NHANES Data Analysis Project

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

August 4, 2023

Title

Introduction

Obesity is an endemic health issue common to many industrialized nations in the world. It is known that obesity is related to cardiovascular disease (high blood pressure, heart attack, stroke), type II diabetes, sleep apnea, metabolic syndrome, fatty liver disease and cancer CDC on obesity. According to the World Health Organization, WHO, obesity has nearly tripled between 1975 and 2021, in a time frame of less than two generations. Obesity can be measured by the BMI, is therefore a relevant health issue. The threshold of calling someone obese is a BMI ≥ 30.

Our personal interest in obesity is comes from the fact that two members of this team have a background in biochemistry and medicine, respectively, and that obesity still represents a health issue which everyone knows and can observe in everyday life. So, while there is a bio-medical relevance of the topic, the concept of BMI or the phenomenon of obesity do not require everyone to have a deep domain-specific knowledge. In fact, none of us directly worked scientifically with obesity before and thus we saw it as a new and challenging topic to work on.

We are using the "the non-institutionalized civilian resident population of the United States" (NHANES) dataset. It has been published by the "American National Health and Nutrition Examination surveys" since the 1960s. The full data thus covers the relevant time frame which saw the drastic increase in obesity as reported by the WHO. The full data can be obtained from CDC.gov. We are using a subset of NHANES which is easily accessible through the R library `NHANES`. It comprises a subset of 10,000 rows and covers a survey period between 2009 and 2012. While not comprehensive it should give us enough material to build a model, select different predictors and reason about its predictions. Our goal is to build an interpretable model which identifies and quantifies the influence of several physical and life style-related predictors on body weight, specifically the BMI.

In order to build an interpretable model we need to be mindful with non-linear data transformations, high-order interactions and also need to keep variance inflation under control. The task is made challenging by a large number of missing values. If we simply omit all `NA`s using `na.omit()`, which discards all rows with any missing value, we will reduce the total information from 10,000 observations to less than 1%.

Our strategy outline for approaching this project is as follows:

1. Variable, that is, predictor selection using a combination of semantic grouping (some variables convey similar information) and collinearity (visually, through `pairs()` plots).
2. With a subset of predictors, which we call `nhanes_select`, we then approach the problem of missing data by using multivariate imputation by chained equations MICE using the `mice` library by Buuren and Groothuls-Outhoorn (2011).
3. After omitting all `NA`s, and with 5 versions of the imputed data, we build our first additive model to check parameter estimates, p-values, significance of regression as well as testing LINE assumptions.

4. Next we apply a Box-Cox transformation to BMI and use different search procedures, backwards, forwards, and stepwise to search for models and select predictors.
5. After this we check which variables are correlated with BMI to further investigate possible interaction terms, and determine if there are other variables we could consider dropping.
6. We then consider if our models of interest could benefit from any data transformations.
7. Finally, we perform outlier assessment and diagnostics on our models of interest to evaluate them and explain which model(s) we prefer.

Methods

Data Import

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

```
## Loading required package: NHANES
```

```
library(NHANES, quiet = TRUE)
```

Initial Variable Selection

Rule out variables by reasoning or by exploratory analysis We have chosen `BMI` (Body mass index (weight/height^2 in $[\frac{kg}{m^2}]$) 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`.

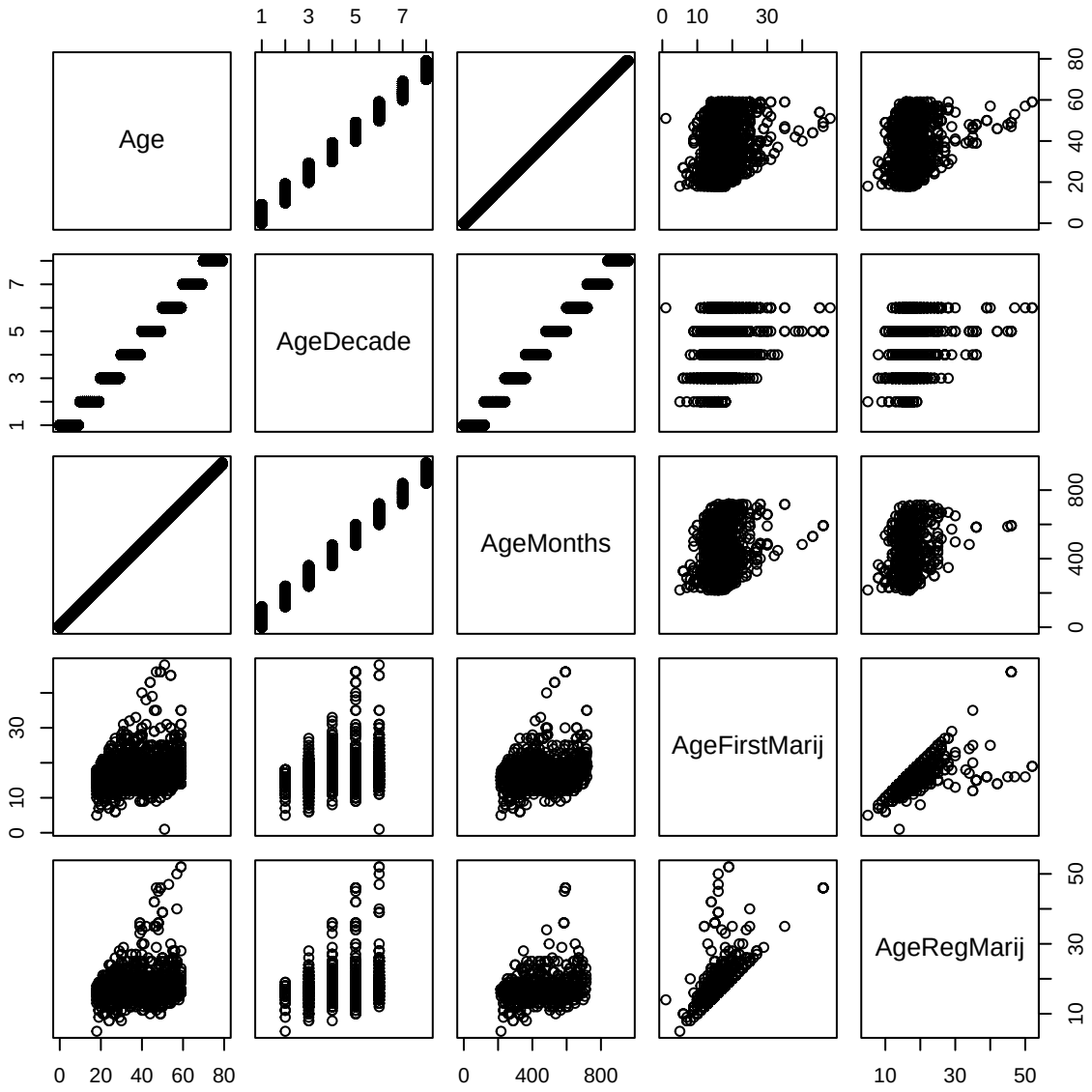
```
sort(names(NHANES)) # alphabetic order
```

```
##  [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 dataframe `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 seem 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:

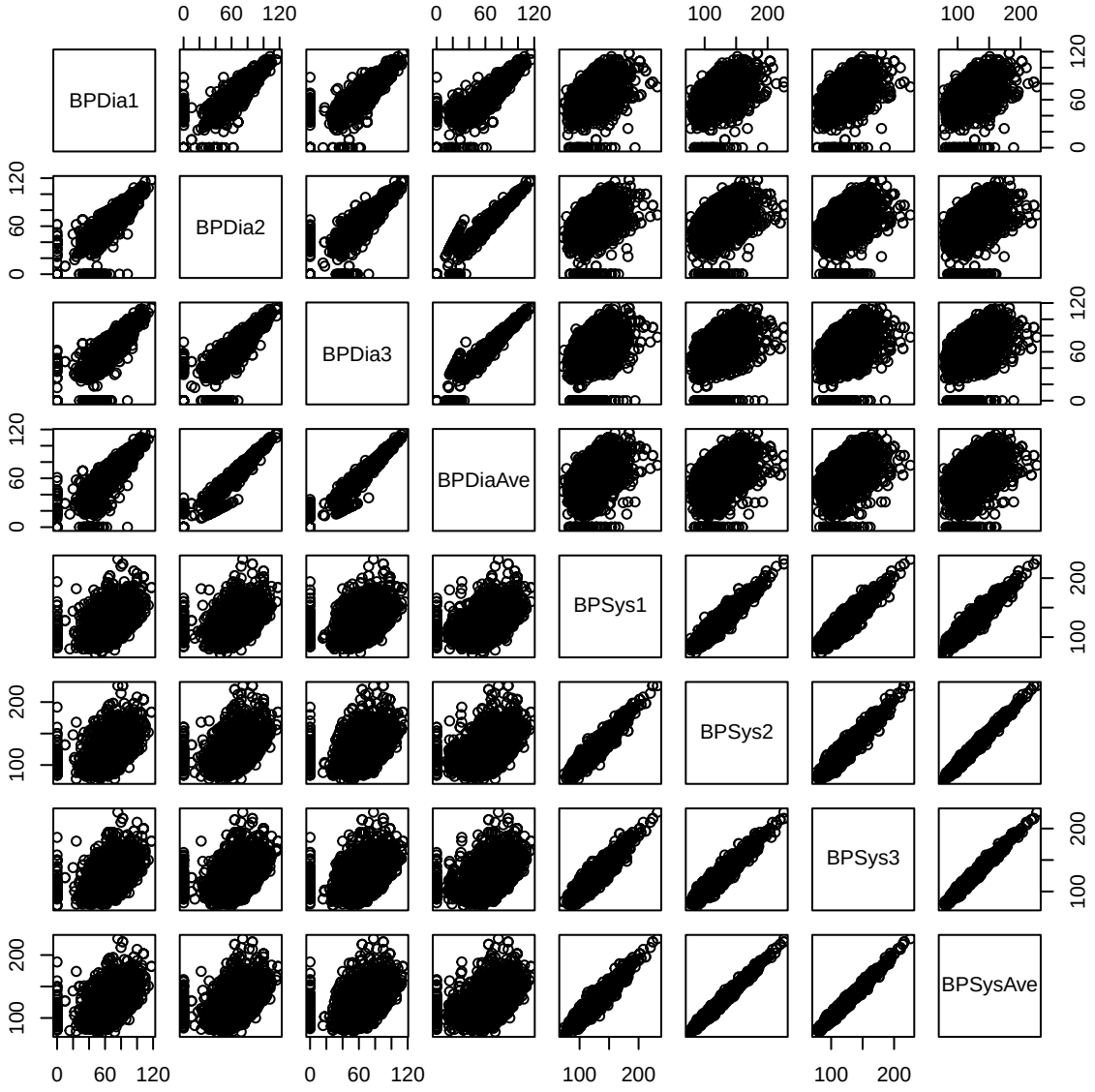
```
pairs(subset(NHANES, select = c('Age', 'AgeDecade', 'AgeMonths',
                                'AgeFirstMarij', 'AgeRegMarij')))
```



`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 decide 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 choose 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 dataframes 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
```

Table 1: Initial Predictors Selected

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

```
                  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'))

#Note: 'SurveyYr' is not a predictor, we will be filtering data by SurveyYr later,after which we
#will remove SurveyYr from the predictor list.
df_keep = data.frame(predictor = c('SurveyYr', 'Age', 'AlcoholYear', 'Marijuana', 'SmokeNow',
                                   'HardDrugs', 'BPDiaAve', 'BPSysAve', 'PhysActiveDays', 'TVHrsDay',
                                   '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.

Furthermore, the NHANES dataset has data for 2 survey years: 2009-10 and 2011-12. There are certain variables, such as TVHrsDay and CompHrsDay which are only present within the later time period (2011_2012). To eliminate this large missing value problem, we will further filter our dataset down by the more recent year, 2011_12.

```
nhanes_select = subset(NHANES, select =c(df_keep$predictor, "BMI"))
nhanes_select = nhanes_select[nhanes_select$SurveyYr == '2011_12', ] #filtering by survey year
nhanes_select = subset(nhanes_select, select = -c(SurveyYr)) #removing SurveyYr as a column for
#model building
```

The resulting dataset, after the initial variable selection and filtering above, consists of 5000 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.

```r
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

```r
library(tidyverse, quiet = 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
```

```r
# 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 dataframe 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     3440         68.80
## PhysActiveDays PhysActiveDays     2614         52.28
## Marijuana           Marijuana     2557         51.14
## HardDrugs           HardDrugs     2118         42.36
```

```
## AlcoholYear       AlcoholYear     2016         40.32
## LittleInterest LittleInterest     1665         33.30
## Education           Education     1416         28.32
## MaritalStatus   MaritalStatus     1415         28.30
## HealthGen           HealthGen     1202         24.04
## SleepHrsNight   SleepHrsNight     1166         23.32
## TotChol               TotChol      775         15.50
## BPDiaAve             BPDiaAve      719         14.38
## BPSysAve             BPSysAve      719         14.38
## UrineVol1           UrineVol1      501         10.02
## Poverty               Poverty      325          6.50
## BMI                       BMI      166          3.32
## TVHrsDay             TVHrsDay      141          2.82
## CompHrsDay         CompHrsDay      137          2.74
## Diabetes             Diabetes       64          1.28
## 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`, `Marijuana`, `HardDrugs` and `AlcoholYr`. 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 166 such observations, to avoid imputation of our response variable. There are 4834 observations left after this operation.

```r
library(mice, quiet = TRUE)
```

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

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

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

```r
# 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 using the variables we decided to keep earlier, to allow for an initial diagnostic evaluation.

```r
# 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 model with 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
## -15.73  -3.86  -0.55   3.09  52.06
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               15.305175   1.250505   12.24  < 2e-16 ***
## Age                        0.064724   0.006595    9.81  < 2e-16 ***
## AlcoholYear               -0.007722   0.000948   -8.14  4.8e-16 ***
```

```
## MarijuanaYes              -0.374769   0.194165   -1.93  0.05365 .
## SmokeNowYes               -2.740417   0.196813  -13.92  < 2e-16 ***
## HardDrugsYes              -0.022609   0.246256   -0.09  0.92685
## BPDiaAve                   0.068707   0.006625   10.37  < 2e-16 ***
## BPSysAve                   0.030473   0.006433    4.74  2.2e-06 ***
## PhysActiveDays             0.055219   0.045900    1.20  0.22902
## TVHrsDay0_to_1_hr         -0.575227   0.630741   -0.91  0.36182
## TVHrsDay1_hr              -0.860637   0.618472   -1.39  0.16412
## TVHrsDay2_hr               0.090020   0.608126    0.15  0.88233
## TVHrsDay3_hr               0.973730   0.619853    1.57  0.11627
## TVHrsDay4_hr               0.708768   0.640128    1.11  0.26825
## TVHrsDayMore_4_hr          0.742210   0.635447    1.17  0.24286
## CompHrsDay0_to_1_hr        0.552657   0.254936    2.17  0.03022 *
## CompHrsDay1_hr             1.696322   0.277612    6.11  1.1e-09 ***
## CompHrsDay2_hr             1.875005   0.315557    5.94  3.0e-09 ***
## CompHrsDay3_hr             2.280344   0.371747    6.13  9.3e-10 ***
## CompHrsDay4_hr             2.875970   0.506751    5.68  1.5e-08 ***
## CompHrsDayMore_4_hr        4.353220   0.425713   10.23  < 2e-16 ***
## TotChol                    0.320162   0.088781    3.61  0.00031 ***
## DiabetesYes                2.030624   0.339931    5.97  2.5e-09 ***
## UrineVol1                  0.002489   0.000984    2.53  0.01142 *
## HealthGenVgood             1.982660   0.269509    7.36  2.2e-13 ***
## HealthGenGood              4.267078   0.275729   15.48  < 2e-16 ***
## HealthGenFair              5.098822   0.371772   13.71  < 2e-16 ***
## HealthGenPoor              5.798022   0.698812    8.30  < 2e-16 ***
## LittleInterestSeveral      0.025121   0.240525    0.10  0.91682
## LittleInterestMost        -0.843029   0.364079   -2.32  0.02063 *
## Poverty                   -0.072035   0.061714   -1.17  0.24317
## SleepHrsNight             -0.261706   0.066796   -3.92  9.1e-05 ***
## Gendermale                -0.031737   0.178418   -0.18  0.85882
## Race1Hispanic             -1.222194   0.407788   -3.00  0.00274 **
## Race1Mexican               0.069518   0.385570    0.18  0.85692
## Race1White                -0.885474   0.279902   -3.16  0.00157 **
## Race1Other                -2.471128   0.382317   -6.46  1.1e-10 ***
## Education9 - 11th Grade    0.131455   0.464995    0.28  0.77742
## EducationHigh School       0.610874   0.449645    1.36  0.17435
## EducationSome College      0.561087   0.449301    1.25  0.21180
## EducationCollege Grad      0.133899   0.473872    0.28  0.77752
## MaritalStatusLivePartner  -1.411047   0.426765   -3.31  0.00095 ***
## MaritalStatusMarried      -1.163745   0.337951   -3.44  0.00058 ***
## MaritalStatusNeverMarried -1.552329   0.373075   -4.16  3.2e-05 ***
## MaritalStatusSeparated    -0.487822   0.668684   -0.73  0.46572
## MaritalStatusWidowed      -3.362105   0.523531   -6.42  1.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.79 on 4788 degrees of freedom
## Multiple R-squared:  0.374,  Adjusted R-squared:  0.368
## F-statistic: 63.5 on 45 and 4788 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, quiet = TRUE)
```

```
##
## 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            3.045  1           1.745
## AlcoholYear    1.186  1           1.089
## Marijuana      1.305  1           1.142
## SmokeNow       1.397  1           1.182
## HardDrugs      1.258  1           1.121
## BPDiaAve       1.454  1           1.206
## BPSysAve       1.818  1           1.348
## PhysActiveDays 1.030  1           1.015
## TVHrsDay       1.426  6           1.030
## CompHrsDay     1.491  6           1.034
## TotChol        1.292  1           1.137
## Diabetes       1.167  1           1.080
## UrineVol1      1.050  1           1.025
## HealthGen      1.540  4           1.055
## LittleInterest 1.173  2           1.041
## Poverty        1.578  1           1.256
## SleepHrsNight  1.094  1           1.046
## Gender         1.148  1           1.072
## Race1          1.537  4           1.055
## Education      2.014  4           1.091
## MaritalStatus  2.309  5           1.087
```

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 diagnostic tests on this model to identify any potential issues.

```r
library(lmtest)
```

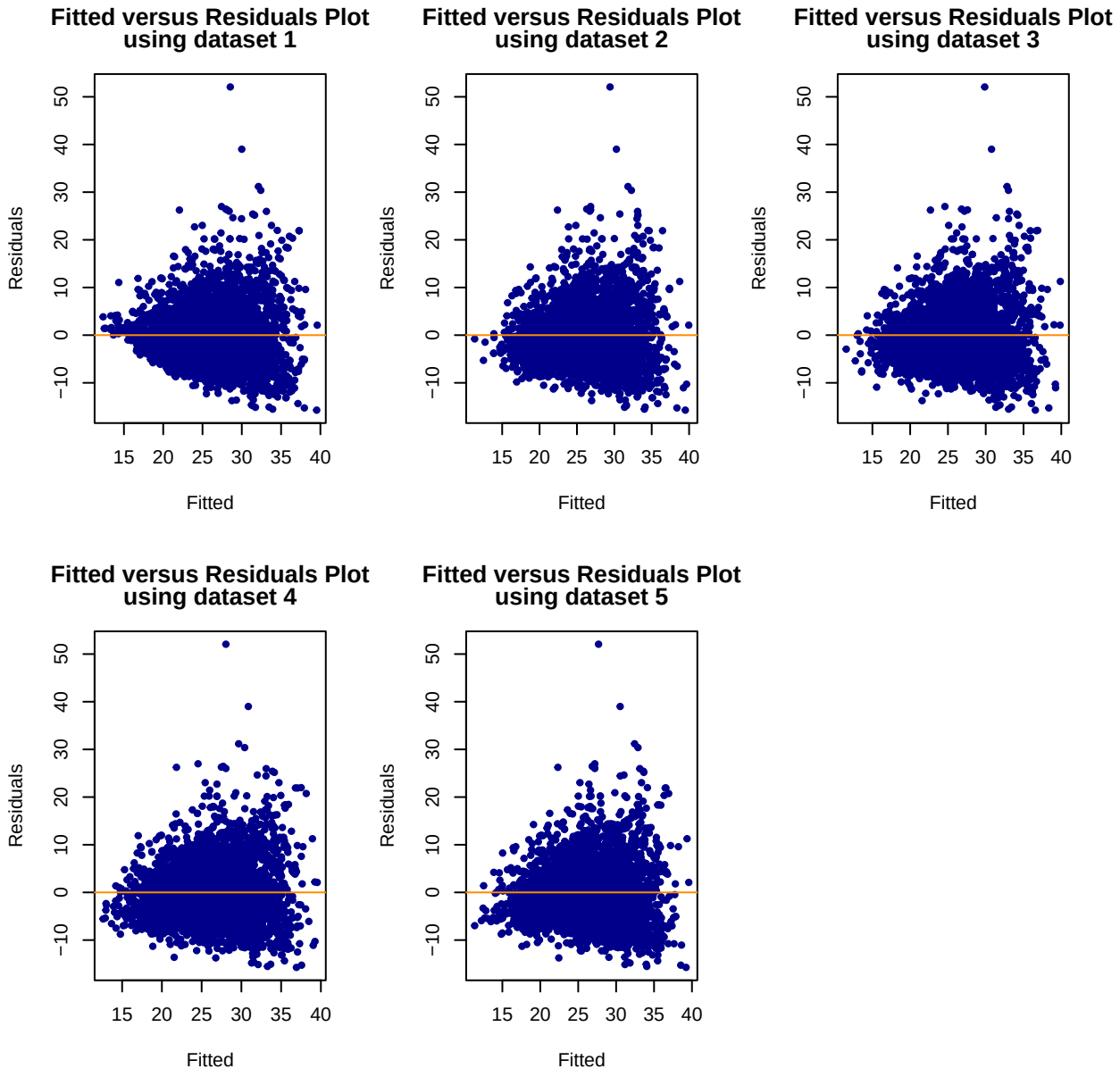
```
## Loading required package: zoo
```

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

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

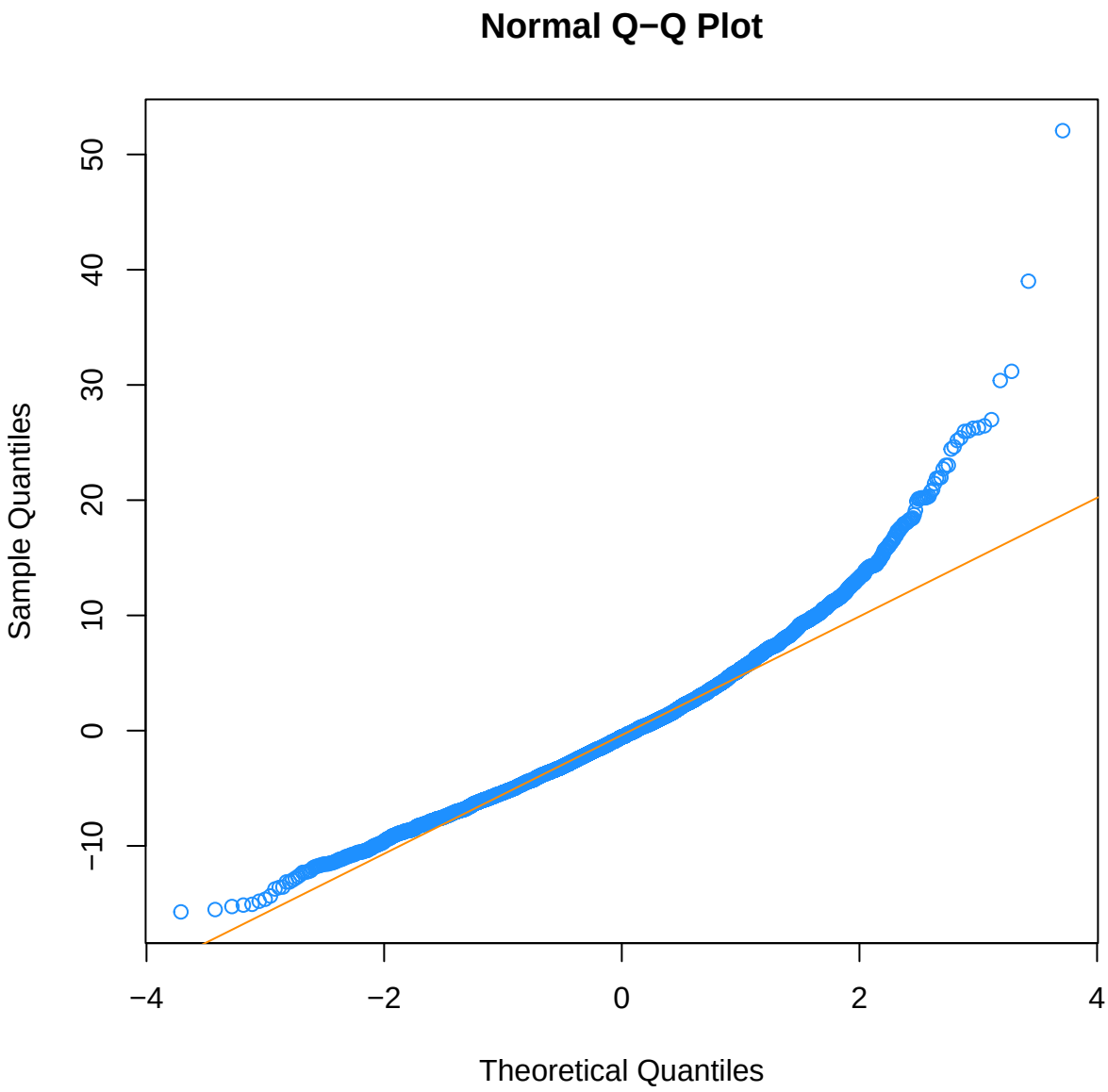
```r
### First, let's define some functions ###

# 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), shapirotest_p = rep(0,5))
  for (i in 1:5){
    fit_summary$bptest_p[i] = unname(bptest(fit$analyses[[i]])$p.value)
    fit_summary$shapirotest_p[i] = shapiro.test(residuals(fit$analyses[[i]]))$p.value
  }
  knitr::kable(fit_summary, col.names = c("BP Test", "Shapiro Test"))
}


# 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
par(mfrow = c(2,3))
#Fitted versus Residuals Plot for the imputed dataset model
for (i in seq(1,5)) {
  title = strwrap(paste("Fitted versus Residuals Plot using dataset ", as.character(i)), width = 30, si
  plot(fitted(fit_add$analyses[[i]]), resid(fit_add$analyses[[1]]), col = "darkblue", pch = 20,
       xlab = "Fitted", ylab = "Residuals", main = title)
  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 Shapiro-Wilk Test for normality, and the Breusch-Pagan Test for Homoscedasticity.

```
model_diagnostics(fit_add)
```

BP Test	Shapiro 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.

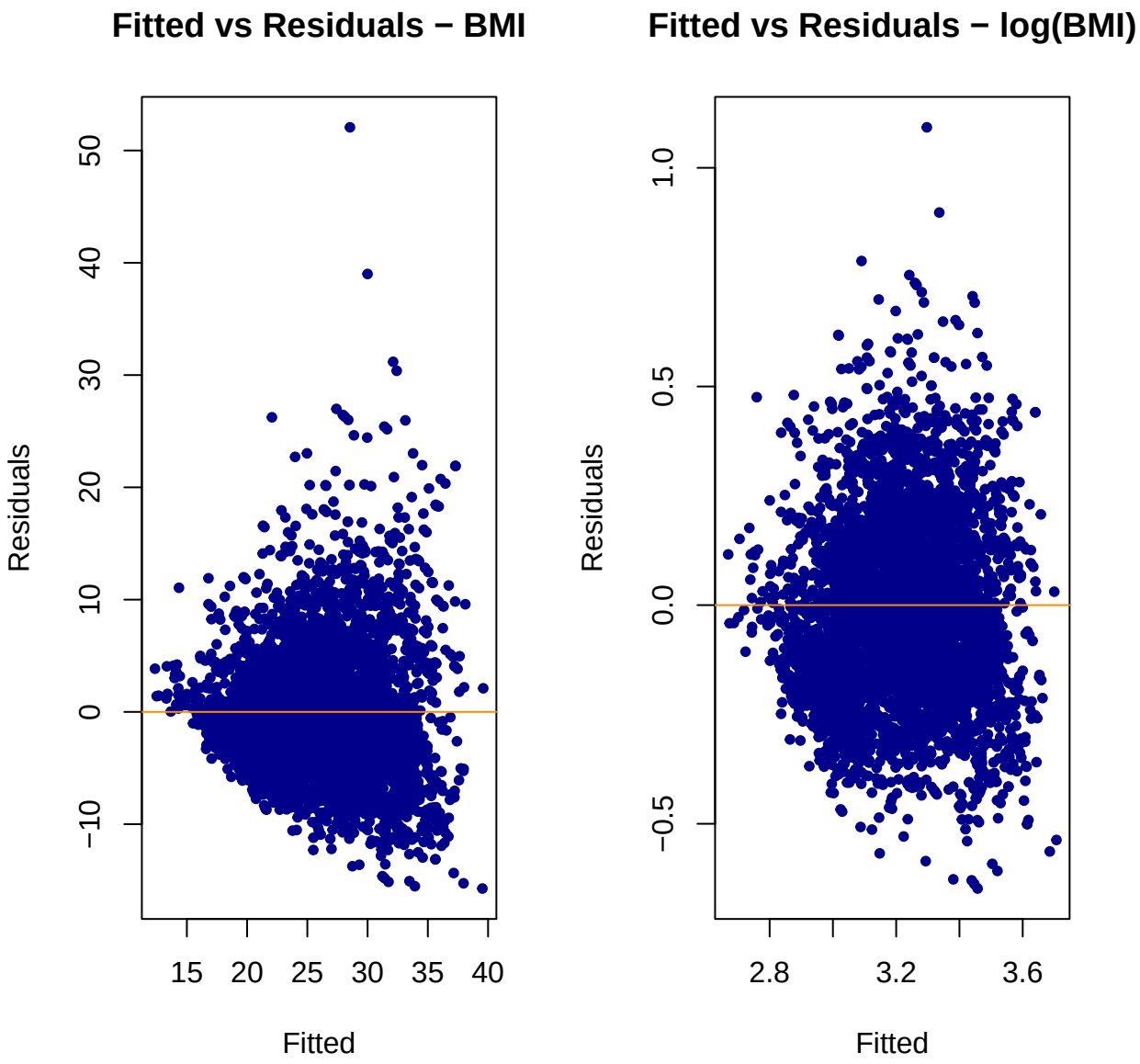
Because of the findings above, we will perform a variance stabilizing log transformation on the response variable (BMI), fit the model again and reassess the diagnostics.

```
# 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]]) #summary of the model with the 1st imputed dataset
```

```
##
## 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
## -0.6480 -0.1470 -0.0048  0.1330  1.0925
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.7725568  0.0448156   61.87  < 2e-16 ***
## Age                        0.0034457  0.0002363   14.58  < 2e-16 ***
## AlcoholYear               -0.0002686  0.0000340   -7.91  3.3e-15 ***
## MarijuanaYes              -0.0115717  0.0069585   -1.66  0.09639 .
## SmokeNowYes               -0.0966763  0.0070534  -13.71  < 2e-16 ***
## HardDrugsYes              -0.0030122  0.0088253   -0.34  0.73288
## BPDiaAve                   0.0028181  0.0002374   11.87  < 2e-16 ***
## BPSysAve                   0.0009281  0.0002305    4.03  5.8e-05 ***
## PhysActiveDays             0.0016255  0.0016450    0.99  0.32312
## TVHrsDay0_to_1_hr         -0.0217552  0.0226045   -0.96  0.33588
## TVHrsDay1_hr              -0.0360923  0.0221648   -1.63  0.10351
## TVHrsDay2_hr               0.0021394  0.0217940    0.10  0.92181
## TVHrsDay3_hr               0.0291594  0.0222143    1.31  0.18937
## TVHrsDay4_hr               0.0196392  0.0229409    0.86  0.39200
## TVHrsDayMore_4_hr          0.0171747  0.0227732    0.75  0.45079
## CompHrsDay0_to_1_hr        0.0272574  0.0091364    2.98  0.00287 **
## CompHrsDay1_hr             0.0719256  0.0099491    7.23  5.6e-13 ***
## CompHrsDay2_hr             0.0850372  0.0113089    7.52  6.5e-14 ***
## CompHrsDay3_hr             0.1002327  0.0133227    7.52  6.3e-14 ***
## CompHrsDay4_hr             0.1217077  0.0181610    6.70  2.3e-11 ***
## CompHrsDayMore_4_hr        0.1589111  0.0152567   10.42  < 2e-16 ***
## TotChol                    0.0126407  0.0031817    3.97  7.2e-05 ***
## DiabetesYes                0.0521338  0.0121824    4.28  1.9e-05 ***
## UrineVol1                  0.0001222  0.0000353    3.47  0.00053 ***
```

```
## HealthGenVgood             0.0770423  0.0096587    7.98  1.9e-15 ***
## HealthGenGood              0.1554180  0.0098816   15.73  < 2e-16 ***
## HealthGenFair              0.1796301  0.0133236   13.48  < 2e-16 ***
## HealthGenPoor              0.1929542  0.0250440    7.70  1.6e-14 ***
## LittleInterestSeveral      0.0005474  0.0086199    0.06  0.94937
## LittleInterestMost        -0.0295603  0.0130479   -2.27  0.02352 *
## Poverty                   -0.0033370  0.0022117   -1.51  0.13142
## SleepHrsNight             -0.0090436  0.0023938   -3.78  0.00016 ***
## Gendermale                 0.0049159  0.0063941    0.77  0.44204
## Race1Hispanic             -0.0336022  0.0146143   -2.30  0.02153 *
## Race1Mexican               0.0167151  0.0138181    1.21  0.22647
## Race1White                -0.0307923  0.0100311   -3.07  0.00215 **
## Race1Other                -0.0869734  0.0137015   -6.35  2.4e-10 ***
## Education9 - 11th Grade    0.0007607  0.0166645    0.05  0.96359
## EducationHigh School       0.0159172  0.0161144    0.99  0.32332
## EducationSome College      0.0140815  0.0161021    0.87  0.38188
## EducationCollege Grad     -0.0007174  0.0169826   -0.04  0.96631
## MaritalStatusLivePartner  -0.0456227  0.0152944   -2.98  0.00287 **
## MaritalStatusMarried      -0.0354756  0.0121115   -2.93  0.00342 **
## MaritalStatusNeverMarried -0.0549756  0.0133703   -4.11  4.0e-05 ***
## MaritalStatusSeparated    -0.0110077  0.0239643   -0.46  0.64601
## MaritalStatusWidowed      -0.1217199  0.0187623   -6.49  9.6e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.207 on 4788 degrees of freedom
## Multiple R-squared:  0.421,  Adjusted R-squared:  0.416
## F-statistic: 77.3 on 45 and 4788 DF,  p-value: <2e-16
```

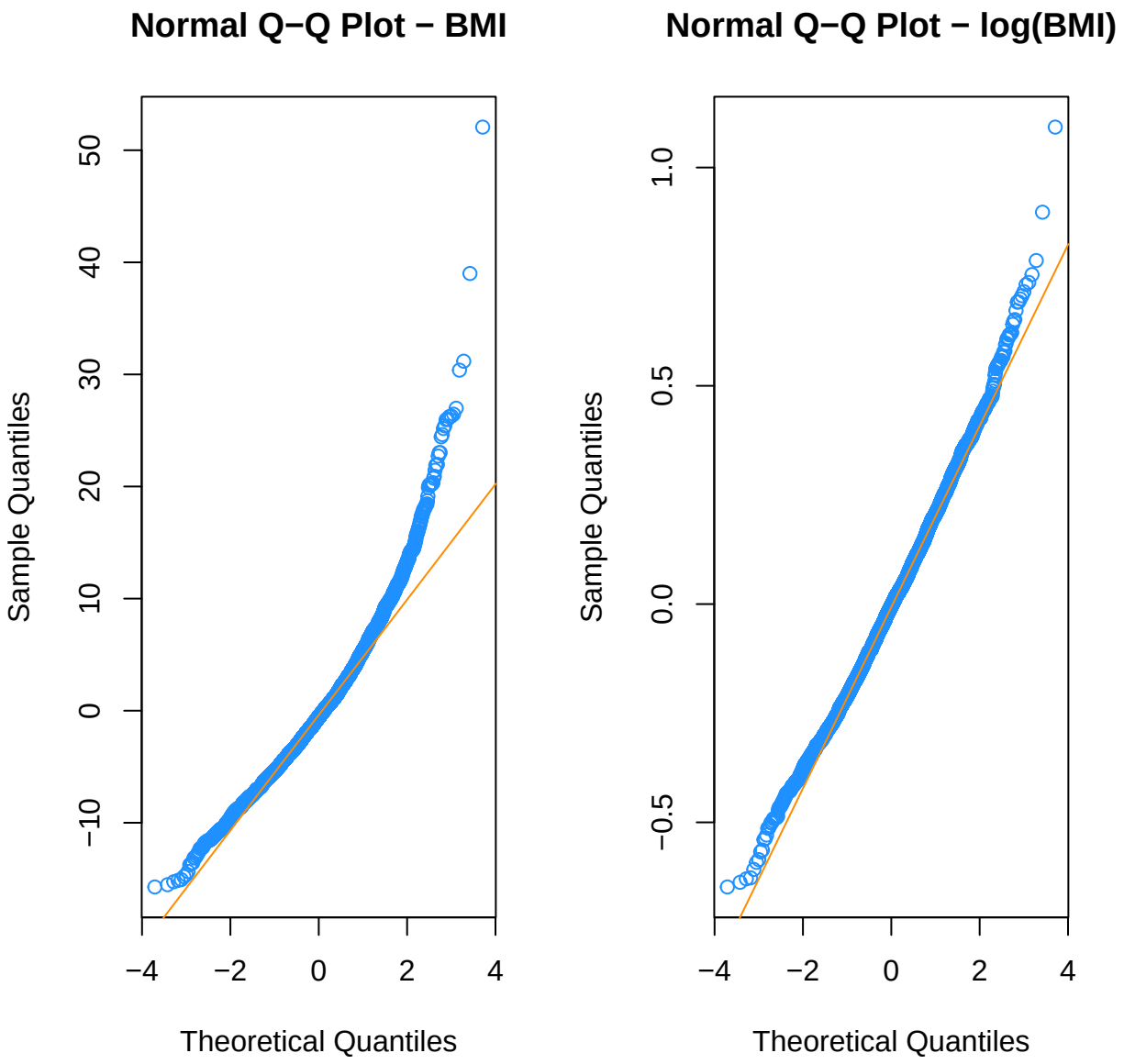
```r
#Comparing the Fitted versus Residuals Plots of the Initial Additive model and the Log(BMI) #Transforma
par(mfrow=c(1,2))
plot(fitted(fit_add$analyses[[1]]), resid(fit_add$analyses[[1]]), col = "darkblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residuals - BMI")
abline(h=0,col = "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 is a much better appearing QQ plot. Moving forward, we will use the log transformed BMI for our model building.

Model Search for 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              1              3              5              5
##      Marijuana  MaritalStatus PhysActiveDays        Poverty          Race1
##              1              5              3              4              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 Education, Gender, and Marijuana. Let's compare the models using anova, leaving out variables with less than 5 votes.

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

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     1.162   1   4  4794  0.3417 0.7217
```

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

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

```
##    test statistic df1 df2 dfcom p.value   riv
##  2 ~~ 1    0.5063   1   4  4795   0.516 2.867
```

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

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

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     2.363   1   4  4796   0.199 0.8313
```

This p-value is again greater than 0.05, and so we will remove `Poverty` for now.

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 +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
summary(fit_add_aic$analyses[[1]])
```

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve +
##     BPSysAve + TVHrsDay + CompHrsDay + TotChol + Diabetes + UrineVol1 +
##     HealthGen + LittleInterest + SleepHrsNight + Race1 + MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6309 -0.1483 -0.0039  0.1346  1.1052
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.7794590  0.0418230   66.46  < 2e-16 ***
## Age                        0.0033349  0.0002323   14.36  < 2e-16 ***
## AlcoholYear               -0.0002872  0.0000327   -8.78  < 2e-16 ***
## SmokeNowYes               -0.0958068  0.0069330  -13.82  < 2e-16 ***
## BPDiaAve                   0.0028119  0.0002352   11.95  < 2e-16 ***
## BPSysAve                   0.0009807  0.0002276    4.31  1.7e-05 ***
## TVHrsDay0_to_1_hr         -0.0255639  0.0225343   -1.13  0.25667
## TVHrsDay1_hr              -0.0392363  0.0220777   -1.78  0.07560 .
## TVHrsDay2_hr               0.0003978  0.0217280    0.02  0.98539
## TVHrsDay3_hr               0.0290326  0.0221500    1.31  0.19001
## TVHrsDay4_hr               0.0190557  0.0228437    0.83  0.40422
## TVHrsDayMore_4_hr          0.0173772  0.0226665    0.77  0.44333
```

```
## CompHrsDay0_to_1_hr        0.0233118  0.0088953    2.62  0.00880 **
## CompHrsDay1_hr             0.0687027  0.0095956    7.16  9.3e-13 ***
## CompHrsDay2_hr             0.0816409  0.0110123    7.41  1.4e-13 ***
## CompHrsDay3_hr             0.0976171  0.0130769    7.46  9.8e-14 ***
## CompHrsDay4_hr             0.1180113  0.0178185    6.62  3.9e-11 ***
## CompHrsDayMore_4_hr        0.1575684  0.0149722   10.52  < 2e-16 ***
## TotChol                    0.0129918  0.0031369    4.14  3.5e-05 ***
## DiabetesYes                0.0545001  0.0121515    4.49  7.5e-06 ***
## UrineVol1                  0.0001222  0.0000348    3.51  0.00045 ***
## HealthGenVgood             0.0769485  0.0095603    8.05  1.0e-15 ***
## HealthGenGood              0.1570562  0.0096750   16.23  < 2e-16 ***
## HealthGenFair              0.1812836  0.0129991   13.95  < 2e-16 ***
## HealthGenPoor              0.1960665  0.0249067    7.87  4.3e-15 ***
## LittleInterestSeveral     -0.0002466  0.0085974   -0.03  0.97711
## LittleInterestMost        -0.0306617  0.0129316   -2.37  0.01778 *
## SleepHrsNight             -0.0094214  0.0023823   -3.95  7.8e-05 ***
## Race1Hispanic             -0.0329306  0.0144673   -2.28  0.02288 *
## Race1Mexican               0.0160020  0.0135288    1.18  0.23694
## Race1White                -0.0347807  0.0099063   -3.51  0.00045 ***
## Race1Other                -0.0895706  0.0135397   -6.62  4.1e-11 ***
## MaritalStatusLivePartner  -0.0460673  0.0152443   -3.02  0.00252 **
## MaritalStatusMarried      -0.0368731  0.0119362   -3.09  0.00202 **
## MaritalStatusNeverMarried -0.0560017  0.0132826   -4.22  2.5e-05 ***
## MaritalStatusSeparated    -0.0103287  0.0239475   -0.43  0.66627
## MaritalStatusWidowed      -0.1190438  0.0185570   -6.42  1.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.208 on 4797 degrees of freedom
## Multiple R-squared:  0.419,  Adjusted R-squared:  0.415
## F-statistic: 96.2 on 36 and 4797 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, full scope with all predictors
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              1              3              5              5
##      Marijuana  MaritalStatus PhysActiveDays        Poverty          Race1
##              1              5              3              4              5
##  SleepHrsNight       SmokeNow        TotChol       TVHrsDay      UrineVol1
##              5              5              5              5              5
```

This appears to yield the same votes as the prior method, which results in the same model.

Lastly, let's try a Stepwise search 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              1              3              5              5
##      Marijuana  MaritalStatus PhysActiveDays        Poverty          Race1
##              1              5              3              4              5
##  SleepHrsNight       SmokeNow        TotChol       TVHrsDay      UrineVol1
##              5              5              5              5              5
```

Here agin we get the same results. For now, our additive model will be `fit_add_aic`

Before we move on, there are some predictors which are not significant individually as we saw from the summary of the model earlier, so we should check for collinearity again.

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

```
##                 GVIF Df GVIF^(1/(2*Df))
## Age            2.938  1           1.714
## AlcoholYear    1.098  1           1.048
## SmokeNow       1.349  1           1.161
## BPDiaAve       1.426  1           1.194
## BPSysAve       1.770  1           1.331
```

```
## TVHrsDay       1.333  6           1.024
## CompHrsDay     1.298  6           1.022
## TotChol        1.255  1           1.120
## Diabetes       1.160  1           1.077
## UrineVol1      1.024  1           1.012
## HealthGen      1.398  4           1.043
## LittleInterest 1.146  2           1.035
## SleepHrsNight  1.082  1           1.040
## Race1          1.321  4           1.035
## MaritalStatus  2.060  5           1.075
```

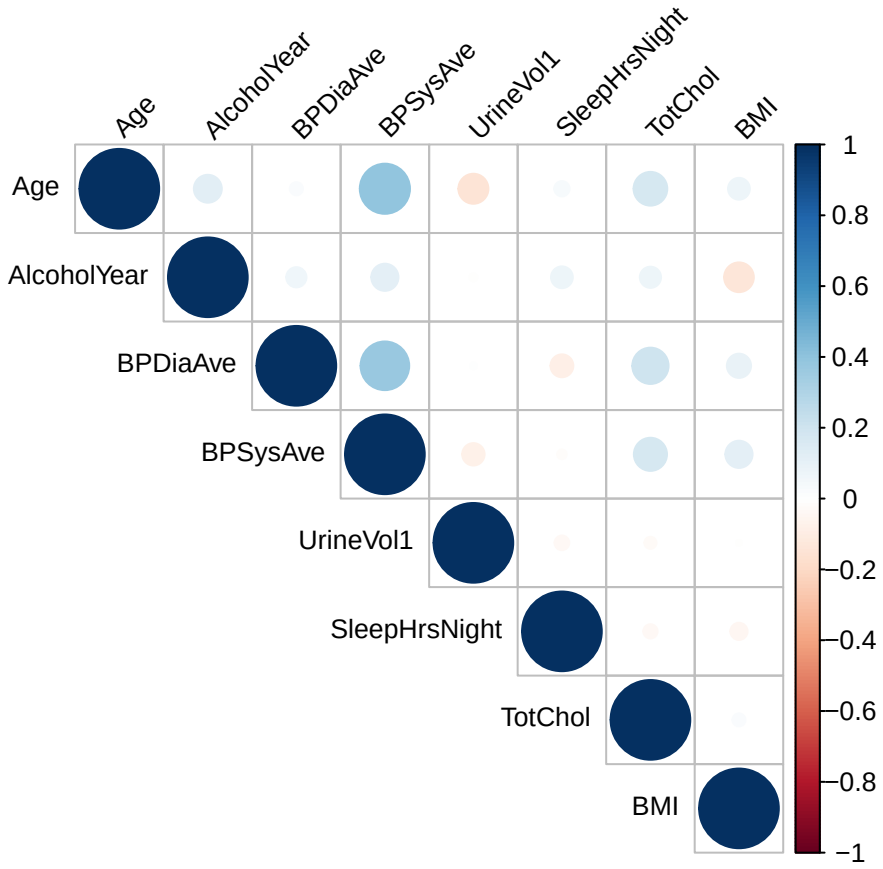
There appear to be no major issues with collinearity in this additive model.

Further Predictor Exploration Additionally, we want to check which variables from this addditive model are correlated with BMI to further investigate possible interaction terms, and determine if there are additional variables we could consider dropping.

```
library("corrplot")
```

```
## corrplot 0.92 loaded
```

```
# Subset the 'nhanes' dataset using the names of numerical columns in 'numerical_data'
nhanes_numerical_subset = nhanes_select[, c('Age', 'AlcoholYear', 'BPDiaAve', 'BPSysAve', 'UrineVol1',
                                            'SleepHrsNight', 'TotChol' ,'BMI')]

# Calculate the correlation matrix for 'nhanes_numerical_subset'
cor_matrix  = cor(nhanes_numerical_subset, use = "complete.obs")

# Create the correlation plot for 'BMI' and other numeric variables
corrplot(cor_matrix, type = "upper", tl.cex = 0.8, tl.col = "black", tl.srt = 45)
```




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

```
## Loading required package: vcd
```

```
## Loading required package: grid
```

```
library(vcd)

# Subset the 'nhanes' dataset using the names of categorical columns in 'categorical_data'
nhanes_categorical_subset = nhanes_select[, c('SmokeNow', 'TVHrsDay', 'CompHrsDay','Diabetes',
                                              'HealthGen', 'LittleInterest', 'Race1',
                                              'MaritalStatus', 'BMI')]

# Assuming 'nhanes_categorical_subset' already contains the 'BMI' column and categorical variables

# Perform a Chi-square test for each categorical variable
for (var in names(nhanes_categorical_subset)) {
  if (is.factor(nhanes_categorical_subset[[var]])) {
    chi_result <- chisq.test(nhanes_categorical_subset[[var]], nhanes_categorical_subset$BMI)
    print(paste("Variable:", var))
    print(chi_result)
  }
}
```

```
## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: SmokeNow"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 550, df = 249, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: TVHrsDay"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 4251, df = 2100, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: CompHrsDay"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 4472, df = 2100, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: Diabetes"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 1125, df = 350, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: HealthGen"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 2901, df = 1284, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect
```

```
## [1] "Variable: LittleInterest"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 1270, df = 612, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: Race1"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 2344, df = 1400, p-value <2e-16

## Warning in chisq.test(nhanes_categorical_subset[[var]],
## nhanes_categorical_subset$BMI): Chi-squared approximation may be incorrect

## [1] "Variable: MaritalStatus"
##
##  Pearson's Chi-squared test
##
## data:  nhanes_categorical_subset[[var]] and nhanes_categorical_subset$BMI
## X-squared = 2971, df = 1545, p-value <2e-16
```

Since all the p-values for the categorical values above are very small, we do not sneed to remove them. However, based upon the findings of the numerical data above, it seems reasonable to try removing both `UrineVol1` and `TotChol`, as their individual correlations with BMI are negligible.

First we will try removing `UrineVol1` at a significance level $\alpha = 0.01$:

```
fit_add_aic_with = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  UrineVol1 +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
# removing UrineVol1
fit_add_aic_without =with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
anova(fit_add_aic_without, fit_add_aic_with)
```

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     7.289   1   4  4797  0.0541 0.2203
```

Since the p-value of the anova test comparing the models with and without `UrineVol` is greater than 0.01, we fail to reject the null hypothesis, and should remove the variable.

Now let's consider removing `TotChol` at a significance level $\alpha = 0.01$.

```
fit_add_aic_with = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
# removing TotChol
fit_add_aic_without =with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
anova(fit_add_aic_without, fit_add_aic_with)
```

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     8.481   1   4  4798 0.04358 0.2853
```

Likewise, the p-value is greater than 0.01, although it is less than 0.05, so for now we can leave it.

So our **final simple additive model** is below, and we will name it `final_add`.

```
final_add = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus))
summary(final_add$analyses[[1]])
```

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve +
##     BPSysAve + TVHrsDay + CompHrsDay + TotChol + Diabetes + HealthGen +
##     LittleInterest + SleepHrsNight + Race1 + MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6426 -0.1493 -0.0043  0.1350  1.1403
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.7968551  0.0415772   67.27  < 2e-16 ***
## Age                        0.0032851  0.0002321   14.15  < 2e-16 ***
## AlcoholYear               -0.0002868  0.0000327   -8.76  < 2e-16 ***
## SmokeNowYes               -0.0974683  0.0069250  -14.07  < 2e-16 ***
## BPDiaAve                   0.0028423  0.0002354   12.08  < 2e-16 ***
## BPSysAve                   0.0009685  0.0002279    4.25  2.2e-05 ***
## TVHrsDay0_to_1_hr         -0.0273176  0.0225553   -1.21   0.2259
## TVHrsDay1_hr              -0.0409924  0.0220981   -1.86   0.0637 .
## TVHrsDay2_hr              -0.0018682  0.0217440   -0.09   0.9315
## TVHrsDay3_hr               0.0264089  0.0221635    1.19   0.2335
## TVHrsDay4_hr               0.0175558  0.0228667    0.77   0.4427
## TVHrsDayMore_4_hr          0.0149139  0.0226823    0.66   0.5109
## CompHrsDay0_to_1_hr        0.0229295  0.0089051    2.57   0.0101 *
## CompHrsDay1_hr             0.0685051  0.0096068    7.13  1.1e-12 ***
## CompHrsDay2_hr             0.0819096  0.0110250    7.43  1.3e-13 ***
## CompHrsDay3_hr             0.0979143  0.0130920    7.48  8.9e-14 ***
## CompHrsDay4_hr             0.1198771  0.0178316    6.72  2.0e-11 ***
```

```
## CompHrsDayMore_4_hr        0.1581283  0.0149890   10.55  < 2e-16 ***
## TotChol                    0.0131992  0.0031401    4.20  2.7e-05 ***
## DiabetesYes                0.0544160  0.0121658    4.47  7.9e-06 ***
## HealthGenVgood             0.0767951  0.0095715    8.02  1.3e-15 ***
## HealthGenGood              0.1575059  0.0096855   16.26  < 2e-16 ***
## HealthGenFair              0.1822406  0.0130115   14.01  < 2e-16 ***
## HealthGenPoor              0.1973725  0.0249333    7.92  3.0e-15 ***
## LittleInterestSeveral     -0.0007298  0.0086064   -0.08   0.9324
## LittleInterestMost        -0.0314470  0.0129449   -2.43   0.0152 *
## SleepHrsNight             -0.0097181  0.0023836   -4.08  4.6e-05 ***
## Race1Hispanic             -0.0311339  0.0144753   -2.15   0.0315 *
## Race1Mexican               0.0167594  0.0135430    1.24   0.2160
## Race1White                -0.0336319  0.0099126   -3.39   0.0007 ***
## Race1Other                -0.0884138  0.0135516   -6.52  7.5e-11 ***
## MaritalStatusLivePartner  -0.0456756  0.0152618   -2.99   0.0028 **
## MaritalStatusMarried      -0.0362126  0.0119488   -3.03   0.0025 **
## MaritalStatusNeverMarried -0.0555292  0.0132976   -4.18  3.0e-05 ***
## MaritalStatusSeparated    -0.0115108  0.0239733   -0.48   0.6311
## MaritalStatusWidowed      -0.1210410  0.0185701   -6.52  7.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.208 on 4798 degrees of freedom
## Multiple R-squared:  0.418,  Adjusted R-squared:  0.414
## F-statistic: 98.4 on 35 and 4798 DF,  p-value: <2e-16
```

Data Interactions To determine if interaction terms might improve our model, we will perform a backward AIC stepwise() function with added `Age` interaction terms. This seemed the most plausible and made the most sense from a domain perspective. It could be understandable for instance, that alcohol use at a young age vs an older age might have combined effects upon BMI.

```
# build the stepwise workflow using our fit_add_aic
# with interaction terms added as a starting point
scope <- list(upper = ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  TotChol +  Diabetes + HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:AlcoholYear +  Age:SmokeNow +  Age:BPDiaAve +
    Age:BPSysAve  + Age:TVHrsDay + Age:CompHrsDay +  Age:TotChol +  Age:Diabetes +  Age:HealthGen +
    Age:LittleInterest +  Age:SleepHrsNight +  Age:Race1 +  Age: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     Age:AlcoholYear        Age:BPDiaAve        Age:BPSysAve
```

```
##                   5                   3                   5                   5
##       Age:Diabetes      Age:HealthGen Age:LittleInterest  Age:MaritalStatus
##                   5                   5                   2                   5
##          Age:Race1  Age:SleepHrsNight       Age:SmokeNow        Age:TotChol
##                   5                   1                   5                   1
##       Age:TVHrsDay        AlcoholYear           BPDiaAve           BPSysAve
##                   4                   5                   5                   5
##         CompHrsDay           Diabetes          HealthGen     LittleInterest
##                   5                   5                   5                   5
##      MaritalStatus               Race1      SleepHrsNight           SmokeNow
##                   5                   5                   5                   5
##            TotChol           TVHrsDay
##                   1                   5
```

We now build a model with all the above predictors with > 3 votes.

```
fit_int_aic = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  + Age:TVHrsDay  +  Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus))
summary(fit_int_aic$analyses[[1]])
```

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve +
##     BPSysAve + TVHrsDay + CompHrsDay + Diabetes + HealthGen +
##     LittleInterest + SleepHrsNight + Race1 + MaritalStatus +
##     Age:SmokeNow + Age:BPDiaAve + Age:BPSysAve + Age:TVHrsDay +
##     Age:Diabetes + Age:HealthGen + Age:Race1 + Age:MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6386 -0.1323 -0.0062  0.1221  1.0231
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.08e+00   7.89e-02   26.43  < 2e-16 ***
## Age                            2.06e-02   1.60e-03   12.82  < 2e-16 ***
## AlcoholYear                   -2.81e-04   3.05e-05   -9.24  < 2e-16 ***
## SmokeNowYes                   -1.36e-01   1.30e-02  -10.47  < 2e-16 ***
## BPDiaAve                       1.86e-03   3.69e-04    5.04  4.9e-07 ***
## BPSysAve                       9.16e-03   4.64e-04   19.76  < 2e-16 ***
## TVHrsDay0_to_1_hr             -1.00e-01   4.16e-02   -2.42  0.01572 *
## TVHrsDay1_hr                  -1.50e-01   4.05e-02   -3.69  0.00023 ***
## TVHrsDay2_hr                  -7.96e-02   4.00e-02   -1.99  0.04702 *
## TVHrsDay3_hr                  -5.44e-02   4.11e-02   -1.32  0.18592
## TVHrsDay4_hr                  -6.91e-02   4.32e-02   -1.60  0.10960
## TVHrsDayMore_4_hr             -5.85e-02   4.31e-02   -1.36  0.17401
## CompHrsDay0_to_1_hr            9.64e-03   8.30e-03    1.16  0.24564
## CompHrsDay1_hr                 4.21e-02   8.97e-03    4.69  2.8e-06 ***
## CompHrsDay2_hr                 4.82e-02   1.03e-02    4.66  3.3e-06 ***
## CompHrsDay3_hr                 6.27e-02   1.22e-02    5.13  3.0e-07 ***
```

```
## CompHrsDay4_hr                 7.89e-02   1.67e-02    4.74  2.2e-06 ***
## CompHrsDayMore_4_hr            1.00e-01   1.42e-02    7.04  2.2e-12 ***
## DiabetesYes                    1.64e-01   4.11e-02    3.99  6.8e-05 ***
## HealthGenVgood                 9.59e-02   1.58e-02    6.08  1.3e-09 ***
## HealthGenGood                  1.75e-01   1.64e-02   10.69  < 2e-16 ***
## HealthGenFair                  3.04e-01   2.59e-02   11.74  < 2e-16 ***
## HealthGenPoor                  3.15e-01   6.03e-02    5.22  1.9e-07 ***
## LittleInterestSeveral          3.49e-03   7.96e-03    0.44  0.66103
## LittleInterestMost            -3.41e-02   1.20e-02   -2.85  0.00442 **
## SleepHrsNight                 -4.13e-03   2.21e-03   -1.87  0.06187 .
## Race1Hispanic                 -3.44e-02   2.49e-02   -1.38  0.16655
## Race1Mexican                   1.88e-02   2.30e-02    0.82  0.41286
## Race1White                     4.33e-03   1.78e-02    0.24  0.80816
## Race1Other                    -3.92e-02   2.36e-02   -1.66  0.09700 .
## MaritalStatusLivePartner      -1.02e-01   4.10e-02   -2.47  0.01339 *
## MaritalStatusMarried          -5.46e-03   3.86e-02   -0.14  0.88761
## MaritalStatusNeverMarried     -1.48e-01   3.84e-02   -3.86  0.00011 ***
## MaritalStatusSeparated        -2.07e-02   6.41e-02   -0.32  0.74643
## MaritalStatusWidowed           1.88e-03   9.00e-02    0.02  0.98333
## Age:SmokeNowYes                1.19e-03   3.01e-04    3.97  7.2e-05 ***
## Age:BPDiaAve                  -2.99e-05   8.88e-06   -3.36  0.00078 ***
## Age:BPSysAve                  -1.54e-04   8.45e-06  -18.21  < 2e-16 ***
## Age:TVHrsDay0_to_1_hr          2.22e-03   9.75e-04    2.28  0.02271 *
## Age:TVHrsDay1_hr               3.44e-03   9.46e-04    3.64  0.00028 ***
## Age:TVHrsDay2_hr               2.62e-03   9.27e-04    2.83  0.00469 **
## Age:TVHrsDay3_hr               2.72e-03   9.38e-04    2.89  0.00381 **
## Age:TVHrsDay4_hr               2.83e-03   9.67e-04    2.93  0.00344 **
## Age:TVHrsDayMore_4_hr          2.67e-03   9.59e-04    2.79  0.00537 **
## Age:DiabetesYes               -1.40e-03   6.97e-04   -2.01  0.04474 *
## Age:HealthGenVgood            -9.48e-04   3.94e-04   -2.41  0.01614 *
## Age:HealthGenGood             -1.17e-03   3.97e-04   -2.95  0.00316 **
## Age:HealthGenFair             -3.53e-03   5.46e-04   -6.47  1.1e-10 ***
## Age:HealthGenPoor             -3.23e-03   1.13e-03   -2.86  0.00426 **
## Age:Race1Hispanic              7.36e-04   6.24e-04    1.18  0.23840
## Age:Race1Mexican              -1.14e-05   6.31e-04   -0.02  0.98557
## Age:Race1White                -6.24e-04   4.29e-04   -1.46  0.14557
## Age:Race1Other                -1.07e-03   5.85e-04   -1.82  0.06878 .
## Age:MaritalStatusLivePartner   1.56e-03   8.69e-04    1.79  0.07286 .
## Age:MaritalStatusMarried      -7.48e-04   7.40e-04   -1.01  0.31233
## Age:MaritalStatusNeverMarried  3.64e-03   7.87e-04    4.63  3.8e-06 ***
## Age:MaritalStatusSeparated    -3.73e-04   1.37e-03   -0.27  0.78483
## Age:MaritalStatusWidowed      -9.50e-04   1.36e-03   -0.70  0.48394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.192 on 4776 degrees of freedom
## Multiple R-squared:  0.506,  Adjusted R-squared:  0.501
## F-statistic:   86 on 57 and 4776 DF,  p-value: <2e-16
```

We conduct an anova test to check if we should keep `Age:TVHrsDay` or not as the individual p-values are not significant.

```
fit_int_aic_with = with(imp, lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  + Age:TVHrsDay  +  Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus))

# removing Age:TVHrsDay
fit_int_aic_without =with(imp, lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  +  Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus))
anova(fit_int_aic_without, fit_int_aic_with)
```

```
##    test statistic df1 df2 dfcom p.value    riv
##  2 ~~ 1     2.172   6 814  4776 0.04367 0.1529
```

The p-value is less than 0.05, so for now, we will keep `Age:TVHrsDay`.

We also notice from the model that the the p-values for `Race1` categories and their interaction terms are not significant, and the estimates are low. Perhaps we can remove Race1, and the interaction term.

```
fit_int_aic_with = with(imp, lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  + Age:TVHrsDay  +  Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus))

# removing Age:Race1 and interaction
fit_int_aic_without =with(imp, lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  +  Age:Diabetes +   Age:HealthGen +  Age:MaritalStatus))
anova(fit_int_aic_without, fit_int_aic_with)
```

```
##    test statistic df1  df2 dfcom  p.value    riv
##  2 ~~ 1     4.669  14 1222  4776 2.32e-08 0.2088
```

The p-value is very low, less than 0.01, so we reject the null hypothesis, and will keep `Race1` and the related interaction term.

So our final model with the addition of interactions is `final_int`, and is given below:

```
final_int = with(imp, lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow +
  BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve
  + Age:TVHrsDay  +  Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus))
summary(final_int$analyses[[1]])
```

```
##
```

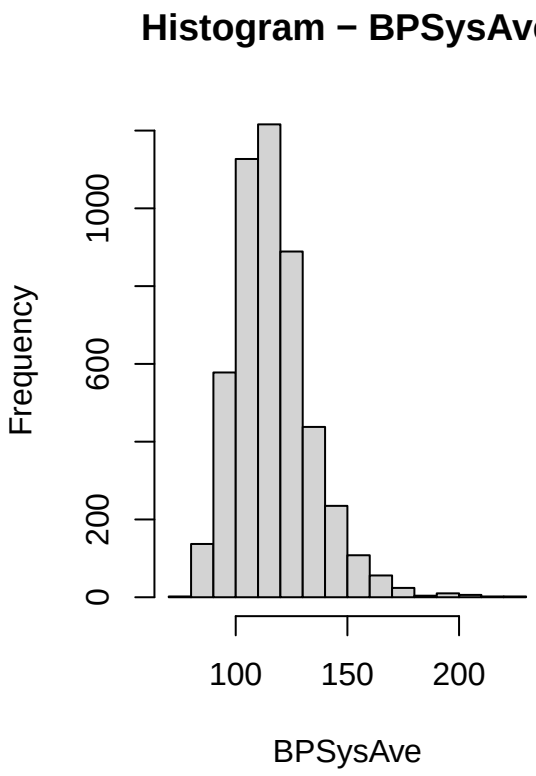
```
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve +
##     BPSysAve + TVHrsDay + CompHrsDay + Diabetes + HealthGen +
##     LittleInterest + SleepHrsNight + Race1 + MaritalStatus +
##     Age:SmokeNow + Age:BPDiaAve + Age:BPSysAve + Age:TVHrsDay +
##     Age:Diabetes + Age:HealthGen + Age:Race1 + Age:MaritalStatus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6386 -0.1323 -0.0062  0.1221  1.0231
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.08e+00   7.89e-02   26.43  < 2e-16 ***
## Age                           2.06e-02   1.60e-03   12.82  < 2e-16 ***
## AlcoholYear                  -2.81e-04   3.05e-05   -9.24  < 2e-16 ***
## SmokeNowYes                  -1.36e-01   1.30e-02  -10.47  < 2e-16 ***
## BPDiaAve                      1.86e-03   3.69e-04    5.04  4.9e-07 ***
## BPSysAve                      9.16e-03   4.64e-04   19.76  < 2e-16 ***
## TVHrsDay0_to_1_hr            -1.00e-01   4.16e-02   -2.42  0.01572 *
## TVHrsDay1_hr                 -1.50e-01   4.05e-02   -3.69  0.00023 ***
## TVHrsDay2_hr                 -7.96e-02   4.00e-02   -1.99  0.04702 *
## TVHrsDay3_hr                 -5.44e-02   4.11e-02   -1.32  0.18592
## TVHrsDay4_hr                 -6.91e-02   4.32e-02   -1.60  0.10960
## TVHrsDayMore_4_hr            -5.85e-02   4.31e-02   -1.36  0.17401
## CompHrsDay0_to_1_hr           9.64e-03   8.30e-03    1.16  0.24564
## CompHrsDay1_hr                4.21e-02   8.97e-03    4.69  2.8e-06 ***
## CompHrsDay2_hr                4.82e-02   1.03e-02    4.66  3.3e-06 ***
## CompHrsDay3_hr                6.27e-02   1.22e-02    5.13  3.0e-07 ***
## CompHrsDay4_hr                7.89e-02   1.67e-02    4.74  2.2e-06 ***
## CompHrsDayMore_4_hr           1.00e-01   1.42e-02    7.04  2.2e-12 ***
## DiabetesYes                   1.64e-01   4.11e-02    3.99  6.8e-05 ***
## HealthGenVgood                9.59e-02   1.58e-02    6.08  1.3e-09 ***
## HealthGenGood                 1.75e-01   1.64e-02   10.69  < 2e-16 ***
## HealthGenFair                 3.04e-01   2.59e-02   11.74  < 2e-16 ***
## HealthGenPoor                 3.15e-01   6.03e-02    5.22  1.9e-07 ***
## LittleInterestSeveral         3.49e-03   7.96e-03    0.44  0.66103
## LittleInterestMost           -3.41e-02   1.20e-02   -2.85  0.00442 **
## SleepHrsNight                -4.13e-03   2.21e-03   -1.87  0.06187 .
## Race1Hispanic                -3.44e-02   2.49e-02   -1.38  0.16655
## Race1Mexican                  1.88e-02   2.30e-02    0.82  0.41286
## Race1White                    4.33e-03   1.78e-02    0.24  0.80816
## Race1Other                   -3.92e-02   2.36e-02   -1.66  0.09700 .
## MaritalStatusLivePartner     -1.02e-01   4.10e-02   -2.47  0.01339 *
## MaritalStatusMarried         -5.46e-03   3.86e-02   -0.14  0.88761
## MaritalStatusNeverMarried    -1.48e-01   3.84e-02   -3.86  0.00011 ***
## MaritalStatusSeparated       -2.07e-02   6.41e-02   -0.32  0.74643
## MaritalStatusWidowed          1.88e-03   9.00e-02    0.02  0.98333
## Age:SmokeNowYes               1.19e-03   3.01e-04    3.97  7.2e-05 ***
## Age:BPDiaAve                 -2.99e-05   8.88e-06   -3.36  0.00078 ***
## Age:BPSysAve                 -1.54e-04   8.45e-06  -18.21  < 2e-16 ***
## Age:TVHrsDay0_to_1_hr         2.22e-03   9.75e-04    2.28  0.02271 *
## Age:TVHrsDay1_hr              3.44e-03   9.46e-04    3.64  0.00028 ***
## Age:TVHrsDay2_hr              2.62e-03   9.27e-04    2.83  0.00469 **
```

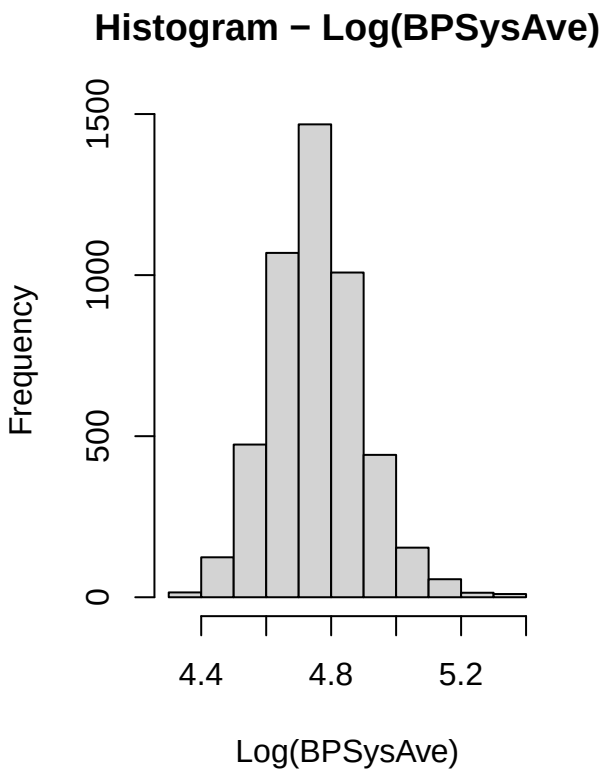
```
## Age:TVHrsDay3_hr                2.72e-03   9.38e-04    2.89  0.00381 **
## Age:TVHrsDay4_hr                2.83e-03   9.67e-04    2.93  0.00344 **
## Age:TVHrsDayMore_4_hr           2.67e-03   9.59e-04    2.79  0.00537 **
## Age:DiabetesYes                -1.40e-03   6.97e-04   -2.01  0.04474 *
## Age:HealthGenVgood             -9.48e-04   3.94e-04   -2.41  0.01614 *
## Age:HealthGenGood              -1.17e-03   3.97e-04   -2.95  0.00316 **
## Age:HealthGenFair              -3.53e-03   5.46e-04   -6.47  1.1e-10 ***
## Age:HealthGenPoor              -3.23e-03   1.13e-03   -2.86  0.00426 **
## Age:Race1Hispanic               7.36e-04   6.24e-04    1.18  0.23840
## Age:Race1Mexican               -1.14e-05   6.31e-04   -0.02  0.98557
## Age:Race1White                 -6.24e-04   4.29e-04   -1.46  0.14557
## Age:Race1Other                 -1.07e-03   5.85e-04   -1.82  0.06878 .
## Age:MaritalStatusLivePartner    1.56e-03   8.69e-04    1.79  0.07286 .
## Age:MaritalStatusMarried       -7.48e-04   7.40e-04   -1.01  0.31233
## Age:MaritalStatusNeverMarried   3.64e-03   7.87e-04    4.63  3.8e-06 ***
## Age:MaritalStatusSeparated     -3.73e-04   1.37e-03   -0.27  0.78483
## Age:MaritalStatusWidowed       -9.50e-04   1.36e-03   -0.70  0.48394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.192 on 4776 degrees of freedom
## Multiple R-squared:  0.506,  Adjusted R-squared:  0.501
## F-statistic:   86 on 57 and 4776 DF,  p-value: <2e-16
```

Data Transformations Now we will consider if our model will benefit from any data transformations. The two numerical variables we are most suspicious of having skewed distributions are those related to blood pressure. Let's graph each of their distributions. We are using the imputed data, but only one of the 5 imputed datasets.

BPSysAve Transformation:

```
par(mfrow=c(1,2))
hist(imp_df[imp_df$.imp == 1, ]$BPSysAve,
     main = "Histogram - BPSysAve",
     xlab = "BPSysAve")
hist(log(imp_df[imp_df$.imp == 1, ]$BPSysAve),
     main = "Histogram - Log(BPSysAve)",
     xlab = "Log(BPSysAve)")
```

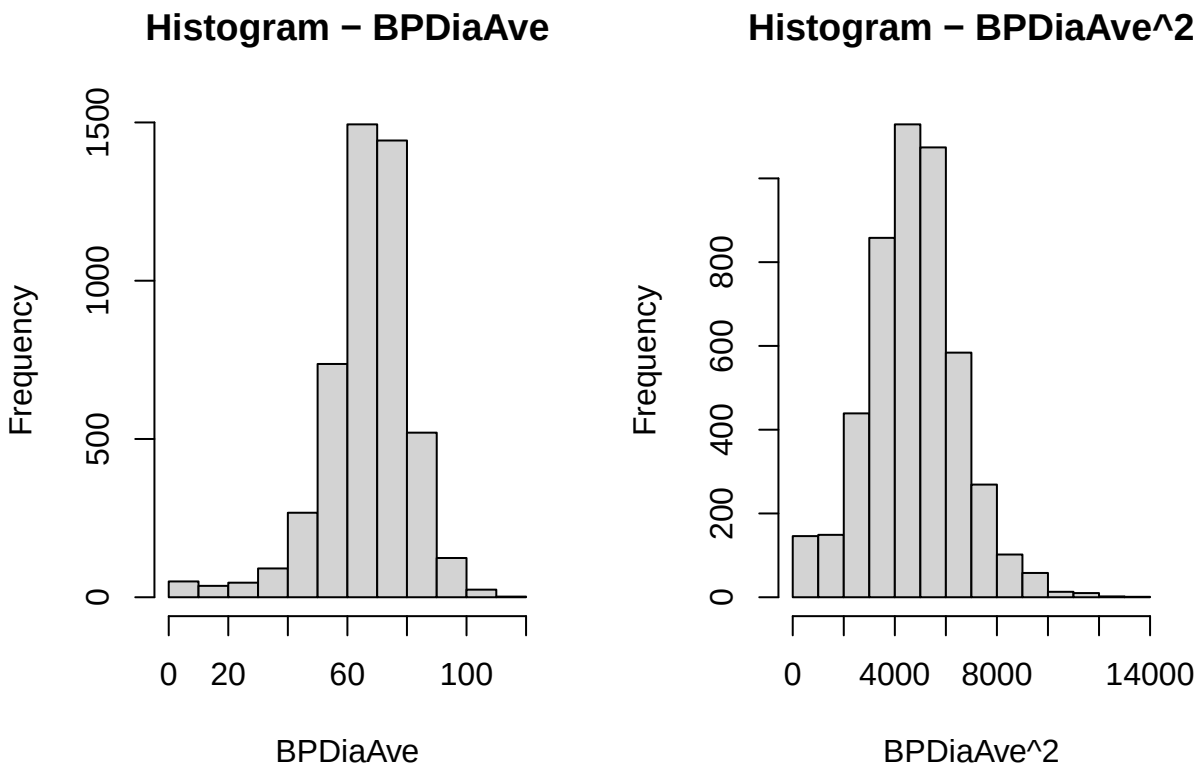




We notice that the non-transformed `BPSysAve` data appears skewed right, so we apply a log transformation, and the distribution improves.

BPDiasAve Trnasformation:

```
par(mfrow=c(1,2))
hist(imp_df[imp_df$.imp == 1, ]$BPDiaAve,
     main = "Histogram - BPDiaAve",
     xlab = "BPDiaAve")
hist((imp_df[imp_df$.imp == 1, ]$BPDiaAve)^2,
     main = "Histogram - BPDiaAve^2",
     xlab = "BPDiaAve^2")
```



In the case of `BPDiaAve`, we notice that the non-transformed data appeared skewed left, so we apply a log transformation, and the distribution improves somewhat, though it is a bit right skewed, but less skewed overall.

Now we will consider our model with the log transformed BP measures:

```
final_trns = with(imp, lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
  I(BPDiaAve^2) + log1p(BPSysAve) + TVHrsDay + CompHrsDay +
  Diabetes +  HealthGen +  LittleInterest +
  SleepHrsNight +  Race1  +  MaritalStatus +  Age:SmokeNow +  Age:I(BPDiaAve^2) +
    Age:log1p(BPSysAve)  + Age:TVHrsDay +  Age:Diabetes +  Age:HealthGen +  Age:Race1 +
    Age:MaritalStatus))

#Note: we apply log1p to get the natural log of (1+BPSysAve), as we were getting infinity
#for 0 and NaN for negative values of BPSysAve with log(BPSysAve)

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

```
##
## Call:
## lm(formula = log(BMI) ~ Age + AlcoholYear + SmokeNow + I(BPDiaAve^2) +
##     log1p(BPSysAve) + TVHrsDay + CompHrsDay + Diabetes + HealthGen +
##     LittleInterest + SleepHrsNight + Race1 + MaritalStatus +
##     Age:SmokeNow + Age:I(BPDiaAve^2) + Age:log1p(BPSysAve) +
##     Age:TVHrsDay + Age:Diabetes + Age:HealthGen + Age:Race1 +
##     Age:MaritalStatus)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6357 -0.1332 -0.0057  0.1225  1.0265
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.50e+00   2.54e-01   -5.91  3.7e-09 ***
## Age                            8.27e-02   4.99e-03   16.58  < 2e-16 ***
## AlcoholYear                   -2.75e-04   3.04e-05   -9.03  < 2e-16 ***
## SmokeNowYes                   -1.34e-01   1.30e-02  -10.33  < 2e-16 ***
## I(BPDiaAve^2)                  2.23e-05   3.75e-06    5.96  2.7e-09 ***
## log1p(BPSysAve)                9.81e-01   5.31e-02   18.48  < 2e-16 ***
## TVHrsDay0_to_1_hr             -1.02e-01   4.16e-02   -2.46  0.01397 *
## TVHrsDay1_hr                  -1.51e-01   4.05e-02   -3.73  0.00020 ***
## TVHrsDay2_hr                  -8.19e-02   4.00e-02   -2.05  0.04088 *
## TVHrsDay3_hr                  -5.75e-02   4.11e-02   -1.40  0.16172
## TVHrsDay4_hr                  -7.09e-02   4.32e-02   -1.64  0.10063
## TVHrsDayMore_4_hr             -6.23e-02   4.30e-02   -1.45  0.14794
## CompHrsDay0_to_1_hr            9.90e-03   8.31e-03    1.19  0.23331
## CompHrsDay1_hr                 4.24e-02   8.97e-03    4.72  2.4e-06 ***
## CompHrsDay2_hr                 4.83e-02   1.03e-02    4.67  3.1e-06 ***
## CompHrsDay3_hr                 6.29e-02   1.22e-02    5.15  2.7e-07 ***
## CompHrsDay4_hr                 7.71e-02   1.67e-02    4.63  3.8e-06 ***
## CompHrsDayMore_4_hr            1.01e-01   1.42e-02    7.07  1.8e-12 ***
## DiabetesYes                    1.67e-01   4.11e-02    4.06  5.0e-05 ***
## HealthGenVgood                 9.52e-02   1.58e-02    6.04  1.7e-09 ***
## HealthGenGood                  1.74e-01   1.64e-02   10.60  < 2e-16 ***
## HealthGenFair                  3.01e-01   2.59e-02   11.64  < 2e-16 ***
## HealthGenPoor                  3.14e-01   6.02e-02    5.21  2.0e-07 ***
## LittleInterestSeveral          3.20e-03   7.96e-03    0.40  0.68802
## LittleInterestMost            -3.34e-02   1.20e-02   -2.79  0.00531 **
## SleepHrsNight                 -4.02e-03   2.21e-03   -1.82  0.06940 .
## Race1Hispanic                 -3.59e-02   2.49e-02   -1.44  0.14992
## Race1Mexican                   1.97e-02   2.30e-02    0.86  0.39030
## Race1White                     1.83e-03   1.78e-02    0.10  0.91813
## Race1Other                    -4.02e-02   2.36e-02   -1.70  0.08921 .
## MaritalStatusLivePartner      -9.83e-02   4.11e-02   -2.39  0.01668 *
## MaritalStatusMarried          -2.08e-03   3.86e-02   -0.05  0.95699
## MaritalStatusNeverMarried     -1.44e-01   3.84e-02   -3.75  0.00018 ***
## MaritalStatusSeparated        -1.72e-02   6.41e-02   -0.27  0.78819
## MaritalStatusWidowed           2.30e-02   9.01e-02    0.26  0.79840
## Age:SmokeNowYes                1.17e-03   3.01e-04    3.89  0.00010 ***
## Age:I(BPDiaAve^2)             -3.57e-07   8.30e-08   -4.30  1.7e-05 ***
## Age:log1p(BPSysAve)           -1.69e-02   1.03e-03  -16.33  < 2e-16 ***
## Age:TVHrsDay0_to_1_hr          2.25e-03   9.75e-04    2.31  0.02101 *
## Age:TVHrsDay1_hr               3.47e-03   9.46e-04    3.67  0.00025 ***
## Age:TVHrsDay2_hr               2.65e-03   9.27e-04    2.86  0.00424 **
## Age:TVHrsDay3_hr               2.77e-03   9.38e-04    2.95  0.00320 **
## Age:TVHrsDay4_hr               2.87e-03   9.67e-04    2.97  0.00302 **
## Age:TVHrsDayMore_4_hr          2.72e-03   9.58e-04    2.84  0.00455 **
## Age:DiabetesYes               -1.45e-03   6.98e-04   -2.08  0.03747 *
## Age:HealthGenVgood            -9.22e-04   3.94e-04   -2.34  0.01925 *
## Age:HealthGenGood             -1.13e-03   3.97e-04   -2.85  0.00433 **
```

```
## Age:HealthGenFair             -3.47e-03   5.46e-04   -6.36  2.2e-10 ***
## Age:HealthGenPoor             -3.18e-03   1.13e-03   -2.82  0.00487 **
## Age:Race1Hispanic              7.25e-04   6.24e-04    1.16  0.24575
## Age:Race1Mexican              -5.28e-05   6.30e-04   -0.08  0.93323
## Age:Race1White                -6.26e-04   4.29e-04   -1.46  0.14427
## Age:Race1Other                -1.09e-03   5.85e-04   -1.87  0.06191 .
## Age:MaritalStatusLivePartner   1.48e-03   8.69e-04    1.71  0.08785 .
## Age:MaritalStatusMarried      -8.18e-04   7.40e-04   -1.11  0.26920
## Age:MaritalStatusNeverMarried  3.55e-03   7.87e-04    4.51  6.7e-06 ***
## Age:MaritalStatusSeparated    -4.26e-04   1.37e-03   -0.31  0.75510
## Age:MaritalStatusWidowed      -1.32e-03   1.36e-03   -0.97  0.33232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.192 on 4776 degrees of freedom
## Multiple R-squared:  0.507,  Adjusted R-squared:  0.501
## F-statistic:   86 on 57 and 4776 DF,  p-value: <2e-16
```

We notice that our R-Squared is the same as without the transformed data, but this will represent our final transformation model and we will more fully compare all three models in the next section.

Final Models Based on our analysis, these are the final 3 models that we will conduct further outlier and diagnostic assessments on:

Additive Model log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay + TotChol + Diabetes + HealthGen + LittleInterest + SleepHrsNight + Race1 + MaritalStatus

Model with Interactions log(BMI) ~ Age + AlcoholYear + SmokeNow + BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay + Diabetes + HealthGen + LittleInterest + SleepHrsNight + Race1 + MaritalStatus + Age:SmokeNow + Age:BPDiaAve + Age:BPSysAve + Age:TVHrsDay + Age:Diabetes + Age:HealthGen + Age:Race1 + Age:MaritalStatus

Model with Transformation & Interactions log(BMI) ~ Age + AlcoholYear + SmokeNow + I(BPDiaAveˆ2) + log1p(BPSysAve) + TVHrsDay + CompHrsDay + Diabetes + HealthGen + LittleInterest + SleepHrsNight + Race1 + MaritalStatus + Age:SmokeNow + Age:I(BPDiaAveˆ2) + Age:log1p(BPSysAve) + Age:TVHrsDay + Age:Diabetes + Age:HealthGen + Age:Race1 + Age:MaritalStatus

We arrived at these models using the following steps: 1. Ruled out many variables by grouping them into similar domains and keeping only one representative variable in case we detected collinearity. This reduced the number of possible predictors from 75 to 21. 2. Dealt with missing data using multiple imputation using the `mice` package. 3. Fitted a first version of an additive model with the selected predictors and detected violations of equal variance and normality. 4. Log-transformed the response in an attempt to alleviate the constant variance violation. 5. Performed AIC and BIC stepwise processes and conducted anova tests to reduce the number of predictors further. 6. Tested interaction of `Age` with other predictors as well as transformations of `BPDiaAve`, `BPSysAve`.

Results

Outlier Assessment For each of the 3 models, we would now like to see if there are any outliers that are influential, and that are having a large effect on our regressions. We will conduct this assessment using Cook's Distance.

```r
#finding influential observations for each model

indexs_trns = which(cooks.distance(final_trns$analyses[[1]]) > 4 /
                      length(cooks.distance(final_trns$analyses[[1]])))

indexs_int = which(cooks.distance(final_int$analyses[[1]]) > 4 /
                     length(cooks.distance(final_int$analyses[[1]])))

indexs_add = which(cooks.distance(final_add$analyses[[1]]) > 4 /
                     length(cooks.distance(final_add$analyses[[1]])))


#removing influential observations for each model using the first imputed dataset

imp_df_1 = imp_df[imp_df$.imp == 1, ]
imp_df_1_trns_rm = imp_df_1[-indexs_trns, ]
imp_df_1_int_rm = imp_df_1[-indexs_int, ]
imp_df_1_add_rm = imp_df_1[-indexs_add, ]
```

Next, we will fit the 3 models afer removing the influential observations.

```r
#Final_add Model fit with influential observations removed

final_add = lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
                 BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
                 TotChol +  Diabetes +  HealthGen +  LittleInterest +
                 SleepHrsNight +  Race1  +  MaritalStatus, data = imp_df_1_add_rm)
```

```r
#Final_int Model fit with influential observations removed

final_int = lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow + BPDiaAve + BPSysAve +
                 TVHrsDay + CompHrsDay + Diabetes +  HealthGen +
                 LittleInterest + SleepHrsNight +  Race1  +  MaritalStatus +
                 Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve  + Age:TVHrsDay +
                 Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus,
               data = imp_df_1_int_rm)
```

```r
#Final_trns Model fit with influential observations removed

final_trns = lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow + I(BPDiaAve^2) +
                  log1p(BPSysAve) + TVHrsDay + CompHrsDay + Diabetes +
                  HealthGen +  LittleInterest + SleepHrsNight +  Race1  +
                  MaritalStatus +  Age:SmokeNow +  Age:I(BPDiaAve^2) +
                  Age:log1p(BPSysAve)  + Age:TVHrsDay +  Age:Diabetes +
                  Age:HealthGen +  Age:Race1 +  Age:MaritalStatus,
                data = imp_df_1_trns_rm)
```

Model Diagnostics

Now that we have removed the influential observations, we would like to conduct different diagnostic tests and see how the 3 models perform.

Diagnostic Tests

```r
#Functions to calculate various diagnostics
get_adj_r2 = function(model) {
  summary(model)$adj.r.squared
}

calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

get_shapiro = function(model) {
  shapiro.test(resid(model))$p.value
}

get_bp = function(model) {
  unname(bptest(model)$p.value)
}
```

```r
# model comparison results
row_names = c("Additive Model", "Model with Interactions", "Model with Transformations")
col_names = c("Model", "BP Test", "Shapiro-Wilk Test", "Adjusted R2")

bp_results = c(get_bp(final_add), get_bp(final_int), get_bp(final_trns))
shapiro_results = c(get_shapiro(final_add), get_shapiro(final_int), get_shapiro(final_trns))
r2_results = c(get_adj_r2(final_add), get_adj_r2(final_int), get_adj_r2(final_trns))

results_table = cbind(row_names, signif(bp_results,6), signif(shapiro_results,6), round(r2_results,6))
knitr::kable(results_table, col.names = col_names, caption = "Model Comparison")
```

Table 3: Model Comparison

Model	BP Test	Shapiro-Wilk Test	Adjusted R2
Additive Model	1.62383e-36	5.56921e-13	0.490983
Model with Interactions	2.60423e-36	5.26166e-09	0.568574
Model with Transformations	1.28736e-34	5.47327e-09	0.569475

As we see from the table above, we have been able to build 3 models which can explain more than 49% of the variability in the dataset, according to adjusted R^2. As far as adjusted R^2 is concerned, the interaction model and the transformed model perform better than the additive model, by explaining about 7.5% more of the variability. All models have issues with heteroskedasticity and violation of normality assumptions according to the Breusch-Pagan and Shapiro-Wilk tests, respectively.

Next, we'll split the data into testing and training sets to see how these 3 models compare when we evaluate the LOOCV-RMSE on the training (unseen) data.

```r
set.seed(420)
#Splitting the data for the Final_add Model
add_idx  = sample(nrow(imp_df_1_add_rm), size = trunc(0.80 * nrow(imp_df_1_add_rm)))
add_trn_data = imp_df_1_add_rm[add_idx, ]
add_tst_data = imp_df_1_add_rm[-add_idx, ]

#Splitting the data for the Final_int Model
int_idx  = sample(nrow(imp_df_1_int_rm), size = trunc(0.80 * nrow(imp_df_1_int_rm)))
```

```
int_trn_data = imp_df_1_int_rm[int_idx, ]
int_tst_data = imp_df_1_int_rm[-int_idx, ]

#Splitting the data for the Final_trns Model
trns_idx  = sample(nrow(imp_df_1_trns_rm), size = trunc(0.80 * nrow(imp_df_1_trns_rm)))
trns_trn_data = imp_df_1_trns_rm[trns_idx, ]
trns_tst_data = imp_df_1_trns_rm[-trns_idx, ]
```

We will re-fit the models using the training data.

```
#Final_add Model fit with influential observations removed, and using training data

final_add2 = lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow +
                BPDiaAve + BPSysAve + TVHrsDay + CompHrsDay +
                TotChol +  Diabetes +  HealthGen +  LittleInterest +
                SleepHrsNight +  Race1  +  MaritalStatus, data = add_trn_data)
```

```
#Final_int Model fit with influential observations removed, and using training data

final_int2 = lm(log(BMI) ~  Age +  AlcoholYear + SmokeNow + BPDiaAve + BPSysAve +
                TVHrsDay + CompHrsDay + Diabetes +  HealthGen +
                LittleInterest + SleepHrsNight +  Race1  +  MaritalStatus +
                Age:SmokeNow +  Age:BPDiaAve +  Age:BPSysAve  + Age:TVHrsDay +
                Age:Diabetes +   Age:HealthGen +  Age:Race1 +  Age:MaritalStatus,
              data = int_trn_data)
```

```
#Final_add Model fit with influential observations removed, and using training data

final_trns2 = lm(log(BMI) ~ Age +  AlcoholYear + SmokeNow + I(BPDiaAve^2) +
                 log1p(BPSysAve) + TVHrsDay + CompHrsDay + Diabetes +
                 HealthGen +  LittleInterest + SleepHrsNight +  Race1  +
                 MaritalStatus +  Age:SmokeNow +  Age:I(BPDiaAve^2) +
                 Age:log1p(BPSysAve)  + Age:TVHrsDay +  Age:Diabetes +
                 Age:HealthGen +  Age:Race1 +  Age:MaritalStatus,
               data = trns_trn_data)
```

```
# model comparison results
row_names = c("Additive Model", "Model with Interactions", "Model with Transformations")
col_names = c("Model", "LOOCV_RMSE", "Avg Pct Error")


predicted_add2 = exp(predict(final_add2, newdata = add_tst_data))
avg_pct_error_add2 = mean(abs(predicted_add2 - add_tst_data$BMI)
                     /predicted_add2) * 100

predicted_int2 = exp(predict(final_int2, newdata = int_tst_data))
avg_pct_error_int2 = mean(abs(predicted_int2 - int_tst_data$BMI)
                     /predicted_int2) * 100

predicted_trns2 = exp(predict(final_trns2, newdata = trns_tst_data))
avg_pct_error_trns2 = mean(abs(predicted_trns2 - trns_tst_data$BMI)
                     /predicted_trns2) * 100
```

```
rmse_results2 = c(calc_loocv_rmse(final_add2), calc_loocv_rmse(final_int2), calc_loocv_rmse(final_trns2)
avg_pct_error2 = c(avg_pct_error_add2, avg_pct_error_int2, avg_pct_error_trns2)

results_table2 = cbind(row_names, round(rmse_results2,6), round(avg_pct_error2,6))
knitr::kable(results_table2, col.names = col_names, caption = "Model Comparison")
```

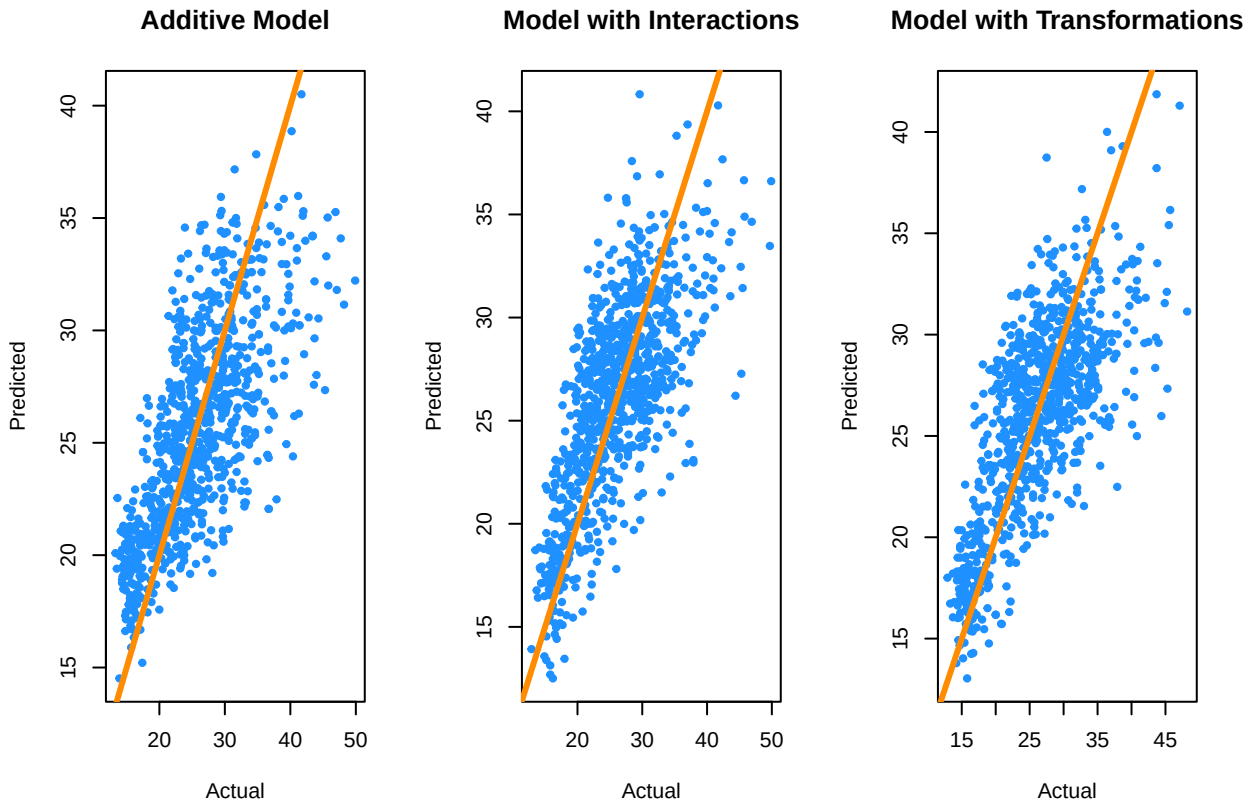
Table 4: Model Comparison

Model	LOOCV_RMSE	Avg Pct Error
Additive Model	0.184747	14.933648
Model with Interactions	0.170929	13.839178
Model with Transformations	0.170808	13.758884

We would like to select a model with the lowest value for LOOCV-RMSE, so it does not overfit or underfit, and well as has the lowest value for Average Percent Error so it is able to perform well while predicting *unseen* data. Based on the above results, although the model with transformations has the lowest values for these metrics, there is not a big difference when compared to the other 2 models, especially the model with interactions.

Let's also look at the plots for the predicted versus the actual values for these 3 models and add the line $y = x$.

```
par(mfrow=c(1,3))
plot(add_tst_data$BMI,
     predicted_add2,
     col = "dodgerblue",
     pch = 20,
     main = "Additive Model",
     xlab = "Actual",
     ylab = "Predicted"
     )
abline(a=0, b=1, col = "darkorange", lwd = 3)

plot(int_tst_data$BMI,
     predicted_int2,
     col = "dodgerblue",
     pch = 20,
     main = "Model with Interactions",
     xlab = "Actual",
     ylab = "Predicted"
     )
abline(a=0, b=1, col = "darkorange", lwd = 3)

plot(trns_tst_data$BMI,
     predicted_trns2,
     col = "dodgerblue",
     pch = 20,
     main = "Model with Transformations",
     xlab = "Actual",
     ylab = "Predicted"
     )
abline(a=0, b=1, col = "darkorange", lwd = 3)
```



We see that the Model with Transformations does better at predicting BMI, however, it does not differ that much from the Model with Interactions. The additive model does not perform as well as the other two models for predicting BMI.

Discussion

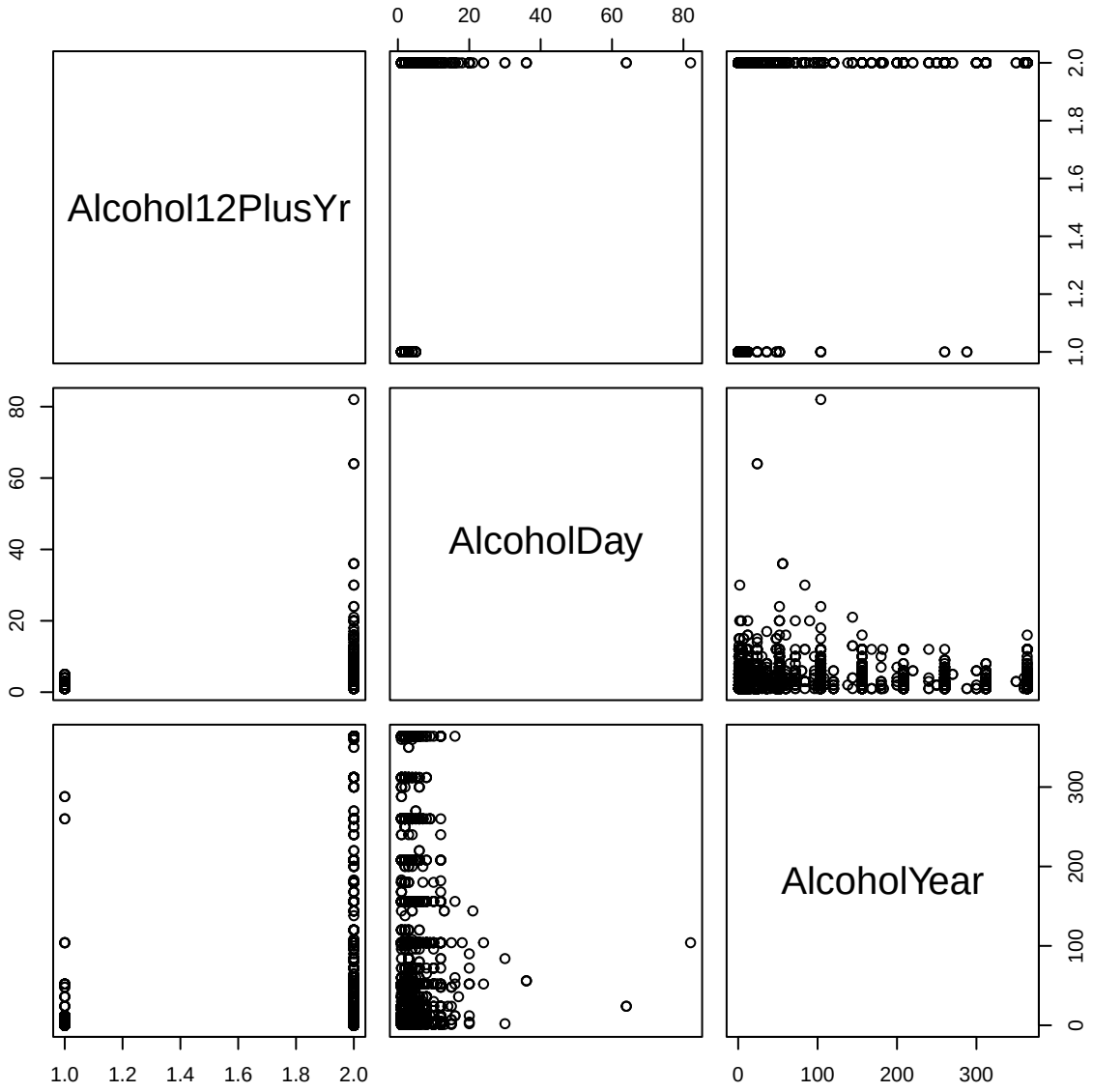
Based on our findings above, we feel that from a prediction and explainability point of view, we would prefer the interaction model, since the transformations do not seem to help much and the interaction model is a little simpler than the model with transformations.

Appendix

Variable Selection

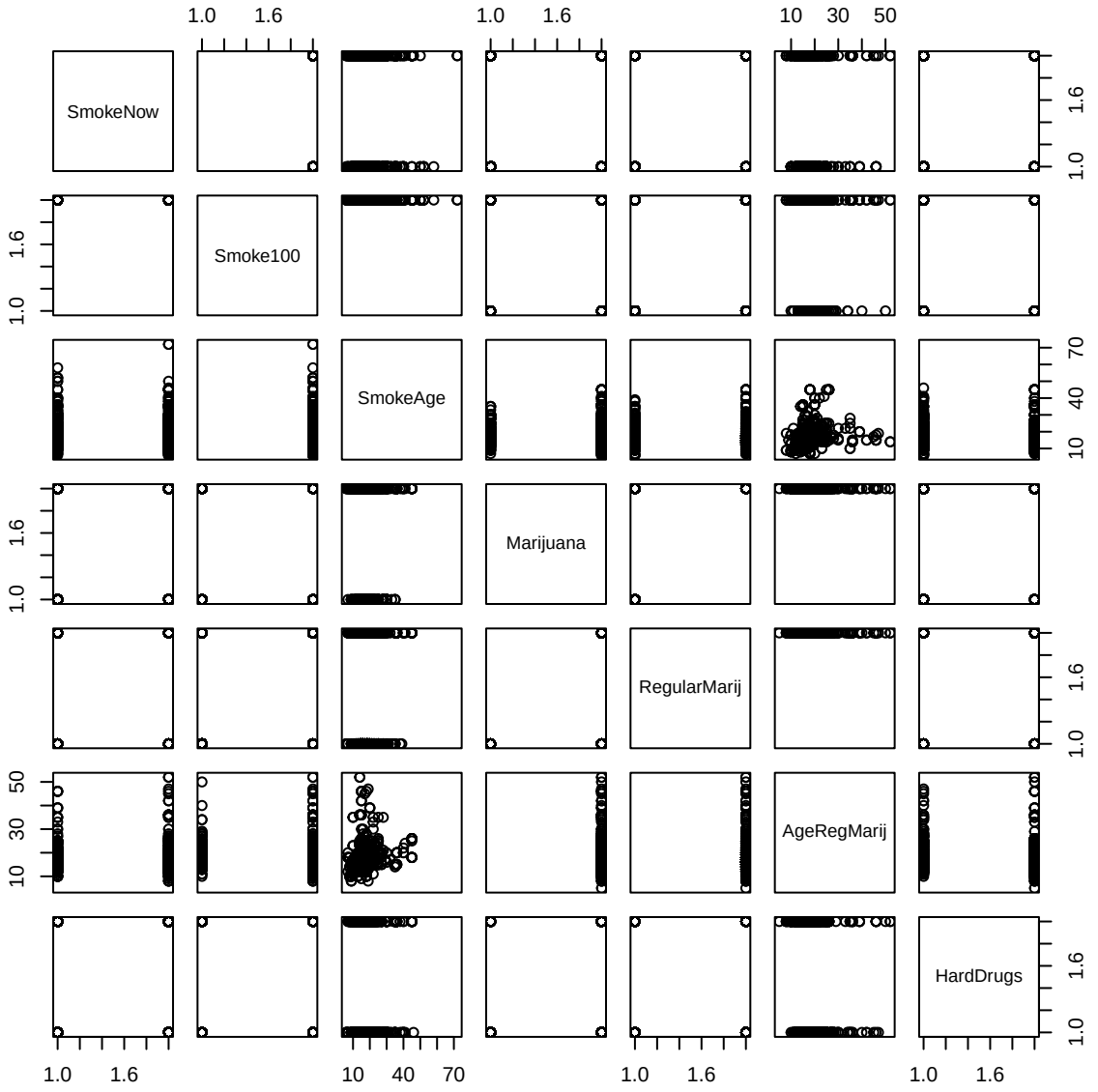
Additional `pairs()` Plots for Collinearity Assessment Alcohol related variables:

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



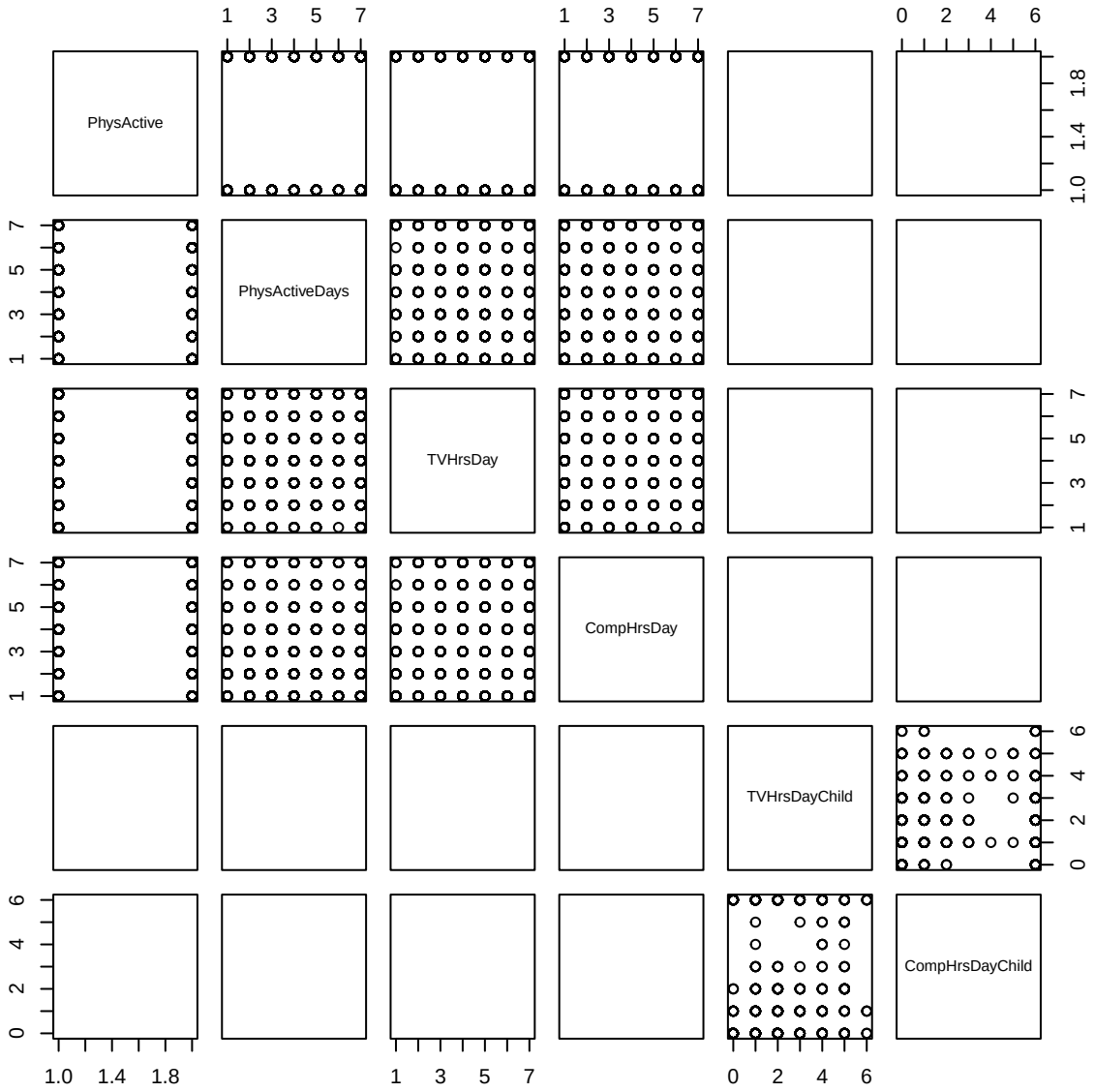
Smoking and Drug related variables:

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



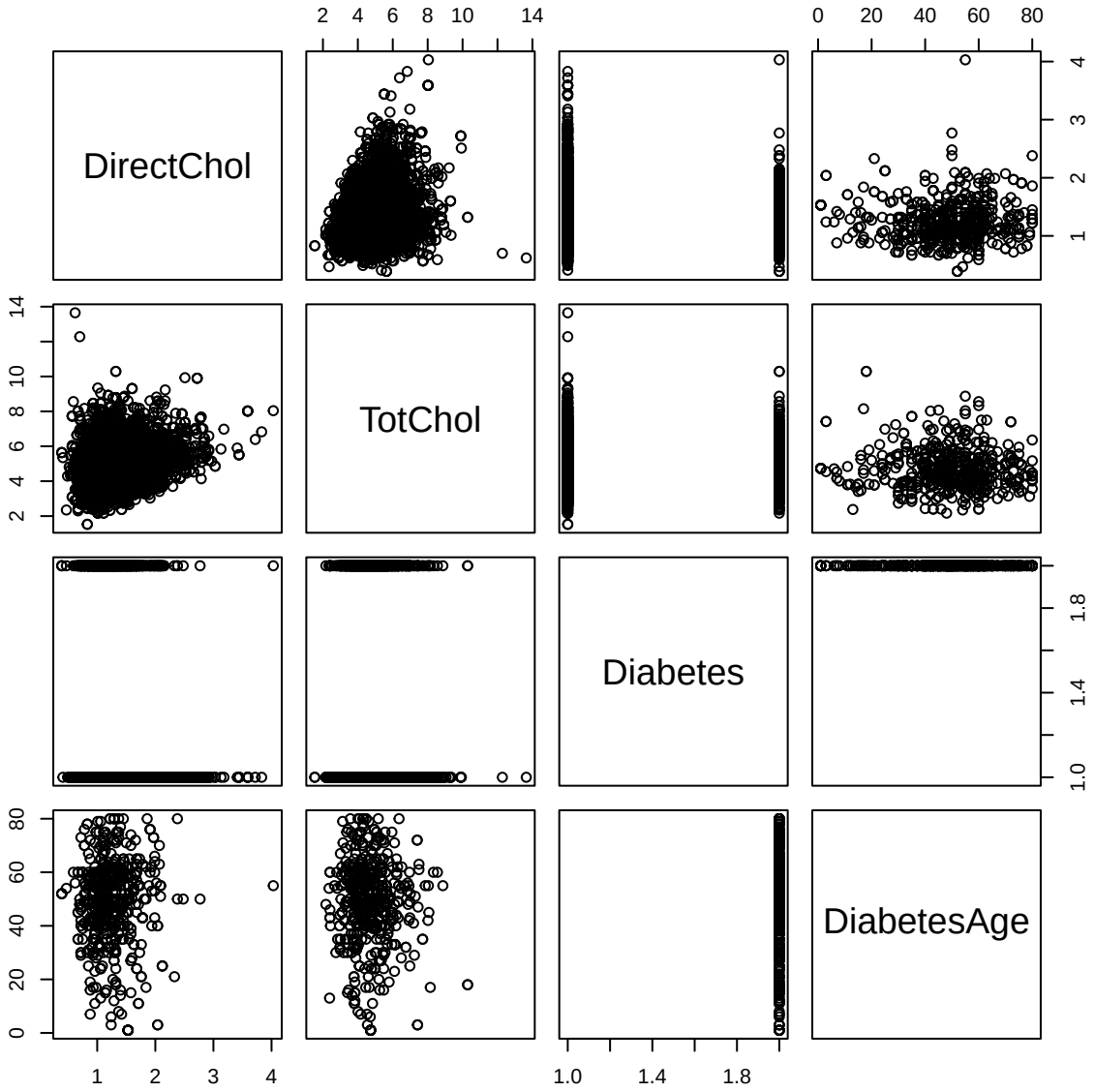
Lifestyle related variables:

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



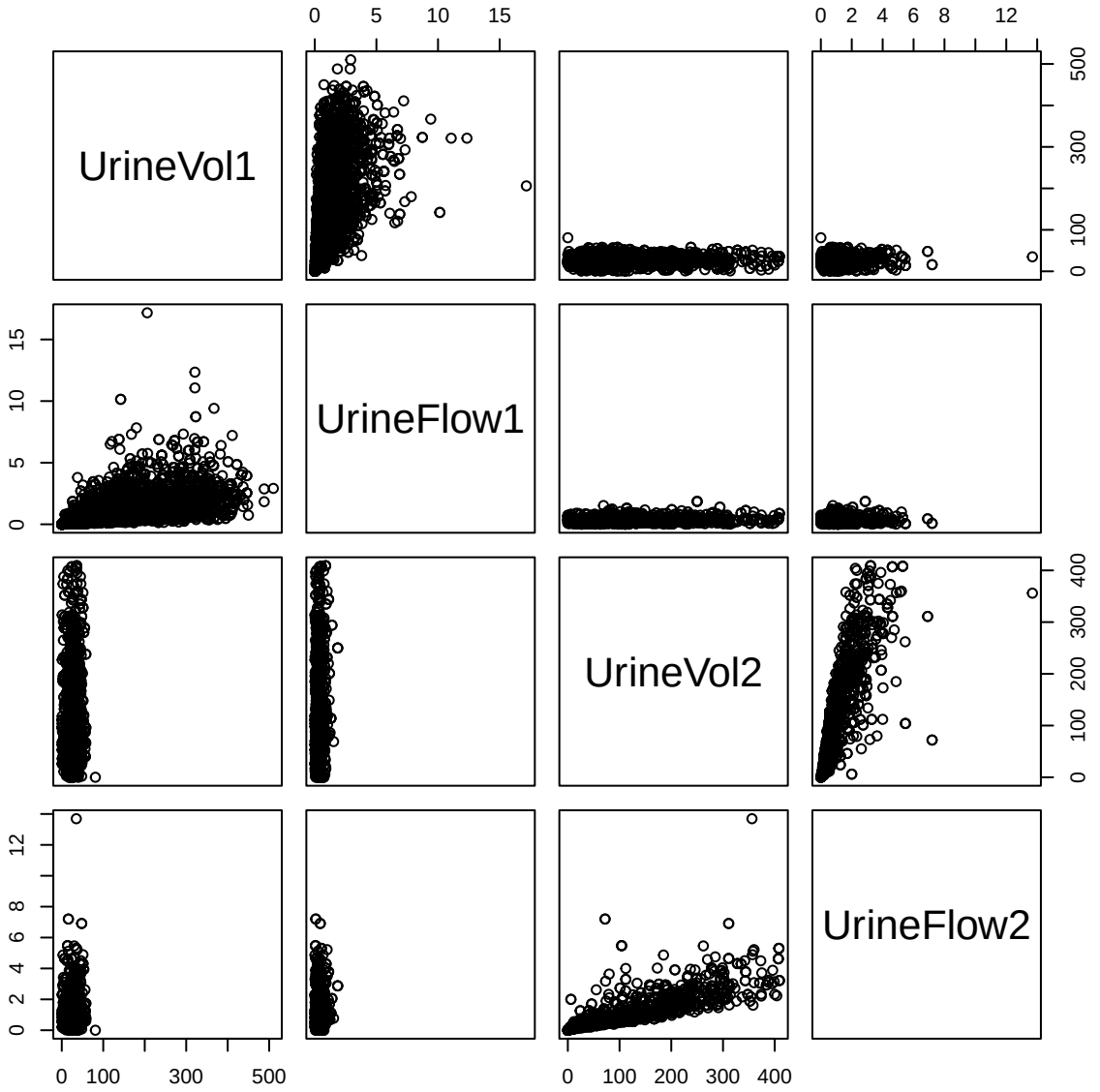
Cholesterol related variables:

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



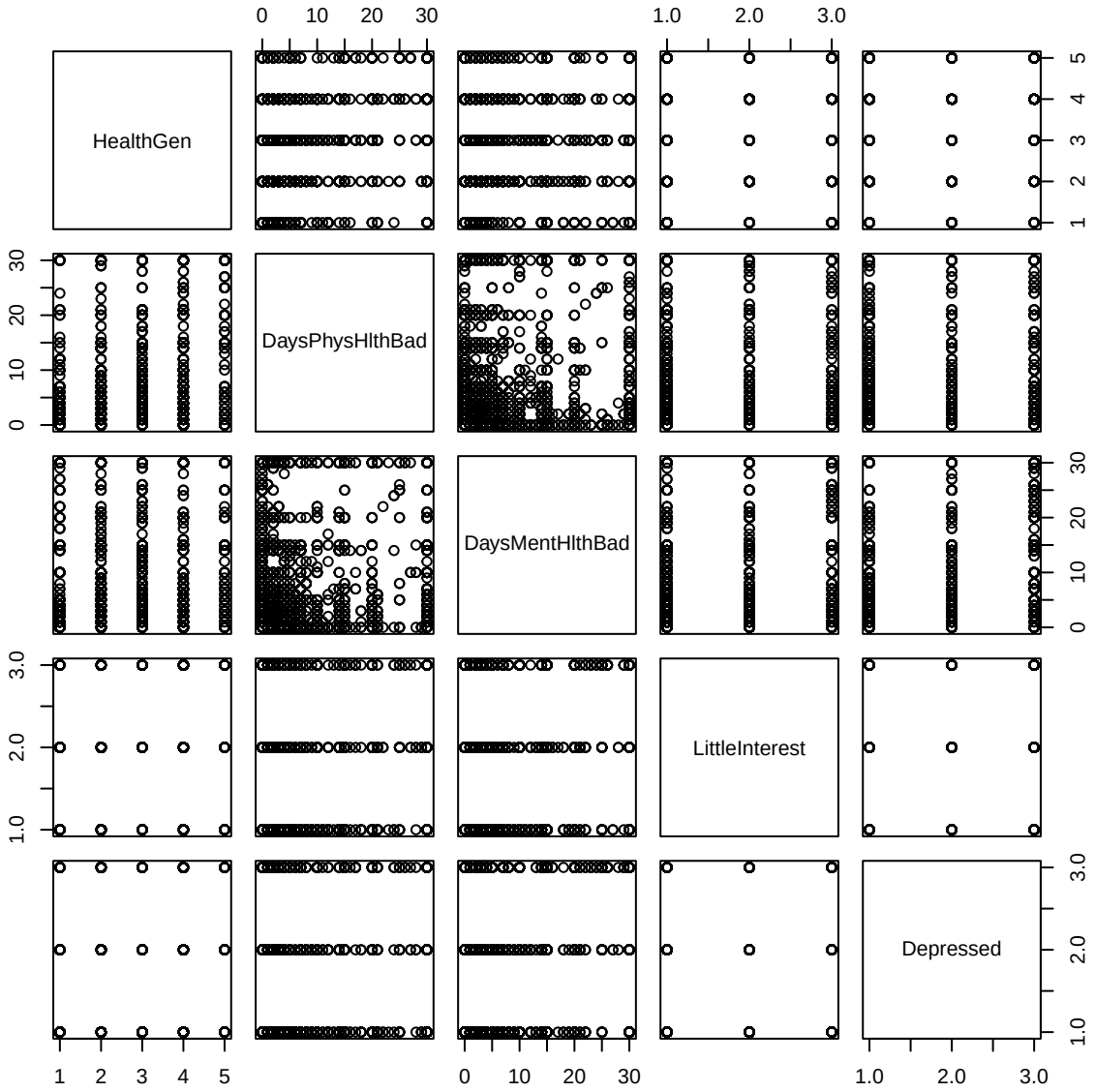
Urine related variables:

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

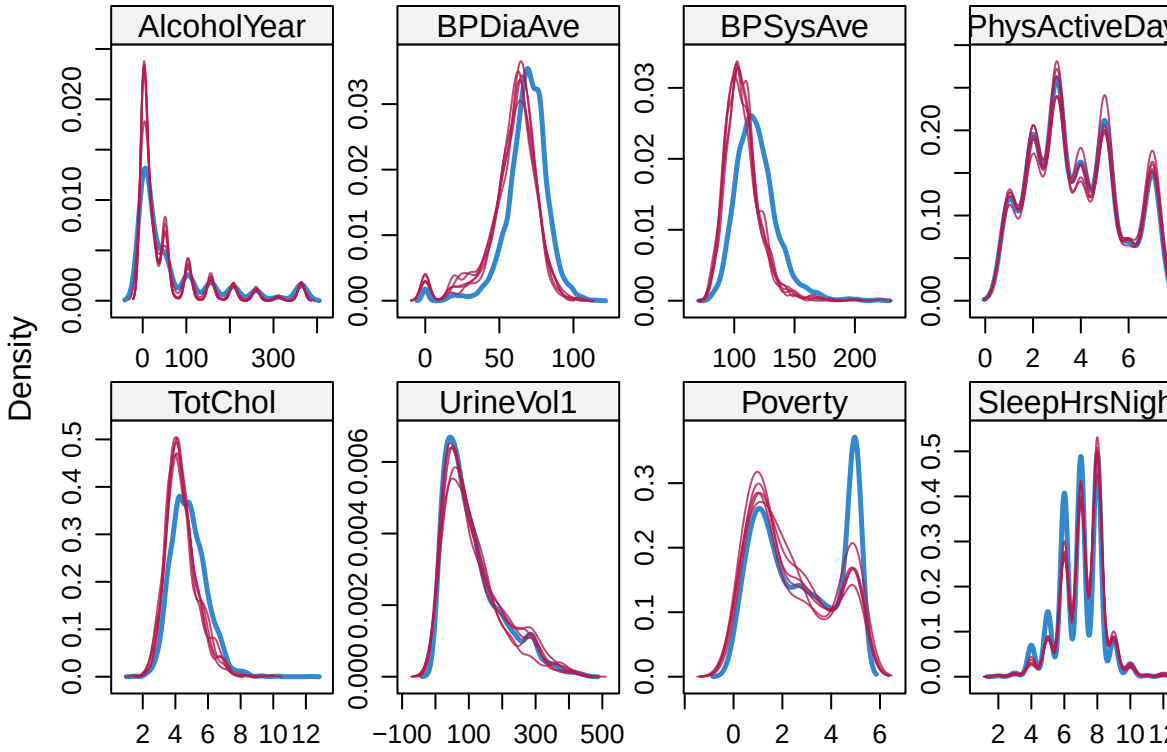


Mental health related variables:

```
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)
```