NHANES Data Analysis Project

STAT 420, Summer 2023, Preeti Agrawal, Thimira Bandara, Michael Conlin, Constatin Kappel

2023-07-26

Title

Introduction

Methods

Data Import

```
if (!require(NHANES)) {
  install.packages("NHANES", quiet = TRUE)
}
```

```
library(NHANES)
# head(NHANES)
```

Initial Variable Selection

Rule out variables by reasoning or by exploratory analysis We have chosen `BMI` (Body mass index (weight/height2 in kg/m2)) as our response variable. In NHANES, this data is reported for participants aged 2 years or older, so we will focus on those participants for our analysis. Provided below are all the variables in NHANES, along with our response variable `BMI`.

```
sort(names(NHANES)) # alphabetic order
```

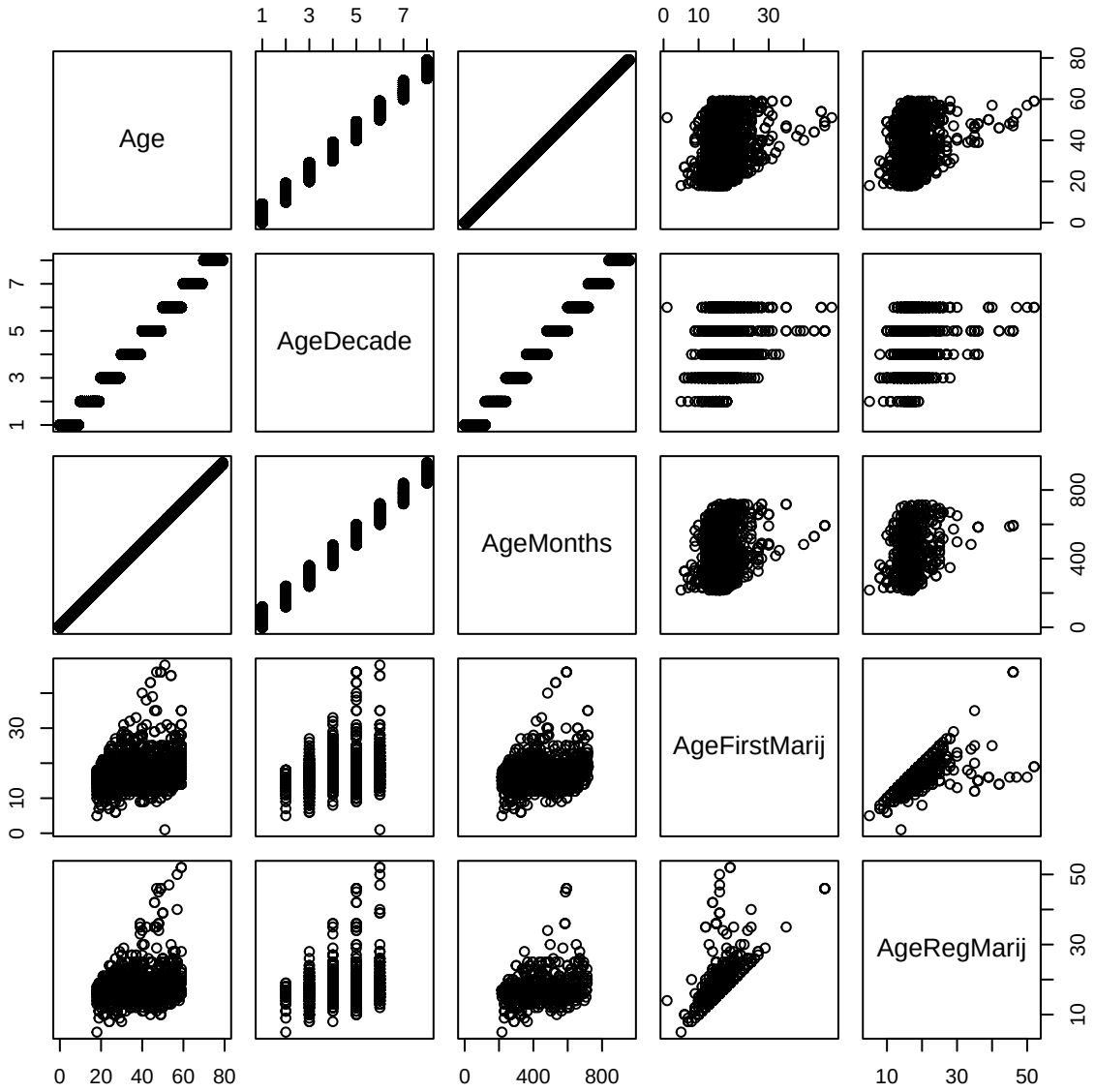
```
##  [1] "Age"              "Age1stBaby"       "AgeDecade"        "AgeFirstMarij"
##  [5] "AgeMonths"        "AgeRegMarij"      "Alcohol12PlusYr"  "AlcoholDay"
##  [9] "AlcoholYear"      "BMI"              "BMI_WHO"          "BMICatUnder20yrs"
## [13] "BPDia1"           "BPDia2"           "BPDia3"           "BPDiaAve"
## [17] "BPSys1"           "BPSys2"           "BPSys3"           "BPSysAve"
## [21] "CompHrsDay"       "CompHrsDayChild"  "DaysMentHlthBad"  "DaysPhysHlthBad"
## [25] "Depressed"        "Diabetes"         "DiabetesAge"      "DirectChol"
## [29] "Education"        "Gender"           "HardDrugs"        "HeadCirc"
## [33] "HealthGen"        "Height"           "HHIncome"         "HHIncomeMid"
## [37] "HomeOwn"          "HomeRooms"        "ID"               "Length"
## [41] "LittleInterest"   "Marijuana"        "MaritalStatus"    "nBabies"
## [45] "nPregnancies"     "PhysActive"       "PhysActiveDays"   "Poverty"
## [49] "PregnantNow"      "Pulse"            "Race1"            "Race3"
## [53] "RegularMarij"     "SameSex"          "SexAge"           "SexEver"
```

```
## [57] "SexNumPartnLife"  "SexNumPartYear"   "SexOrientation"   "SleepHrsNight"
## [61] "SleepTrouble"     "Smoke100"         "Smoke100n"        "SmokeAge"
## [65] "SmokeNow"         "SurveyYr"         "Testosterone"     "TotChol"
## [69] "TVHrsDay"         "TVHrsDayChild"    "UrineFlow1"       "UrineFlow2"
## [73] "UrineVol1"        "UrineVol2"        "Weight"           "Work"
```

We will add all omitted variables to a data frame `df_exclude`. The variables we would like to use as predictors will be kept in a dataframe `df_keep`. The following is our reasoning for ruling out or keeping certain variables as predictors:

1. Some predictors can be ruled out right away. Our response variable is `BMI`, so we should not use body `Weight` or `Height` as predictors, because `BMI` is calculated by dividing the `Weight` by `Height`.
2. The next group of predictors seems very closely related either by name or logic deduction, for example, age related variables such as `Age`, `AgeDecade`, `AgeMonths`. Let's quickly double-check if they are linearly related:

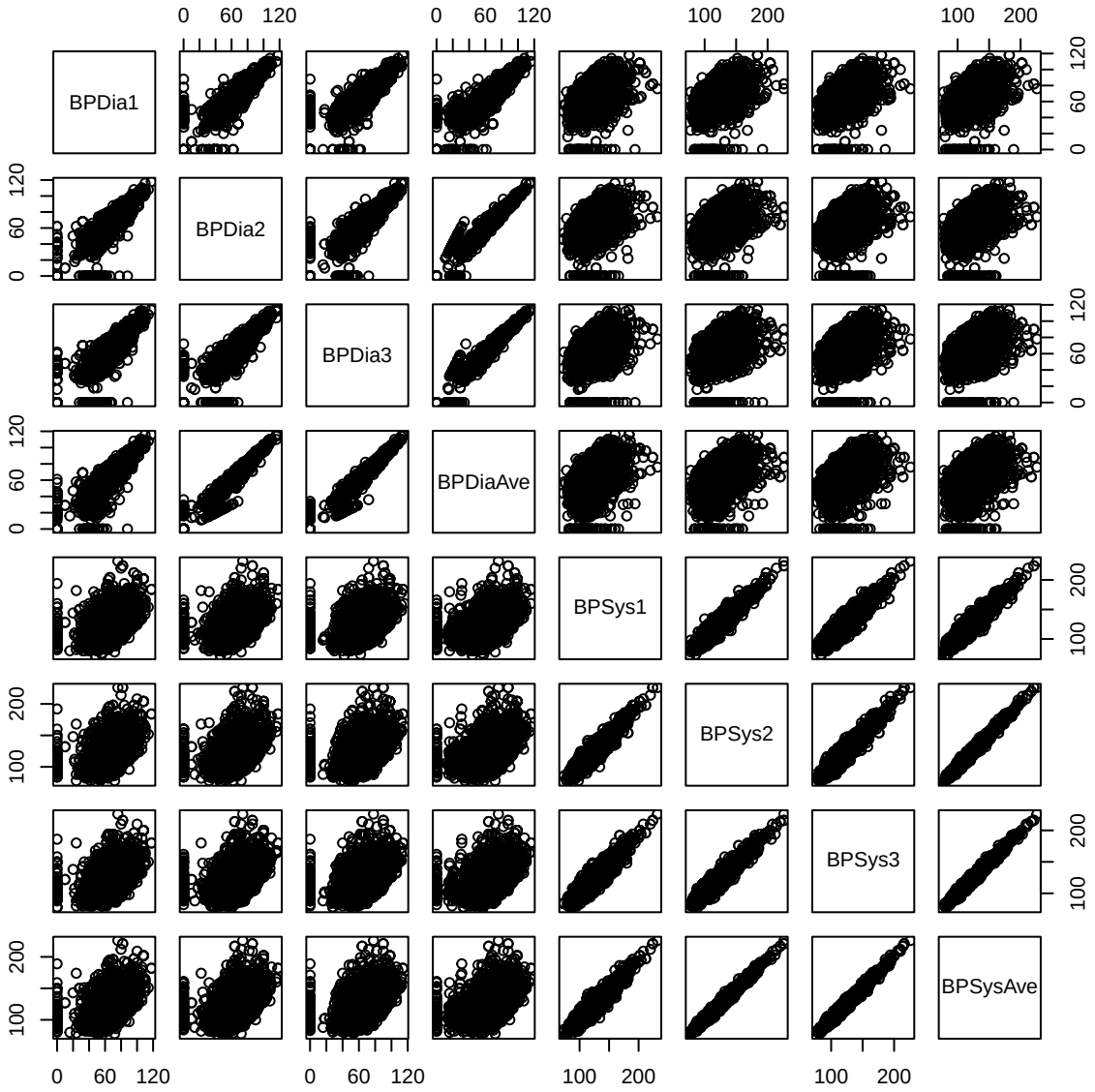
```
pairs(subset(NHANES, select = c('Age', 'AgeDecade', 'AgeMonths',
                                'AgeFirstMarij', 'AgeRegMarij')))
```



`Age`, `AgeDecade` and `AgeMonth` are clearly collinear, so we will only keep `Age`. Likewise, both variables for Marijuana use appear collinear, so we keep only one, say `AgeRegMarij` and we may decide to drop it later if it is not useful.

3. Now let's check for collinearity between different blood pressure related variables:

```
to_test = c("BPDia1", "BPDia2", "BPDia3", "BPDiaAve", "BPSys1", "BPSys2", "BPSys3", "BPSysAve" )
pairs(subset(NHANES, select=to_test))
```



The blood pressure variables fall into two groups: diastolic and systolic blood pressure readings. We would expect there to be strongly collinearity within each group, which is the case. So, we only keep the average in each group `BPDiaAve` and `BPSysAve`.

Refer to the Appendix for the `pairs()` plots assessment of the following variables' collinearity:

4. Let's check all variables related to alcohol: We again performed a `pairs()` plot to visualize possible collinearity, and this graph is in the Appendix. Collinearity is not as clear in this case, but we believe one predictor related to alcohol consumption may be sufficient. We will keep `AlcoholYear`.

5. Let's now investigate the collinearity of other drug-related variables: Most of these predictors are categorical, so collinearity cannot be seen, except for `SmokeAge` and `AgeRegMarij`. The latter makes sense as this drug is usually consumed via smoking. We can thus use one as a proxy for the other.

(Note: `AgeRegMarij` was in the age related group above as well and we kept it). Let's keep `SmokeNow` and `HardDrugs` as proxies for drug abuse and its potential effect on BMI.

6. Next, let's investigate a few life-style variables related to being physically active or the opposite thereof, screen time: Due to the nature of these variables being categorical, a clear picture of collinearity is not observable. Let's keep half of these parameters for now, which are the ones with a bit denser levels, `PhysActiveDays`, `TVHrsDay`, `CompHrsDay`.

7. Now let's look into some other health related variables, such as cholesterol and diabetes related predictors: `DirectChol` and `TotChol` appear to be collinear, let's keep `TotChol`. Out of the diabetes related ones, we keep `Diabetes`.

8. Let's analyze more health related variables, such as those related to urine volume and flow below: Urine volume and urine flow appear collinear. Moreover, there might be collinearity between the first and second urine measurement, respectively. Let's keep `UrineVol1` for now.

9. Next we analyze a somewhat heterogenic group of variables related to health or mental health. For example, somebody who is depressed might show little interest in doing things. Again, collinearity is not easy to spot in categorical variables. Let's pick `LittleInterest` as a mild form of mental health issue which might lead to little physical activity and obesity, and `HealthGen` as a general health rating.

10. We decided to keep `Poverty` which is a ratio of family income to poverty guidelines, and drop `HHIncomeMid` and `HHIncome`, as they both capture similar information to what the `Poverty` variable captures. Similarly, we chose to keep `Race1` instead of `Race3` as they both capture similar information, and `Race1` has more data compared to `Race3`.

11. Finally, let's add `Poverty`, `SleepHrsNight`, `Gender`, `Race1`, `Education`, and `MartialStatus` as we believe they can have an effect on BMI, and we do not suspect collinearity.

```
#Setting up the data frames with the variables we will be excluding and keeping for model building

df_exclude = data.frame(predictor = c('Weight', 'Height', 'Age1stBaby', 'AgeDecade', 'AgeMonth',
                                      'AgeRegMarij', 'Alcohol12PlusYr', 'AlcoholDay', 'Smoke100',
                                      'SmokeAge', 'Marijuana', 'RegularMarij', "BPDia1", "BPDia2",
                                      "BPDia3", "BPSys1", "BPSys2", "BPSys3", 'PhysActive',
                                      'TVHrsDayChild', 'CompHrsDayChild', 'DirectChol',
                                      'DiabetesAge', "UrineFlow1", "UrineVol2", "UrineFlow2",
                                      "DaysPhysHlthBad", "DaysMentHlthBad", "Depressed", "Race3",
                                      "nPregnancies"),
reason_to_omit = c('linear dependence with BMI','linear dependence with BMI', 'specific by Gender',
                   'collinear with Age', 'collinear with Age',
                   'redundant with Marijuana', 'more sparse than AlcoholYear', 'redundant with
                   AlcoholYear', 'redundant with SmokeNow', 'collinear with AgeRegMarij',
                   'redundant with AgeRegMarij, the two might be swapped', 'redundant with Marijuana',
                   'collinear with other blood pressure predictors', 'collinear with other blood
                   pressure predictors', 'collinear with other blood pressure predictors', 'collinear
                   with other blood pressure predictors', 'collinear with other blood pressure
                   predictors', 'collinear with other blood pressure predictors', 'Redundant with
                   PhysActiveDays', 'redundant with TVHrsDay', 'redundant with CompHrsDay', 'collinear
                   with TotChol', 'redundant with Diabetes', 'collinear with UrineVol1', 'collinear
                   with UrineVol1','collinear with UrineVol1', 'redundant with HealthGen', 'redundant
                   with HealthGen','redundant with HealthGen', 'redundant with Race1', 'specific by
                   Gender'))

df_keep = data.frame(predictor = c('Age', 'AlcoholYear', 'Marijuana', 'SmokeNow', 'HardDrugs',
                                   'BPDiaAve', 'BPSysAve', 'PhysActiveDays', 'TVHrsDay',
```

Table 1: Initial Predictors Selected

Predictor	Predictor
Age	Diabetes
AlcoholYear	UrineVol1
Marijuana	HealthGen
SmokeNow	LittleInterest
HardDrugs	Poverty
BPDiaAve	SleepHrsNight
BPSysAve	Gender
PhysActiveDays	Race1
TVHrsDay	Education
CompHrsDay	MaritalStatus
TotChol	

```
                                'CompHrsDay', 'TotChol', 'Diabetes', 'UrineVol1', 'HealthGen',
                                'LittleInterest', 'Poverty', 'SleepHrsNight', 'Gender',
                                'Race1', 'Education', 'MaritalStatus' ))

opts <- options(knitr.kable.NA = "")
knitr::kable(list(df_keep[1:11,], df_keep[12:22,]), caption = "Initial Predictors Selected",
             col.names = "Predictor", booktabs = TRUE)
```

Next, let's build a dataset `nhanes_select` using just the above `df_keep` variables.

```
nhanes_select = subset(NHANES, select =c(df_keep$predictor, "BMI"))
```

The resulting dataset, after the initial variable selection above, consists of 10000 observations (rows) and 22 variables (columns) including `BMI` and the chosen predictors.

Convert Categorical Variables into Factor Variables

We will now convert the categorical predictors into factors.

```
nhanes_select$Marijuana = as.factor(nhanes_select$Marijuana)
nhanes_select$SmokeNow = as.factor(nhanes_select$SmokeNow)
nhanes_select$HardDrugs = as.factor(nhanes_select$HardDrugs)
nhanes_select$Diabetes = as.factor(nhanes_select$Diabetes)
nhanes_select$TVHrsDay = as.factor(nhanes_select$TVHrsDay)
nhanes_select$CompHrsDay = as.factor(nhanes_select$CompHrsDay)
nhanes_select$HealthGen = as.factor(nhanes_select$HealthGen)
nhanes_select$LittleInterest = as.factor(nhanes_select$LittleInterest)
nhanes_select$Gender = as.factor(nhanes_select$Gender)
nhanes_select$Race1 = as.factor(nhanes_select$Race1)
nhanes_select$Education = as.factor(nhanes_select$Education)
nhanes_select$MaritalStatus = as.factor(nhanes_select$MaritalStatus)
```

Address Missing Values It would be helpful to have a dataset which is devoid of NAs (missing values) before we conduct our regression analysis. First let's get a quick idea of how many missing values are present in our initial dataset.

Identify which variables have majority Nan values

```
library(tidyverse, quietly = TRUE)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Count the NA values in each column
na_counts = colSums(is.na(nhanes_select))

# Calculate the percentage of NA values in each column
total_rows = nrow(nhanes_select)
na_percentage = (na_counts / total_rows) * 100

# Create a data frame to store the results
na_summary = data.frame(Column = names(na_counts), NA_Count = na_counts, NA_Percentage = na_percentage)
na_summary = na_summary %>%
  arrange(desc(NA_Percentage))

# Print the summary
print(na_summary)
```

```
##                         Column NA_Count NA_Percentage
## SmokeNow             SmokeNow     6789         67.89
## PhysActiveDays PhysActiveDays     5337         53.37
## TVHrsDay             TVHrsDay     5141         51.41
## CompHrsDay         CompHrsDay     5137         51.37
## Marijuana           Marijuana     5059         50.59
## HardDrugs           HardDrugs     4235         42.35
## AlcoholYear       AlcoholYear     4078         40.78
## LittleInterest LittleInterest     3333         33.33
## Education           Education     2779         27.79
## MaritalStatus   MaritalStatus     2769         27.69
## HealthGen           HealthGen     2461         24.61
## SleepHrsNight   SleepHrsNight     2245         22.45
## TotChol               TotChol     1526         15.26
## BPDiaAve             BPDiaAve     1449         14.49
## BPSysAve             BPSysAve     1449         14.49
## UrineVol1           UrineVol1      987          9.87
## Poverty               Poverty      726          7.26
## BMI                       BMI      366          3.66
## Diabetes             Diabetes      142          1.42
## Age                       Age        0          0.00
## Gender                 Gender        0          0.00
## Race1                   Race1        0          0.00
```

The table above is sorted according to NA percentage in descending order. The top 5 predictors as far as NAs are concerned are: `SmokeNow`, `PhysActiveDays`, `TVHrsDay`, `CompHrsDay` and `Marijuana`. Half of all predictors have greater than 25% missing values. If we eliminated all rows with any missing value, we would be left with only 419, which is not enough observations to be meaningful. We cannot simply proceed using this data, as any regression tools we will use will need to eliminate many observations in order to proceed with the statistical calculations. It would also be inappropriate to simply eliminate these observations, although this was previously the standard approach. Eliminating this many observations would bring into question how well our study models represent the underlying population. Interpretation of our results would become more difficult, and suspicious of selective observation elimination introducing bias. This data was also costly to produce - we prefer to not simply cast it aside. We therefore decided to perform data imputation for the missing data.

Data imputation involves the substitution of missing data with a different value. Although there are simple methods of replacing missing values with the mean or median of the variable in question, the most robust method is multiple imputation. Multiple imputation involves the generation of multiple complete datasets by replacing the missing values with data values which are modeled for each missing entry, from a plausible distribution. The imputation process can use a variety of methods for computing the imputed values, depending upon the underlying distribution of the observed values, and the relationship of those observed values and the other variables in the observation. Once the multiple complete datasets are generated, any analysis can be performed (such as linear regression) and the results of each analyses are pooled into one set of results.

We will perform the multiple imputation process with the `mice` package below. More information regarding the `mice` package can be read at the book website Flexible Imputation of Missing Data

Data Imputation with the `mice` Package

```
if (!require(mice)) {
  install.packages("mice", quiet = TRUE)
}
```

Here we will perform the imputation. Given the size of the data, this will take a bit of processing time. First we will remove the observations where there is no entry for BMI as there are only 366 such observations, to avoid imputation of our response variable.

```
library(mice, quietly = TRUE)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
# remove the rows which have NAs for BMI
nhanes_imp = nhanes_select[!is.na(nhanes_select$BMI), ]

# perform the multiple imputation (5 datasets)
imp = mice(nhanes_imp, seed = 420, m = 5, print = FALSE)
```

See Appendix for density plots comparing the imputed and observed values.

Initial Model Building and Diagnostics

Now that imputation is complete to address the missing data, we will build a complete additive model, to allow an initial diagnostic evaluation.

```
# perform the linear regression with each of the 5 imputed datasets
fit_add <- with(imp, lm(BMI ~  Age +  AlcoholYear + Marijuana + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus))

summary(fit_add$analyses[[1]]) #summary of the 1st imputed dataset
```

```
##
## Call:
## lm(formula = BMI ~ Age + AlcoholYear + Marijuana + SmokeNow +
##     HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay +
##     CompHrsDay + TotChol + Diabetes + UrineVol1 + HealthGen +
##     LittleInterest + Poverty + SleepHrsNight + Gender + Race1 +
##     Education + MaritalStatus)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17.72  -4.02  -0.66   3.08  51.07
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                15.971309   0.902813   17.69  < 2e-16 ***
## Age                         0.068493   0.004923   13.91  < 2e-16 ***
## AlcoholYear                -0.007182   0.000682  -10.53  < 2e-16 ***
## MarijuanaYes               -0.110896   0.141892   -0.78  0.43450
## SmokeNowYes                -1.844623   0.145042  -12.72  < 2e-16 ***
## HardDrugsYes               -0.535036   0.179859   -2.97  0.00294 **
## BPDiaAve                    0.076354   0.004811   15.87  < 2e-16 ***
## BPSysAve                    0.028586   0.004629    6.18  6.9e-10 ***
## PhysActiveDays             -0.120430   0.033572   -3.59  0.00034 ***
## TVHrsDay0_to_1_hr          -1.028120   0.441569   -2.33  0.01992 *
## TVHrsDay1_hr               -1.466664   0.432008   -3.39  0.00069 ***
## TVHrsDay2_hr               -0.138588   0.422957   -0.33  0.74317
## TVHrsDay3_hr                0.820815   0.432167    1.90  0.05755 .
## TVHrsDay4_hr                0.137412   0.448163    0.31  0.75915
## TVHrsDayMore_4_hr           0.135049   0.443332    0.30  0.76066
## CompHrsDay0_to_1_hr         0.756341   0.186713    4.05  5.1e-05 ***
## CompHrsDay1_hr              2.080135   0.203571   10.22  < 2e-16 ***
## CompHrsDay2_hr              1.995214   0.232239    8.59  < 2e-16 ***
## CompHrsDay3_hr              2.278443   0.273593    8.33  < 2e-16 ***
## CompHrsDay4_hr              3.198275   0.376541    8.49  < 2e-16 ***
## CompHrsDayMore_4_hr         5.487766   0.315182   17.41  < 2e-16 ***
## TotChol                     0.194370   0.064065    3.03  0.00242 **
## DiabetesYes                 2.191698   0.248486    8.82  < 2e-16 ***
## UrineVol1                   0.001975   0.000695    2.84  0.00447 **
```

```
## HealthGenVgood             1.522558   0.202334    7.52  5.7e-14 ***
## HealthGenGood              3.432174   0.204019   16.82  < 2e-16 ***
## HealthGenFair              4.319456   0.261867   16.49  < 2e-16 ***
## HealthGenPoor              5.911734   0.468975   12.61  < 2e-16 ***
## LittleInterestSeveral      0.312386   0.165730    1.88  0.05947 .
## LittleInterestMost        -0.374333   0.263538   -1.42  0.15552
## Poverty                   -0.068833   0.044977   -1.53  0.12595
## SleepHrsNight             -0.152794   0.047284   -3.23  0.00124 **
## Gendermale                -0.232813   0.129759   -1.79  0.07281 .
## Race1Hispanic             -1.352504   0.311397   -4.34  1.4e-05 ***
## Race1Mexican              -0.506982   0.279392   -1.81  0.06962 .
## Race1White                -1.187517   0.203948   -5.82  6.0e-09 ***
## Race1Other                -2.874504   0.286073  -10.05  < 2e-16 ***
## Education9 - 11th Grade   -0.018180   0.322924   -0.06  0.95510
## EducationHigh School       0.253344   0.310195    0.82  0.41411
## EducationSome College      0.098083   0.310173    0.32  0.75184
## EducationCollege Grad     -0.388556   0.329384   -1.18  0.23817
## MaritalStatusLivePartner  -0.551224   0.314469   -1.75  0.07966 .
## MaritalStatusMarried      -0.383803   0.244461   -1.57  0.11645
## MaritalStatusNeverMarried -1.556719   0.272387   -5.72  1.1e-08 ***
## MaritalStatusSeparated     0.315622   0.469454    0.67  0.50140
## MaritalStatusWidowed      -2.033354   0.377985   -5.38  7.6e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.98 on 9588 degrees of freedom
## Multiple R-squared:  0.345,  Adjusted R-squared:  0.342
## F-statistic:  112 on 45 and 9588 DF,  p-value: <2e-16
```

We will next construct a dataframe of all of our 5 imputed datasets, with the additional values added of columns `.imp` for the imputation number, and `.i` for the observation number within that imputation.

```
imp_df = mice::complete(imp, action = "long")
```

Collinearity When we built the additive model above, a few parameters had large p-values. Let's check the variance inflation factors for all the predictors in this model, to see if there is any effect of collinearity on the variance of our regression estimates.

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

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

```
car::vif(fit_add$analyses[[1]])
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## Age             3.099  1           1.760
## AlcoholYear     1.175  1           1.084
## Marijuana       1.325  1           1.151
## SmokeNow        1.414  1           1.189
## HardDrugs       1.233  1           1.110
## BPDiaAve        1.371  1           1.171
## BPSysAve        1.708  1           1.307
## PhysActiveDays 1.040  1           1.020
## TVHrsDay        1.460  6           1.032
## CompHrsDay      1.492  6           1.034
## TotChol         1.259  1           1.122
## Diabetes        1.191  1           1.091
## UrineVol1       1.064  1           1.031
## HealthGen       1.427  4           1.045
## LittleInterest 1.159  2           1.038
## Poverty         1.537  1           1.240
## SleepHrsNight   1.065  1           1.032
## Gender          1.133  1           1.064
## Race1           1.586  4           1.059
## Education       1.899  4           1.083
## MaritalStatus   2.432  5           1.093
```

None of the variable appear to have a large (>5) variance inflation factor which is good to see.

Variance and Normality Assessment Now let's do some tests on this model to identify any potential issues.

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

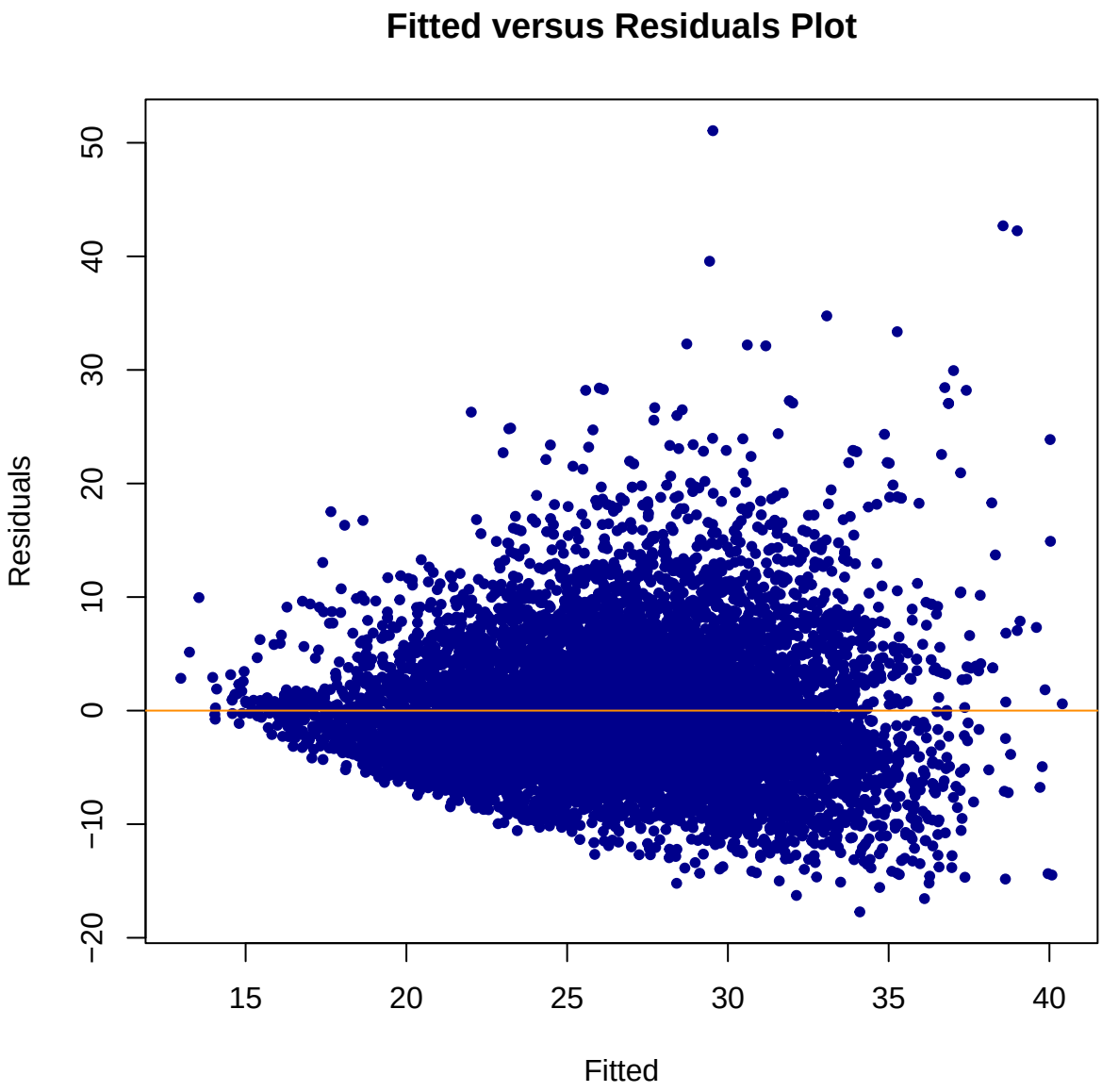
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
if (!require(nortest)) {
  install.packages("nortest", quiet = TRUE)
}
```

```
## Loading required package: nortest
```

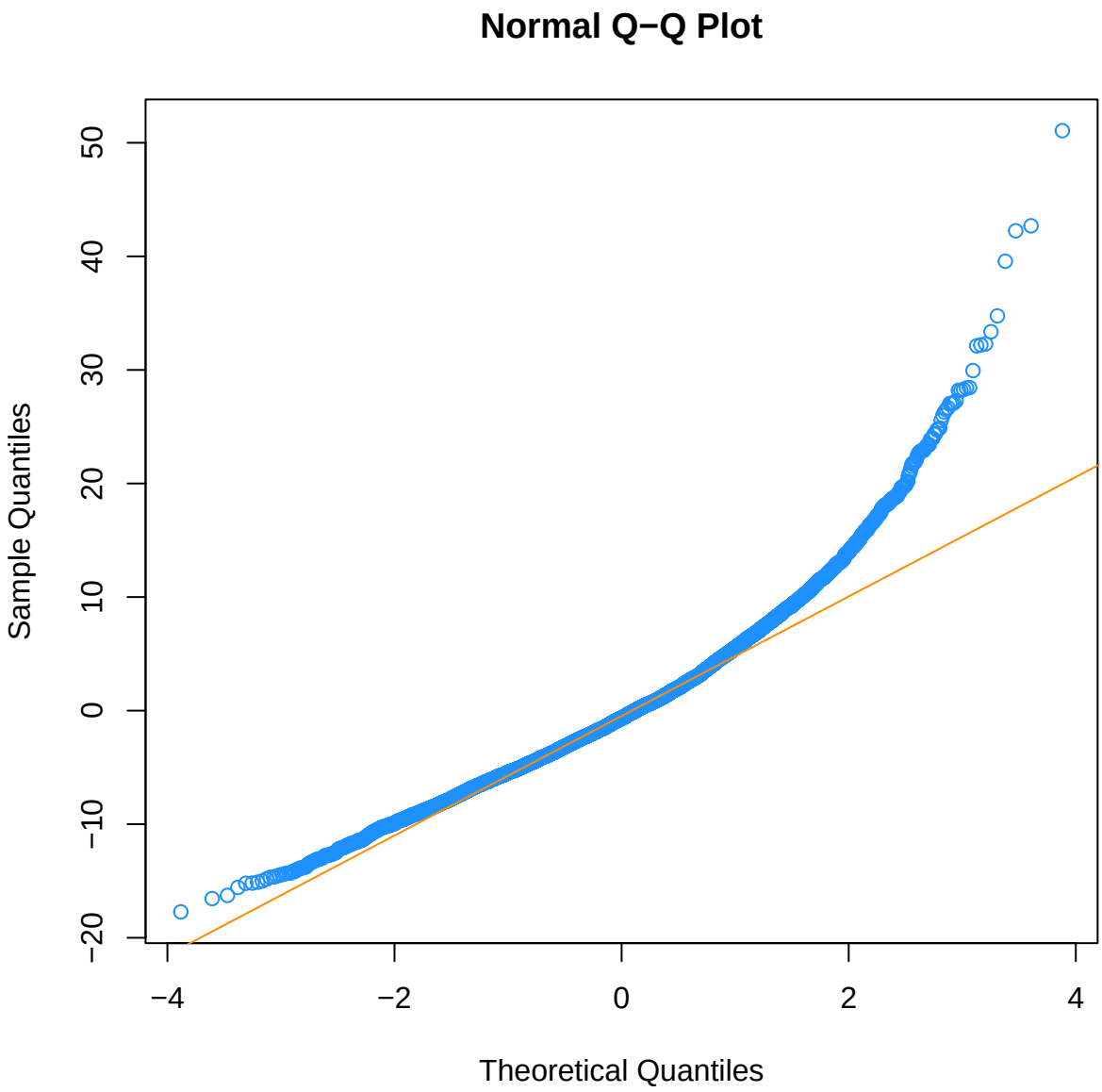
```r
# install.packages("nortest", quiet = TRUE)
library(nortest, quietly = TRUE)

### First, let's define some functions ###

# Function to calculate the LOOCVRMSE
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

# model diagnostics
model_diagnostics = function(fit){
  fit_summary <- data.frame(bptest_p = rep(0,5), ad_test = rep(0,5))
  for (i in 1:5){
    fit_summary$bptest_p[i] = unname(bptest(fit$analyses[[i]])$p.value)
    ad.test(residuals(fit$analyses[[i]]))$p.value
  }
  knitr::kable(fit_summary, col.names = c("BP Test", "AD Test"))
}

# cooks distance to check for influential observations
cooks_function = function(fit){
  cook_thresh = 4 * cooks.distance(fit$analyses[[1]]) / nrow(imp)
  mean(cooks.distance(fit$analyses[[1]]) > cook_thresh)
}

# model assessments
model_assess = function(fit){
  fit_summary <- data.frame(adj_r_squared = rep(0,5), loocv_rmse = rep(0,5))
  for (i in 1:5){
    fit_summary$adj_r_squared[i] = summary(fit$analyses[[i]])$adj
    fit_summary$loocv_rmse[i] = calc_loocv_rmse(fit$analyses[[i]])
  }
  knitr::kable(fit_summary, col.names = c("Adj. R-Squared", "LOOCV-RMSE"))
}
```

```r
#Fitted versus Residuals Plot for the 1st imputed dataset model
plot(fitted(fit_add$analyses[[1]]), resid(fit_add$analyses[[1]]), col = "darkblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals Plot")
abline(h=0,col = "darkorange")
```



The Fitted versus Residuals plot reveals deviation from homoscedasticity (constant variance).

```
#Normal Q-Q Plot for the 1st imputed dataset model
qqnorm(resid(fit_add$analyses[[1]]), col = "dodgerblue")
qqline(resid(fit_add$analyses[[1]]), col = "darkorange")
```



The Q-Q-Plot also shows deviations from normality.

Let's now look at the p-values from the AD Test for normality, and the Breusch-Pagan Test for Homoscedasticity.

```
model_diagnostics(fit_add)
```

BP Test	AD Test
0	0
0	0
0	0
0	0
0	0

The p-values for these tests, using each of the 5 imputed dataset models, are all very low, essentially 0. So we reject the null hypothesis, calling into question, both normality and homoscedasticity. However, both of these tests are susceptible to the influence of large sample sizes, so they may be less reliable in this setting.

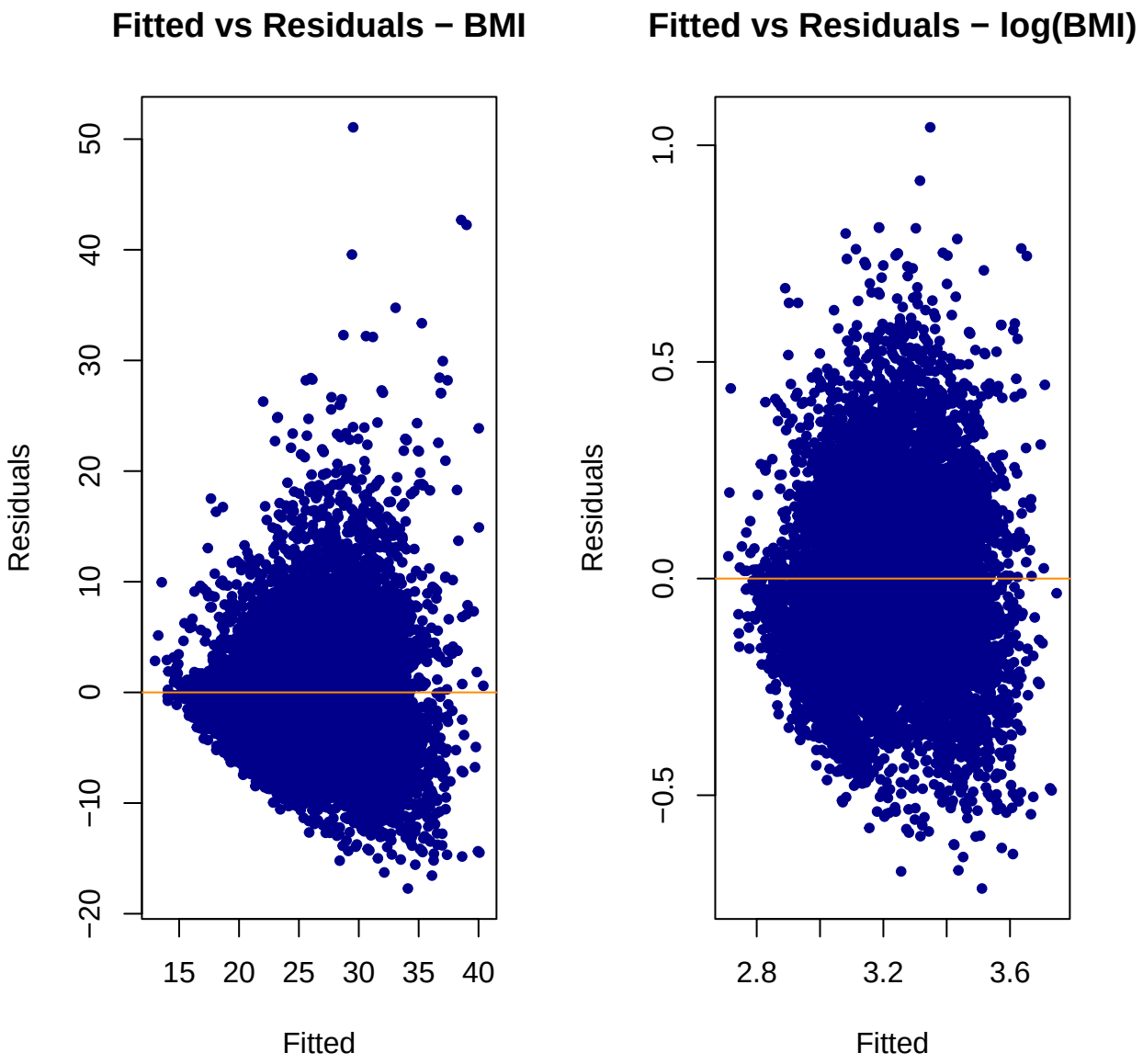
Because of the findings above, we will perform a variance stabilizing log transformation on the response variable (BMI), fit the model again and reassess the diagnostics.

```
# perform the linear regression with each of the 5 imputed datasets
# and the log() transform of BMI
fit_add_log <- with(imp, lm(log(BMI) ~  Age +  AlcoholYear + Marijuana + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus))

summary(fit_add_log$analyses[[1]])
```

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + Marijuana + SmokeNow +
##     HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay +
##     CompHrsDay + TotChol + Diabetes + UrineVol1 + HealthGen +
##     LittleInterest + Poverty + SleepHrsNight + Gender + Race1 +
##     Education + MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7153 -0.1488 -0.0075  0.1338  1.0414
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.8000078  0.0321729   87.03  < 2e-16 ***
## Age                        0.0034561  0.0001755   19.70  < 2e-16 ***
## AlcoholYear               -0.0002533  0.0000243  -10.42  < 2e-16 ***
## MarijuanaYes              -0.0037485  0.0050565   -0.74  0.45852
## SmokeNowYes               -0.0648351  0.0051688  -12.54  < 2e-16 ***
## HardDrugsYes              -0.0173073  0.0064095   -2.70  0.00694 **
## BPDiaAve                   0.0029675  0.0001715   17.31  < 2e-16 ***
## BPSysAve                   0.0009464  0.0001650    5.74  9.9e-09 ***
## PhysActiveDays            -0.0048322  0.0011964   -4.04  5.4e-05 ***
## TVHrsDay0_to_1_hr         -0.0360859  0.0157359   -2.29  0.02186 *
## TVHrsDay1_hr              -0.0547425  0.0153952   -3.56  0.00038 ***
## TVHrsDay2_hr              -0.0047818  0.0150726   -0.32  0.75106
## TVHrsDay3_hr               0.0255240  0.0154008    1.66  0.09749 .
## TVHrsDay4_hr               0.0031905  0.0159708    0.20  0.84166
## TVHrsDayMore_4_hr          0.0000724  0.0157987    0.00  0.99634
## CompHrsDay0_to_1_hr        0.0311888  0.0066537    4.69  2.8e-06 ***
## CompHrsDay1_hr             0.0802491  0.0072545   11.06  < 2e-16 ***
## CompHrsDay2_hr             0.0805117  0.0082761    9.73  < 2e-16 ***
## CompHrsDay3_hr             0.0882696  0.0097498    9.05  < 2e-16 ***
## CompHrsDay4_hr             0.1226157  0.0134185    9.14  < 2e-16 ***
## CompHrsDayMore_4_hr        0.1859430  0.0112319   16.55  < 2e-16 ***
## TotChol                    0.0087585  0.0022830    3.84  0.00013 ***
## DiabetesYes                0.0569458  0.0088551    6.43  1.3e-10 ***
## UrineVol1                  0.0001056  0.0000248    4.27  2.0e-05 ***
```

```
## HealthGenVgood             0.0599624  0.0072104    8.32  < 2e-16 ***
## HealthGenGood              0.1262192  0.0072705   17.36  < 2e-16 ***
## HealthGenFair              0.1545469  0.0093319   16.56  < 2e-16 ***
## HealthGenPoor              0.1970503  0.0167125   11.79  < 2e-16 ***
## LittleInterestSeveral      0.0092696  0.0059060    1.57  0.11656
## LittleInterestMost        -0.0096256  0.0093915   -1.02  0.30542
## Poverty                   -0.0028051  0.0016028   -1.75  0.08012 .
## SleepHrsNight             -0.0058481  0.0016850   -3.47  0.00052 ***
## Gendermale                -0.0010097  0.0046241   -0.22  0.82715
## Race1Hispanic             -0.0359625  0.0110970   -3.24  0.00120 **
## Race1Mexican              -0.0012320  0.0099565   -0.12  0.90153
## Race1White                -0.0401927  0.0072679   -5.53  3.3e-08 ***
## Race1Other                -0.0984992  0.0101946   -9.66  < 2e-16 ***
## Education9 - 11th Grade   -0.0020877  0.0115078   -0.18  0.85604
## EducationHigh School       0.0081584  0.0110542    0.74  0.46051
## EducationSome College      0.0040085  0.0110534    0.36  0.71688
## EducationCollege Grad     -0.0112692  0.0117380   -0.96  0.33705
## MaritalStatusLivePartner  -0.0160915  0.0112065   -1.44  0.15106
## MaritalStatusMarried      -0.0118043  0.0087117   -1.36  0.17545
## MaritalStatusNeverMarried -0.0611841  0.0097069   -6.30  3.0e-10 ***
## MaritalStatusSeparated     0.0096622  0.0167296    0.58  0.56358
## MaritalStatusWidowed      -0.0798839  0.0134700   -5.93  3.1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 9588 degrees of freedom
## Multiple R-squared:  0.389,  Adjusted R-squared:  0.386
## F-statistic:  136 on 45 and 9588 DF,  p-value: <2e-16
```

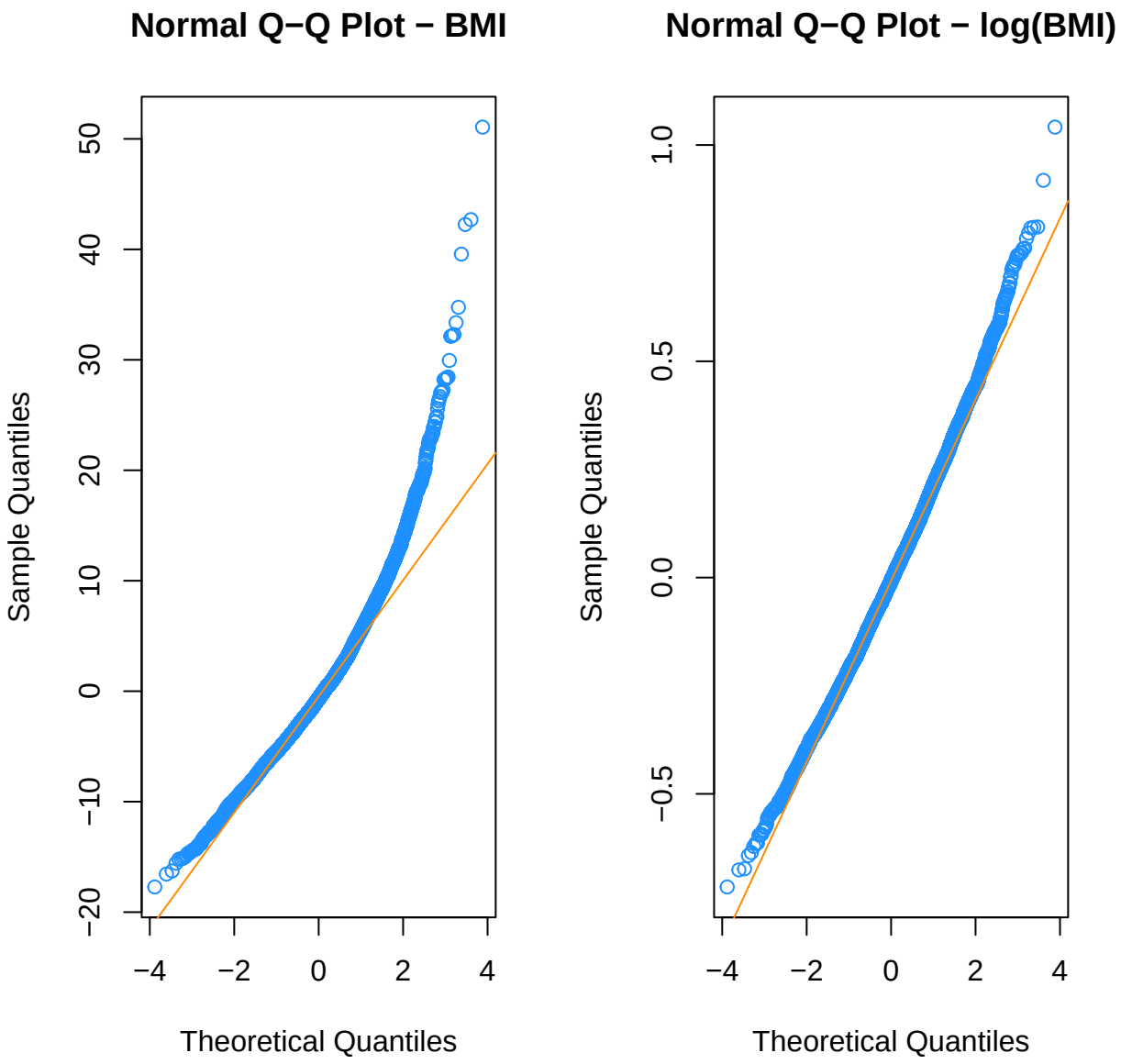
```r
#Comparing the Fitted versus Residuals Plots of the Initial Additive model and the Log(BMI) Transformat
par(mfrow=c(1,2))
plot(fitted(fit_add$analyses[[1]]), resid(fit_add$analyses[[1]]), col = "darkblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residuals - BMI")
abline(h=0,col = "darkorange")
plot(fitted(fit_add_log$analyses[[1]]), resid(fit_add_log$analyses[[1]]), col = "darkblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residuals - log(BMI)")
abline(h=0,col = "darkorange")
```



The log transformation of BMI model looks much better, though still not perfect.

Now let's look at the Q-Q plots:

```
#Comparing the Normal Q-Q Plots of the Initial Additive model and the Log(BMI) Transformation Model
par(mfrow=c(1,2))
# no transformation
qqnorm(resid(fit_add$analyses[[1]]), col = "dodgerblue", main = "Normal Q-Q Plot - BMI")
qqline(resid(fit_add$analyses[[1]]), col = "darkorange")
# log transformation
qqnorm(resid(fit_add_log$analyses[[1]]), col = "dodgerblue", main = "Normal Q-Q Plot - log(BMI)")
qqline(resid(fit_add_log$analyses[[1]]), col = "darkorange")
```



Again, the log transformation of BMI results in a much better appearing QQ plot. Moving forward, we will use the log transformed BMI for our model building.

Model Selection Now we use the different search procedures, backwards, forwards, and stepwise to search for models and select predictors. Notice that our 5 datasets with observed and imputed data are passed to the stepwise function using `with()` which in this case returns a `mira` object from the `mice` package.

First we will start with the additive model and perform a backward AIC model search.

```
# build the stepwise workflow
scope <- list(upper = ~ Age +  AlcoholYear + Marijuana + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus,
              lower = ~ 1)
```

```
expr <- expression(f1 <- lm(log(BMI) ~ 1),
                   f2 <- step(f1, scope = scope, trace = 0))
# perform the stepwise selection with each of the 5 imputed datasets
fit <- with(imp, expr)

# count the votes for variables to keep
formulas <- lapply(fit$analyses, formula)
terms <- lapply(formulas, terms)
votes <- unlist(lapply(terms, labels))
table(votes)
```

```
## votes
##            Age    AlcoholYear       BPDiaAve       BPSysAve     CompHrsDay
##              5              5              5              5              5
##       Diabetes      Education      HardDrugs      HealthGen LittleInterest
##              5              4              5              5              4
##      Marijuana  MaritalStatus PhysActiveDays        Poverty          Race1
##              1              5              5              3              5
##  SleepHrsNight       SmokeNow        TotChol       TVHrsDay      UrineVol1
##              5              5              5              5              5
```

If we use the criterion of more than half of the datasets resulted in selection of a variable, we end up only dropping Marijuana. Let's compare the models using anova, leaving out variables with less than 5 votes.

```
# remove Education
model_without = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
model_with = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus))
anova(model_without, model_with)
```

```
##    test statistic df1   df2 dfcom p.value    riv
##  2 ~~ 1     1.645   4 104.1  9589  0.1686 0.4589
```

This p-value is not significant, so we fail to reject the null hypothesis and we can discard `Education`.

```
# remove LittleInterest
model_without = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen + Poverty +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
model_with = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
anova(model_without, model_with)
```

```
##    test statistic df1   df2 dfcom p.value    riv
##  2 ~~ 1     1.862   2 18.38  9593  0.1835 0.8327
```

Again, we fail to reject the null hypothesis based on the p-value, and can remove `LittleInterest`.

```
# remove Poverty
model_without = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
model_with = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen + Poverty +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
anova(model_without, model_with)
```

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     5.658   1   4  9595  0.0761 0.1411
```

This p-value is getting close to 0.05. We should consider dropping `Poverty`, but for now we will keep it.

Here is the final model of this process which we will call `fit_add_aic`

```
fit_add_aic = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen + Poverty +
  SleepHrsNight +  Gender +  Race1 +  MaritalStatus))
summary(fit_add_aic$analyses[[1]])
```

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + HardDrugs +
##     BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
##     TotChol + Diabetes + UrineVol1 + HealthGen + Poverty + SleepHrsNight +
##     Gender + Race1 + MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7075 -0.1483 -0.0081  0.1345  1.0531
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.7998629  0.0307142   91.16  < 2e-16 ***
## Age                        0.0034141  0.0001745   19.56  < 2e-16 ***
## AlcoholYear               -0.0002615  0.0000239  -10.93  < 2e-16 ***
## SmokeNowYes               -0.0636024  0.0051182  -12.43  < 2e-16 ***
## HardDrugsYes              -0.0175517  0.0060359   -2.91  0.00365 **
## BPDiaAve                   0.0029479  0.0001709   17.25  < 2e-16 ***
## BPSysAve                   0.0009912  0.0001644    6.03  1.7e-09 ***
## PhysActiveDays            -0.0050202  0.0011918   -4.21  2.5e-05 ***
## TVHrsDay0_to_1_hr         -0.0379522  0.0156881   -2.42  0.01557 *
## TVHrsDay1_hr              -0.0561498  0.0153620   -3.66  0.00026 ***
## TVHrsDay2_hr              -0.0045574  0.0150447   -0.30  0.76196
## TVHrsDay3_hr               0.0262886  0.0153949    1.71  0.08774 .
## TVHrsDay4_hr               0.0046249  0.0159340    0.29  0.77163
## TVHrsDayMore_4_hr          0.0015405  0.0157845    0.10  0.92225
```

```
## CompHrsDay0_to_1_hr        0.0298719  0.0065714    4.55  5.5e-06 ***
## CompHrsDay1_hr             0.0796477  0.0071133   11.20  < 2e-16 ***
## CompHrsDay2_hr             0.0788360  0.0081333    9.69  < 2e-16 ***
## CompHrsDay3_hr             0.0859230  0.0096117    8.94  < 2e-16 ***
## CompHrsDay4_hr             0.1202162  0.0132678    9.06  < 2e-16 ***
## CompHrsDayMore_4_hr        0.1840787  0.0110627   16.64  < 2e-16 ***
## TotChol                    0.0090033  0.0022788    3.95  7.8e-05 ***
## DiabetesYes                0.0567895  0.0088489    6.42  1.4e-10 ***
## UrineVol1                  0.0001047  0.0000248    4.23  2.4e-05 ***
## HealthGenVgood             0.0607387  0.0071743    8.47  < 2e-16 ***
## HealthGenGood              0.1282799  0.0072059   17.80  < 2e-16 ***
## HealthGenFair              0.1565532  0.0091507   17.11  < 2e-16 ***
## HealthGenPoor              0.1966292  0.0164770   11.93  < 2e-16 ***
## Poverty                   -0.0038076  0.0015188   -2.51  0.01219 *
## SleepHrsNight             -0.0059988  0.0016825   -3.57  0.00037 ***
## Gendermale                -0.0012345  0.0046001   -0.27  0.78843
## Race1Hispanic             -0.0345086  0.0109893   -3.14  0.00169 **
## Race1Mexican              -0.0005764  0.0096893   -0.06  0.95256
## Race1White                -0.0398583  0.0072545   -5.49  4.0e-08 ***
## Race1Other                -0.0996015  0.0100864   -9.87  < 2e-16 ***
## MaritalStatusLivePartner  -0.0172976  0.0111960   -1.54  0.12239
## MaritalStatusMarried      -0.0124965  0.0086863   -1.44  0.15028
## MaritalStatusNeverMarried -0.0622401  0.0097018   -6.42  1.5e-10 ***
## MaritalStatusSeparated     0.0099329  0.0167216    0.59  0.55252
## MaritalStatusWidowed      -0.0787033  0.0133598   -5.89  4.0e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 9595 degrees of freedom
## Multiple R-squared:  0.388,  Adjusted R-squared:  0.386
## F-statistic:  160 on 38 and 9595 DF,  p-value: <2e-16
```

Let's try a forward search using BIC, and see if we get a smaller model:

```
# build the stepwise workflow
scope <- list(upper = ~ Age +  AlcoholYear + Marijuana + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus,
              lower = ~ 1)
expr <- expression(f1 <- lm(log(BMI) ~ 1),
                   f2 <- step(f1, scope = scope, direction = "forward",
                              K = log(nrow(imp[["data"]])), trace = 0))
# perform the stepwise selection with each of the 5 imputed datasets
fit <- with(imp, expr)

# count the votes for variables to keep
formulas <- lapply(fit$analyses, formula)
terms <- lapply(formulas, terms)
votes <- unlist(lapply(terms, labels))
table(votes)
```

```
## votes
##            Age     AlcoholYear       BPDiaAve       BPSysAve     CompHrsDay
```

```
##             5             5             5             5             5
##      Diabetes     Education     HardDrugs     HealthGen LittleInterest
##             5             4             5             5             4
##     Marijuana MaritalStatus PhysActiveDays       Poverty         Race1
##             1             5             5             3             5
##  SleepHrsNight     SmokeNow      TotChol     TVHrsDay     UrineVol1
##             5             5             5             5             5
```

This appears to yield the same votes as the prior method.

Lastly, let's try a Stepwise seach in both directions using AIC.

```
# build the stepwise workflow
scope <- list(upper = ~ Age +  AlcoholYear + Marijuana + SmokeNow +
  HardDrugs + BPDiaAve + BPSysAve + PhysActiveDays + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest + Poverty +
  SleepHrsNight +  Gender +  Race1 +  Education +  MaritalStatus,
              lower = ~ 1)
expr <- expression(f1 <- lm(log(BMI) ~ 1),
                   f2 <- step(f1, scope = scope, direction = "both",
                              trace = 0))
# perform the stepwise selection with each of the 5 imputed datasets
fit <- with(imp, expr)

# count the votes for variables to keep
formulas <- lapply(fit$analyses, formula)
terms <- lapply(formulas, terms)
votes <- unlist(lapply(terms, labels))
table(votes)
```

```
## votes
##             Age     AlcoholYear       BPDiaAve       BPSysAve     CompHrsDay
##               5               5              5              5              5
##       Diabetes      Education      HardDrugs      HealthGen LittleInterest
##               5               4              5              5              4
##      Marijuana  MaritalStatus PhysActiveDays        Poverty          Race1
##               1               5              5              3              5
##   SleepHrsNight       SmokeNow        TotChol       TVHrsDay      UrineVol1
##               5               5              5              5              5
```

Still same results. For now, our additive model will be `fit_add_aic`

Model Diagnostics

Data Interactions

Data Transformations

Outlier Assessment

Final Model

Results

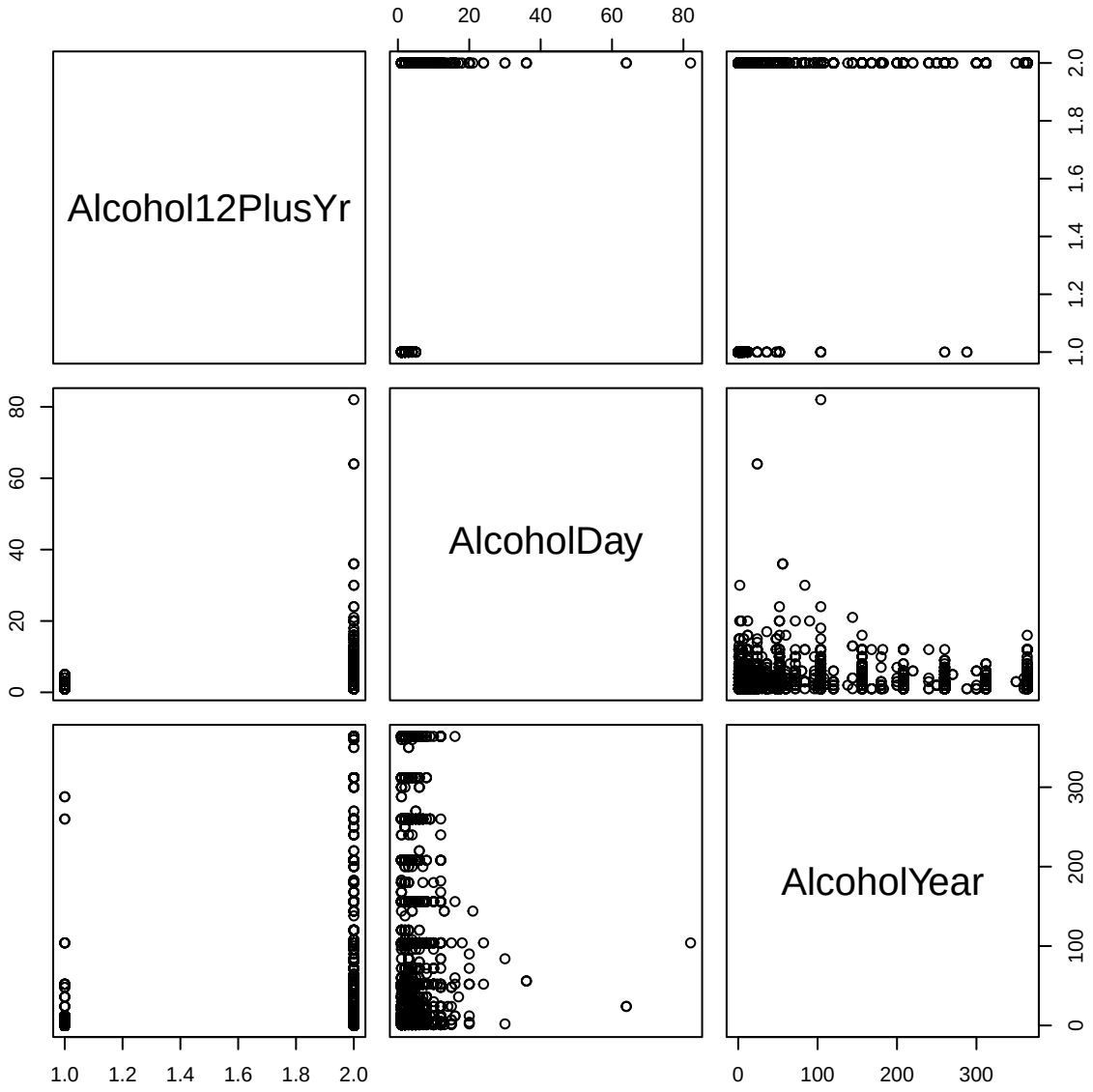
Discussion

Appendix

Variable Selection

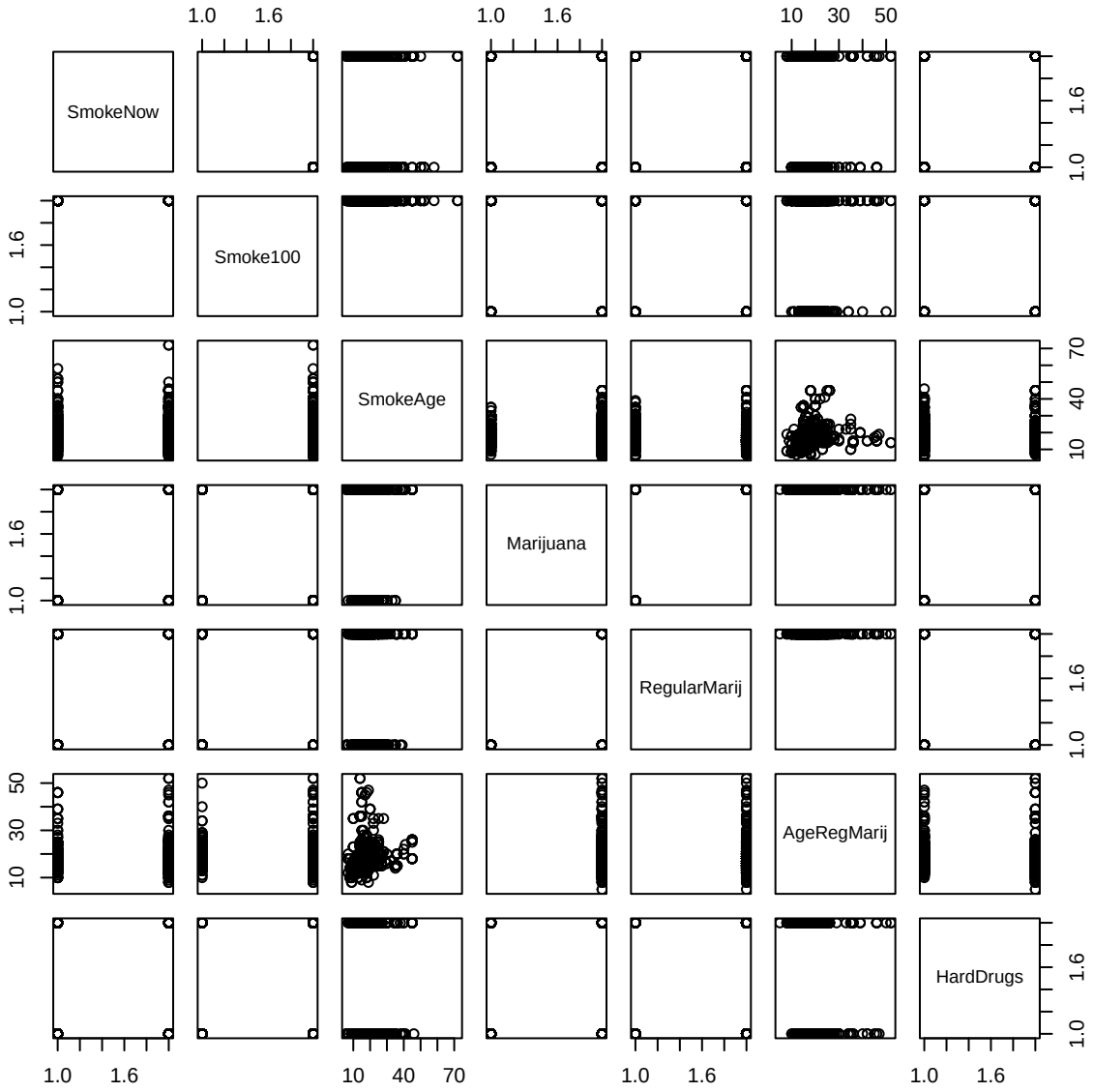
Additional `pairs()` Plots for Collinearity Assessment Alcohol related variables:

```
to_test = c("Alcohol12PlusYr", "AlcoholDay", "AlcoholYear")
pairs(subset(NHANES, select = to_test))
```



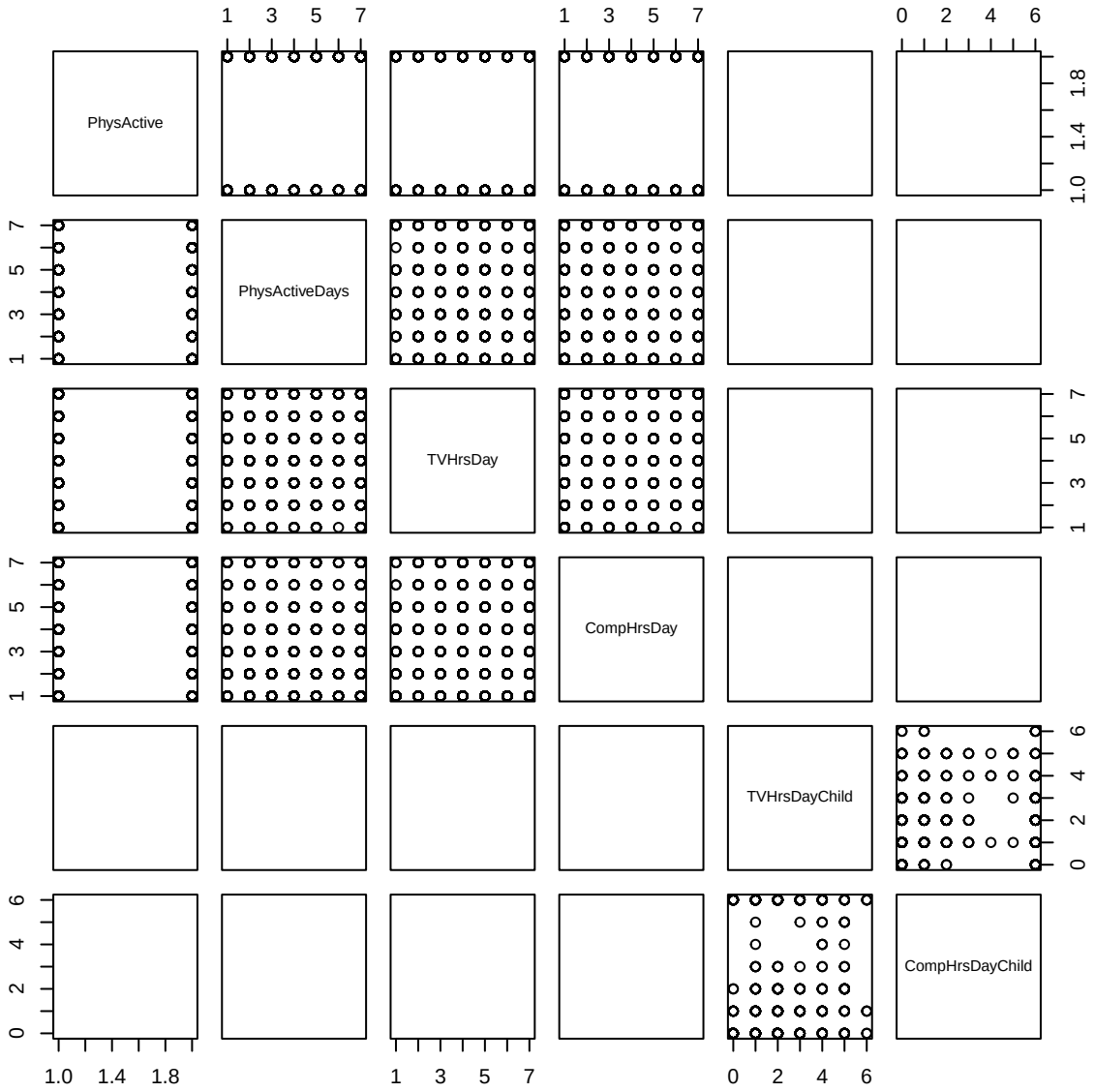
Smoking and Drug related variables:

```
to_test = c("SmokeNow", "Smoke100", "SmokeAge", "Marijuana", "RegularMarij", "AgeRegMarij", "HardDrugs"
pairs(subset(NHANES, select = to_test))
```



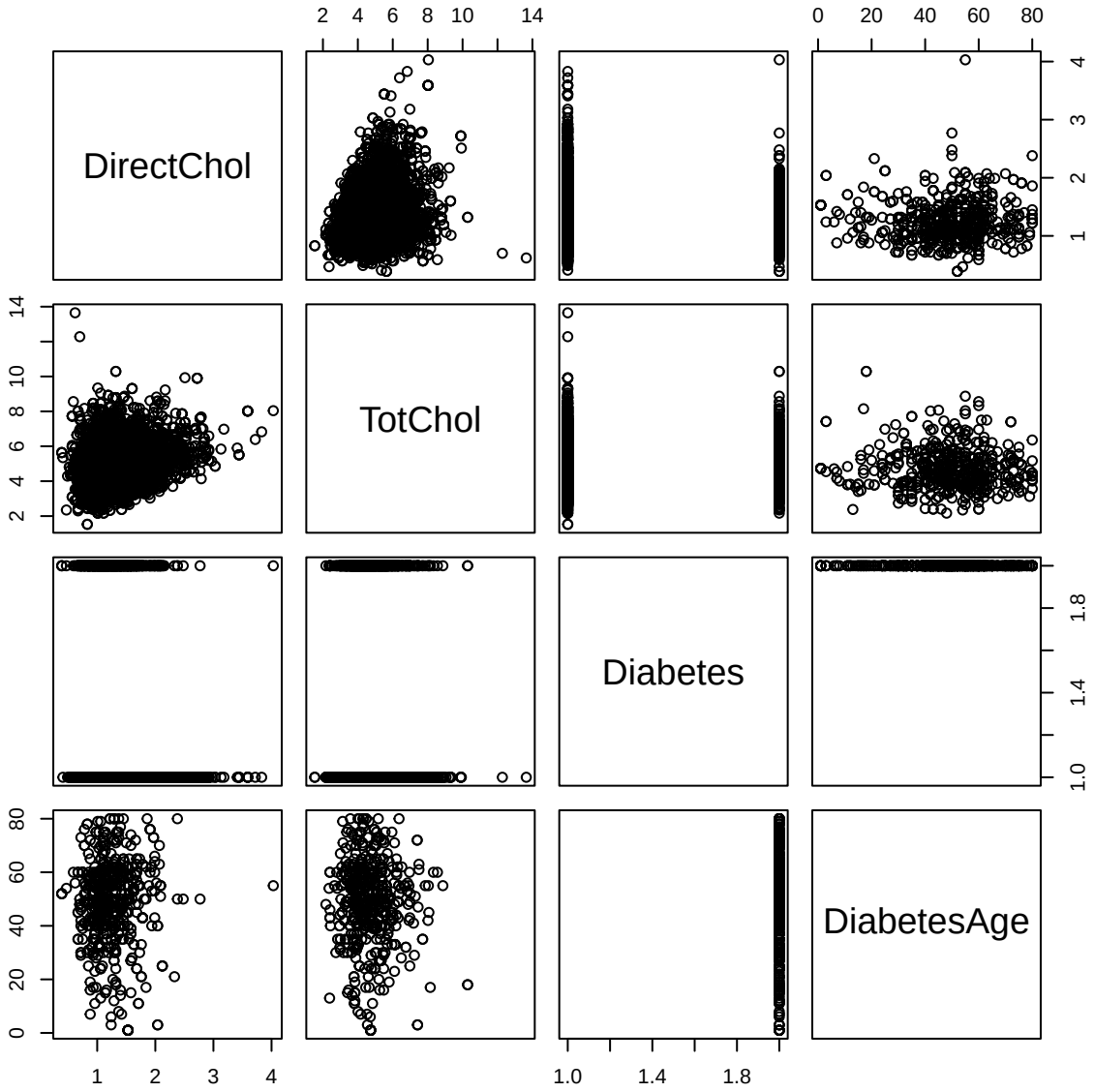
Lifestyle related variables:

```
to_test = c("PhysActive", "PhysActiveDays", "TVHrsDay", "CompHrsDay", "TVHrsDayChild", "CompHrsDayChild
pairs(subset(NHANES, select=to_test))
```



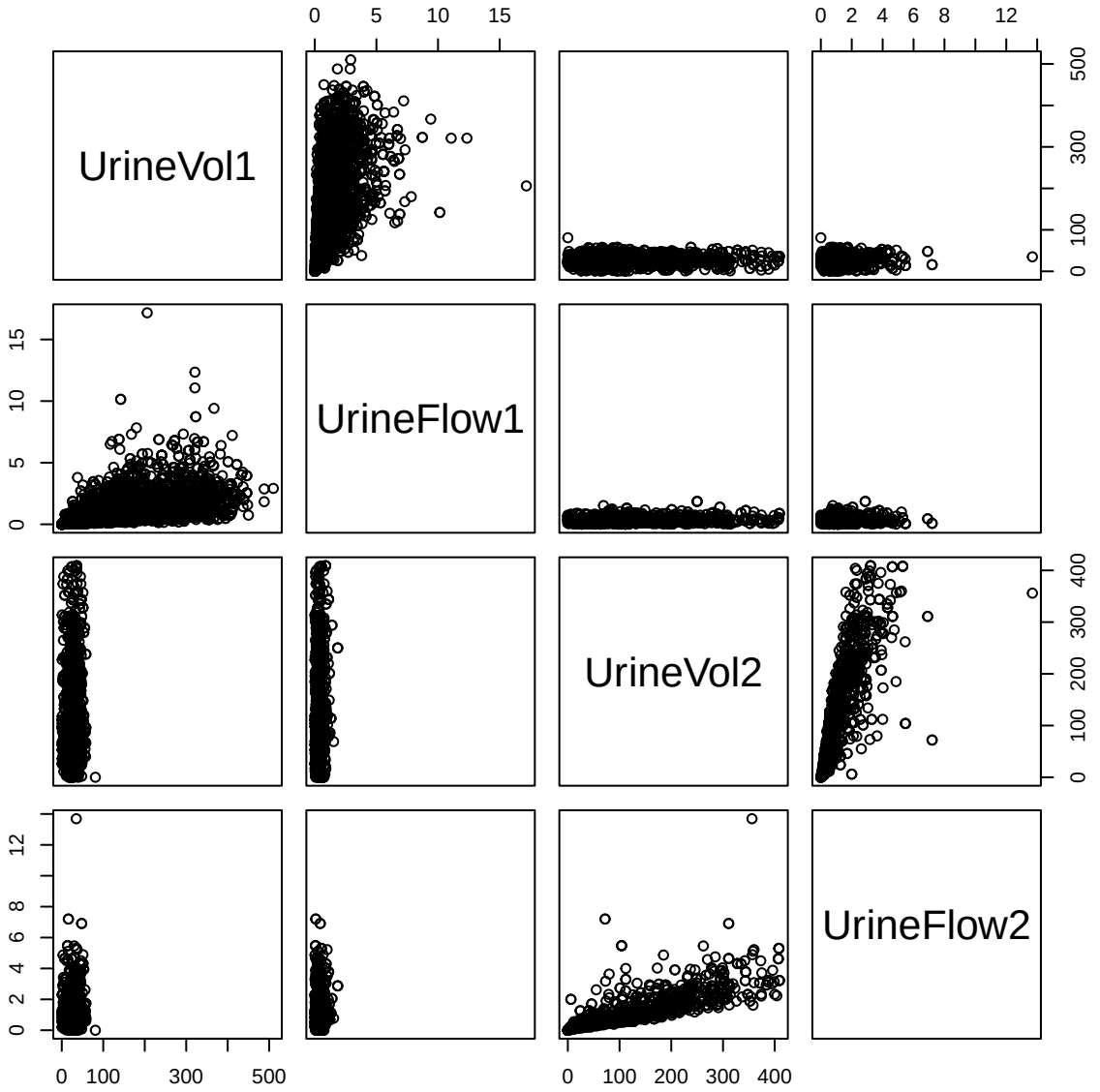
Cholesterol related variables:

```
to_test = c("DirectChol", "TotChol", "Diabetes", "DiabetesAge")
pairs(subset(NHANES, select = to_test))
```



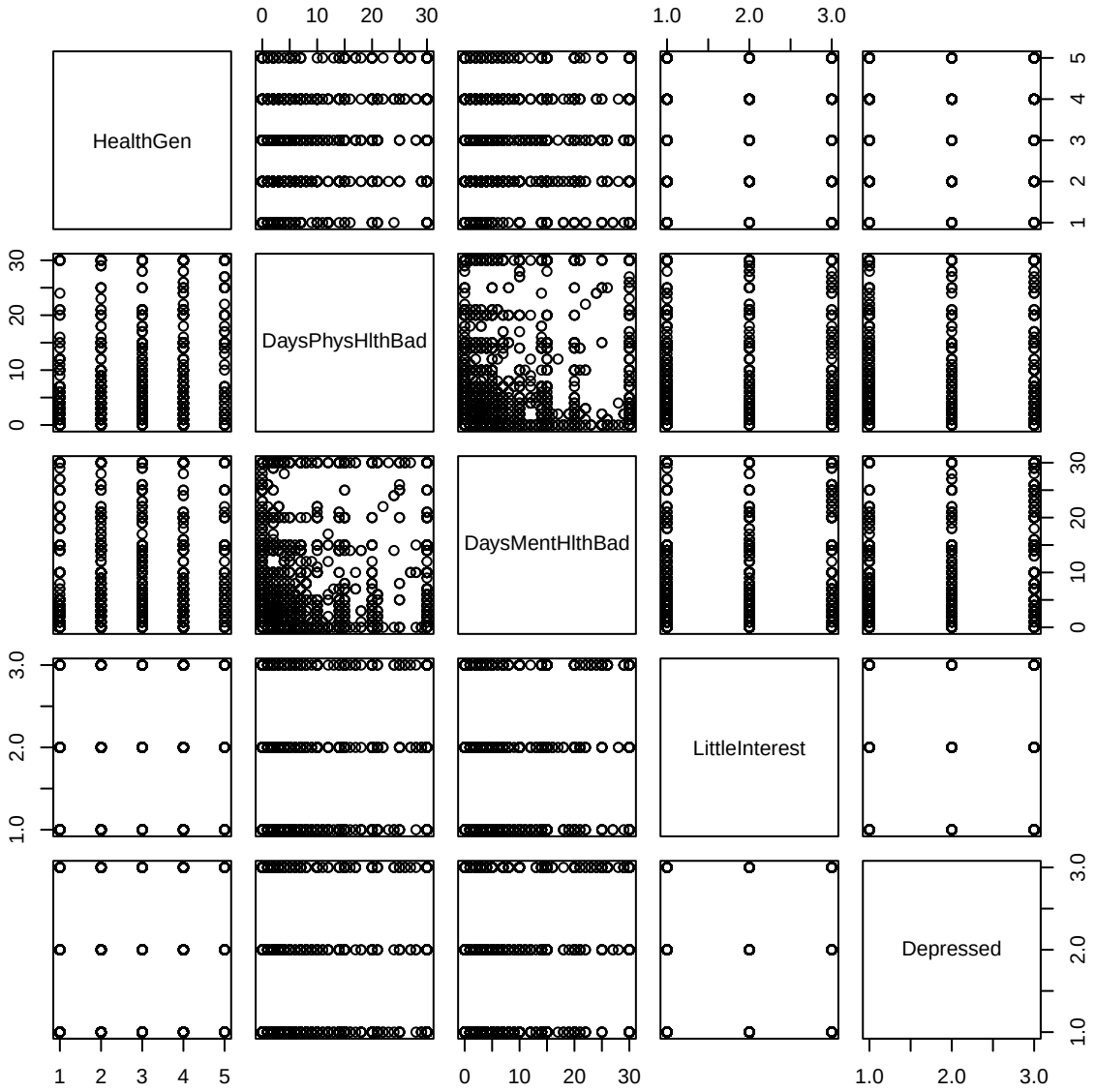
Urine related variables:

```
to_test = c("UrineVol1", "UrineFlow1", "UrineVol2", "UrineFlow2")
pairs(subset(NHANES, select = to_test))
```

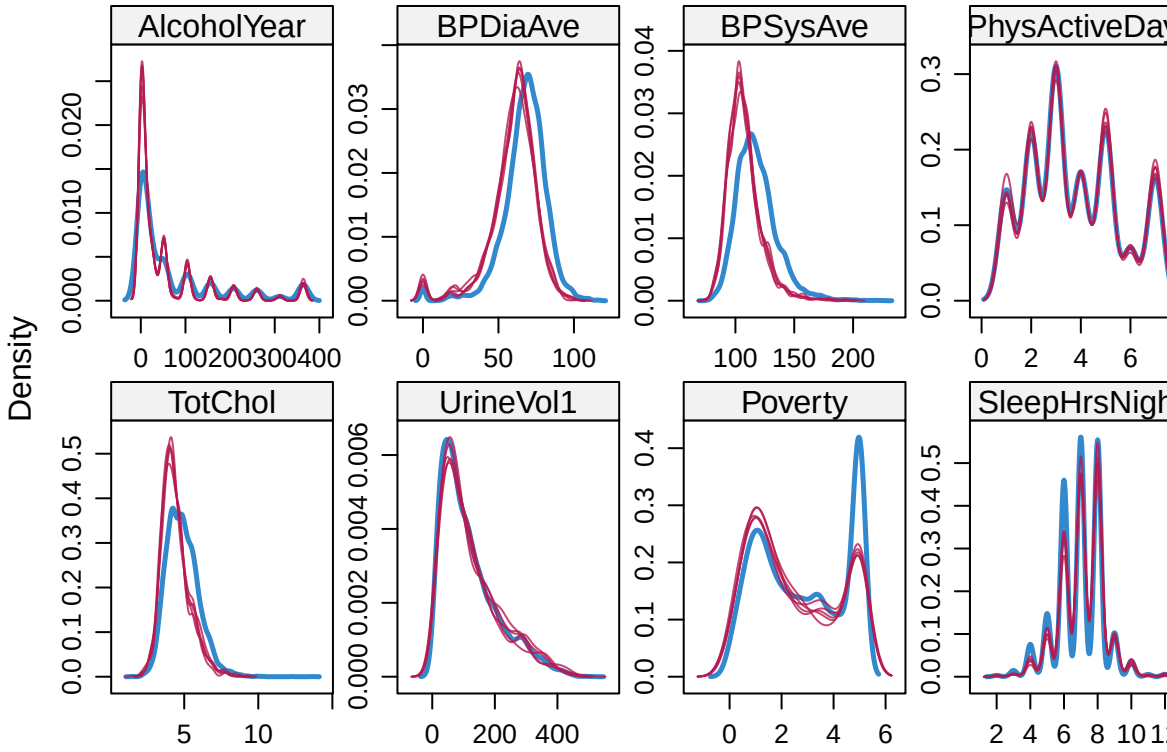


Mental health related variables:

```
to_test = c("HealthGen", "DaysPhysHlthBad", "DaysMentHlthBad", "LittleInterest", "Depressed" )
pairs(subset(NHANES, select = to_test))
```




```
# Compare the imputed variables (red) and observed (blue)
densityplot(imp)
```



Data Imputation

```
# summary(imp)
```