Project Title

A STAT 139 Final Project

Yuyue Wang, Xiangru Shu, Chengye Liu, Chia Chi (Michelle) Ho

Due December 13, 2017

Abstract

Introduction

- Obesity is an exerbating problem in the US.
- Explore the association of 21 different factors with bmi, 13 of which are behavior-related factors such as the typical number of hours sleep per night

Methods

- Data description
  - data source is NHANES 2013-2014
  - Variables of interest
  - Only consider adults of age 20 or above
- Data preprocessing & assumptions
  - Merge data by participant sequence number
  - Exclude don't know/refused/missing values — discuss implications in limitations
- Perform EDA
- Fit regression models
- Check assumptions

Results

Exploratory Data Analysis

Limitations

Conclusions

Appendix

Appendix I: Data preprocessing

```
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
```

```
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 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", "PAQ665",
               "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,activity=PAQ665)

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

# subset the data to exclude refused/don't know/missing data
# demographics variables
df_adult = df_adult[which(df_adult$edu!=7 & df_adult$edu!=9),]
```

```r
df_adult = df_adult[which(df_adult$activity!=7 & df_adult$activity!=9),]
df_adult = df_adult[which(df_adult$marriage!=77 & df_adult$marriage!=99),]
df_adult = df_adult[which(df_adult$famincome!=77 & df_adult$famincome!=99),]

# alcohol use variables
df_adult = df_adult[which(df_adult$alcohol12yr!=7 & df_adult$alcohol12yr!=9),]
df_adult = df_adult[which(df_adult$alcoholfrq!=777 & df_adult$alcoholfrq!=999),]

# consumer behavior variables
df_adult = df_adult[which(df_adult$grocery!=777777 & df_adult$grocery!=999999),]
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),]

# diet behavior variables
df_adult = df_adult[which(df_adult$meals_nothome != 5555 & df_adult$meals_nothome != 7777 & df_adul
df_adult = df_adult[which(df_adult$meals_fastfood != 5555 & df_adult$meals_fastfood != 7777 & df_ad

# physical activity variables
#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),]

# mental health variables
df_adult = df_adult[which(df_adult$depressed!=7 & df_adult$depressed!=9),]
df_adult = df_adult[which(df_adult$sleep_trouble!=7 & df_adult$sleep_trouble!=9),]

# sleeping behavior variables
df_adult = df_adult[which(df_adult$sleep_hr != 99),]

# smoking behavior variables
df_adult$smoke[which(is.na(df_adult$smoke))] = "missing"
df_adult$smoke[which(is.na(df_adult$activity))] = "missing"

# after dropping observations missing bmi, there were only 3 missing height
# so we drop these observations too
drop_obs = c("bmi")

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


# feature engineering
df_adult$marriage[which(df_adult$marriage==1)]="married/living_with_partner"
df_adult$marriage[which(df_adult$marriage==6)]="married/living_with_partner"
df_adult$marriage[which(df_adult$marriage==3)]="divorced/separated"
df_adult$marriage[which(df_adult$marriage==4)]="divorced/separated"
```

```r
df_adult$marriage[which(df_adult$marriage==2)]="widowed"
df_adult$marriage[which(df_adult$marriage==5)]="never_married"


df_adult$tv_hrs[which(df_adult$tv_hrs == 0)]= "1hr_or_less"
df_adult$tv_hrs[which(df_adult$tv_hrs==1)] ="1hr_or_less"
df_adult$tv_hrs[which(df_adult$tv_hrs == 8)] = "0hr"
df_adult$tv_hrs[which(df_adult$tv_hrs == 2)]  = "2-4hrs"
df_adult$tv_hrs[which(df_adult$tv_hrs == 3)]= "2-4hrs"
df_adult$tv_hrs[which(df_adult$tv_hrs == 4)]= "2-4hrs"
df_adult$tv_hrs[which(df_adult$tv_hrs == 5)] = "5hrs_or_more"

new_bmi = df_adult$bmi
df_adult["bmi_class"] = new_bmi
df_adult$bmi_class[which(df_adult$bmi_class >= 40)] = "class 3 obesity"
df_adult$bmi_class[which(df_adult$bmi_class >= 30 & df_adult$bmi_class < 40)] = "obese"
df_adult$bmi_class[which(df_adult$bmi_class >= 25 & df_adult$bmi_class < 30)] = "overweight"
df_adult$bmi_class[which(df_adult$bmi_class >= 18.5 & df_adult$bmi_class < 25)] = "healthy"
df_adult$bmi_class[which(df_adult$bmi_class < 18.5)] = "underweight"

# save variable names into lists of categorical or numeric faetures
categorical_features = c("gender", "race", "edu", "marriage",
                         "famincome", "alcohol12yr", "milk", "depressed",
                         "sleep_trouble", "smoke", "tv_hrs","activity")

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

#drop famincome 12,13
df_adult = df_adult[!(df_adult$famincome==12 | df_adult$famincome==13),]

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

levels(df_adult$activity) = c(1, 2)
levels(df_adult$activity) = c("Yes", "No")

#edu
levels(df_adult$edu) = c(1, 1, 2, 3, 3)
levels(df_adult$edu) = c("High School Below", "High School", "High School Above")
#famincome
levels(df_adult$famincome) = c(rep(1, 5), rep(2, 3), rep(3, 2), 4, 5 )
levels(df_adult$famincome) = c("below 25k", "25k~55k", "55k~75k", "75k~100k", "above 100k")


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       activity         tv_hrs       sleep_hr
##              0              0              0              0              0
##         smoke            bmi      bmi_class
##              0              0              0
```

```
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       activity         tv_hrs       sleep_hr
##       "factor"       "factor"       "factor"       "factor"      "integer"
##          smoke            bmi      bmi_class
##       "factor"      "numeric"    "character"
```

Appendix II: 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
```

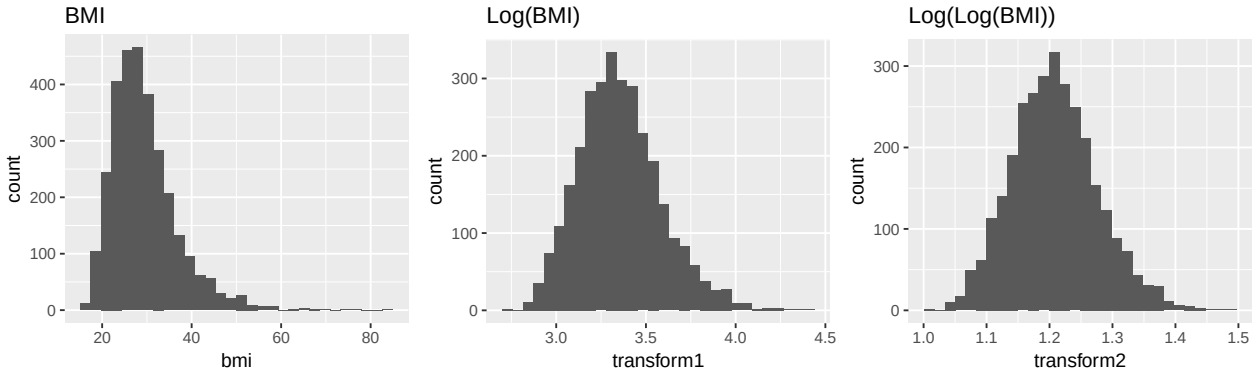
```
transform1 = log(df_adult$bmi)
transform2 = log(log(df_adult$bmi))

# 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`.
```

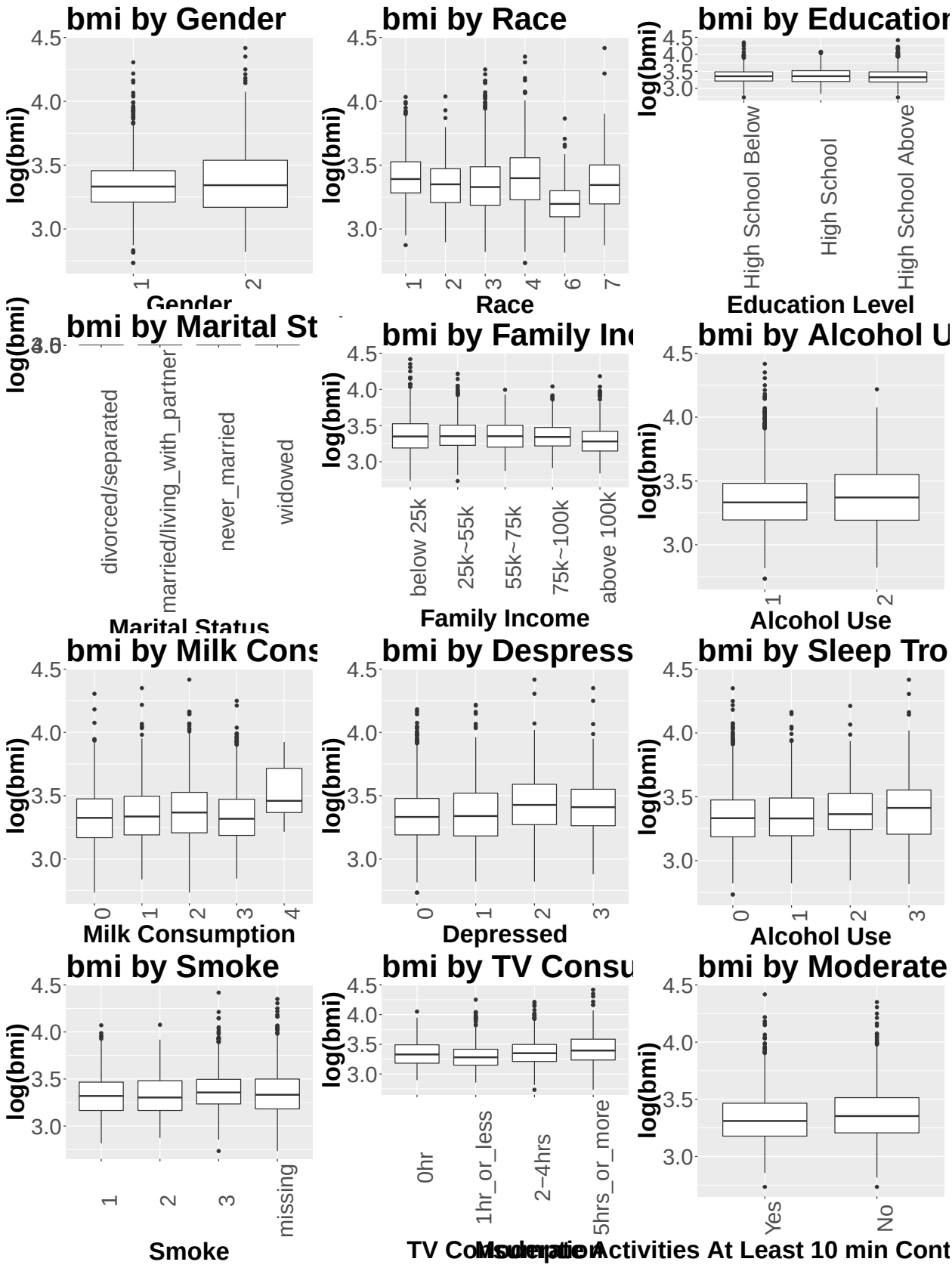


Predictor Variables

Categorical Variables

```
plot1 = ggplot(df_adult, aes(x=gender, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Gender", y="log(bmi)") +
  ggtitle("bmi by Gender") + theme(axis.text=element_text(size=25),axis.title=element_text(size=28,
plot2 = ggplot(df_adult, aes(x=race, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Race", y="log(bmi)") +
  ggtitle("bmi by Race") + theme(axis.text=element_text(size=25),axis.title=element_text(size=28,fa
plot3 = ggplot(df_adult, aes(x=edu, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Education Level", y="log(bmi)") +
  ggtitle("bmi by Education Level") + theme(axis.text=element_text(size=25),axis.title=element_text
plot4 = ggplot(df_adult, aes(x=marriage, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Marital Status", y="log(bmi)") +
  ggtitle("bmi by Marital Status") + theme(axis.text=element_text(size=25),axis.title=element_text(
plot5 = ggplot(df_adult, aes(x=famincome, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Family Income", y="log(bmi)") +
  ggtitle("bmi by Family Income") + theme(axis.text=element_text(size=25),axis.title=element_text(s
plot6 = ggplot(df_adult, aes(x=alcohol12yr, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
```

```
  labs(x="Alcohol Use", y="log(bmi)") +
  ggtitle("bmi by Alcohol Use") + theme(axis.text=element_text(size=25),axis.title=element_text(siz
plot7 = ggplot(df_adult, aes(x=milk, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Milk Consumption", y="log(bmi)") +
  ggtitle("bmi by Milk Consumption") + theme(axis.text=element_text(size=25),axis.title=element_tex
plot8 = ggplot(df_adult, aes(x=depressed, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Depressed", y="log(bmi)") +
  ggtitle("bmi by Despressed") + theme(axis.text=element_text(size=25),axis.title=element_text(size
plot9 = ggplot(df_adult, aes(x=sleep_trouble, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Alcohol Use", y="log(bmi)") +
  ggtitle("bmi by Sleep Trouble") + theme(axis.text=element_text(size=25),axis.title=element_text(s
plot10 = ggplot(df_adult, aes(x=smoke, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Smoke", y="log(bmi)") +
  ggtitle("bmi by Smoke") + theme(axis.text=element_text(size=25),axis.title=element_text(size=28,f
plot11 = ggplot(df_adult, aes(x=tv_hrs, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="TV Consumption", y="log(bmi)") +
  ggtitle("bmi by TV Consumption") + theme(axis.text=element_text(size=25),axis.title=element_text(
plot12 = ggplot(df_adult, aes(x=activity, y = log(bmi))) + geom_boxplot() +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x="Moderate Activities At Least 10 min Continuously per Week", y="log(bmi)") +
  ggtitle("bmi by Moderate Recreational Activities") + theme(axis.text=element_text(size=25),axis.t
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, plot11, plot12,
```

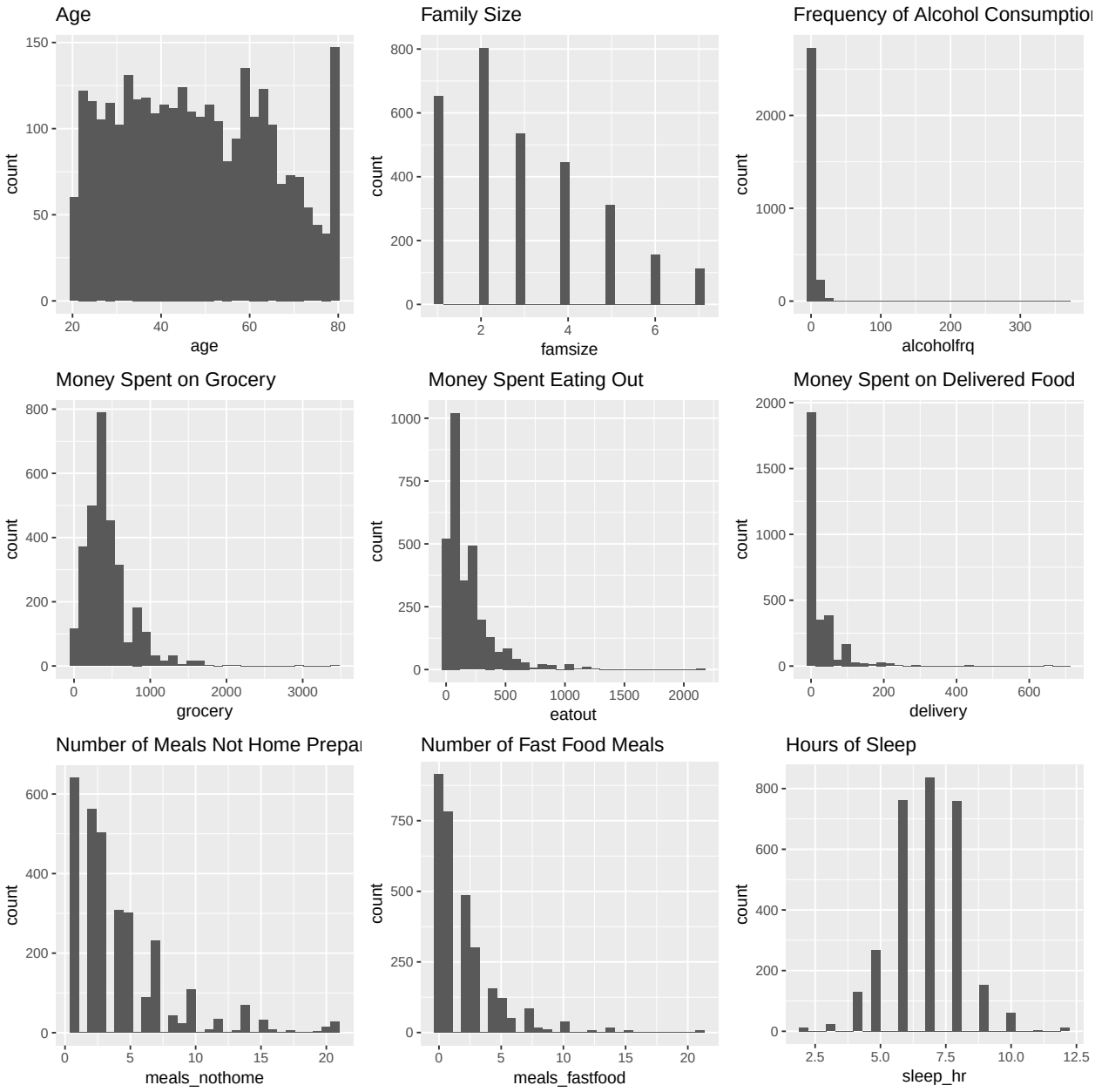

bmi by Gender
log(bmi)
4.5
4.0
3.5
3.0
1
2
Gender
bmi by Race
log(bmi)
4.5
4.0
3.5
3.0
1
2
3
4
6
7
Race
bmi by Educatior
log(bmi)
4.5
4.0
3.5
3.0
High School Below
High School
High School Above
Education Level
bmi by Marital St
log(bmi)
divorced/separated
married/living_with_partner
never_married
widowed
Marital Status
bmi by Family In
log(bmi)
4.5
4.0
3.5
3.0
below 25k
25k~55k
55k~75k
75k~100k
above 100k
Family Income
bmi by Alcohol U
log(bmi)
4.5
4.0
3.5
3.0
1
2
Alcohol Use
bmi by Milk Cons
log(bmi)
4.5
4.0
3.5
3.0
0
1
2
3
4
Milk Consumption
bmi by Despress
log(bmi)
4.5
4.0
3.5
3.0
0
1
2
3
Depressed
bmi by Sleep Tro
log(bmi)
4.5
4.0
3.5
3.0
0
1
2
3
Alcohol Use
bmi by Smoke
log(bmi)
4.5
4.0
3.5
3.0
1
2
3
missing
Smoke
bmi by TV Consu
log(bmi)
4.5
4.0
3.5
3.0
0hr
1hr_or_less
2~4hrs
5hrs_or_more
TV Co
bmi by Moderate
log(bmi)
4.5
4.0
3.5
3.0
Yes
No
Activities At Least 10 min Cont

Numeric Variables

Distribtuion of numeric variables

```
# numeric predictor variable distributions
plot1 = ggplot(df_adult, aes(age)) + geom_histogram() +
  ggtitle("Age")
plot3 = ggplot(df_adult, aes(famsize)) + geom_histogram() +
  ggtitle("Family Size")
plot4 = ggplot(df_adult, aes(alcoholfrq)) + geom_histogram() +
  ggtitle("Frequency of Alcohol Consumption")
plot5 = ggplot(df_adult, aes(grocery)) + geom_histogram() +
  ggtitle("Money Spent on Grocery")
plot6 = ggplot(df_adult, aes(eatout)) + geom_histogram() +
  ggtitle("Money Spent Eating Out")
plot7 = ggplot(df_adult, aes(delivery)) + geom_histogram() +
  ggtitle("Money Spent on Delivered Food")
plot8 = ggplot(df_adult, aes(meals_nothome)) + geom_histogram() +
  ggtitle("Number of Meals Not Home Prepared")
plot9 = ggplot(df_adult, aes(meals_fastfood)) + geom_histogram() +
  ggtitle("Number of Fast Food Meals")
plot10 = ggplot(df_adult, aes(sleep_hr)) + geom_histogram() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, 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`.
```




```
# transformations on numeric predictor variables
plot1 = ggplot(df_adult, aes(alcoholfrq)) + geom_histogram() +
  ggtitle("Frequency of Alcohol Consumption")
plot2 = ggplot(df_adult, aes(log(alcoholfrq+1))) + geom_histogram() +
  ggtitle("Log Frequency of Alcohol Consumption")

plot3 = ggplot(df_adult, aes(grocery)) + geom_histogram() +
  ggtitle("Money Spent on Grocery")
plot4 = ggplot(df_adult, aes(log(grocery+1))) + geom_histogram() +
  ggtitle("Log Money Spent on Grocery")

plot5 = ggplot(df_adult, aes(eatout)) + geom_histogram() +
```

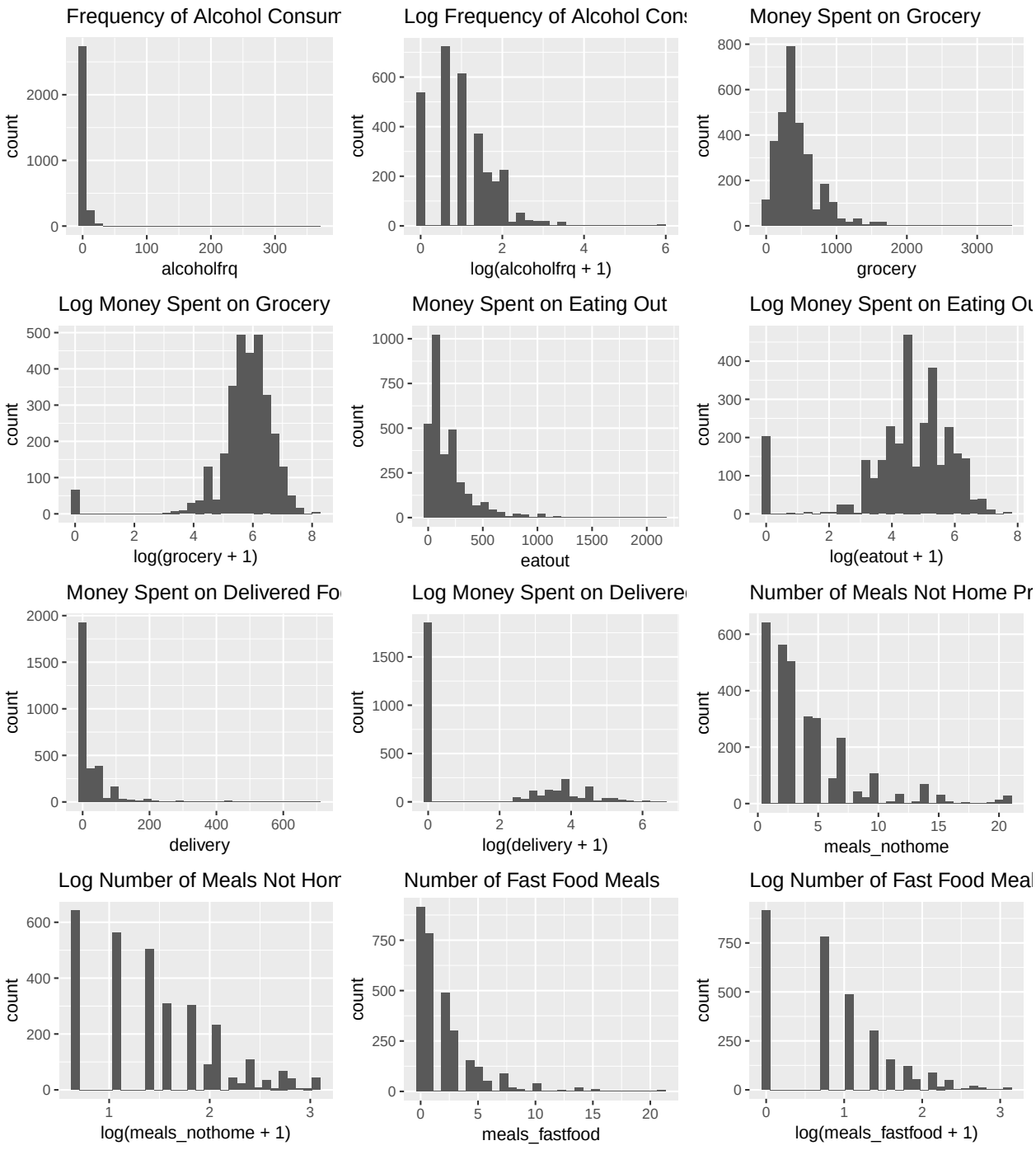
```r
  ggtitle("Money Spent on Eating Out")
plot6 = ggplot(df_adult, aes(log(eatout+1))) + geom_histogram() +
  ggtitle("Log Money Spent on Eating Out")

plot7 = ggplot(df_adult, aes(delivery)) + geom_histogram() +
  ggtitle("Money Spent on Delivered Food")
plot8 = ggplot(df_adult, aes(log(delivery+1))) + geom_histogram() +
  ggtitle("Log Money Spent on Delivered Food")

plot9 = ggplot(df_adult, aes(meals_nothome)) + geom_histogram() +
  ggtitle("Number of Meals Not Home Prepared")
plot10 = ggplot(df_adult, aes(log(meals_nothome+1))) + geom_histogram() +
  ggtitle("Log Number of Meals Not Home Prepared")

plot11 = ggplot(df_adult, aes(meals_fastfood)) + geom_histogram() +
  ggtitle("Number of Fast Food Meals")
plot12 = ggplot(df_adult, aes(log(meals_fastfood+1))) + geom_histogram() +
  ggtitle("Log Number of Fast Food Meals")
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, plot11, plot12,
```

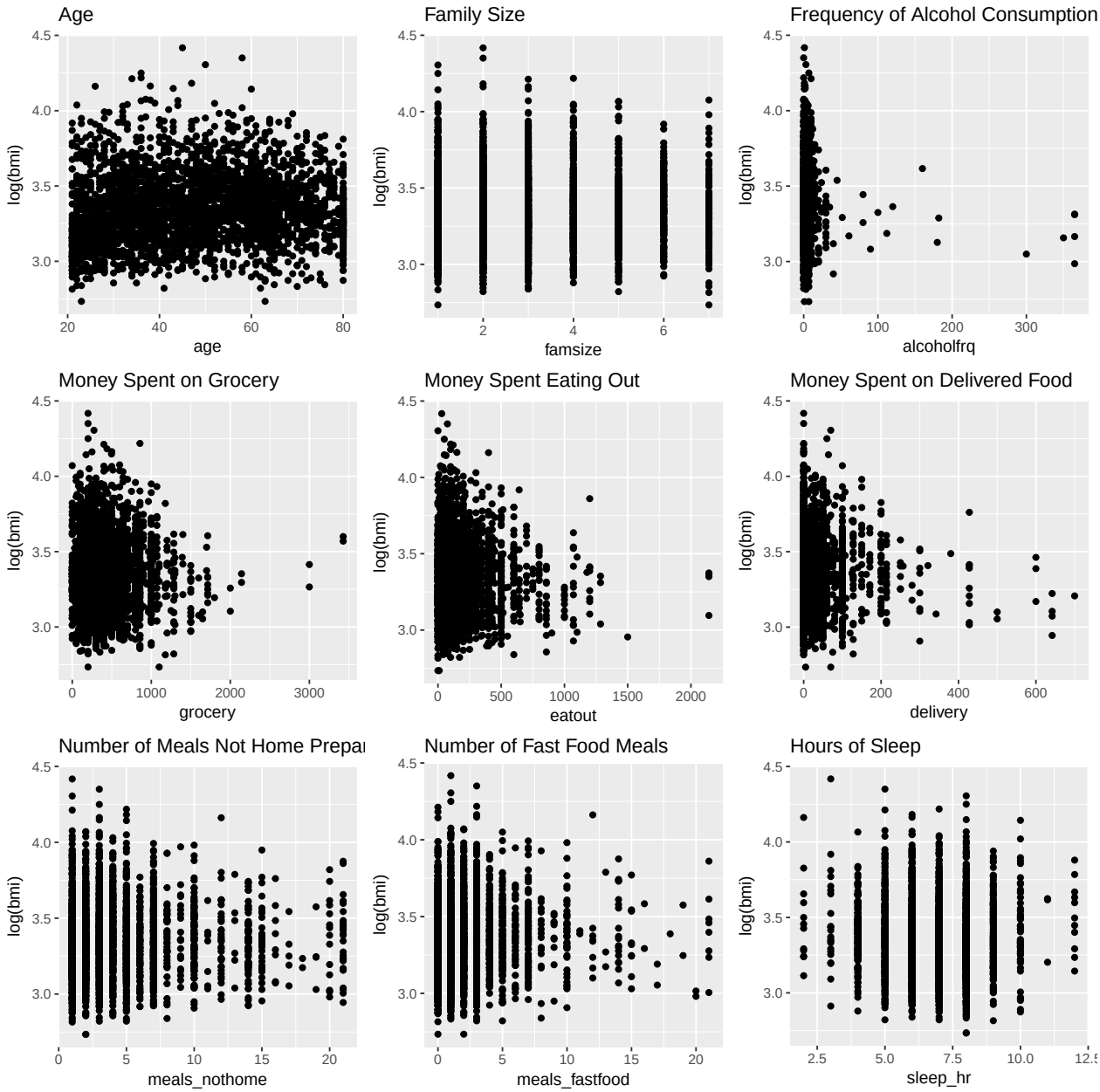
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using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

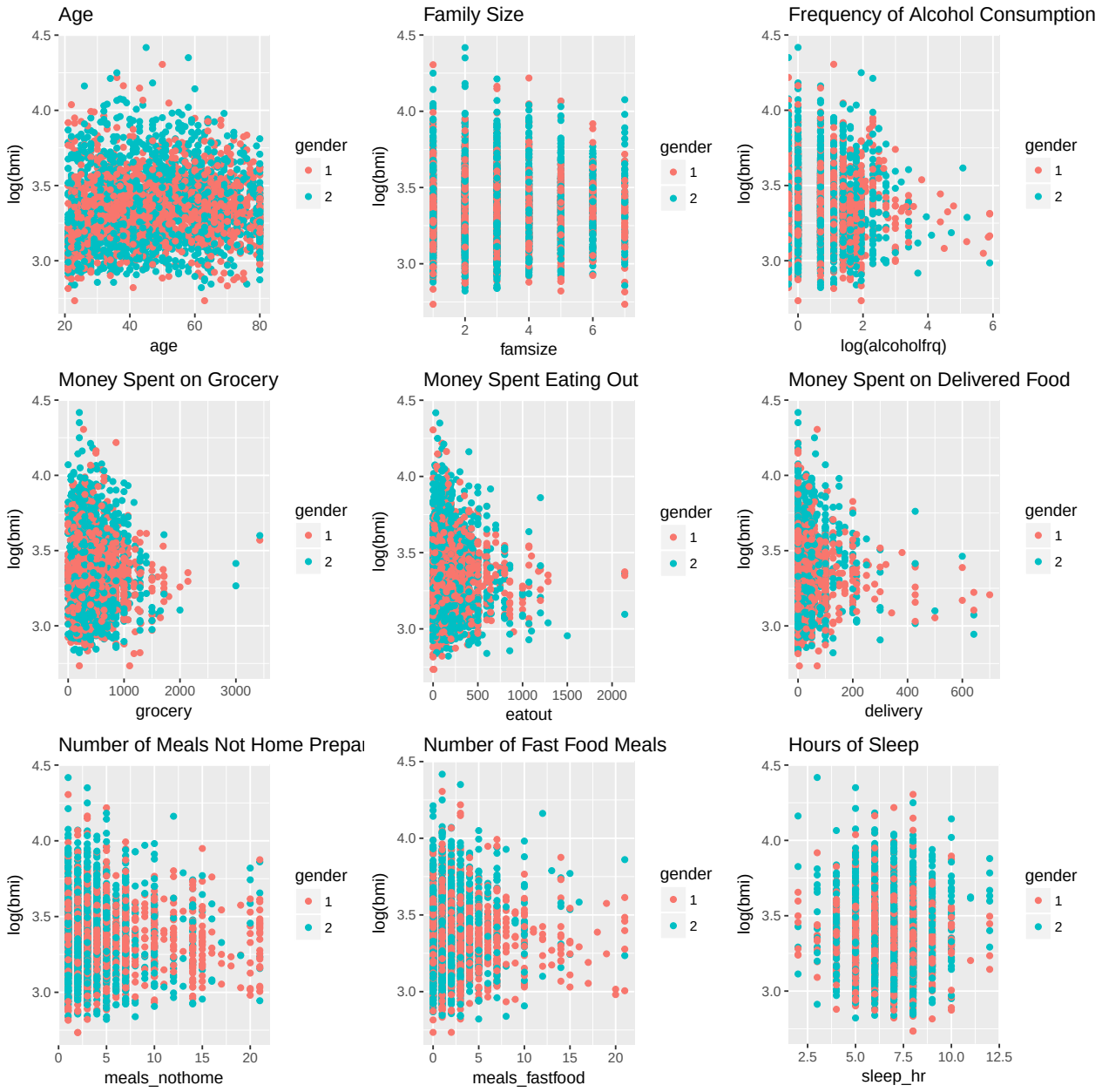
```
plot1 = ggplot(df_adult, aes(x=age, y=log(bmi))) + geom_point() +
  ggtitle("Age")
plot3 = ggplot(df_adult, aes(x=famsize, y=log(bmi))) + geom_point() +
  ggtitle("Family Size")
plot4 = ggplot(df_adult, aes(x=alcoholfrq, y=log(bmi))) + geom_point() +
  ggtitle("Frequency of Alcohol Consumption")
plot5 = ggplot(df_adult, aes(x=grocery, y=log(bmi))) + geom_point() +
```

```
  ggtitle("Money Spent on Grocery")
plot6 =ggplot(df_adult, aes(x=eatout, y=log(bmi))) + geom_point() +
  ggtitle("Money Spent Eating Out")
plot7 = ggplot(df_adult, aes(x=delivery, y=log(bmi))) + geom_point() +
  ggtitle("Money Spent on Delivered Food")
plot8 =ggplot(df_adult, aes(x=meals_nothome, y=log(bmi))) + geom_point() +
  ggtitle("Number of Meals Not Home Prepared")
plot9 = ggplot(df_adult, aes(x=meals_fastfood, y=log(bmi))) + geom_point() +
  ggtitle("Number of Fast Food Meals")
plot10 = ggplot(df_adult, aes(x=sleep_hr, y=log(bmi))) + geom_point() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, ncol=3)
```




```
plot1 = ggplot(df_adult, aes(x=age, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Age")
plot3 = ggplot(df_adult, aes(x=famsize, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Family Size")
plot4 = ggplot(df_adult, aes(x=log(alcoholfrq), y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Frequency of Alcohol Consumption")
plot5 = ggplot(df_adult, aes(x=grocery, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Money Spent on Grocery")
plot6 =ggplot(df_adult, aes(x=eatout, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Money Spent Eating Out")
plot7 = ggplot(df_adult, aes(x=delivery, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Money Spent on Delivered Food")
```

```
plot8 =ggplot(df_adult, aes(x=meals_nothome, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Number of Meals Not Home Prepared")
plot9 = ggplot(df_adult, aes(x=meals_fastfood, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Number of Fast Food Meals")
plot10 = ggplot(df_adult, aes(x=sleep_hr, y=log(bmi), colour=gender)) + geom_point() +
  ggtitle("Hours of Sleep")
grid.arrange(plot1, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, ncol=3)
```




```
# response
df_adult$log.bmi = log(df_adult$bmi)
```

```r
# numeric predictors
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)
df_adult$log.nothome = log(df_adult$meals_nothome + 1)
df_adult$log.fastfood = log(df_adult$meals_fastfood + 1)
```

parallel corrd plot to figure out if interaction may be helpful

```r
# library(MASS)
# library(RColorBrewer)
# k <- adjustcolor(brewer.pal(5, "Set2")[df_adult$sleep_trouble], alpha=.6)
# predictor = c("sleep_hr", "age", "log.delivery", "meals_nothome", "meals_fastfood")
# parcoord(df_adult[,predictor], col=k)
```