project_eda

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# lab = read.csv("data/health/labs.csv")
demographics = read.csv("data/health/demographic.csv")
ques = read.csv("data/health/questionnaire.csv")
exam = read.csv("data/health/examination.csv")

# join tables by participant ID
df = inner_join(demographics, ques, by="SEQN")
df = inner_join(df, exam, by="SEQN")
dim(df)
```

```
## [1] 9813 1222
```

```
# subset df_adult to consider only the variables we're interested in
response = "BMXBMI"
predictors = c("RIAGENDR", "RIDAGEYR", "RIDRETH3", "DMDEDUC2", "DMDMARTL", "DMDFMSIZ", "INDFMIN2",
               "ALQ101", "ALQ120Q",
               "CBD070", "CBD120", "CBD130",
               "DBD895", "DBD900", "DBQ197",
               "DPQ020", "DPQ030",
               "PAQ710",
               "SLD010H", "SMQ040")

columns = c(predictors, response)

df = df[names(df) %in% columns]

# rename columns to more intuitive names
df = rename(df, gender = RIAGENDR, age = RIDAGEYR, race = RIDRETH3, edu = DMDEDUC2,
            marriage = DMDMARTL, famsize = DMDFMSIZ, famincome = INDFMIN2,
            alcohol12yr = ALQ101, alcoholfrq = ALQ120Q, grocery = CBD070, eatout = CBD120,
            delivery = CBD130, milk = DBQ197, meals_nothome = DBD895, meals_fastfood = DBD900,
            depressed = DPQ020, sleep_trouble = DPQ030, tv_hrs = PAQ710,
            sleep_hr = SLD010H, smoke = SMQ040, bmi = BMXBMI)

# subset the df to consider only adults aged 18 or above
df_adult = df[df$age > 20,]

df_adult = df_adult[which(df_adult$grocery!=777777 & df_adult$grocery!=999999),]
```

```r
df_adult = df_adult[which(df_adult$eatout!=777777 & df_adult$eatout!=999999),]
df_adult = df_adult[which(df_adult$delivery!=777777 & df_adult$delivery!=999999),]

df_adult = df_adult[which(df_adult$alcoholfrq != 999),]

df_adult = df_adult[which(df_adult$meals_nothome != 5555 & df_adult$meals_nothome != 7777 & df_adult$mea

df_adult = df_adult[which(df_adult$meals_fastfood != 5555 & df_adult$meals_fastfood != 7777 & df_adult$m

df_adult$tv_hrs[which(df_adult$tv_hrs == 0)] = 1
df_adult$tv_hrs[which(df_adult$tv_hrs == 8)] = 0
df_adult = df_adult[which(df_adult$tv_hrs != 77 & df_adult$tv_hrs != 99),]

df_adult = df_adult[which(df_adult$sleep_hr != 99),]

df_adult$smoke[which(is.na(df_adult$smoke))] = "missing"

# drop observations where number of missing value in certain columns is < 100
drop_obs = c("famincome", "grocery", "eatout", "delivery",
             "sleep_hr","depressed", "sleep_trouble", "bmi")

for (feature in drop_obs){
  df_adult = df_adult[!is.na(df_adult[feature]),]
}

categorical_features = c("gender", "race", "edu", "marriage",
                         "famincome", "alcohol12yr", "milk", "depressed",
                         "sleep_trouble", "smoke", "tv_hrs")

numeric_features = c("age", "famsize", "alcoholfrq", "grocery", "eatout",
                     "delivery", "meals_nothome", "meals_fastfood", "sleep_hr")

# convert categorical variables into factors
df_adult[categorical_features] = lapply(df_adult[categorical_features], factor)

apply(df_adult, 2, function(x) sum(is.na(x))) # check how many missing data
```

```
##         gender            age           race            edu       marriage
##              0              0              0              0              0
##        famsize      famincome    alcohol12yr     alcoholfrq        grocery
##              0              0              0              0              0
##         eatout       delivery           milk  meals_nothome meals_fastfood
##              0              0              0              0              0
##      depressed  sleep_trouble         tv_hrs       sleep_hr          smoke
##              0              0              0              0              0
##            bmi
##              0
```

```r
sapply(df_adult, class) # check data classes
```

```
##         gender            age           race            edu       marriage
##       "factor"      "integer"       "factor"       "factor"       "factor"
##        famsize      famincome    alcohol12yr     alcoholfrq        grocery
##      "integer"       "factor"       "factor"      "integer"      "integer"
```

```
##         eatout       delivery           milk  meals_nothome meals_fastfood
##      "integer"      "integer"       "factor"      "integer"      "integer"
##      depressed  sleep_trouble         tv_hrs       sleep_hr          smoke
##       "factor"       "factor"       "factor"      "integer"       "factor"
##            bmi
##      "numeric"
```

Exploratory Data Analysis

Response Variable (BMI)

```
library(ggplot2)
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

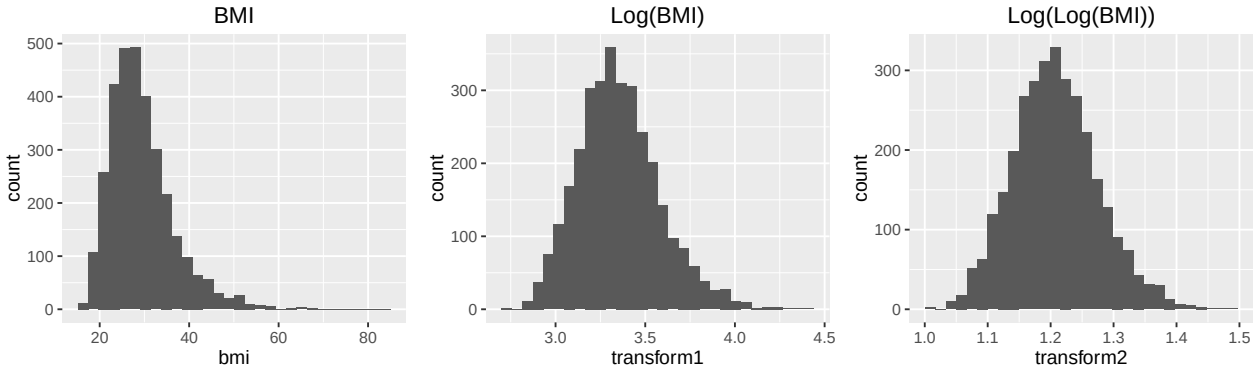
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
transform1 = log(df_adult$bmi)
transform2 = log(log(df_adult$bmi))

# response variable distribution
plot1 = ggplot(df_adult, aes(bmi)) +
  geom_histogram() +
  ggtitle("BMI")
plot2 = ggplot(df_adult, aes(transform1)) +
  geom_histogram() +
  ggtitle("Log(BMI)")
plot3 = ggplot(df_adult, aes(transform2)) +
  geom_histogram() +
  ggtitle("Log(Log(BMI))")
grid.arrange(plot1, plot2, plot3, ncol=3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

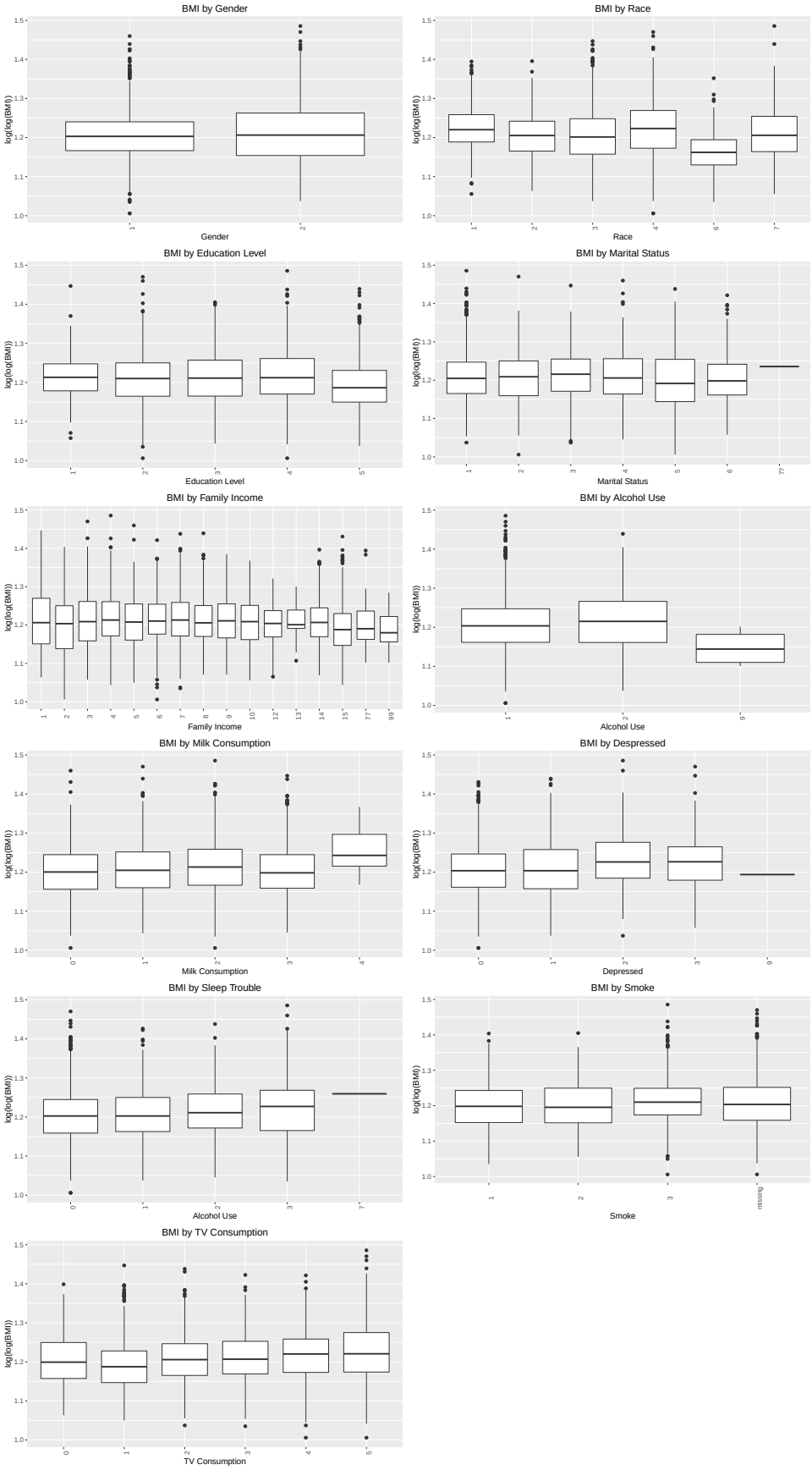


```
df_adult$log2bmi = log(log(df_adult$bmi))
```

Predictor Variables

Categorical Variables

```
plot1 = ggplot(df_adult, aes(x=gender, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Gender", y="log(log(BMI))") +
  ggtitle("BMI by Gender")
plot2 = ggplot(df_adult, aes(x=race, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Race", y="log(log(BMI))") +
  ggtitle("BMI by Race")
plot3 = ggplot(df_adult, aes(x=edu, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Education Level", y="log(log(BMI))") +
  ggtitle("BMI by Education Level")
plot4 = ggplot(df_adult, aes(x=marriage, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Marital Status", y="log(log(BMI))") +
  ggtitle("BMI by Marital Status")
plot5 = ggplot(df_adult, aes(x=famincome, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Family Income", y="log(log(BMI))") +
  ggtitle("BMI by Family Income")
plot6 = ggplot(df_adult, aes(x=alcohol12yr, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Alcohol Use", y="log(log(BMI))") +
  ggtitle("BMI by Alcohol Use")
plot7 = ggplot(df_adult, aes(x=milk, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Milk Consumption", y="log(log(BMI))") +
  ggtitle("BMI by Milk Consumption")
plot8 = ggplot(df_adult, aes(x=depressed, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Depressed", y="log(log(BMI))") +
  ggtitle("BMI by Despressed")
plot9 = ggplot(df_adult, aes(x=sleep_trouble, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Alcohol Use", y="log(log(BMI))") +
  ggtitle("BMI by Sleep Trouble")
plot10 = ggplot(df_adult, aes(x=smoke, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Smoke", y="log(log(BMI))") +
  ggtitle("BMI by Smoke")
plot11 = ggplot(df_adult, aes(x=tv_hrs, y = log2bmi)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="TV Consumption", y="log(log(BMI))") +
  ggtitle("BMI by TV Consumption")
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, plot11, ncol=2)
```

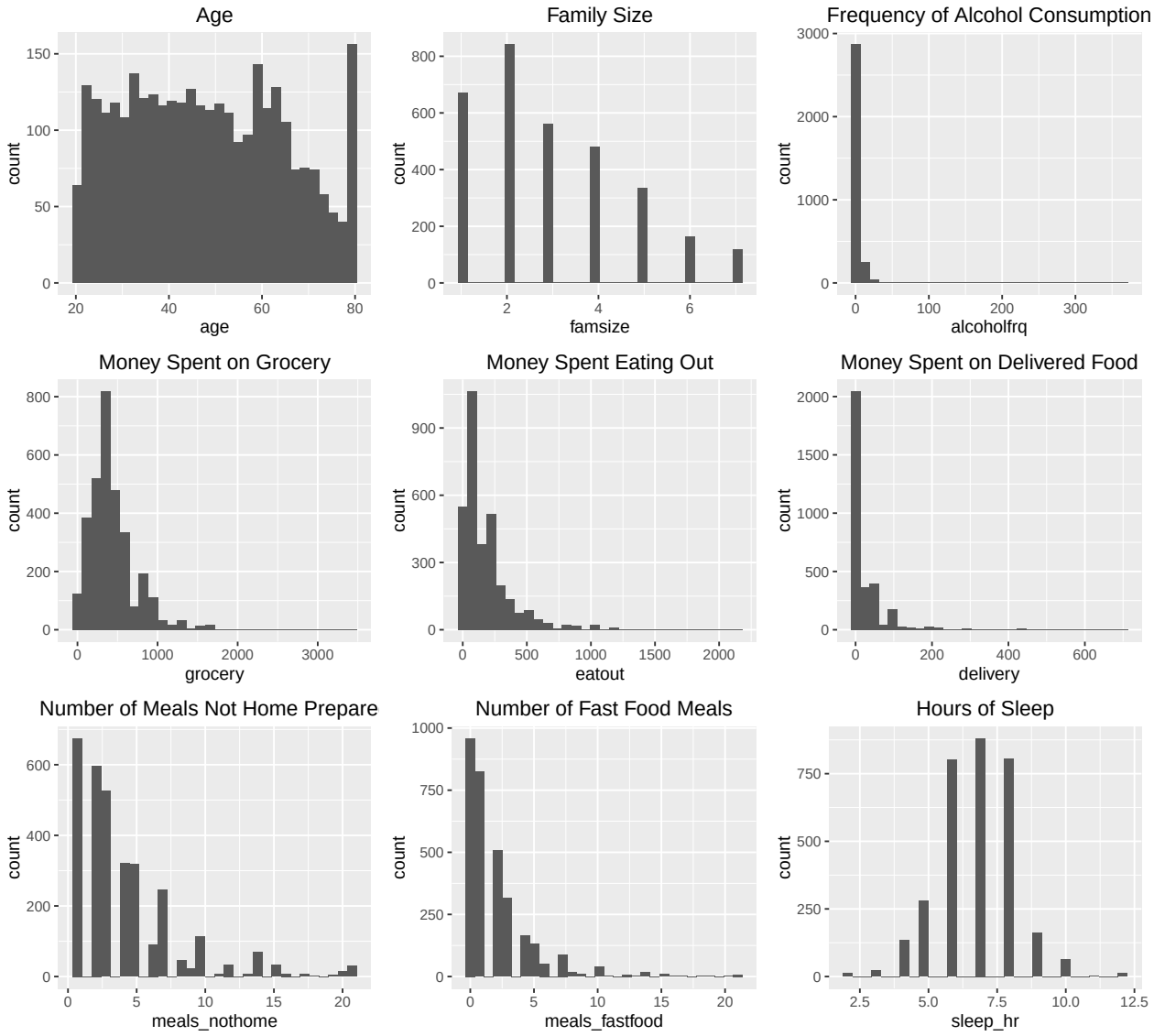

BMI by Gender
log(log(BMI))
Gender
BMI by Race
Race
BMI by Education Level
Education Level
BMI by Marital Status
Marital Status
BMI by Family Income
Family Income
BMI by Alcohol Use
Alcohol Use
BMI by Milk Consumption
Milk Consumption
BMI by Despressed
Depressed
BMI by Sleep Trouble
Alcohol Use
BMI by Smoke
Smoke
missing
BMI by TV Consumption
TV Consumption

Numeric Variables

Distribtuion of numeric variables

```
# numeric predictor variable distributions
plot1 = ggplot(df_adult, aes(age)) + geom_histogram() +
  ggtitle("Age")
plot2 = ggplot(df_adult, aes(famsize)) + geom_histogram() +
  ggtitle("Family Size")
plot3 = ggplot(df_adult, aes(alcoholfrq)) + geom_histogram() +
  ggtitle("Frequency of Alcohol Consumption")
plot4 = ggplot(df_adult, aes(grocery)) + geom_histogram() +
  ggtitle("Money Spent on Grocery")
plot5 = ggplot(df_adult, aes(eatout)) + geom_histogram() +
  ggtitle("Money Spent Eating Out")
plot6 = ggplot(df_adult, aes(delivery)) + geom_histogram() +
  ggtitle("Money Spent on Delivered Food")
plot7 = ggplot(df_adult, aes(meals_nothome)) + geom_histogram() +
  ggtitle("Number of Meals Not Home Prepared")
plot8 = ggplot(df_adult, aes(meals_fastfood)) + geom_histogram() +
  ggtitle("Number of Fast Food Meals")
plot9 = ggplot(df_adult, aes(sleep_hr)) + geom_histogram() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9,ncol=3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

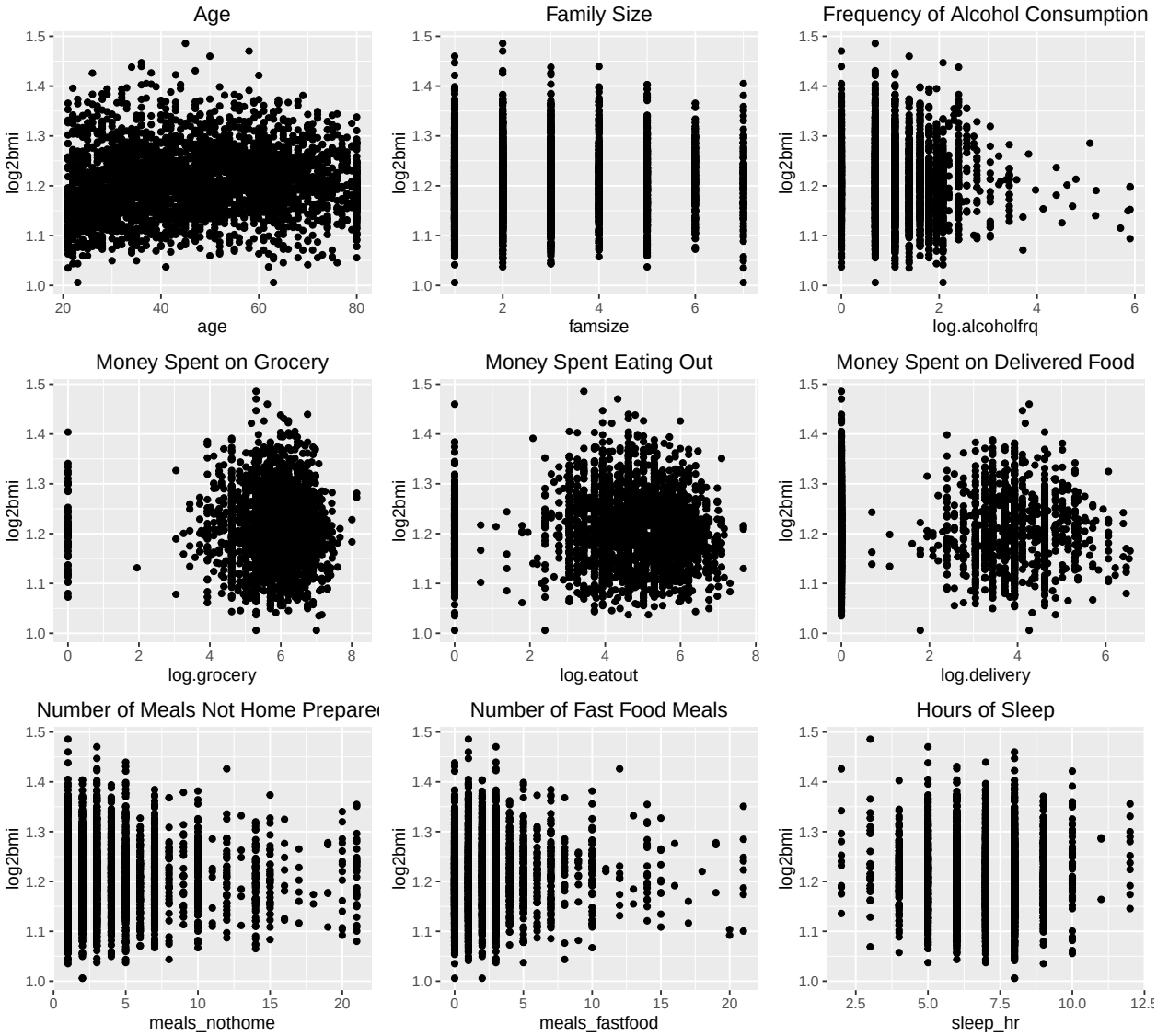


Response vs. numeric distribution

```
df_adult$log.alcoholfrq = log(df_adult$alcoholfrq + 1)
df_adult$log.grocery = log(df_adult$grocery + 1)
df_adult$log.eatout = log(df_adult$eatout + 1)
df_adult$log.delivery = log(df_adult$delivery + 1)

# numeric predictor variable distributions
plot1 = ggplot(df_adult, aes(x=age, y=log2bmi)) + geom_point() +
  ggtitle("Age")
plot2 = ggplot(df_adult, aes(x=famsize, y=log2bmi)) + geom_point() +
  ggtitle("Family Size")
plot3 = ggplot(df_adult, aes(x=log.alcoholfrq, y=log2bmi)) + geom_point() +
  ggtitle("Frequency of Alcohol Consumption")
plot4 = ggplot(df_adult, aes(x=log.grocery, y=log2bmi)) + geom_point() +
  ggtitle("Money Spent on Grocery")
plot5 =ggplot(df_adult, aes(x=log.eatout, y=log2bmi)) + geom_point() +
  ggtitle("Money Spent Eating Out")
plot6 = ggplot(df_adult, aes(x=log.delivery, y=log2bmi)) + geom_point() +
```

```
  ggtitle("Money Spent on Delivered Food")
plot7 =ggplot(df_adult, aes(x=meals_nothome, y=log2bmi)) + geom_point() +
  ggtitle("Number of Meals Not Home Prepared")
plot8 = ggplot(df_adult, aes(x=meals_fastfood, y=log2bmi)) + geom_point() +
  ggtitle("Number of Fast Food Meals")
plot9 = ggplot(df_adult, aes(x=sleep_hr, y=log2bmi)) + geom_point() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9,ncol=3)
```



color by some factor

```
# numeric predictor variable distributions
plot1 = ggplot(df_adult, aes(x=age, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Age")
plot2 = ggplot(df_adult, aes(x=famsize, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Family Size")
plot3 = ggplot(df_adult, aes(x=log.alcoholfrq, y=log2bmi, colour=edu)) + geom_point() +
```

```
  ggtitle("Frequency of Alcohol Consumption")
plot4 = ggplot(df_adult, aes(x=log.grocery, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Money Spent on Grocery")
plot5 =ggplot(df_adult, aes(x=log.eatout, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Money Spent Eating Out")
plot6 = ggplot(df_adult, aes(x=log.delivery, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Money Spent on Delivered Food")
plot7 =ggplot(df_adult, aes(x=meals_nothome, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Number of Meals Not Home Prepared")
plot8 = ggplot(df_adult, aes(x=meals_fastfood, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Number of Fast Food Meals")
plot9 = ggplot(df_adult, aes(x=sleep_hr, y=log2bmi, colour=edu)) + geom_point() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9,ncol=3)
```