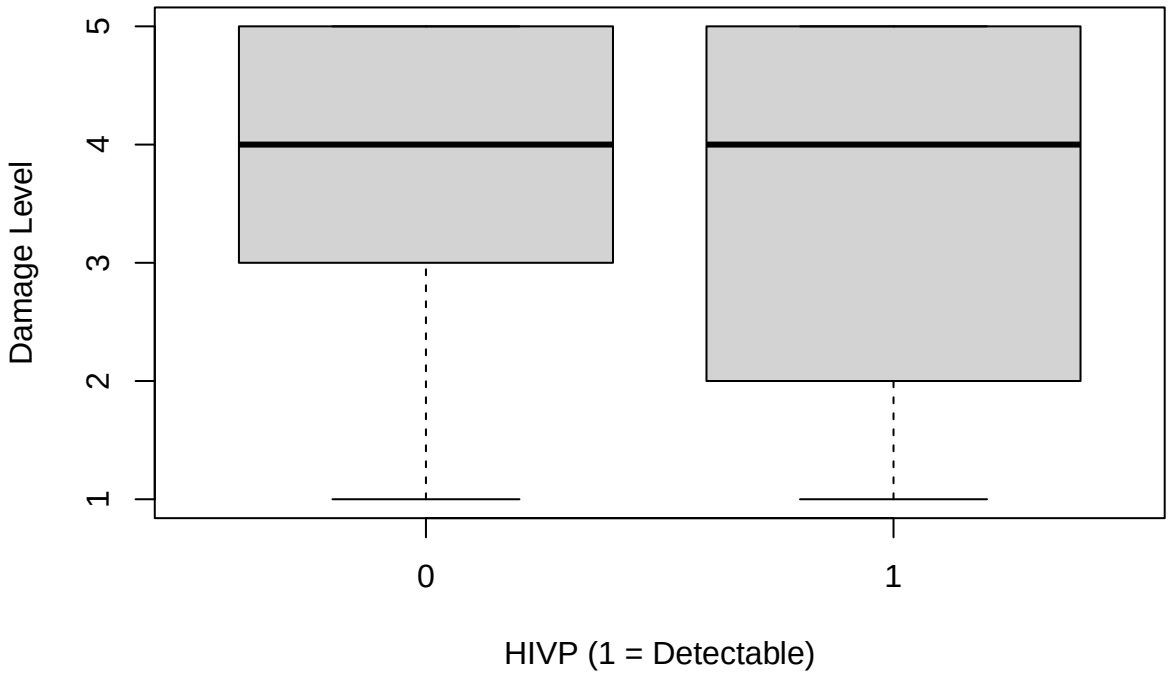

```
# Income vs HIVP
boxplot(income ~ HIVP, data = R, main = "Income by HIV Status", xlab = "HIVP (1 = Detectable)", ylab =
```




```
# Healthy vs HIVP
boxplot(healthy ~ HIVP, data = R, main = "Physical Health by HIV Status", xlab = "HIVP (1 = Detectable)
```

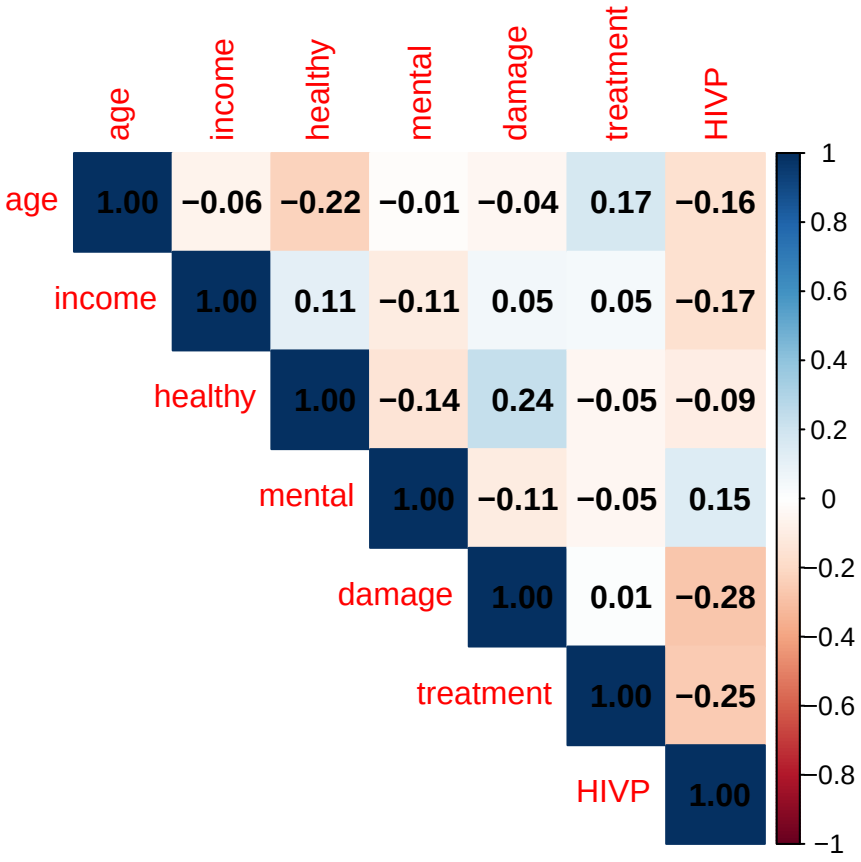



```r
# Damage vs HIVP
boxplot(damage ~ HIVP, data = R, main = "CD4 Damage by HIV Status", xlab = "HIVP (1 = Detectable)", ylab
```



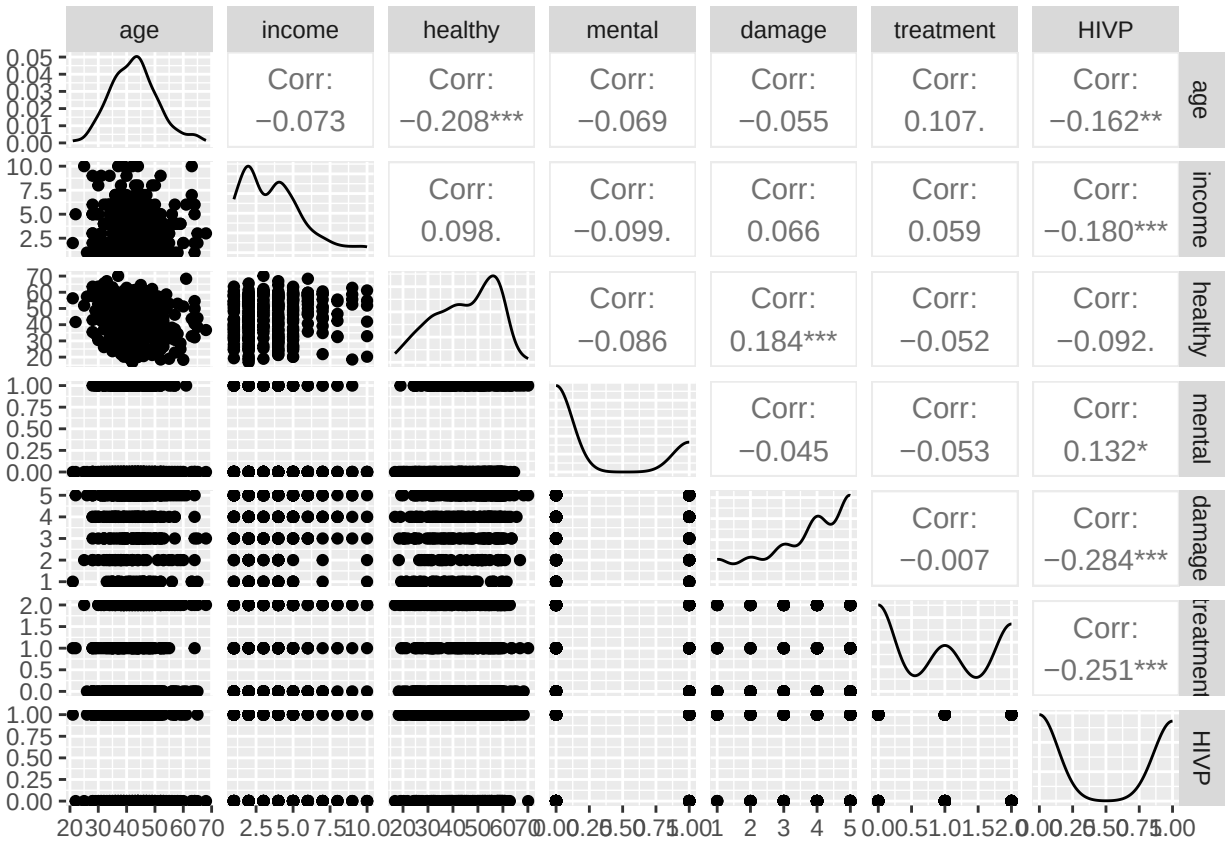
2.2.2 Correlations

```
# Compute the correlation matrix
cor_matrix <- cor(R, use="pairwise.complete.obs")
corrplot(cor_matrix, type="upper", method = "color", addCoef.col = "black")
```



2.2.3 Scatterplots

```
ggpairs(cc(R))
```



2.3 Multivariate EDA

```
Paul <- PCA(cc(R), graph = FALSE)
plot(Paul, choix = "var", axes = c(1, 2))
```

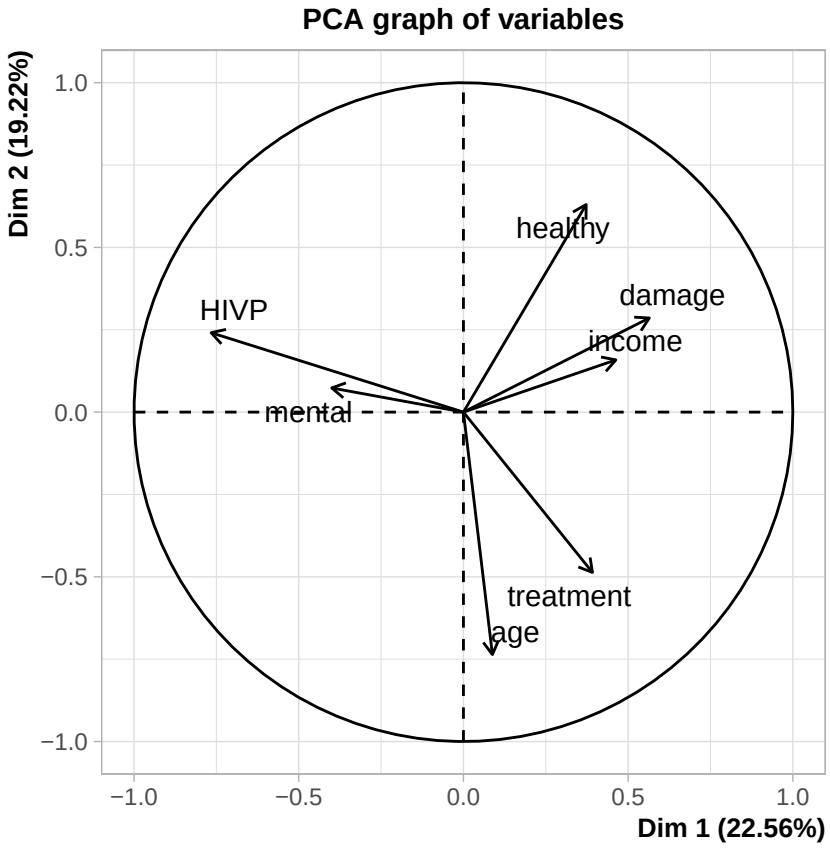


Figure 1: Principal Component Analysis Table

3 The missingness

3.1 The extent and distribution of the missingness

```
sum(complete.cases(R))
```

```
## [1] 335
```

```
miss_summary <- miss_var_summary(R)

# Display the missing value summary as a table
knitr::kable(miss_summary, caption = "Missing Value Summary for the Dataset")
```

Table 1: Missing Value Summary for the Dataset

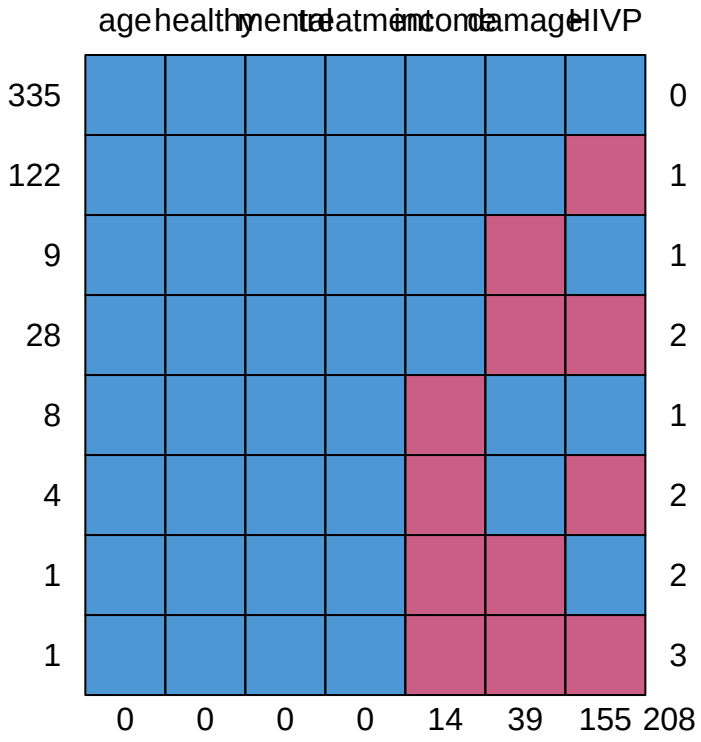
variable	n_miss	pct_miss
HIVP	155	30.5
damage	39	7.68
income	14	2.76
age	0	0
healthy	0	0
mental	0	0
treatment	0	0

```
OVERALLP=sum(is.na(R))/length(is.na(R))*100
knitr::kable(OVERALLP,caption="Overall Missingness")
```

Table 2: Overall Missingness

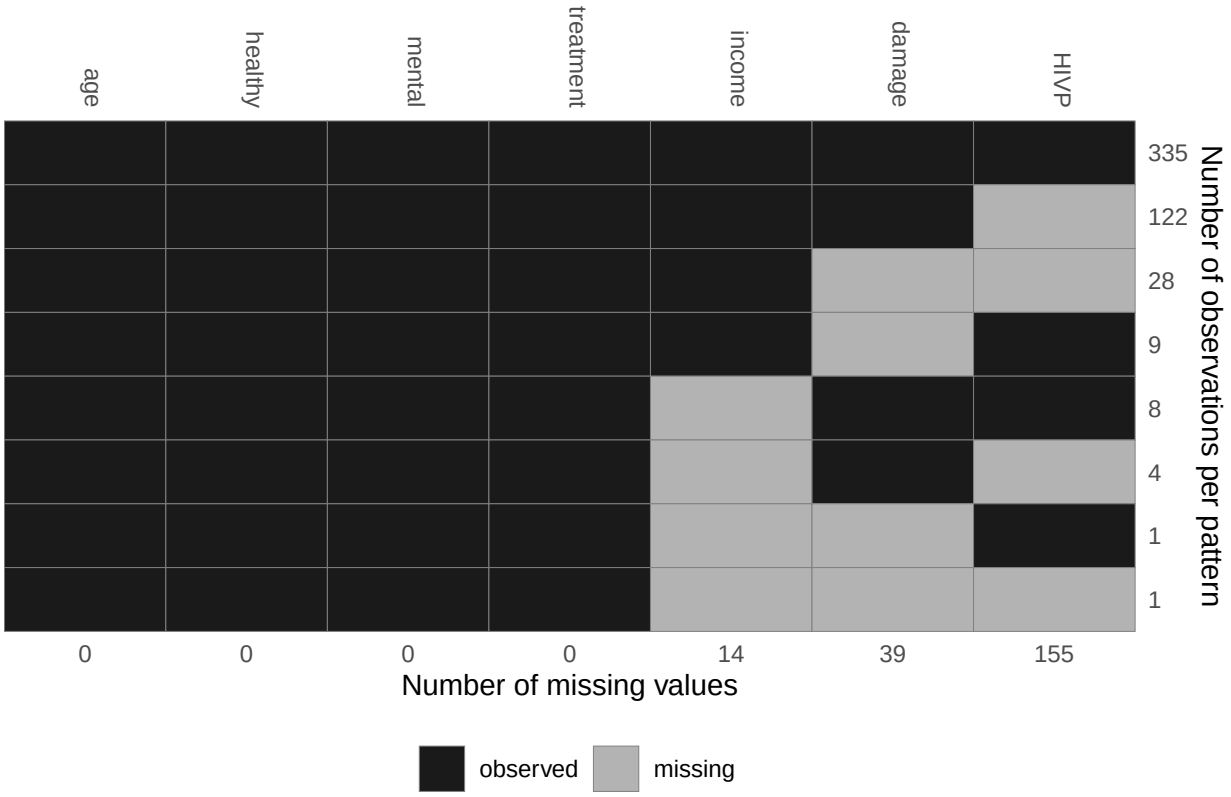
x
5.849269

```
md.pattern(R)
```

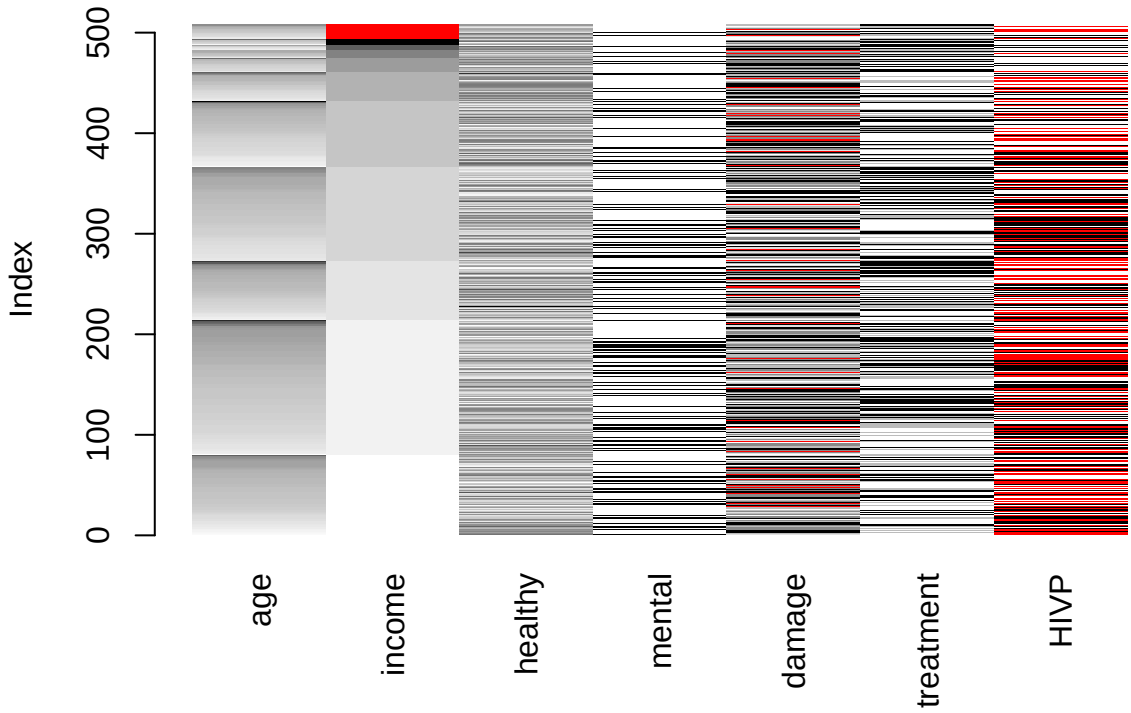



```
##     age healthy mental treatment income damage HIVP
## 335   1       1      1         1      1      1    1   0
## 122   1       1      1         1      1      1    0   1
## 9     1       1      1         1      1      0    1   1
## 28    1       1      1         1      1      0    0   2
## 8     1       1      1         1      0      1    1   1
## 4     1       1      1         1      0      1    0   2
## 1     1       1      1         1      0      0    1   2
## 1     1       1      1         1      0      0    0   3
##       0       0      0         0     14     39  155 208
```

```
md_pattern(R)
```




```
matrixplot(R,sortby="income")
```



3.2 The mechanism of missingness

```r
# MCAR test
mcar_result <- mcar_test(R)
knitr::kable(mcar_result, caption = "MCAR Test Results for the Dataset")
```

Table 3: MCAR Test Results for the Dataset

statistic	df	p.value	missing.patterns
59.32977	37	0.0113477	8

3.2.1 Model the missingness

```r
M <- as.numeric(is.na(R$HIVP))
missing_model <- glm(M ~ age + healthy + mental + treatment + income, data = R, family = binomial)

knitr::kable(summary(missing_model)$coefficients, caption = "Logistic Regression for Missing 'HIVP' Data
```

Table 4: Logistic Regression for Missing 'HIVP' Data

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.0199042	0.7486105	1.3623962	0.1730729
age	-0.0107780	0.0126308	-0.8533101	0.3934874
healthy	-0.0162753	0.0085299	-1.9080193	0.0563887
mental	0.1210276	0.2217475	0.5457903	0.5852101
treatment	-0.1999244	0.1189198	-1.6811701	0.0927299
income	-0.1692775	0.0564400	-2.9992482	0.0027065

```r
M <- as.numeric(is.na(R$HIVP))
missing_model2 <- glm(M ~ age + healthy + mental + treatment + income + damage, data = R, family = binor

knitr::kable(summary(missing_model2)$coefficients, caption = "Logistic Regression for Missing 'HIVP' Da
```

Table 5: Logistic Regression for Missing 'HIVP' Data

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.5697990	0.8336653	1.8830088	0.0596992
age	-0.0094097	0.0135465	-0.6946235	0.4872913
healthy	-0.0151569	0.0094125	-1.6102965	0.1073331
mental	0.0825905	0.2413986	0.3421332	0.7322507
treatment	-0.1072503	0.1287049	-0.8333042	0.4046732
income	-0.2092139	0.0642717	-3.2551454	0.0011333
damage	-0.2251393	0.0814014	-2.7657909	0.0056785

4 Complete case analysis (CCA)

```r
cat("Complete cases:", sum(complete.cases(R)), "\nTotal rows:", nrow(R))
```

```
## Complete cases: 335
## Total rows: 508
```

```r
CC <- na.omit(R)
```

4.1 Saturated Model

```r
# Saturated model
LM_CC <- glm(HIVP ~ age + income + healthy + mental + damage + treatment,
             data = CC, family = binomial)

kable(summary(LM_CC)$coefficients, caption = "Saturated Model using CCA")
```

Table 6: Saturated Model using CCA

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.8317000	1.0841075	5.379264	0.0000001
age	-0.0523040	0.0160161	-3.265716	0.0010919
income	-0.1781393	0.0623693	-2.856201	0.0042874
healthy	-0.0152292	0.0105784	-1.439659	0.1499638
mental	0.4106581	0.2859038	1.436351	0.1509026
damage	-0.5055123	0.1019747	-4.957231	0.0000007
treatment	-0.6355727	0.1452864	-4.374620	0.0000122

```r
# Check multicollinearity
vif_values <- vif(LM_CC)

kable(as.data.frame(vif_values), caption = "VIF for Saturated Model")
```

Table 7: VIF for Saturated Model

	vif_values
age	1.089769
income	1.027111
healthy	1.091156
mental	1.021421
damage	1.052603
treatment	1.039844

4.2 BIC-selected model

```r
# Stepwise model selection using BIC
best_model_CCA <- step(LM_CC, direction = "both", k = log(nrow(CC)), trace = 0)

kable(summary(best_model_CCA)$coefficients, caption = "BIC-selected Model using CCA")
```

Table 8: BIC-selected Model using CCA

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.2212393	0.8832304	5.911525	0.0000000
age	-0.0490734	0.0155519	-3.155454	0.0016025
income	-0.1924142	0.0619273	-3.107096	0.0018893

	Estimate	Std. Error	z value	Pr(>|z|)
damage	-0.5240140	0.1007199	-5.202687	0.0000002
treatment	-0.6228971	0.1436224	-4.337048	0.0000144

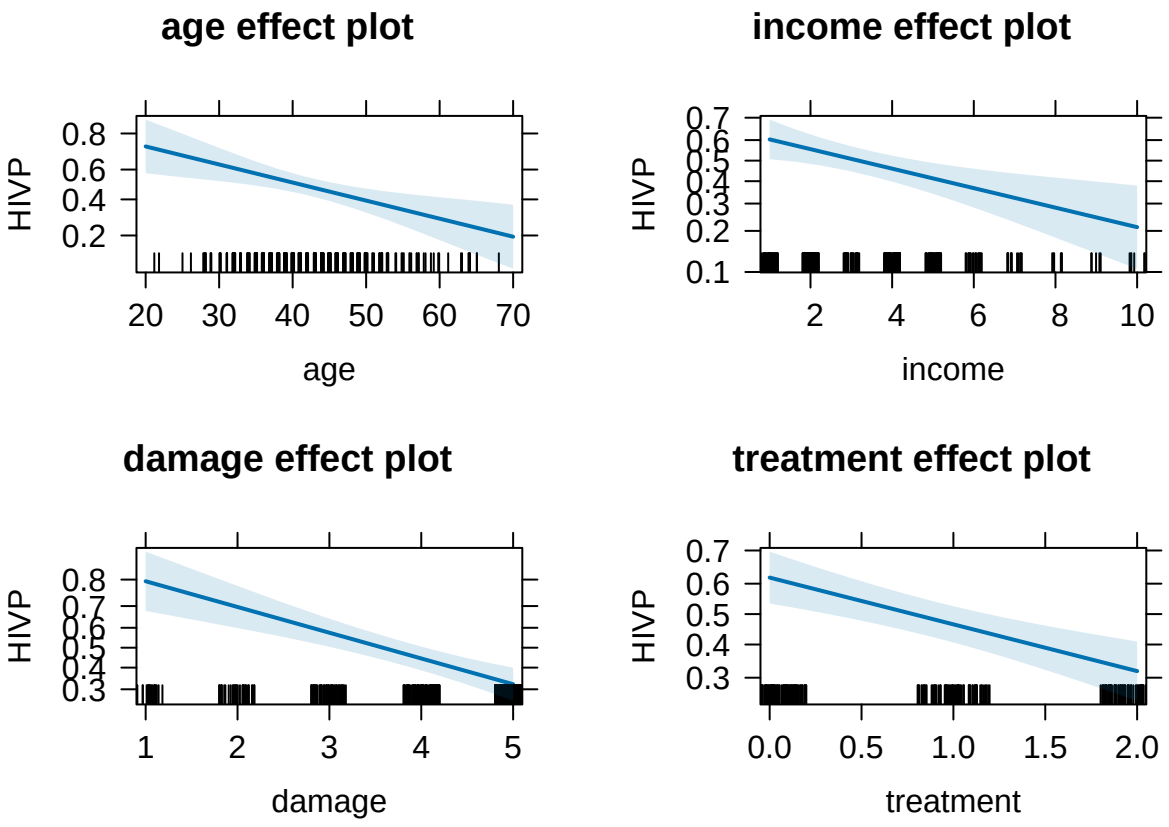
```
# VIF for selected model
vif_selected <- vif(best_model_CCA)

kable(as.data.frame(vif_selected), caption = "VIF for BIC-selected Model")
```

Table 9: VIF for BIC-selected Model

	vif_selected
age	1.034386
income	1.017622
damage	1.041555
treatment	1.027401

4.3 Effect plots

```
# Effect plots
all_effects <- allEffects(best_model_CCA)

plot(all_effects)
```



4.4 Cross-validation

```
# Define misclassification error function
cost_function <- function(r, pi = 0) mean(abs(r - (pi > 0.5)))

# Cross-validation on BIC-selected model
cv_result_bic <- cv.glm(data = CC, glmfit = best_model_CCA,
                        cost = cost_function, K = 10)

# Cross-validation on full model
cv_result_full <- cv.glm(data = CC, glmfit = LM_CC,
                         cost = cost_function, K = 10)

# Display CV errors
cv_errors <- data.frame(
  Model = c("BIC-selected model", "Saturated model"),
  CV_Error = round(c(cv_result_bic$delta[1], cv_result_full$delta[1]), 4),
  Bias_Corrected = round(c(cv_result_bic$delta[2], cv_result_full$delta[2]), 4)
)

knitr::kable(cv_errors, caption = "10-Fold Cross-Validation Errors for CCA Models")
```

Table 10: 10-Fold Cross-Validation Errors for CCA Models

Model	CV_Error	Bias_Corrected
BIC-selected model	0.3015	0.2967
Saturated model	0.2866	0.2842

5 Single Imputation

5.1 Stochastic

```
## Single Imputation (stochastic)
meth        <- make.method(R)
meth["HIVP"]<- ""

MCE <- mice(R, m = 1, method = meth, print = FALSE)   # single SI
COM <- complete(MCE)

# Saturated logistic model
LM_SI      <- glm(HIVP ~ ., data = COM, family = binomial)
SUM_SI_SAT <- summary(LM_SI)$coefficients

# Step-wise BIC model
PM     <- step(LM_SI, k = log(nrow(COM)), trace = 0)
SUM_SI <- summary(PM)$coefficients
```

```
kable(SUM_SI_SAT, caption = "Saturated Model Coefficients using SI")
```

Table 11: Saturated Model Coefficients using SI

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.3140224	1.0301755	5.158366	0.0000002
age	-0.0481773	0.0154527	-3.117739	0.0018224
income	-0.1619040	0.0605420	-2.674242	0.0074898
healthy	-0.0149088	0.0101891	-1.463205	0.1434113
mental	0.5056410	0.2751669	1.837579	0.0661244
damage	-0.4614042	0.0973504	-4.739621	0.0000021
treatment	-0.6201214	0.1405134	-4.413254	0.0000102

```
kable(SUM_SI, caption = "Stepwise BIC Model Selection Coefficients using SI")
```

Table 12: Stepwise BIC Model Selection Coefficients using SI

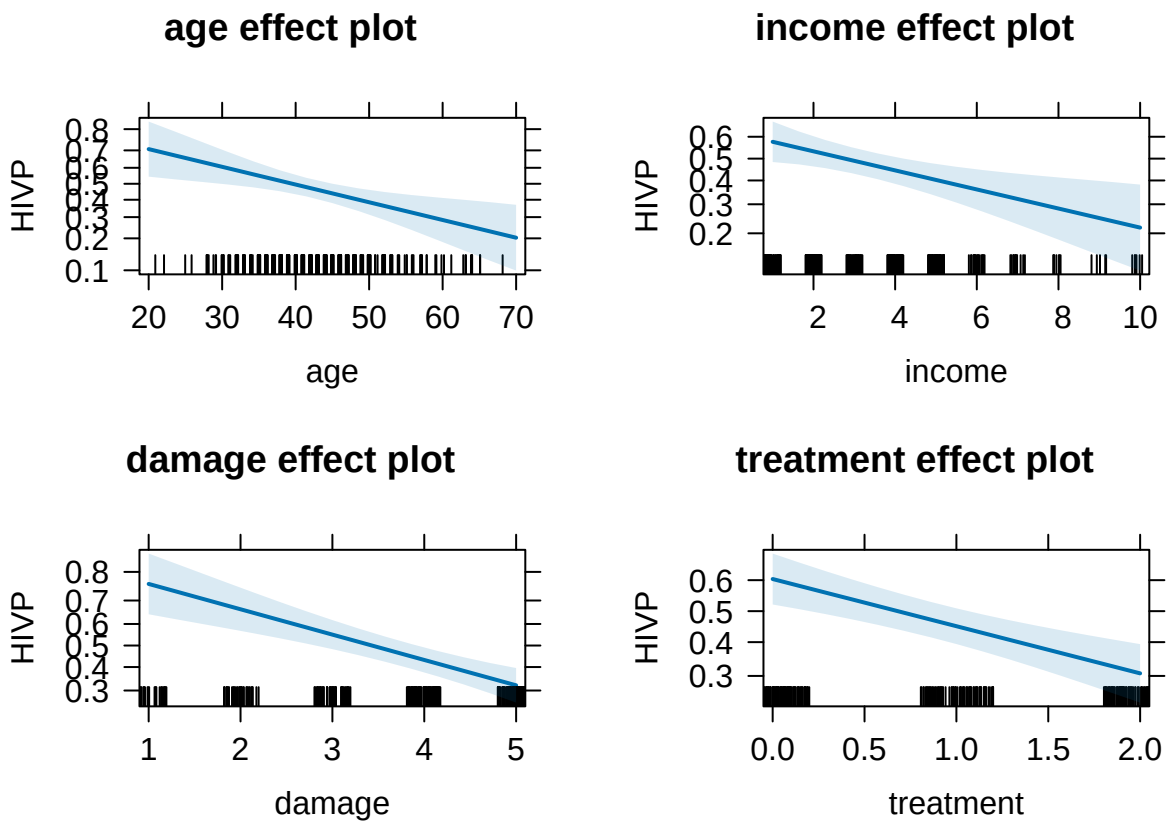
	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	4.7488116	0.8319329	5.708166	0.0000000
age	-0.0449898	0.0149328	-3.012818	0.0025883
income	-0.1768977	0.0600870	-2.944026	0.0032397
damage	-0.4773270	0.0960506	-4.969535	0.0000007
treatment	-0.6170490	0.1390040	-4.439073	0.0000090

```
## Diagnostics stochastic SI
# VIF
vif_si  <- car::vif(PM)
kable(data.frame(Variable = names(vif_si), VIF = vif_si),
      caption = "VIF - BIC model (stochastic SI)")
```

Table 13: VIF – BIC model (stochastic SI)

	Variable	VIF
age	age	1.024080
income	income	1.010727
damage	damage	1.030472
treatment	treatment	1.021313

```
# Effect plots
plot(allEffects(PM))
```



5.2 Deterministic

```
## Single Imputation (deterministic)
meth_det            <- make.method(R)
meth_det["HIVP"]    <- ""
meth_det[meth_det == "norm"] <- "norm.predict"

MCE_det <- mice(R, m = 1, method = meth_det, print = FALSE)   # single SI
COM_det <- complete(MCE_det)

# Saturated logistic model
LM_SI_DET      <- glm(HIVP ~ ., data = COM_det, family = binomial)
SUM_SI_DET_SAT <- summary(LM_SI_DET)$coefficients

# Step-wise BIC model
PM_DET     <- step(LM_SI_DET, k = log(nrow(COM_det)), trace = 0)
SUM_SI_DET <- summary(PM_DET)$coefficients
```

```
kable(SUM_SI_DET_SAT, caption = "Saturated Model Coefficients using deterministic SI")
```

Table 14: Saturated Model Coefficients using deterministic SI

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.4867862	1.0407895	5.271754	0.0000001
age	-0.0493646	0.0155502	-3.174531	0.0015008
income	-0.1771486	0.0607068	-2.918104	0.0035217
healthy	-0.0142021	0.0102417	-1.386698	0.1655339

	Estimate	Std. Error	z value	Pr(>|z|)
mental	0.4819846	0.2768764	1.740793	0.0817198
damage	-0.4843630	0.0985241	-4.916187	0.0000009
treatment	-0.6268934	0.1412404	-4.438487	0.0000091

```
kable(SUM_SI_DET, caption = "Stepwise BIC Model Selection Coefficients using deterministic SI")
```

Table 15: Stepwise BIC Model Selection Coefficients using deterministic SI

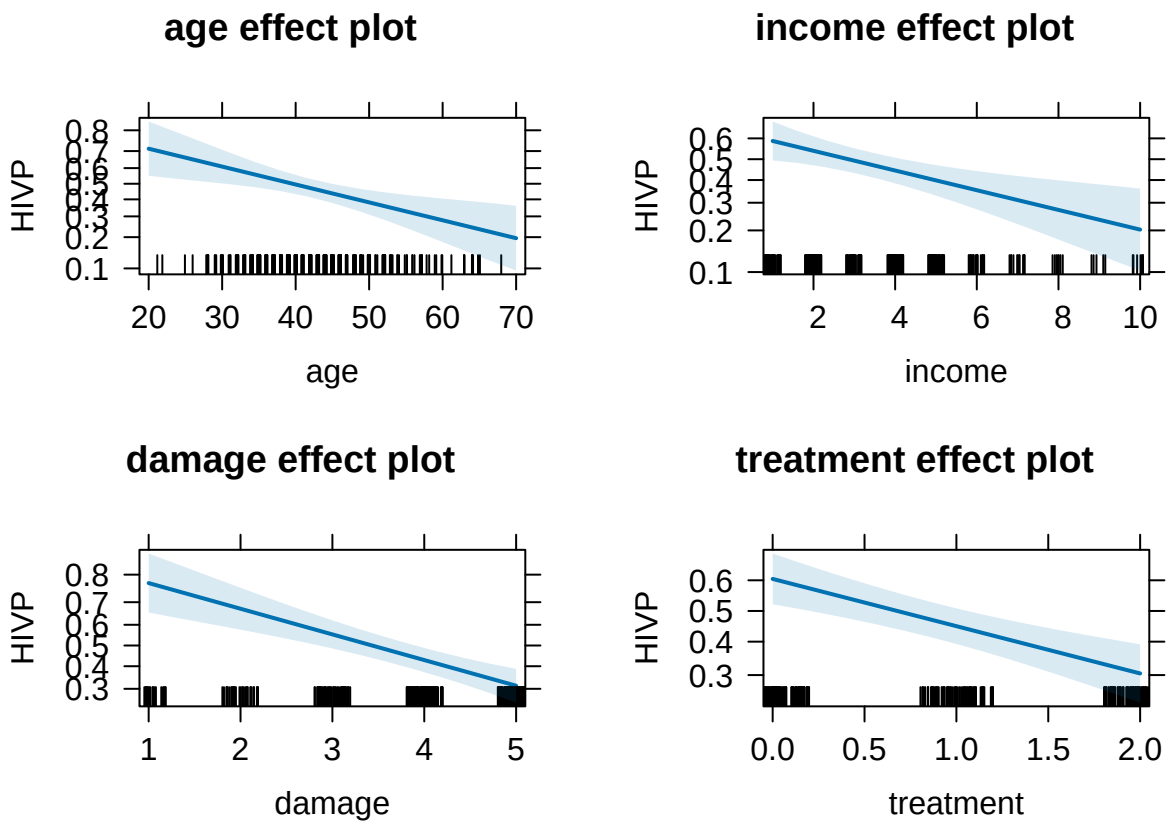
	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	4.9651847	0.8438714	5.883817	0.0000000
age	-0.0464420	0.0150371	-3.088489	0.0020118
income	-0.1917661	0.0601765	-3.186730	0.0014389
damage	-0.5021417	0.0973841	-5.156300	0.0000003
treatment	-0.6248004	0.1398939	-4.466246	0.0000080

```
## Diagnostics deterministic SI
# VIF
vif_det <- car::vif(PM_DET)
kable(data.frame(Variable = names(vif_det), VIF = vif_det),
      caption = "VIF - BIC model (deterministic SI)")
```

Table 16: VIF – BIC model (deterministic SI)

	Variable	VIF
age	age	1.027845
income	income	1.014898
damage	damage	1.033447
treatment	treatment	1.024405

```
# Effect plots
plot(allEffects(PM_DET))
```



6 Bayesian Imputation

6.1 Saturated Bayesian Logistic Model

```
JSAT <- glm_imp(
  HIVP ~ age + income + healthy + mental + damage + treatment,
  data            = R,
  family          = binomial,
  n.iter          = 2000,
  n.adapt         = 1000,
  thin            = 5,
  monitor_params  = c(imps = TRUE)
)
```

```
##
## The variables "HIVP", "mental" were converted to factors.
```

```
GR_df <- as.data.frame(GR_crit(JSAT)[[1]])
knitr::kable(GR_df,
             caption = "Gelman-Rubin diagnostic values - saturated model")
```

Table 17: Gelman–Rubin diagnostic values – saturated model

	Point est.	Upper C.I.
(Intercept)	1.0050265	1.0107354
age	1.0023966	1.0121925
income	0.9991434	1.0005026

	Point est.	Upper C.I.
healthy	1.0012779	1.0023697
mental1	0.9989307	0.9993094
damage	1.0016809	1.0100134
treatment	1.0019985	1.0090122

```
knitr::kable(summary(JSAT)[[6]]$HIVP$regcoef,
             caption = "Regression coefficients - saturated model")
```

Table 18: Regression coefficients – saturated model

	Mean	SD	2.5%	97.5%	tail-prob.	GR-crit	MCE/SD
(Intercept)	5.5519038	1.0519160	3.6477876	7.7070146	0.0000000	1.0121928	0.0288675
age	-0.0509254	0.0157711	-0.0834671	-0.0206223	0.0016667	1.0164780	0.0288675
income	-0.1680799	0.0612438	-0.2872711	-0.0534958	0.0083333	1.0004215	0.0288675
healthy	-0.0137520	0.0103767	-0.0341208	0.0060350	0.1850000	1.0017681	0.0288675
mental1	0.5001854	0.2761671	-0.0302877	1.0395611	0.0733333	0.9999222	0.0288675
damage	-0.5000467	0.1039971	-0.7038929	-0.2922658	0.0000000	1.0032804	0.0288675
treatment	-0.6350735	0.1405422	-0.9117774	-0.3666701	0.0000000	1.0100514	0.0288675

6.2 Reduced Model

```
J1 <- glm_imp(
  HIVP ~ age + income + damage + treatment,
  data            = R,
  auxvars         = ~ healthy + mental,    # auxiliary
  family          = binomial,
  n.iter          = 2000,
  n.adapt         = 1000,
  thin            = 5,
  monitor_params  = c(imps = TRUE)
)
```

```
##
## The variables "HIVP", "mental" were converted to factors.
```

```
GR_df_J1 <- as.data.frame(GR_crit(J1)[[1]])
knitr::kable(GR_df_J1,
             caption = "Gelman-Rubin diagnostic values - reduced model")
```

Table 19: Gelman-Rubin diagnostic values - reduced model

	Point est.	Upper C.I.
(Intercept)	0.9998255	1.003619
age	0.9996416	1.002550
income	0.9998993	1.002250
damage	1.0037296	1.009195
treatment	1.0024499	1.002513

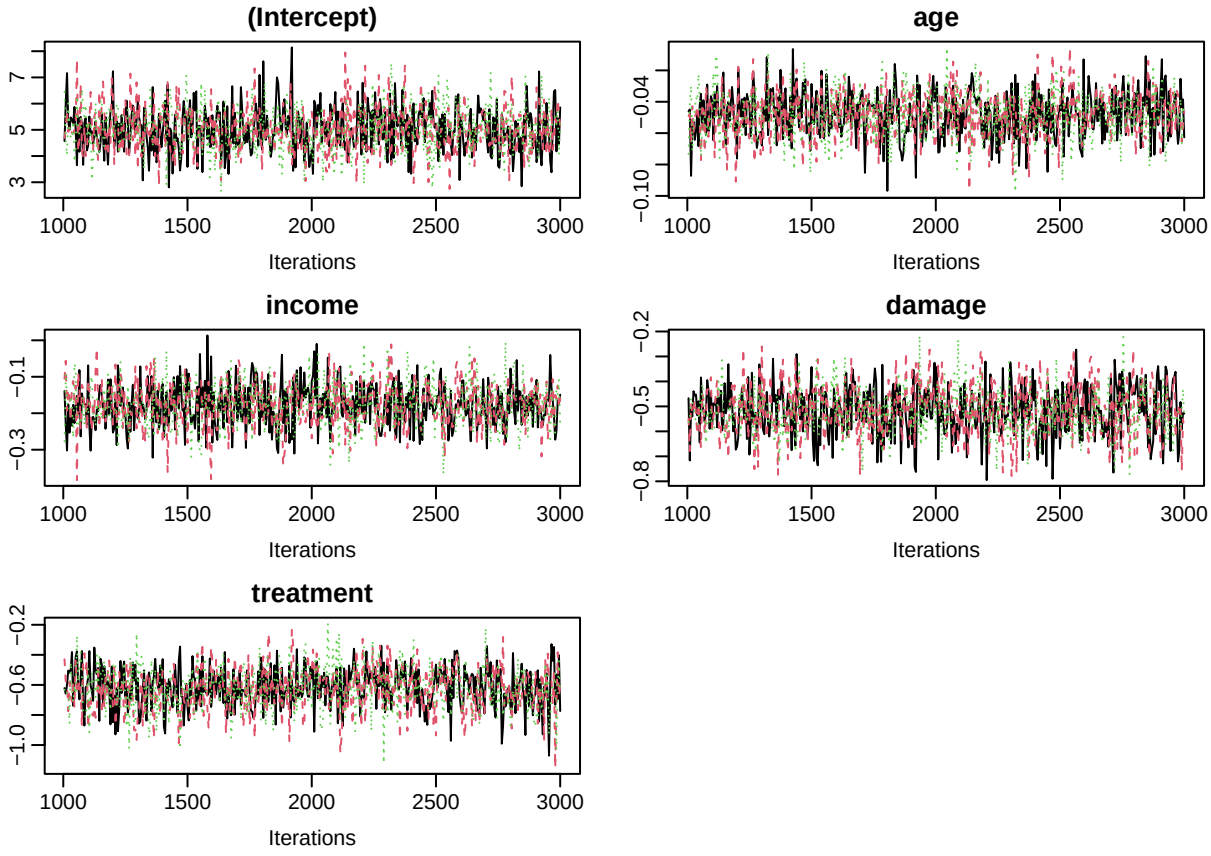
```
knitr::kable(summary(J1)[[6]]$HIVP$regcoef,
             caption = "Regression coefficients - reduced model")
```

Table 20: Regression coefficients – reduced model

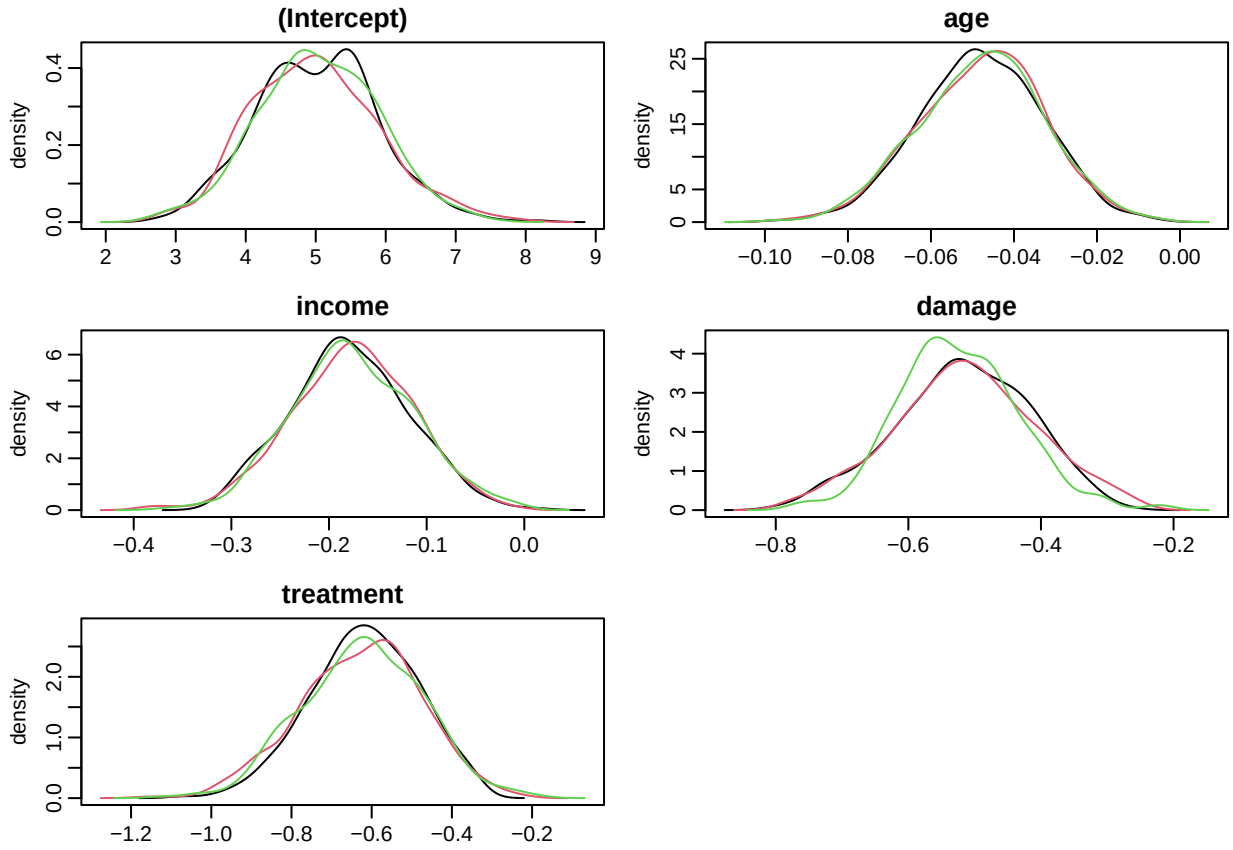
	Mean	SD	2.5%	97.5%	tail-prob.	GR-crit	MCE/SD
(Intercept)	5.0193397	0.8739617	3.3969652	6.7967141	0.0000000	1.0011211	0.0288675
age	-0.0479345	0.0149779	-0.0782206	-0.0202510	0.0000000	0.9990898	0.0288675
income	-0.1768333	0.0613763	-0.2942233	-0.0560158	0.0016667	1.0006232	0.0288675
damage	-0.5183720	0.0993337	-0.7164109	-0.3220768	0.0000000	1.0109484	0.0288675
treatment	-0.6230546	0.1446939	-0.9096149	-0.3515081	0.0000000	1.0049185	0.0288675

6.3 Convergence Visualisations

```
JointAI::traceplot(J1)
```




```
JointAI::densplot(J1)
```



6.4 Model selection via DIC

```
get_DIC <- if (exists("DIC", where = asNamespace("JointAI"), inherits = FALSE)) {
  JointAI:::DIC
} else {
  function(obj, n.iter = 2000) {
    ds <- rjags::dic.samples(obj$model, n.iter, type = "pD")
    mean(ds$deviance + ds$penalty)
  }
}

dic_tab <- data.frame(
  Model = c("Saturated", "Reduced"),
  DIC   = c(get_DIC(JSAT), get_DIC(J1))
)

knitr::kable(dic_tab,
             caption = "Deviance Information Criterion")
```

Table 21: Deviance Information Criterion

Model	DIC
Saturated	3.106351
Reduced	3.107453

7 Multiple Imputation

7.1 Predictor Matrix

```
M <- mice(R, m = 10, maxit = 10, print = FALSE)
pred_matrix <- quickpred(R, mincor = 0.1)

# Convert to a data frame
pred_df <- as.data.frame(pred_matrix)

knitr::kable(pred_df, caption = "Predictor Matrix (mincor = 0.1)")
```

Table 22: Predictor Matrix (mincor = 0.1)

	age	income	healthy	mental	damage	treatment	HIVP
age	0	0	0	0	0	0	0
income	0	0	1	1	0	0	1
healthy	0	0	0	0	0	0	0
mental	0	0	0	0	0	0	0
damage	0	0	1	1	0	0	1
treatment	0	0	0	0	0	0	0
HIVP	1	1	0	1	1	1	0

7.2 Pooled Full Model

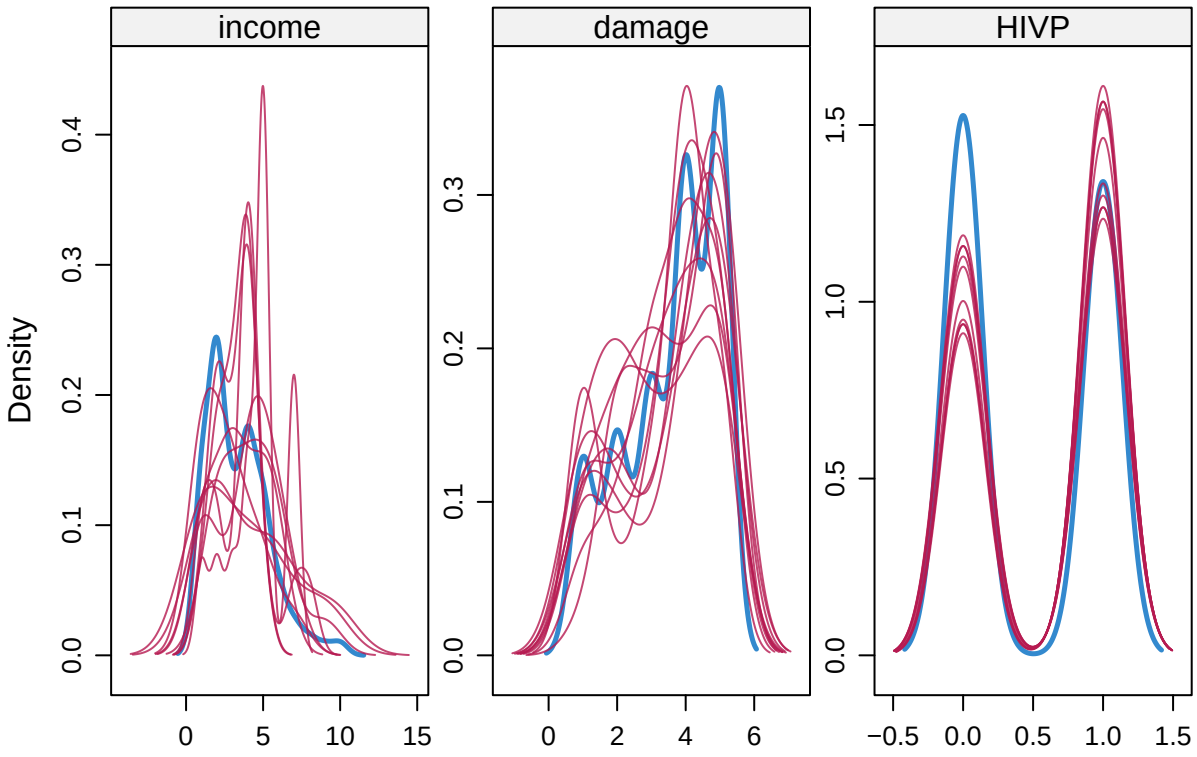
```
# Fit full logistic model across imputations
mi_full <- with(M,
  glm(HIVP ~ age + income + healthy + mental + damage + treatment,
      family = binomial, data = R))

# Pool results and extract estimate, SE, t-statistic, p-value
pooled_full <- pool(mi_full)
sum_full <- summary(pooled_full)[, c(1:3, 6)]
knitr::kable(sum_full, caption = "Summary of Pooled Full Model")
```

Table 23: Summary of Pooled Full Model

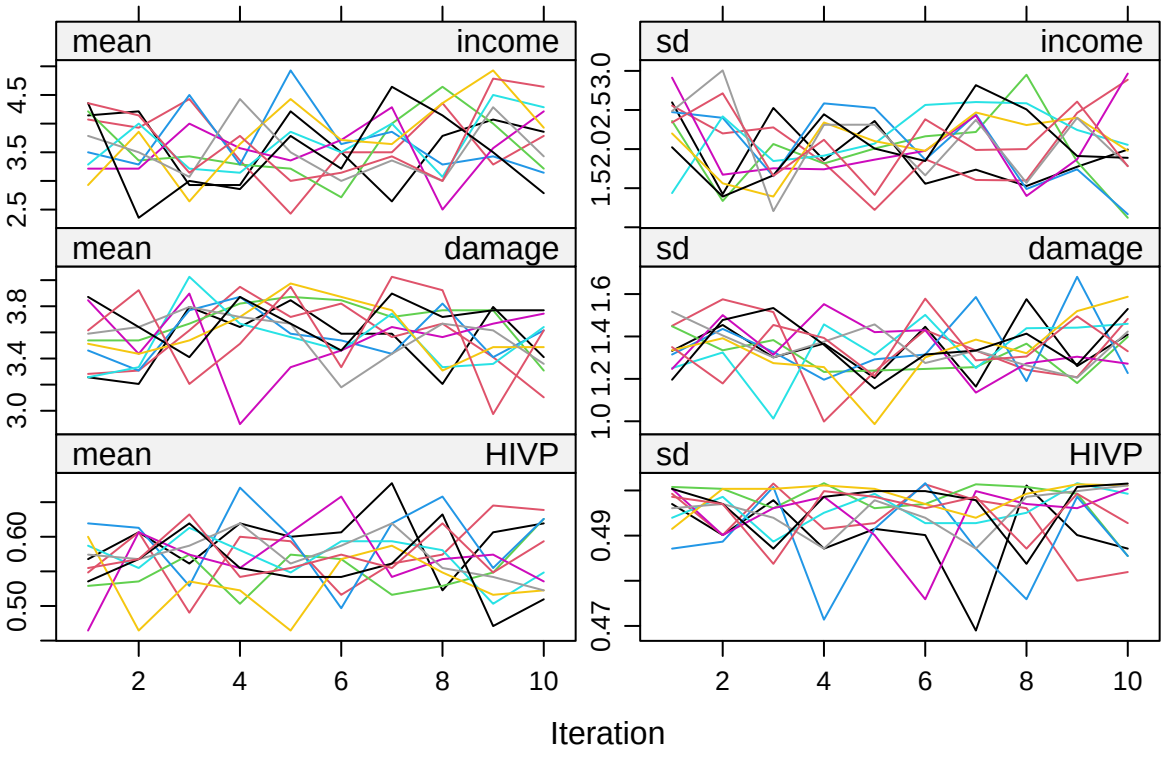
term	estimate	std.error	p.value
(Intercept)	5.8317000	1.0841075	0.0000001
age	-0.0523040	0.0160161	0.0012080
income	-0.1781393	0.0623693	0.0045626
healthy	-0.0152292	0.0105784	0.1509234
mental	0.4106581	0.2859038	0.1518615
damage	-0.5055123	0.1019747	0.0000012
treatment	-0.6355727	0.1452864	0.0000164

7.3 Convergence Diagnostics

```
# Check that the chained equations have converged
densityplot(M)
```




```
plot(M)
```



7.4 Reduced Model

```r
# Drop non-significant covariates to get parsimonious model
mi_reduced <- with(M,
  glm(HIVP ~ age + income + damage + treatment,
      family = binomial, data = R))

pooled_red <- pool(mi_reduced)
sum_red <- summary(pooled_red)[, c(1:3, 6)]

knitr::kable(sum_red, caption = "Summary of Pooled Reduced Model")
```

Table 24: Summary of Pooled Reduced Model

term	estimate	std.error	p.value
(Intercept)	5.2212393	0.8832304	0.0000000
age	-0.0490734	0.0155519	0.0017510
income	-0.1924142	0.0619273	0.0020545
damage	-0.5240140	0.1007199	0.0000003
treatment	-0.6228971	0.1436224	0.0000192

7.5 Model Comparison

```r
# Model comparison via D1 across imputations
D1_res <- D1(mi_full, mi_reduced)
D1_df <- as.data.frame(D1_res$result)
colnames(D1_df) <- c("test_statistic", "df1", "df2", "p_value", "r.v.i")

knitr::kable(D1_df, caption = "D1 Test: Full vs Reduced Model")
```

Table 25: D1 Test: Full vs Reduced Model

test_statistic	df1	df2	p_value	r.v.i
2.296626	2	326.0181	0.1022231	0

8 Conclusion

```r
# CCA (BIC-selected model)
cca_tab <- tidy(best_model_CCA) %>%
  select(term, estimate, std.error) %>%
  mutate(method = "CCA")

# Single imputation (deterministic SI and BIC model)
si_tab <- tidy(PM_DET) %>%
  select(term, estimate, std.error) %>%
  mutate(method = "SI")

# Bayesian
bayes_raw <- summary(J1)[[6]]$HIVP$regcoef
bayes_tab <- as.data.frame(bayes_raw) %>%
  rownames_to_column("term") %>%
  rename(estimate = Mean, std.error = SD) %>%
```

```r
  select(term, estimate, std.error) %>%
  mutate(method = "Bayesian")

# Multiple imputation
mi_tab <- summary(pooled_red) %>%
  select(term, estimate, std.error) %>%
  mutate(method = "MI")

# Combine and filter to key terms
keep_terms <- c("(Intercept)", "age", "income", "damage", "treatment")
comp_tab <- bind_rows(cca_tab, si_tab, bayes_tab, mi_tab) %>%
  filter(term %in% keep_terms) %>%
  arrange(term, method)

knitr::kable(
  comp_tab,
  caption = "Comparison of Coefficients and Standard Errors Across Methods",
  digits  = 3,
  align   = c("l","r","r","l")
)
```

Table 26: Comparison of Coefficients and Standard Errors Across Methods

term	estimate	std.error	method
(Intercept)	5.019	0.874	Bayesian
(Intercept)	5.221	0.883	CCA
(Intercept)	5.221	0.883	MI
(Intercept)	4.965	0.844	SI
age	-0.048	0.015	Bayesian
age	-0.049	0.016	CCA
age	-0.049	0.016	MI
age	-0.046	0.015	SI
damage	-0.518	0.099	Bayesian
damage	-0.524	0.101	CCA
damage	-0.524	0.101	MI
damage	-0.502	0.097	SI
income	-0.177	0.061	Bayesian
income	-0.192	0.062	CCA
income	-0.192	0.062	MI
income	-0.192	0.060	SI
treatment	-0.623	0.145	Bayesian
treatment	-0.623	0.144	CCA
treatment	-0.623	0.144	MI
treatment	-0.625	0.140	SI

```r
# misclassification cost
cost_fn <- function(actual, pred_prob) {
  mean(abs(actual - (pred_prob > 0.5)))
}

# CCA reduced (BIC-selected)
dat_cca <- model.frame(best_model_CCA)
```

```r
cv_cca <- cv.glm(
  data   = dat_cca,
  glmfit = best_model_CCA,
  cost   = cost_fn,
  K      = 10
)$delta[1]

# SI reduced (deterministic and BIC-selected)
dat_si <- model.frame(PM_DET)
cv_si <- cv.glm(
  data   = dat_si,
  glmfit = PM_DET,
  cost   = cost_fn,
  K      = 10
)$delta[1]

# MI reduced
m        <- M$m
cv_vals  <- numeric(m)
for(i in seq_len(m)) {
  d_i         <- complete(M, action = i)
  fit_i       <- glm(HIVP ~ age + income + damage + treatment,
                     data = d_i, family = binomial)
  cv_vals[i]  <- cv.glm(d_i, fit_i, cost_fn, K = 10)$delta[1]
}
cv_mi <- mean(cv_vals)

# Bayesian reduced
dat_bayes <- na.omit(R[, c("HIVP","age","income","damage","treatment")])
coefs     <- summary(J1)[[6]]$HIVP$regcoef[,"Mean"]
Xb        <- model.matrix(~ age + income + damage + treatment, data = dat_bayes)
p_hat     <- plogis(Xb %*% coefs)
cv_bayes  <- mean(abs(dat_bayes$HIVP - (p_hat > 0.5)))

# Summarise
cv_summary <- data.frame(
  Method   = c("CCA", "SI", "MI", "Bayesian"),
  CV_Error = c(cv_cca, cv_si, cv_mi, cv_bayes)
)

knitr::kable(
  cv_summary,
  digits  = 3,
  caption = "10-Fold CV Misclassification Rate for Reduced Models"
)
```

Table 27: 10-Fold CV Misclassification Rate for Reduced Models

Method	CV_Error
CCA	0.301
SI	0.292
MI	0.296
Bayesian	0.275