APSTA-2013: Final Project

William Spagnola

3/20/2019

Part A: Introduction

Dataset

My dataset is a subset of the Behavioral Risk Factor Surveillance System (BRFSS), which I obtained from the naniar package. The dataset included 245 observations on a number of health measures. I kept four of the 34 variables from the original datset: age, BMI, health insurance coverage (*health_cover*), and self-reported general health (*health_general*). *Age* and *BMI* are continuous variables and are self-explanatory. *Health_cover* is a binary variable that is coded 1 for subjects who reported having health insurance and 0 for subjects who did not have insurance. *Health_general* was an ordinal factor that was measured on a five point scale: Poor, Fair, Good, Very Good and Excellent. In order to simplify my analysis, I recoded *health_general* as a binary variable called *health_good*. A *health_general* rating of 'fair' or 'poor' was coded as 0 and a rating of 'good', 'very good', or 'excellent' rating was coded as 1.

Analysis

My analysis plan was to predict 'good health' (*health_good*) based on each subject's age, BMI, and health insurance coverage (*health_cover*). Since *health_good* was a binary variable, I opted to fit a binomial logistic regression model on the data.

```
set.seed(2013)

### Load Data
risk_sub <- read.csv('data/risk_sub.csv')
rownames(risk_sub) <- NULL

#Convert Health_general and health_cover to binary variables coded 1 and 0
risk_sub$health_cover <- ifelse(risk_sub$health_cover=='Yes', 1, 0)
risk_sub$health_good <- ifelse(risk_sub$health_general %in% c('Excellent', 'VeryGood', 'Good'), 1, 0)

#Drop Id, Health_general and Smoking Variables
risk_sub <- risk_sub[, -which(names(risk_sub) %in% c('id', 'health_general', 'smoke_days'))]
head(risk_sub)
```

```
##   age   bmi health_cover health_good
## 1  49 32.68            1           1
## 2  48 25.90            1           0
## 3  55 28.04            1           0
## 4  42 29.59            1           1
## 5  66 24.74            1           1
## 6  66 23.72            1           1
```

Missing Data

In the original dataset, all the variables were completely observed except BMI, in which 4.5% cases were missing. Consequently, I had to generate missing data in order to explore different imputation methods.

```
 sapply(risk_sub, function(x) mean(is.na(x)))
```

```
##          age          bmi health_cover  health_good
##   0.00000000   0.04489796   0.00000000   0.00000000
```

Use Ampute Function

I used the ampute function from the mice package to generate missing data at random. After generating the missing data, I replaced the age variable with the completely observed column from the original dataset. Having one completely observed continuous variable was necessary in order to perform all the imputation methods.

```
#Get index of Rows to Generate MAR
missing_row_idx <- complete.cases(risk_sub[ , ])

#Create new data.frame to use for MAR data
risk_MAR <- risk_sub

#Use ampute function to generate missing data in bmi and health_cover columns
a <- ampute(risk_sub[ missing_row_idx , ],
            bycases = F,
            prop = .16)

#Replace data with MAR data generated by ampute function
risk_MAR[missing_row_idx , ] <-  a$amp

#Replace age with original data so that I have one completely observed continous variable
risk_MAR$age <- risk_sub$age
```

Examine Missing Data

After using the ampute function to generate missing data at random, 22.4% of the *bmi* values were missing, 16.3% of the *health_cover* values were missing and 16.3% of the *health_good* values were missing. *Age* was completely observed. Out of all the cases, 44.9% were completely observed. Overall, 13.8% of the values were missing.

```
miss_prop_summary(risk_MAR)
```

```
## # A tibble: 1 x 3
##      df   var  case
##   <dbl> <dbl> <dbl>
## 1 0.138  0.75 0.551
```

```
miss_var_summary(risk_MAR)
```

```
## # A tibble: 4 x 3
##   variable     n_miss pct_miss
##   <chr>         <int>    <dbl>
## 1 bmi              55     22.4
## 2 health_cover     40     16.3
## 3 health_good      40     16.3
## 4 age               0      0
```

Part B: Methods

Since there were very few missing variables in the original dataset, I decided to first fit a model on the full dataset for comparison. Then I fit models using listwise deletion, mean/mode imputation, random imputation, the dummy variable method, hotdecking, regression imputation, stochastic imputation (regression with noise), and multiple imputation using the mi package. For each model, I printed out a table of the coefficient estimates. The tables also included the standard errors of each of the coefficients. In addition, I included equations based on the coefficients produced by each model.

Because my data was cross-sectional, I omitted the LVCF method, which is generally used for longitudinal data. In addition, as a caveat, despite setting the seed, the *mi* function produced slightly different coefficients every time I ran the program. In the fourth section, I explain in further detail the range of numbers produced by the *mi* function for multiple imputation.

Full Dataset

```
risk_full  <- risk_sub[complete.cases(risk_sub) ,]
fit_full <- glm(formula = health_good ~ health_cover + age + bmi, data = risk_full,
                family = binomial)
summary(fit_full)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + age + bmi, family = binomial,
##     data = risk_full)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3777  -0.5527   0.6472   0.7788   1.3196
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.274852   0.969979   3.376 0.000735 ***
## health_cover  1.187222   0.461667   2.572 0.010123 *
## age          -0.028407   0.009978  -2.847 0.004413 **
## bmi          -0.054041   0.023050  -2.345 0.019052 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 264.27  on 233  degrees of freedom
## Residual deviance: 247.89  on 230  degrees of freedom
## AIC: 255.89
##
## Number of Fisher Scoring iterations: 4
```

1. Listwise deletion.

For the listwise deletion method, we simply perform the analysis on the complete cases only and disregard the cases with incomplete information.

```
risk_CC <- cc(risk_MAR)
fit_list_wise <-glm(health_good ~ bmi + age + health_cover,
```

```
            data =risk_CC, family = binomial(link = 'logit'))
summary(fit_list_wise)
```

```
##
## Call:
## glm(formula = health_good ~ bmi + age + health_cover, family = binomial(link = "logit"),
##     data = risk_CC)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2398  -0.9585   0.5442   0.7835   1.3511
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.66105    1.40750   2.601 0.009293 **
## bmi          -0.02153    0.03169  -0.679 0.496924
## age          -0.05366    0.01612  -3.328 0.000874 ***
## health_cover  1.18477    0.68866   1.720 0.085362 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 126.90  on 109  degrees of freedom
## Residual deviance: 112.32  on 106  degrees of freedom
## AIC: 120.32
##
## Number of Fisher Scoring iterations: 4
```

Report Formula for listwise deletion model

$$HEALTH\_GOOD = 3.36 + -0.02 * BMI + -0.05 * AGE + 1.18 * HEALTH\_COVER$$

2. Mean/mode imputation.

For the mean/mode imputation, we use the mean of the observed values for the imputed values for all the continous variables with missing data. For categorical variables, we do the same thing but use the mode instead of the mean.

```
#Function for Mean Imputation
mean.imp <- function (a){
  missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- mean(a.obs)
  return (imputed)
}

#Function for Mode
mode = function(x) {
  ta = table(x)
  tam = max(ta)
  if (all(ta == tam))
    mod = NA
```

```
  else
    mod = names(ta)[ta == tam]
  return(mod)
}


#Function for Mode Imputation
mode.imp <- function (a) {
  missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- mode(a.obs)
  # Output the imputed vector
  return(imputed)
}

#Create A Duplicate Dataset
risk_imp_mean <- risk_MAR

#Apply Mode Imputation to Indicator Variables (Health_Good & Health_Cover)
risk_imp_mean$health_cover <- as.integer(mode.imp(risk_MAR$health_cover))
risk_imp_mean$health_good <- as.integer(mode.imp(risk_MAR$health_good))

#Apply Mean Imputation to Continuous Variables (BMI)
risk_imp_mean$bmi <- mean.imp(risk_MAR$bmi)

#Perform Analysis
fit_imp_mean_mode <- glm(health_good ~ health_cover +bmi + age,
                         data = risk_imp_mean,
                         family = binomial(link = 'logit'))
summary(fit_imp_mean_mode)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age, family = binomial(link = "logit"),
##     data = risk_imp_mean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2234   0.4931   0.6282   0.7073   1.0747
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.059610   1.028552   2.975  0.00293 **
## health_cover  0.630248   0.528898   1.192  0.23341
## bmi          -0.042250   0.025128  -1.681  0.09268 .
## age          -0.018462   0.009943  -1.857  0.06336 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 250.64  on 244  degrees of freedom
## Residual deviance: 244.09  on 241  degrees of freedom
```

```
## AIC: 252.09
##
## Number of Fisher Scoring iterations: 4
```

Equation: Mean/Mode Imputation

$$HEALTH\_GOOD = 3.05 + 0.63 * HEALTH\_COVER + -0.04 * BMI + -0.02 * AGE$$

3. Random imputation.

For random imputation, we randomly draw observed values to impute the missing values on a given variable.

```
## Simple random imputation
random.imp <- function (a){
  missing <- is.na(a)
  n.missing <- sum(missing)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- sample(a.obs, n.missing, replace=TRUE)
  return (imputed)
}

#Perform Random Imputation
risk_imp_rand <- risk_MAR
risk_imp_rand$bmi <- random.imp(risk_MAR$bmi)
risk_imp_rand$health_cover <- random.imp(risk_MAR$health_cover)
risk_imp_rand$health_good <- random.imp(risk_MAR$health_good)

#Perform Analysis
fit_imp_rand <- glm(health_good ~ health_cover + bmi + age,
                        data = risk_imp_rand,
                        family = binomial(link = 'logit'))
summary(fit_imp_rand)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age, family = binomial(link = "logit"),
##     data = risk_imp_rand)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1871  -1.0680   0.6781   0.8049   1.1441
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.925507   0.915556   3.195   0.0014 **
## health_cover  0.512311   0.471206   1.087   0.2769
## bmi          -0.032535   0.020207  -1.610   0.1074
## age          -0.022859   0.009355  -2.444   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 277.18  on 244  degrees of freedom
## Residual deviance: 268.30  on 241  degrees of freedom
## AIC: 276.3
##
## Number of Fisher Scoring iterations: 4
```

Equation: Random Imputation

$$HEALTH\_GOOD = 2.93 + 0.51 * HEALTH\_COVER + -0.03 * BMI + -0.02 * AGE$$

4. Dummy variable on predictor variables.

For the dummy variable method, we assign an arbitrary value to all the missing values (e.g. zero or the mean). Then we create an indicator variable for each missing variable. These indicator variables tell whether a given value was observed or missing. Then we perform the analysis using both the original variables and the missing data indicator variables.

```r
#Create Indicator Variabler
health_cover_missing <- is.na(risk_MAR$health_cover)*1
bmi_missing <- is.na(risk_MAR$bmi)*1

risk_dummy <- risk_MAR

#Add arbitrary number for missing values of predictors
risk_dummy[bmi_missing == 1,] <- 0
risk_dummy[health_cover_missing == 1,] <- 0

#Add indicator variables for to inicate rows with missing values on predictor variables
risk_dummy$health_cover_miss <- health_cover_missing
risk_dummy$bmi_miss <- bmi_missing

#Remove Rows with Missing Outcome Variable
risk_dummy_cc <- risk_dummy[!is.na(risk_dummy$health_good) , ]

#Perform Analysis
fit_dummy <- glm(health_good ~ health_cover + bmi + age + health_cover_miss + bmi_miss,
                         data = risk_dummy_cc,
                         family = binomial(link = 'logit'))
summary(fit_dummy)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age + health_cover_miss +
##     bmi_miss, family = binomial(link = "logit"), data = risk_dummy_cc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.23976  -0.00008  -0.00008   0.56970   1.35114
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.66105    1.40754   2.601 0.009294 **
```

```
## health_cover         1.18477    0.68868   1.720 0.085367 .
## bmi                 -0.02153    0.03170  -0.679 0.496928
## age                 -0.05366    0.01612  -3.328 0.000874 ***
## health_cover_miss  -23.22712 1700.35933  -0.014 0.989101
## bmi_miss           -23.22712 1450.07150  -0.016 0.987220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.10  on 204  degrees of freedom
## Residual deviance: 112.32  on 199  degrees of freedom
## AIC: 124.32
##
## Number of Fisher Scoring iterations: 18
```

Equation: Dummy Variable Method

$$HEALTH\_GOOD = 3.66 + 1.18 * HEALTH\_COVER + -0.02 * BMI + -0.05 * AGE$$
$$+ - 23.23 * MISSING\_HEALTH\_COVER + -22.23 * MISSING\_BMI$$

5. LVCF (if applicable to your data).

My data is cross-sectional so the LVCF method is not applicable.

Last value carried forward simply means replacing a missing value with the previously observed value in a longitudinal study.

6. Hotdecking (nearest neighbor).

Hotdecking involves using a distance metric to find the nearest neighbor among the complete cases for each case with a missing value. This nearest neighbor serves as a 'donor' for the missing value. For example, if two cases are similar on health coverage and age, but one case is missing bmi, we can use the observed value of bmi on the complete case as a 'donor' for the case that is missing bmi.

```
# Using Manhattan distance and variance as weight to rescale variables:
risk_hot_deck <- impute.NN_HD(DATA=risk_MAR,
                              distance="eukl",
                              donor_limit = 2,
                              optimal_donor = 'mmin',
                              weights="var",
                              diagnose = 'diagnostics')
#Perform Analysis
fit_hot_deck <- glm(health_good ~ health_cover + bmi + age,
                    data = risk_hot_deck,
                    family = binomial(link = 'logit'))
summary(fit_hot_deck)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age, family = binomial(link = "logit"),
##     data = risk_hot_deck)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3686  -1.2222   0.6727   0.8065   1.3169
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.210668   0.884635   2.499  0.01246 *
## health_cover  1.000239   0.420322   2.380  0.01733 *
## bmi          -0.013103   0.021205  -0.618  0.53661
## age          -0.028896   0.009365  -3.086  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 287.47  on 244  degrees of freedom
## Residual deviance: 273.10  on 241  degrees of freedom
## AIC: 281.1
##
## Number of Fisher Scoring iterations: 4
```

Equation: Hotdecking Method

$$HEALTH\_GOOD = 2.21 + 1.00 * HEALTH\_COVER + -0.01 * BMI + -0.03 * AGE$$

7. Regression imputation.

For regression imputation, we use the complete cases to regress each variable with missing data on some combination of the other variables. Then we use the models derived from this process to impute values in the cases that are missing data.

```
risk_imp_reg <- risk_MAR
df_cc <- risk_MAR[complete.cases(risk_MAR) , ] #Subset complete cases (cc)

#BMI: Regression Imputation
Ry <- as.numeric(!is.na(risk_MAR$bmi))# Missing data indicator
bmi_miss <- risk_MAR[Ry == 0 , ] #Subset rows with missing BMI values
mod_reg_imp <- lm(bmi ~ age + health_good + health_cover, data = df_cc) #Run model on cc data
bmi_imp_reg <- predict(mod_reg_imp, newdata = bmi_miss) #Use model to predict missing values
risk_imp_reg$bmi[Ry ==0] <- bmi_imp_reg  #Replace missing values

#HEALTH INSURANCE COVERAGE: Regression Imputation
Ry <- as.numeric(!is.na(risk_MAR$health_cover)) # Missing data indicator
health_cover_miss <- risk_imp_reg[Ry == 0 , ] #Subset rows with missing HEALTH_COVER values
health_cover_mod <- glm(health_cover ~ age + bmi + health_good,
                        data = df_cc,
                        family = binomial(link = 'logit')) #Fit model on cc data
health_cover_probs <- predict(health_cover_mod,
                              newdata = health_cover_miss,
                              type = 'response') #Predict Probability for missing data
heath_cover_imp_reg <-  as.integer(round(health_cover_probs)) #Round probs to 0/1
risk_imp_reg$health_cover[Ry == 0] <- heath_cover_imp_reg  #Replace missing values
```

```r
#GOOD HEALTH: Regression Imputation
Ry <- as.numeric(!is.na(risk_MAR$health_good)) # Missing data indicator
health_good_miss <- risk_imp_reg[Ry == 0 , ] #Subset rows with missing HEALTH_COVER values
health_good_mod <- glm(health_good ~ age + bmi + health_cover,
                       data = df_cc,
                       family = binomial(link = 'logit')) #Fit model on cc data
health_good_probs <- predict(health_good_mod,
                             newdata = health_good_miss,
                             type = 'response') #Predict Probability for missing data
heath_good_imp_reg <-  as.integer(health_good_probs) #Round probs to 0/1
risk_imp_reg$health_good[Ry == 0] <- heath_good_imp_reg  #Replace missing COVER values

#Perform Analysis
fit_imp_reg <- glm(health_good ~ health_cover + bmi + age,
                       data = risk_imp_reg,
                       family = binomial(link = 'logit'))
summary(fit_imp_reg)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age, family = binomial(link = "logit"),
##     data = risk_imp_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0380  -1.2614   0.7869   0.9565   1.4341
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.100052   0.890315   2.359  0.01834 *
## health_cover  0.925836   0.478924   1.933  0.05322 .
## bmi          -0.031391   0.022560  -1.391  0.16409
## age          -0.026066   0.008506  -3.065  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 323.26  on 244  degrees of freedom
## Residual deviance: 310.17  on 241  degrees of freedom
## AIC: 318.17
##
## Number of Fisher Scoring iterations: 4
```

Equation: Regression Imputation

$$HEALTH\_GOOD = 2.10 + 0.93 * HEALTH\_COVER + -0.03 * BMI + -0.03 * AGE$$

8. Regression imputation with Noise

Perform only on numerical and dichotomous variables. Regression imputation with noise is the same thing as regression imputation except we add random noise to each predicition. Adding noise is important because it

helps to produce imputed values with the same variance as the observed data.

```
risk_imp_stoch <- risk_MAR

#BMI: Stochastic Imputation
Ry <- as.numeric(!is.na(risk_MAR$bmi))# Missing data indicator
bmi_miss <- risk_MAR[Ry == 0 , ] #Subset rows with missing BMI values
df_cc <- risk_MAR[complete.cases(risk_MAR) , ] #Subset complete cases (cc)
mod_reg_imp <- lm(bmi ~ age + health_good + health_cover, data = df_cc) #Run model on cc data
bmi_imp <- predict(mod_reg_imp, newdata = bmi_miss) #Use model to predict missing BMI values
noise  <- rnorm(length(bmi_imp), 0, summary(mod_reg_imp)$sigma) #Generate noise
bmi_imp_stoch <- bmi_imp + noise #Combine predictions with noise
risk_imp_stoch$bmi[Ry ==0] <- bmi_imp_stoch  #Replace missing values

#HEALTH INSURANCE COVERAGE: Stochastic Imputation
Ry <- as.numeric(!is.na(risk_MAR$health_cover)) # Missing data indicator
health_cover_miss <- risk_MAR[Ry == 0 , ] #Subset rows with missing HEALTH_COVER values
health_cover_mod <- glm(health_cover ~ age + bmi + health_good,
                        data = df_cc,
                        family = binomial(link = 'logit')) #Fit model on cc data
health_cover_probs <- predict(health_cover_mod,
                              newdata = health_cover_miss,
                              type = 'response') #Predict Probability for missing data
heath_cover_imp_stoch <-  rbinom(n = length(health_cover_probs),
                            size = 1,
                            prob = health_cover_probs) #Draw from binomial distribution
risk_imp_stoch$health_cover[Ry == 0] <- heath_cover_imp_stoch  #Replace missing values

#GOOD HEALTH: Stochastic Imputation
Ry <- as.numeric(!is.na(risk_MAR$health_good)) # Missing data indicator
health_good_miss <- risk_MAR[Ry == 0 , ] #Subset rows with missing HEALTH_COVER values
health_good_mod <- glm(health_good ~ age + bmi + health_cover,
                        data = df_cc,
                        family = binomial(link = 'logit')) #Fit model on cc data
health_good_probs <- predict(health_good_mod,
                              newdata = health_good_miss,
                              type = 'response') #Predict Probability for missing data
heath_good_imp_stoch <-  rbinom(n = length(health_good_probs),
                            size = 1,
                            prob = health_good_probs) #Draw from binomial distribution
risk_imp_stoch$health_good[Ry == 0] <- heath_good_imp_stoch  #Replace missing COVER values

#Perform Analysis
fit_imp_stoch <- glm(health_good ~ health_cover +bmi + age,
                        data = risk_imp_stoch,
                        family = binomial(link = 'logit'))
summary(fit_imp_stoch)
```

```
##
## Call:
## glm(formula = health_good ~ health_cover + bmi + age, family = binomial(link = "logit"),
##     data = risk_imp_stoch)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.2877  -1.1049   0.6389   0.7925   1.4421
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.232945   0.900385   3.591  0.00033 ***
## health_cover  1.145455   0.458713   2.497  0.01252 *
## bmi          -0.050673   0.020727  -2.445  0.01450 *
## age          -0.031009   0.009499  -3.264  0.00110 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.41  on 244  degrees of freedom
## Residual deviance: 268.42  on 241  degrees of freedom
## AIC: 276.42
##
## Number of Fisher Scoring iterations: 4
```

Equation: Regression with Noise Imputation

$$HEALTH\_GOOD = 3.23 + 1.15 * HEALTH\_COVER + -0.05 * BMI + -0.03 * AGE$$

Multiple imputation with either mi or mice package:

In multiple imputation, we perform imputations with a number of chains and then average the values created by the chains. This is done with Monte Carlo Markov Chain using either a posterior predictive distribution or predictive mean matching.

9. Load your data into the package. Obtain summary, histogram and image of the data.

```
risk_mdf <- missing_data.frame(risk_MAR)
```

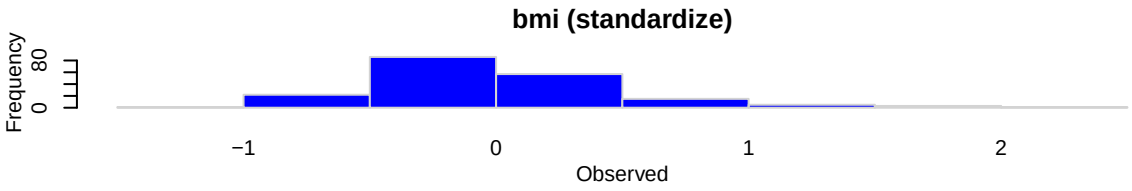
A) Summary

```
summary(risk_mdf)
```

```
##       age              bmi         health_cover     health_good
##  Min.   : 7.00   Min.   :14.21   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:48.00   1st Qu.:23.25   1st Qu.:1.0000   1st Qu.:1.0000
##  Median :59.00   Median :26.66   Median :1.0000   Median :1.0000
##  Mean   :58.11   Mean   :27.86   Mean   :0.8927   Mean   :0.7512
##  3rd Qu.:70.00   3rd Qu.:31.38   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :97.00   Max.   :55.72   Max.   :1.0000   Max.   :1.0000
##                  NA's   :55      NA's   :40       NA's   :40
```

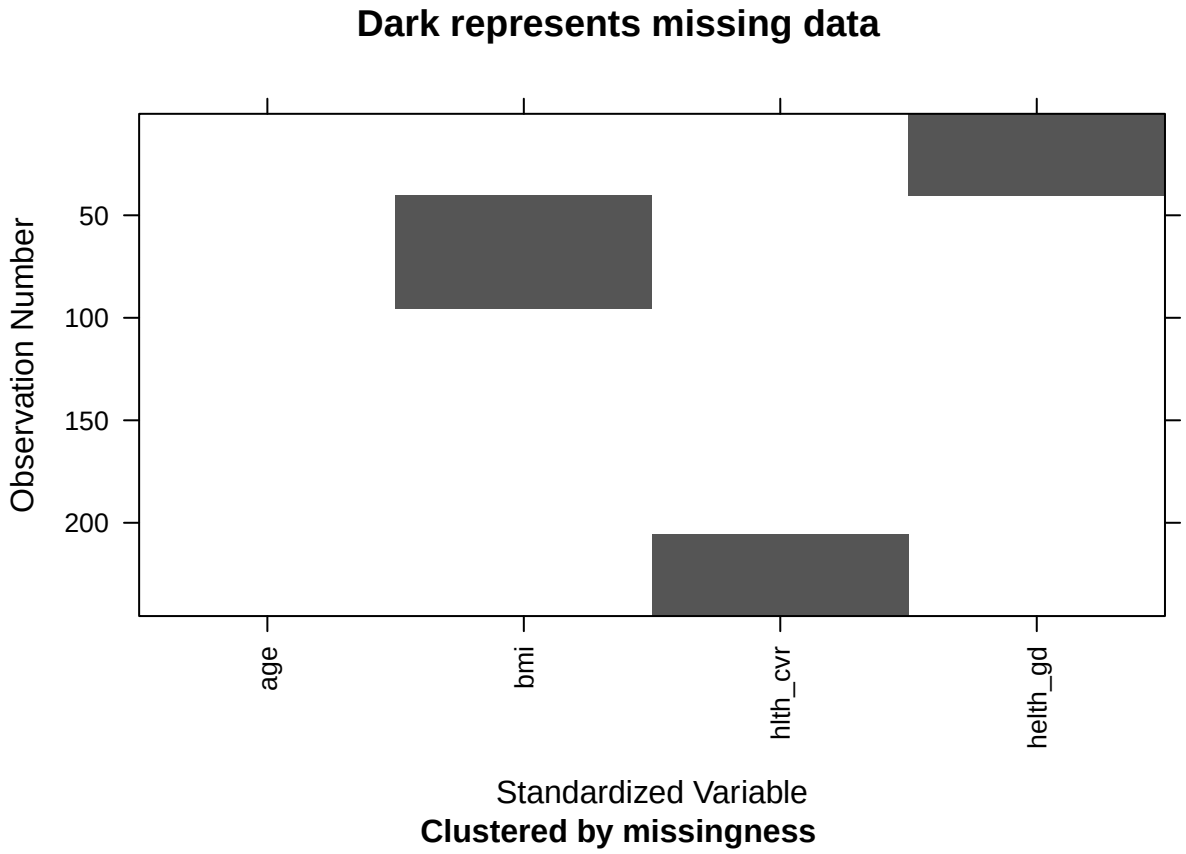
B) Histogram

```
hist(risk_mdf, ask = F)
```



C) Image

```
image(risk_mdf, grayscale = TRUE)
```



10. Check your data types and make changes if necessary.

There did not seem to be any need for changes. *BMI* was continuous, so the imputation method would be the identity link, i.e. regular linear regression. *Health_Cover* and *Health_Good* were both binary variables, so they should be imputed with binomial logistic regression, which uses a logit link. *Age* was not missing any data, so there was no need to impute any values in that column, hence the NAs in the table below.

```
show(risk_mdf)
```

```
## Object of class missing_data.frame with 245 observations on 4 variables
##
## There are 4 missing data patterns
##
## Append '@patterns' to this missing_data.frame to access the corresponding pattern for every observati
##
##                    type missing method  model
## age          continuous       0   <NA>   <NA>
## bmi          continuous      55    ppd linear
## health_cover     binary      40    ppd  logit
## health_good      binary      40    ppd  logit
##
##                family     link transformation
## age               <NA>     <NA>    standardize
## bmi          gaussian identity    standardize
## health_cover binomial    logit           <NA>
## health_good  binomial    logit           <NA>
```

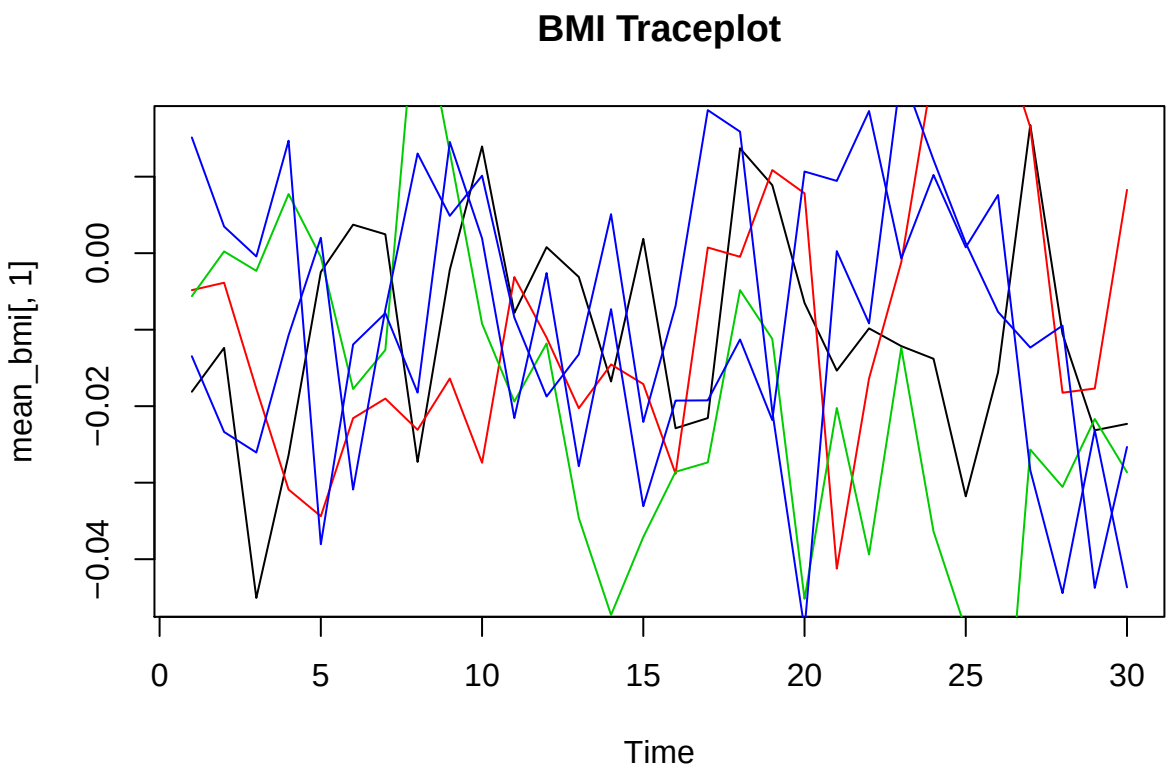
11. Run the mi command and check convergence by traceplots.

```
risk_imp_MI <- mi(risk_mdf, n.chains = 5, n.iter = 30, verbose = F, seed = 2013)
```

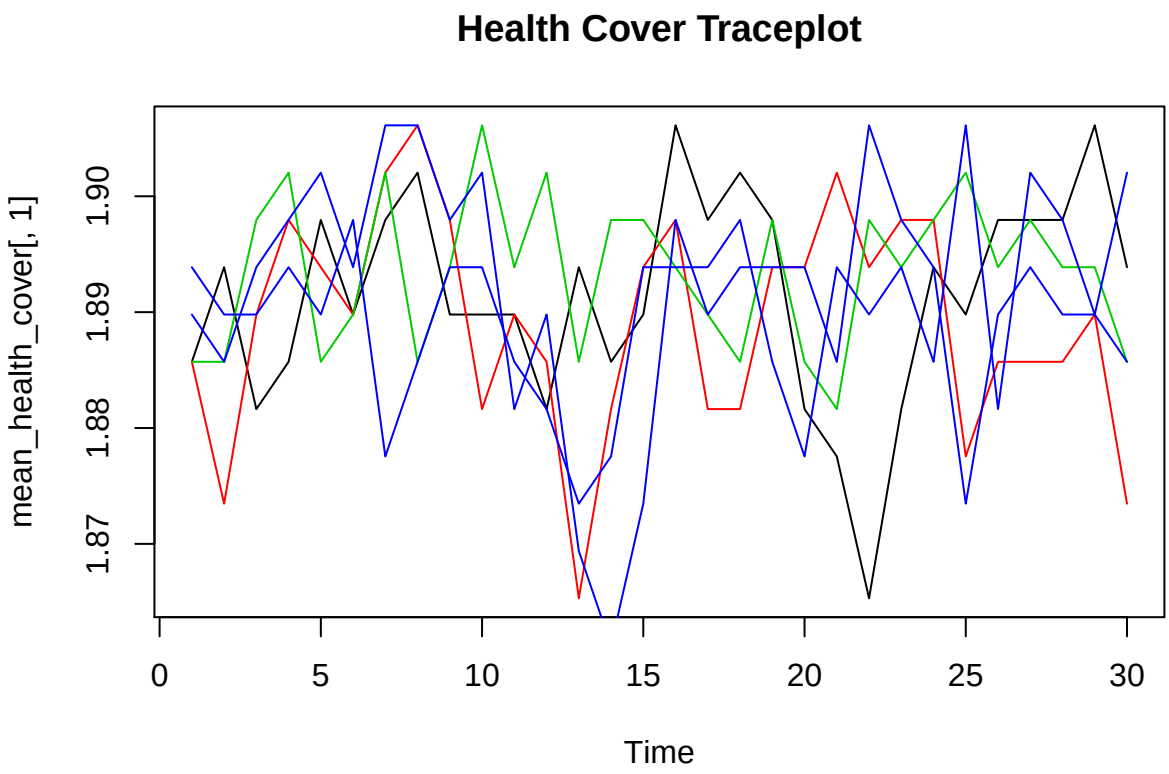
Trace Plots

None of the trace plots provided clear evidence of divergence. However, some of the lines extended beyond the y-limits of the plots. Also, I was a little concerned about the red line at the end of the *health_cover* plot. It seemed possible that diverged on the 30th iteration.

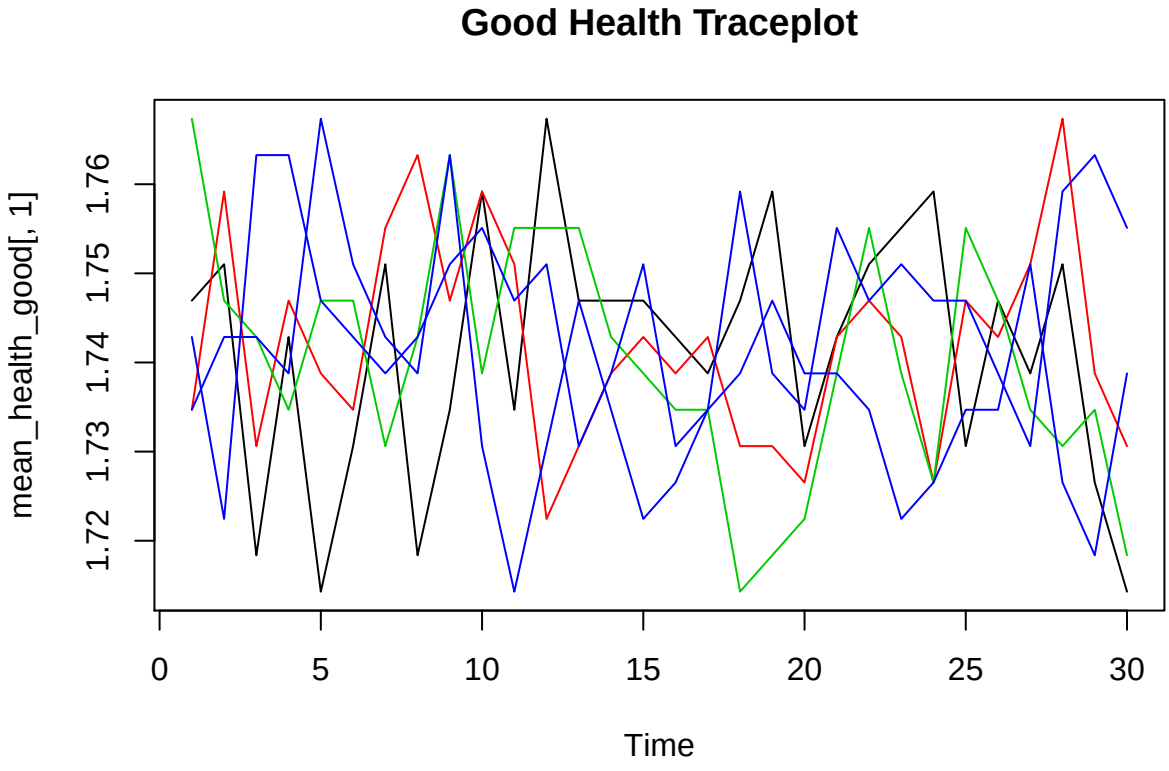
```
converged <- mi2BUGS(risk_imp_MI)

mean_bmi <- converged[, , 1]
mean_health_cover <- converged[, , 2]
mean_health_good <- converged[, , 3]

ts.plot(mean_bmi[,1], col=1, main = 'BMI Traceplot')
lines(mean_bmi[,2], col= 2)
lines(mean_bmi[,3], col= 3)
lines(mean_bmi[,4], col= 4)
lines(mean_bmi[,5], col= 4)
```




```
ts.plot(mean_health_cover[,1], col=1,  main = 'Health Cover Traceplot')
lines(mean_health_cover[,2], col= 2)
lines(mean_health_cover[,3], col= 3)
lines(mean_health_cover[,4], col= 4)
lines(mean_health_cover[,5], col= 4)
```




```
ts.plot(mean_health_good[,1], col=1,  main = 'Good Health Traceplot')
lines(mean_health_good[,2], col= 2)
lines(mean_health_good[,3], col= 3)
lines(mean_health_good[,4], col= 4)
lines(mean_health_good[,5], col= 4)
```



12. Check r-hats.

All of the r-hats were close to 1 and less than 1.1, which was a good sign.

```
Rhats(risk_imp_MI)
```

```
##          mean_bmi mean_health_cover  mean_health_good            sd_bmi
##         1.0192877         0.9997305         0.9876479         1.0004799
##   sd_health_cover    sd_health_good
##         0.9994022         0.9877448
```

13. Increase number of imputations if necessary.

The r-hats were close to one, which was a good sign. Nevertheless, although there was no clear sign of divergence in the trace plots, I was a little concerned about some of the anomalies I saw, particularly the possible divergence of one of the imputations at the end of the *health_cover* plot. I decided to increase the number of iterations from 30 to 60 just to be sure that the imputations converged. Because the model was relatively simple and the dataset relatively small, the increase in iterations was not too much of a burden on my computer.

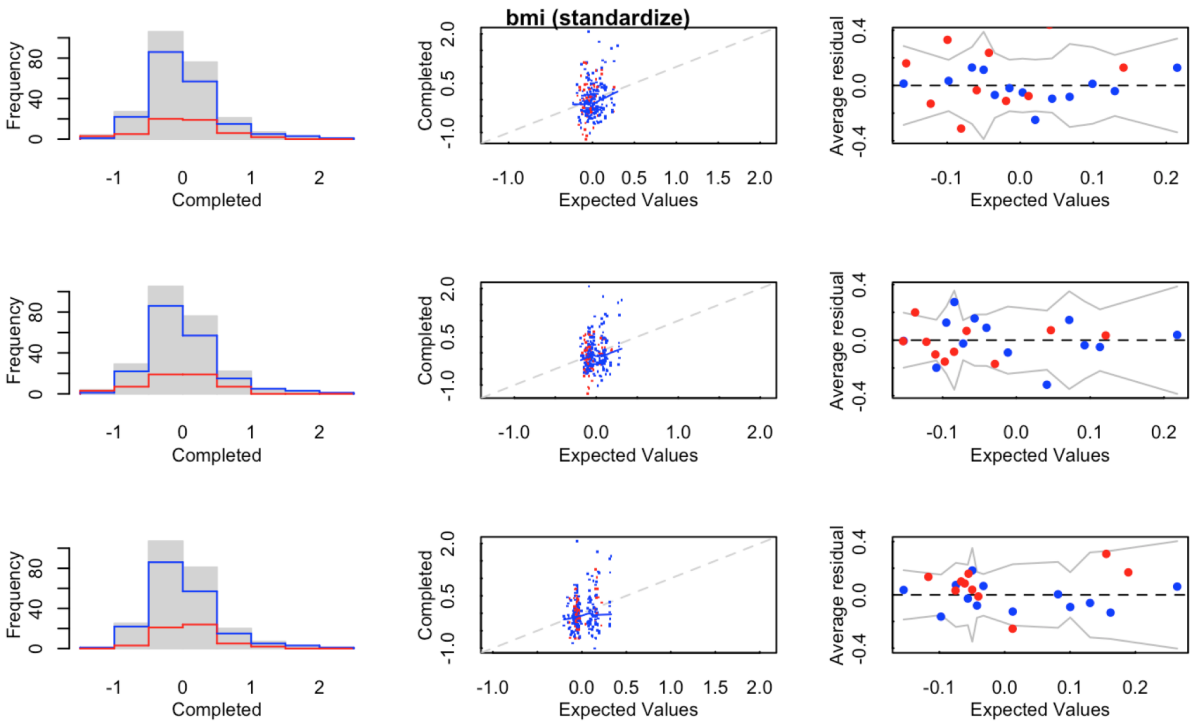
```
set.seed(2013)
risk_imp_mi_60 <- mi(risk_mdf, verbose = F, n.iter = 60, seed = 2013)
```

14. Plot diagnostics.

The barplots/histograms, scatter plots and residual plots of imputed values seemed consistent with the observed values on the three variables with missing data.

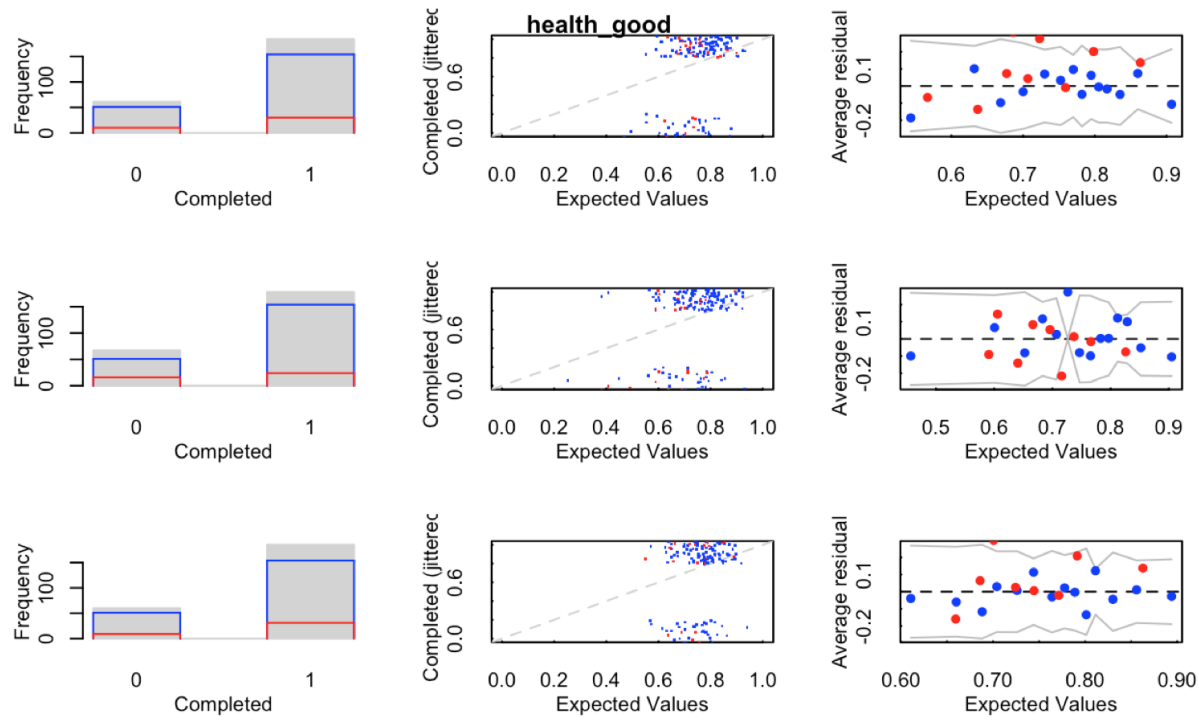
14i. Diagnostic Plot: BMI PLOT

```
include_graphics('img/diagnostic_plot_bmi.png')
```



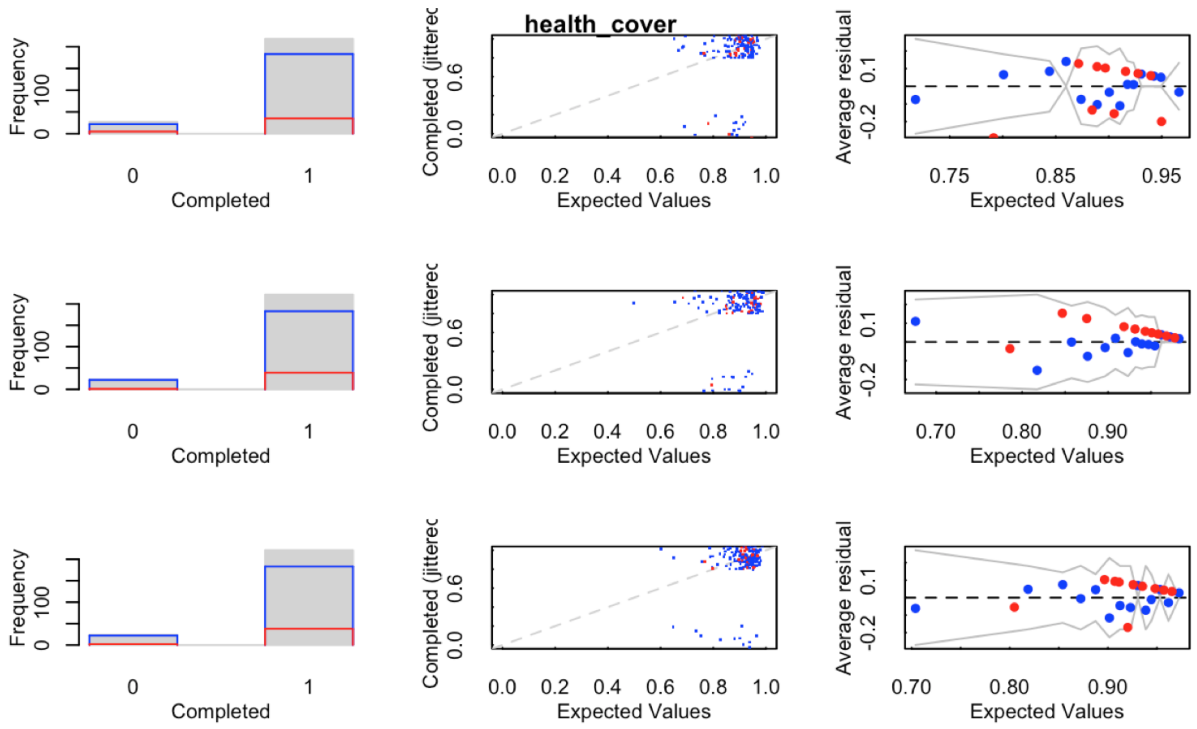
14ii. Diagnostic Plot: Health_Good

```
include_graphics('img/diagnostic_plot_health_good.png')
```



14ii. Diagnostic Plot: Health_Cover

```
include_graphics('img/diagnostic_plot_health_cover.png')
```



15. Change imputation models if necessary.

Because the imputed data from the second multiple imputation model seemed consistent with the observed data, I decided not to alter the imputation model further.

16. Run pooled analysis.

```
fit_risk_imp_mi_60 <- mi::pool(health_good ~ health_cover + age + bmi, data = risk_imp_mi_60)
display(fit_risk_imp_mi_60)
```

```
## bayesglm(formula = health_good ~ health_cover + age + bmi, data = risk_imp_mi_60)
##               coef.est coef.se
## (Intercept)    3.07     1.21
## health_cover1  0.51     0.78
## age           -0.03     0.01
## bmi           -0.03     0.03
## n = 241, k = 4
## residual deviance = 264.5, null deviance = 276.6 (difference = 12.1)
```

Part C: Summary Table

Table with results from all imputation methods.

```
prepare_table <- function(x,  model_name, vars = 4){
  require(magrittr)
  estimates <-   coef(summary(x))[1:vars, 1]  %>% t
```

```
  standard_errors <- coef(summary(x))[1:vars, 2]  %>%  t
  colnames(standard_errors) <- paste('SE', colnames(standard_errors), sep = '_')
  d <- cbind(estimates, standard_errors)
  d <- round(d, 2)
  d <- cbind(model_name, d)
  d <- as.data.frame(d)
  return(d)

}

#Get Coefs from Each Model
listwise_coefs <- prepare_table(fit_list_wise, model_name ='Listwise')
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
mean_coefs <-prepare_table(fit_imp_mean_mode, model_name = 'Mean')
random_coefs <- prepare_table(fit_imp_rand, model_name = 'Random')
dummy_coefs <- prepare_table(fit_dummy, model_name = 'Dummy')
hot_deck_coefs <- prepare_table( fit_hot_deck, model_name = 'Hot Deck')
regression_coefs <- prepare_table(fit_imp_reg, model_name = 'Regression')
stoch_coefs <- prepare_table(fit_imp_stoch, model_name = 'Stochastic')
mi_coefs <-  prepare_table( fit_risk_imp_mi_60, model_name = 'Multi. Imp.')
names(mi_coefs)[grepl(names(mi_coefs), pattern ='health_cover')] <- c("health_cover", "SE_health_cover")
full_coefs <- prepare_table(fit_full, "Full Data")

#Bind All Coefficients into One Table
summ_table <- listwise_coefs %>%
                  rbind(mean_coefs) %>%
                  rbind(random_coefs) %>%
                  rbind(dummy_coefs) %>%
                  rbind(hot_deck_coefs) %>%
                  rbind(regression_coefs) %>%
                  rbind(stoch_coefs) %>%
                  rbind(mi_coefs) %>%
                  rbind(full_coefs)

summ_table <- summ_table %>%
  select(model_name, `(Intercept)`, `SE_(Intercept)`, age, SE_age, bmi, SE_bmi,
         health_cover, SE_health_cover)

names(summ_table) <- c('Model', rep(c('Est.', 'SE'), 4))
kable(summ_table) %>%
add_header_above(c(" " = 1, "Intercept" = 2, "Age" = 2, "BMI" = 2,
                   'Health Cover' = 2)) %>%
row_spec(9, italic= T)
```

	Intercept		Age		BMI		Health Cover	
Model	Est.	SE	Est.	SE	Est.	SE	Est.	SE
Listwise	3.66	1.41	-0.05	0.02	-0.02	0.03	1.18	0.69
Mean	3.06	1.03	-0.02	0.01	-0.04	0.03	0.63	0.53
Random	2.93	0.92	-0.02	0.01	-0.03	0.02	0.51	0.47
Dummy	3.66	1.41	-0.05	0.02	-0.02	0.03	1.18	0.69
Hot Deck	2.21	0.88	-0.03	0.01	-0.01	0.02	1	0.42
Regression	2.1	0.89	-0.03	0.01	-0.03	0.02	0.93	0.48
Stochastic	3.23	0.9	-0.03	0.01	-0.05	0.02	1.15	0.46
Multi. Imp.	3.07	1.21	-0.03	0.01	-0.03	0.03	0.51	0.78
Full Data	*3.27*	*0.97*	*-0.03*	*0.01*	*-0.05*	*0.02*	*1.19*	*0.46*

Part D: Comparisons

The table above shows all the estimates from all the models including their standard errors. The last row (in *italics*) represents the 'true' estimates and standard errors from the model that was fit on the complete data set (i.e. The original dataset before I applied the ampute function). The 'stochastic' model refers to the regression with noise imputation model.

1) Intercept

Out of all the methods, stochastic imputation generated an estimate of the intercept that was closest to the intercept from the full data model(3.23 versus 3.27). Despite setting a seed, multiple imputation tended to produce different intercept estimates on different runs. Nevertheless, multiple imputation generally resulted in estimates of the intercept that were slightly lower than the full data model's estimate. The listwise deletion, dummy variable method, and stochastic models overestimated the intercept and the regression imputation, hot deck, mean imputation and random imputation models underestimated the intercept.

The listwise deletion and the dummy variable models tended to overestimate the standard errors of the intercept coefficient. The multiple imputation gave a standard error that varied quite a bit but was generally close to 1. This was close to the standard error of the intercept from the full model (0.97). All the other models tended to provide standard errors that were slightly less than the full model and were generally around 0.90.

2) Age

Age was completely observed. Consequently, most of the imputation models performed well by estimating a coefficient that was similar to the one from the model fit on the full dataset (-0.03). The only exceptions were the listwise deletion and dummy variable method models, which overestimated the effect of age (-0.05). The coefficient of the multiple imputation tended to vary between -0.02 and -0.03, which was fairly close to the coefficient estimate from the full data model. All of the models produced estimates of the standard error that were either 0.01 or 0.02, which were all fairly close to the standard error of the full data model (0.01).

3) BMI

The estimate of BMI from the stochastic model matched closely to the full data model (-0.05). The coefficient from the multiple imputation model also generally matched the coefficient from the full data model, varying between -0.04 and -0.05. The other models tended to underestimate the effect of BMI on health and produced coefficients ranging from -0.01 to -0.03. The standard errors of all the models varied betwen .02 and .03, which was pretty close to the .02 SE from the full data model.

4) Health Cover

Surprisingly, the listwise deletion method and dummy variable models resulted in estimates of health_cover that were closest to the full data model (1.18 vs. 1.19). However, the standard error of the coefficient from the listwise deletion model was higher than the standard error from the full data model (0.69 versus 0.46) The next closest estimate was the stochastic imputation model (1.15). The stochastic model also produced a standard error that matched the full data model. The other models generally underestimated the coefficient for *health_cover*. The random imputation and regression imputation models produced standard errors that were close to the original whereas the mean imputation model overestimated the standard error (.53) and the hotdecking method underestimated the standard error (.42). The multiple imputation model tended to underestimate the effect of health coverage and overestimate the standard error.

It should be noted that although the listwise deletion model produced a reasonably close estimate for the *health_cover* effect, it probably would not produce accurate predictions for *good_health*, the outcome variable, because the other coefficients were not close to the full data model. The stochastic and the multiple imputation models produced coefficients for BMI, age, and the intercept that were closer to the full data model; consequently, they would likely produce better predictions.

Note on Standard Errors The stochastic imputation model most closely matched the estimates from the full data model. The multiple imputation method produced accurate coefficients for age and bmi, but it underestimated the effect of *health_cover*. The multiple imputation method generally produced higher standard errors than the stochastic imputation model. These higher standard errors could reflect the additional uncertainty due to the imputation process. Finally, although the diagnostic plots seemed to be reasonable, increasing the number of iterations or changing the imputation method from post predictive distributions to predictive mean matching may have resulted in stabler estimates.

The listwise deletion model tended to have the highest standard errors. The lower standard errors are a result of this model being fit on only around half of the data. Having a smaller number of observations will result in a higher standard error if everything else is held equal.

Health_Cover From the barplot in section 9B, we can see that among the observed cases, most subjects had health insurance. The fact that there was very little variance on this variable may have caused the multiple imputation method to underestimate its effect.

Other Limitations In order to simplify the interpretation of the models, I dichotimized *health_general*, which was originally a 5-point ordinal variable, into a binary variable coded 1 for 'good health' and 0 for 'fair' or 'poor' health. Although this change simplified the interpretation of the coefficients of the model, some information would have been lost as the result of binning multiple categories together. This could have in turn affected the estimates of the predictor variables across all the variables.

Part E: Wow Factor

Changing Imputation Model

As I wrote in the previous section, I was concerned about the lack of stability of the estimates from the multiple imputation model across different runs. I decided to convert the imputation method used for health_cover and health_good to predictive mean matching in the hope that this might lead to stabler estimates

```
risk_mdf_wow <- change(risk_mdf, y = c("health_cover", 'health_good'), what = "imputation_method", to =
risk_imp_mi_wow <- mi(risk_mdf_wow, verbose = F, n.iter = 60, seed = 2013)
Rhats(risk_imp_mi_wow)
```

```
##          mean_bmi mean_health_cover  mean_health_good           sd_bmi
##         1.0056393         1.0073533         0.9983880         1.0088694
##   sd_health_cover    sd_health_good
##         1.0075085         0.9982524
```

Result of Changing Imputation Model The r-hats seemed to be closer to 1. However, the estimates produced by the new model still seemed to underestimate the effect of health_cover as seen below.

```
fit_risk_imp_mi_wow <- mi::pool(health_good ~ health_cover + age + bmi, data = risk_imp_mi_wow)
display(fit_risk_imp_mi_wow )
```

```
## bayesglm(formula = health_good ~ health_cover + age + bmi, data = risk_imp_mi_wow)
##               coef.est coef.se
## (Intercept)    3.12     1.38
## health_cover1  0.55     0.49
## age           -0.02     0.01
## bmi           -0.04     0.03
## n = 241, k = 4
## residual deviance = 262.4, null deviance = 273.1 (difference = 10.7)
```

Interaction

My next decision was to evaluate whether or not the model should have any interaction terms. I fit new models with interactions between *health_cover* and *age* and *bmi* and then performed likelihood ratio tests to evaluate whether the interaction terms significantly improved the model. Only the model with the *health_cover/bmi* interaction produced a statistically significant improvement over the model without interaction terms.

```
fit_full_1 <- glm(formula = health_good ~ health_cover*age + bmi,
             data = risk_full,
             family = binomial)
anova(fit_full,fit_full_1,  test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: health_good ~ health_cover + age + bmi
## Model 2: health_good ~ health_cover * age + bmi
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       230     247.88
## 2       229     245.31  1   2.5742   0.1086
```

```
fit_full_2 <- glm(formula = health_good ~ age + health_cover*bmi, data = cc(risk_sub))
anova(fit_full, fit_full_2, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: health_good ~ health_cover + age + bmi
## Model 2: health_good ~ age + health_cover * bmi
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       230    247.885
## 2       229     41.084  1    206.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing Imputation Model with New Interaction Term

I decided to compare the stochastic imputation model with the multiple imputation model using the new formula. I chose these models because stochastic imputation performed the best with the previous main terms only formula despite the fact that multiple imputation is generally considered to be superior.

Full Data Model with Interaction term

```r
fit_full_interaction <- glm(health_good ~ age + health_cover*bmi, data = risk_full,
            family = binomial)

summary(fit_full_interaction)
```

```
##
## Call:
## glm(formula = health_good ~ age + health_cover * bmi, family = binomial,
##     data = risk_full)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3811  -0.5743   0.6461   0.7788   1.3032
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.126071   1.998250   1.564  0.11772
## age              -0.028541   0.010107  -2.824  0.00475 **
## health_cover      1.364788   2.139887   0.638  0.52361
## bmi              -0.048339   0.070839  -0.682  0.49500
## health_cover:bmi -0.006412   0.075428  -0.085  0.93225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 264.27  on 233  degrees of freedom
## Residual deviance: 247.88  on 229  degrees of freedom
## AIC: 257.88
##
## Number of Fisher Scoring iterations: 4
```

Stochastic Regression with Interaction term

```r
fit_stoch_interaction <- glm(health_good ~ age + health_cover*bmi, data = risk_imp_stoch,
            family = binomial)
summary(fit_stoch_interaction)
```

```
##
## Call:
## glm(formula = health_good ~ age + health_cover * bmi, family = binomial,
##     data = risk_imp_stoch)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3100  -1.0586   0.6281   0.7967   1.3862
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.184980   1.719509   1.271 0.203835
## age              -0.032271   0.009714  -3.322 0.000894 ***
## health_cover      2.430381   1.887232   1.288 0.197815
## bmi              -0.012035   0.058113  -0.207 0.835932
## health_cover:bmi -0.044171   0.062760  -0.704 0.481555
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.41  on 244  degrees of freedom
## Residual deviance: 267.95  on 240  degrees of freedom
## AIC: 277.95
##
## Number of Fisher Scoring iterations: 4
```

Multiple Imputation with Interaction term

```
fit_mi_interaction <- mi::pool(health_good ~ age + health_cover*bmi, data = risk_imp_mi_60)
display(fit_mi_interaction )
```

```
## bayesglm(formula = health_good ~ age + health_cover * bmi, data = risk_imp_mi_60)
##                    coef.est coef.se
## (Intercept)         2.96     1.47
## age                -0.03     0.01
## health_cover1       0.62     1.34
## bmi                -0.03     0.05
## health_cover1:bmi   0.00     0.05
## n = 240, k = 5
## residual deviance = 264.4, null deviance = 276.6 (difference = 12.2)
```

Summary Table with Interaction Terms

```
full_row <- prepare_table(fit_full_interaction, model_name = 'Full Data', vars = 5)
stoch_row <- prepare_table(fit_stoch_interaction, model_name = 'Stochastic', vars = 5)
mi_row <- prepare_table(fit_mi_interaction, model_name = 'Multi. Imp.', vars = 5)
names(mi_row) <- str_replace_all(names(mi_row), pattern = 'health_cover1', replacement = 'health_cover')

summ_table_interaction <- stoch_row %>%
                            rbind(mi_row) %>%
                            rbind(full_row)

summ_table_interaction <- summ_table_interaction %>%
  select(model_name, `(Intercept)`, `SE_(Intercept)`, age, SE_age, bmi, SE_bmi,
         health_cover, SE_health_cover, `health_cover:bmi`,  `SE_health_cover:bmi`)

names(summ_table_interaction) <- c('Model', rep(c('Est.', 'SE'), 5))
kable(summ_table_interaction) %>%
add_header_above(c(" " = 1, "Intercept" = 2, "Age" = 2, "BMI" = 2,
                   'Health Cover' = 2, 'BMI:Health Cover' = 2)) %>%
row_spec(3, italic= T)
```

	Intercept		Age		BMI		Health Cover		BMI:Health Cover	
Model	Est.	SE	Est.	SE	Est.	SE	Est.	SE	Est.	SE
Stochastic	2.18	1.72	-0.03	0.01	-0.01	0.06	2.43	1.89	-0.04	0.06
Multi. Imp.	2.96	1.47	-0.03	0.01	-0.03	0.05	0.62	1.34	0	0.05
Full Data	*3.13*	*2*	*-0.03*	*0.01*	*-0.05*	*0.07*	*1.36*	*2.14*	*-0.01*	*0.08*

Comparing Models with Interaction Terms

Using the new formula with the interaction term between *bmi* and *health_cover*, the multiple imputation method generally produced estimates closer to the model fit on the full dataset.

Since the interaction model was a significant improvement over the main terms only model, it seems likely that the best method of analysis would have been to use the interaction model and then use multiple imputation for the missing values.

One Final Note The original dataset was almost entirely complete. However, 4% of the BMI values were missing. It is possible that these missing values could affect the model if they were MNAR (Missing Not At Random).