The question we've been asked 1,000 times: Do people still watch TV? Why?

Kelly Dixon, Jessi Gronsbell, Kathleen Keshishian, Kate Williams

2020-01-10

Contents

1 Abstract

DONT IGNORE ME

2 Background

When we tell people we work at Nielsen, we inevitably get questions. If we're speaking to someone under 40, the first question is, "What is Nielsen?" The second question is "Does anyone still watch TV?" Yes, in fact they do! This is supported both by Nielsen's data and a 2018 article in *The Atlantic* marveled at this fact (Madrigal 2018).

> "Over the last 8 years, all the new, non-TV things—Facebook, phones, YouTube, Netflix—have only cut about an hour per day from the dizzying amount of TV that the average household watches. Americans are still watching more than 7 hours and 50 minutes per household per day."

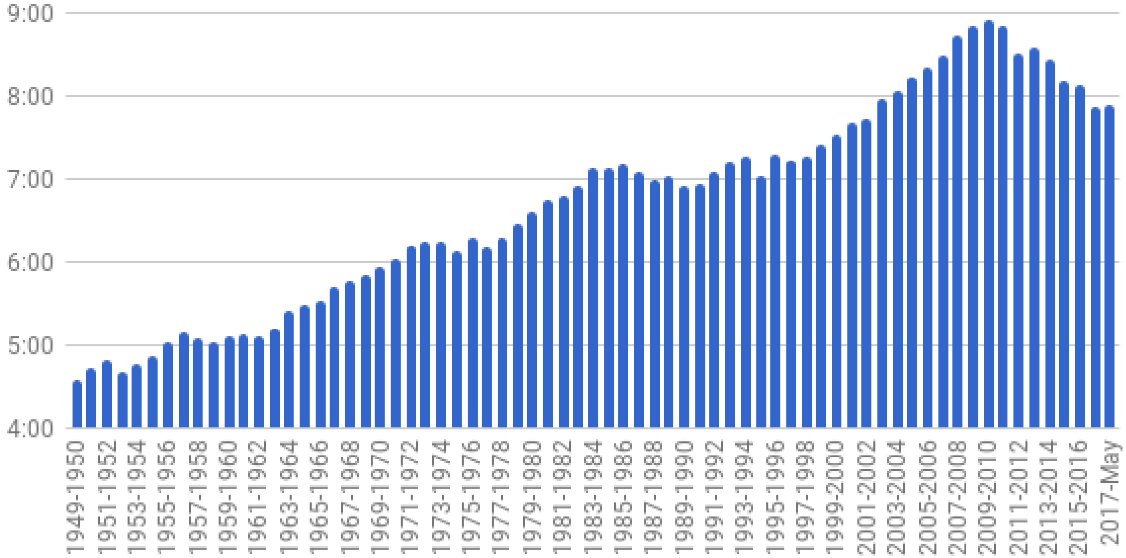


Figure 1: Hours of TV American Households Watch Per Day

Nielsen uses principles of survey methodology to build best-in-class representative panels to measure television watching - the "what" (Nielsen 2020). But how do we analyze the "why"? Nielsen measures many attributes about the household and captures minute-level TV viewing via meters connected into television in the home, but we have limited visibility into the "why".

Nielsen has made this decision because we do not want to be overly-intrusive into our panelists' lives. Nielsen meters are a predominately passive measurement collection tool after the initial installation into the home, and we ask our panelists minimal questions and none on attitudes or behavior. For example, as of late, citizenship status is considered a highly sensitive question; therefore, Nielsen does not ask it so that we can increase participation of Hispanic households. By limiting the data we collect, Nielsen reduces non-response bias which improves data quality.

2.1 Social Science Context

The decsion to limit data collection on feelings and attitudes of Nielsen's panelists allows us to best measure media behavior without introducing bias is in the media industry and Nielsen's. After being introduced to the GSS survey data and understanding the social science studies conducted from this dataset, we were excited about the opportunity to examine this dataset. It could provide us with additional insight about the drivers of television viewing behavior beyond the typical demographic factors and prior media behavior that we typically study.

2.2 Key Research Question

This leads us to the question, *what feelings and attitudes are associated with higher self-reports of average television viewing in a week?* As we explore the GSS data it is important to note that self-reported television viewing is subject to more response bias than the passive measurement via the Nielsen television meter.

2.3 General Social Survey Data

The GSS survey measures a nationally representative panel's feelings and attitudes about a variety of topics. This survey also asks about television watching in a typical week. We wanted to explore other factors besides the demographic factors typically associated with teleivision watching. We use exploratory data analysis, generalized linear modeling, logistic regression with Bayesian techniques and random forest models to explore this question. The results of the analysis are mixed leaving further research questions to explore.

The General Social Survey (GSS) is a project of the independent research organization NORC at the University of Chicago, with principal funding from the National Science Foundation.

From the GSS website ("About the GSS NORC" 2016):
>For more than four decades, the General Social Survey (GSS) has studied the growing complexity of American society. It is the only full-probability, personal-interview survey designed to monitor changes in both social characteristics and attitudes currently being conducted in the United States

Data for this project were downloaded using NORC's GSS data explorer ("GSS Data Explorer NORC at the University of Chicago" 2020). We downloaded 96 variables representing television viewing, demographics, life satisfaction, family life, politics, and religion from surveys conducted during the period from 2008 to 2018.

GSS questions vary somewhat from year to year. An example question set from 2014 is available on their website ("GSS Data Explorer NORC at the University of Chicago Questionnaire 2014 Gss V1 English" 2020).

3 Data Exploration

Self reported daily hours of televison watched, the dependent variable, was based on GSS' question regarding the number of hours per day the respondent spends watching television, TV Hours. Unfortunately, this dependent variable had 4,639 pieces of missing data in 13,794 cases. In the interst of time, these were omitted, leaving 9,155 cases. Time allowing, sophisticated methods to address missing data, such as multiple imputations, should be used for a more thorough investigation.

In order to better understand the GSS dataset, we explored more than 25 variables of interest from question categories including life satisfaction, family life, politics, religion, and demographics.

3.1 In Horizon Variables

Over 70 years of media research behavior has repeatedly shown that drivers of media consumption are geography and key demographic factors including age, race, gender, language spoken in the home, education, and household income (Nielsen 2020). Additionally, the media business will analyze the data according to these demographic breaks, so it is essential that models contain these variables. We will consider the following as "in horizon" explanatiory variables in initial model trials.

- Gender
- Race
- Age
- Household income or education
- Geograhic region

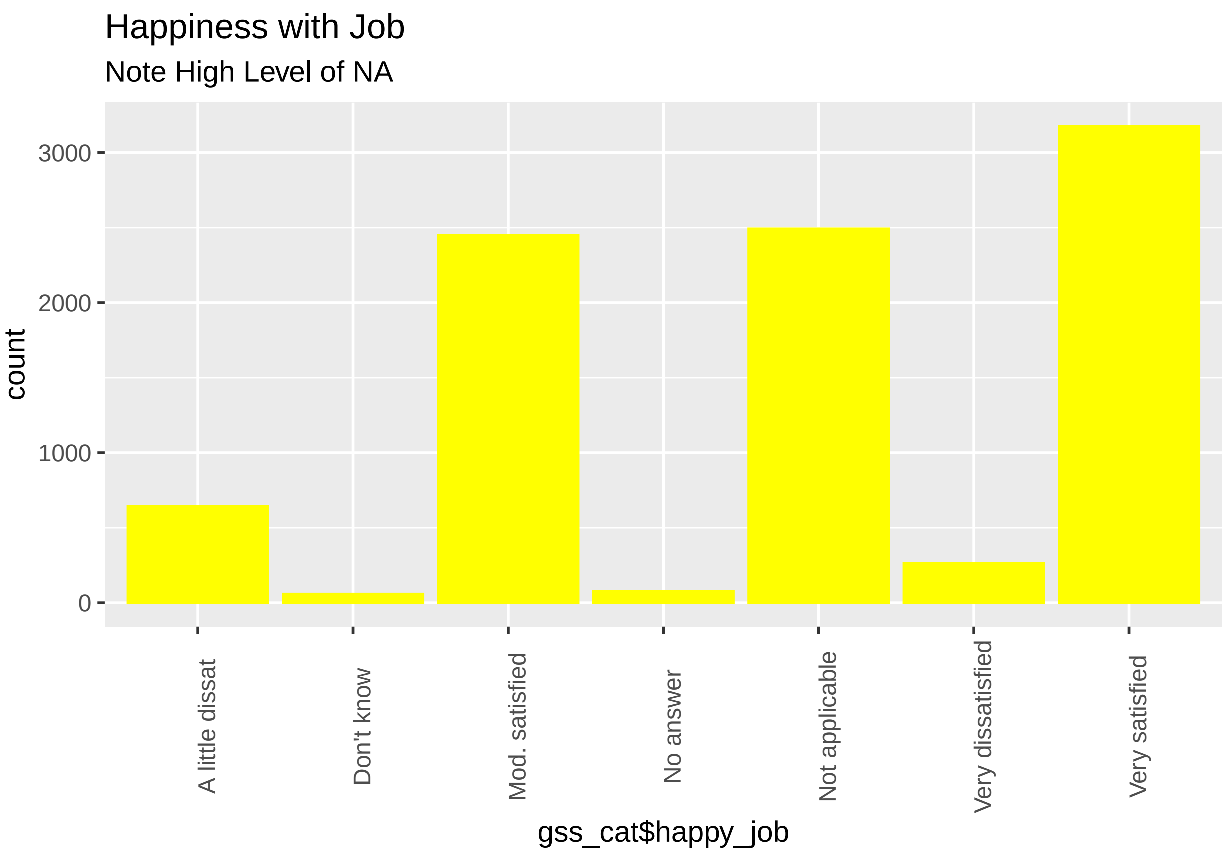
3.2 Beyond the Horizon Variables

When considering the GSS data, we are looking for questions that are capturing more about the respondent's attitude, behavior, feelings or opinions that could infulence television watching. After an initial look at the data and consideration of human behavior we chose to focus on a subset of measures that theoretically offered more value.

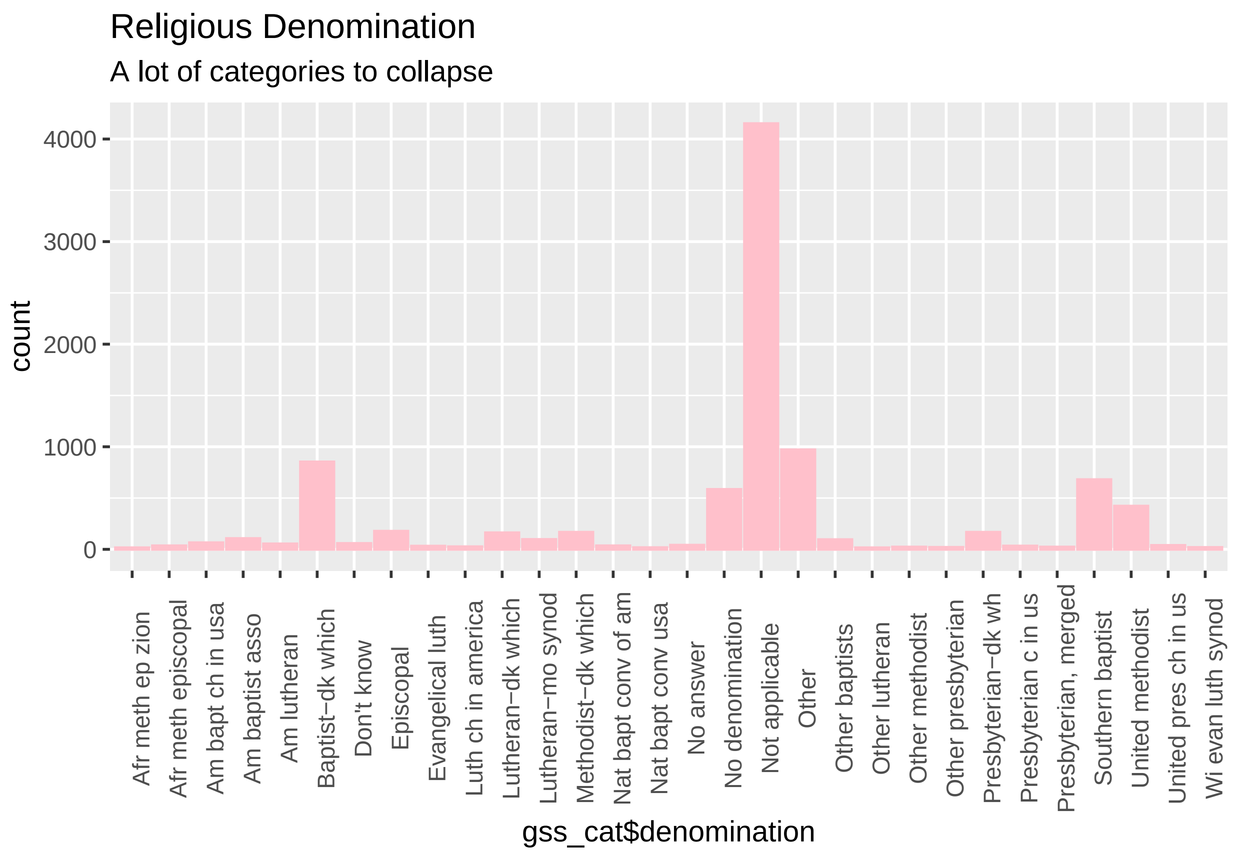
For our visualization we wrote R code to create bar graphs using ggplots for all factor variables, example shown below.

After this exploratory analysis, we chose to focus on 2 of these feeling/attitude variables for the remainder of the project denoted as happy and pray.
* Happy: "Taken all together, how would you say things are these days–would you say that you are very happy, pretty happy, or not too happy? * Pray:"About how often do you pray?"

Our decision to reduce to to 2 was based on the following reasoning. 1. The data was either missing NA or Don't Know for > 3,000 respondents. An example of this can be seen with the Happiness of Job variable. While this variable is interesting, it would take our sample size down to approximately 6,000 respondents.



2. Collapsing the categories would require knowledge/information that we do not have time to research given the timelines of the project



3. Most of the information in the data was captured by another variable.

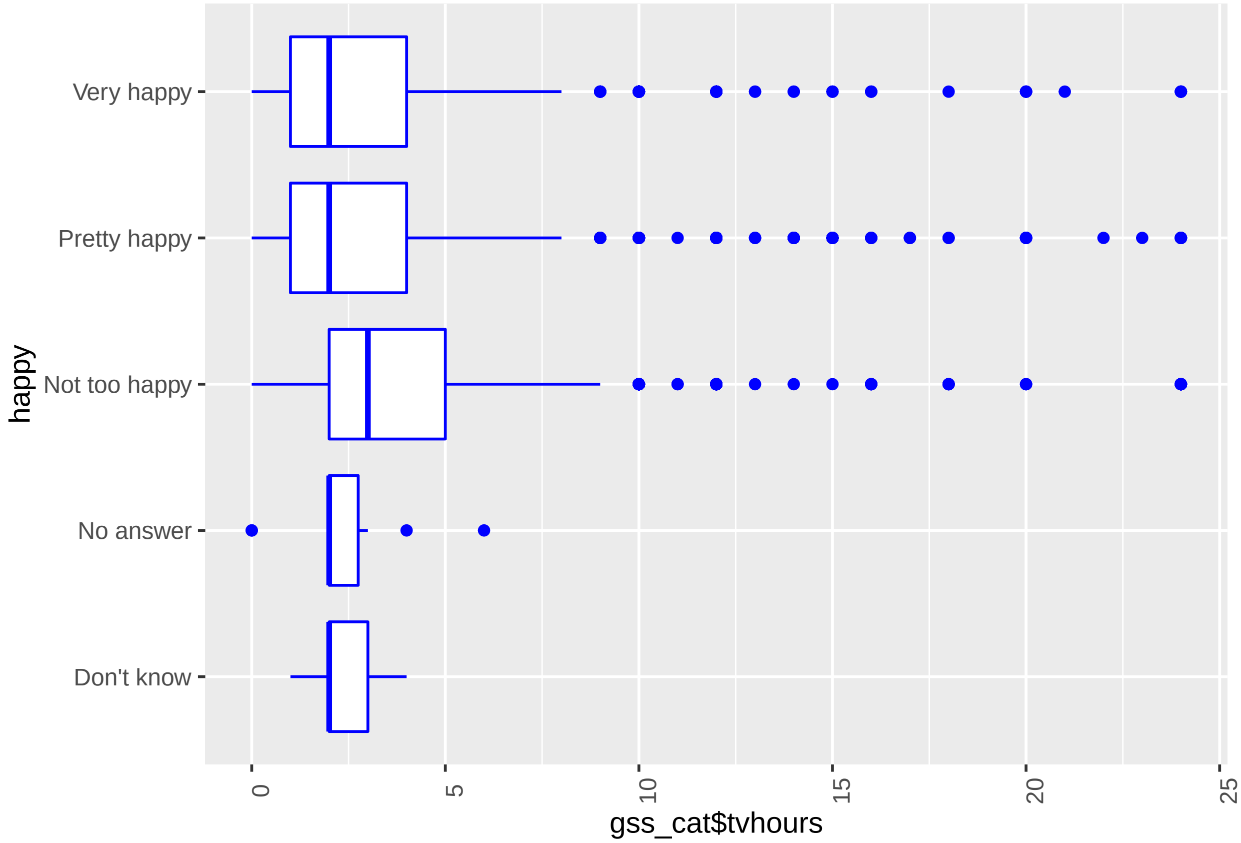
```
##
##                  Don't know No answer Not applicable Not too happy Pretty happy
##   Don't know              1         0              3             1            1
##   No answer               0         3              4             1            0
##   Not too happy           0         5            854            59          149
##   Pretty happy            2        12           3155            69         1125
##   Very happy              0         2           1116            11          170
##
##                  Very happy
##   Don't know              1
##   No answer               2
##   Not too happy          96
##   Pretty happy          940
##   Very happy           1373
```

```
##
##  Pearson's Chi-squared test
##
## data:  mytable
## X-squared = 2298.7, df = 20, p-value < 2.2e-16
```

From this point forward, we subset our dataset to consider rows that had a valid value >=0 for "tvhours" and had the "happy" and "pray" variables populated.

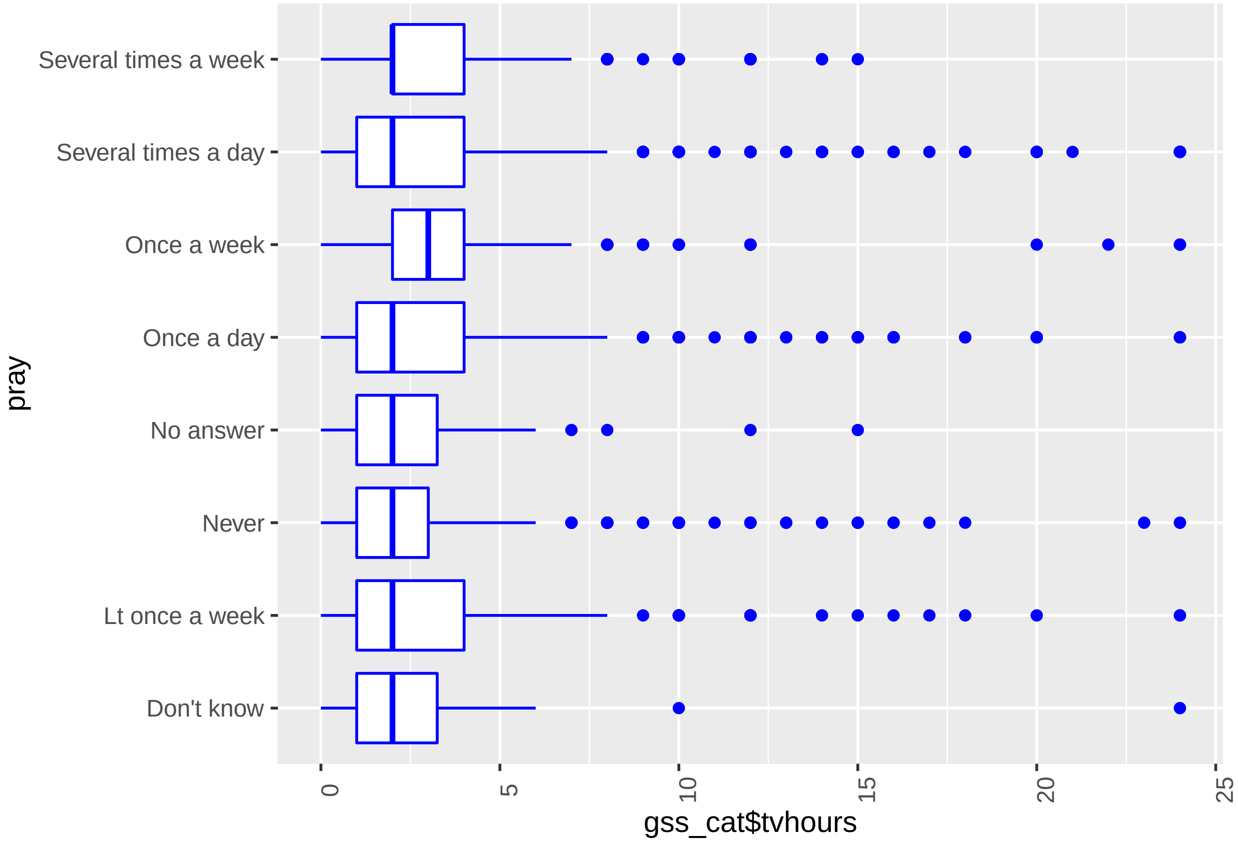
As seen in the plot below, the mean of TV hours watched is lower for the respondents who answer “very happy” and “pretty happy”.

```
ggplot(gss_cat, aes(x = happy, y = gss_cat$tvhours)) +
  geom_boxplot(color = "blue") +
  theme(axis.text.x = element_text(angle = 90)) +
  coord_flip()
```



As seen in the plot below, the mean of TV hours watched is higher for the respondents who pray “once a week or less”.

```
ggplot(gss_cat, aes(x = pray, y = gss_cat$tvhours)) +
      geom_boxplot(color = "blue") +
      theme(axis.text.x = element_text(angle = 90)) +
      coord_flip()
```

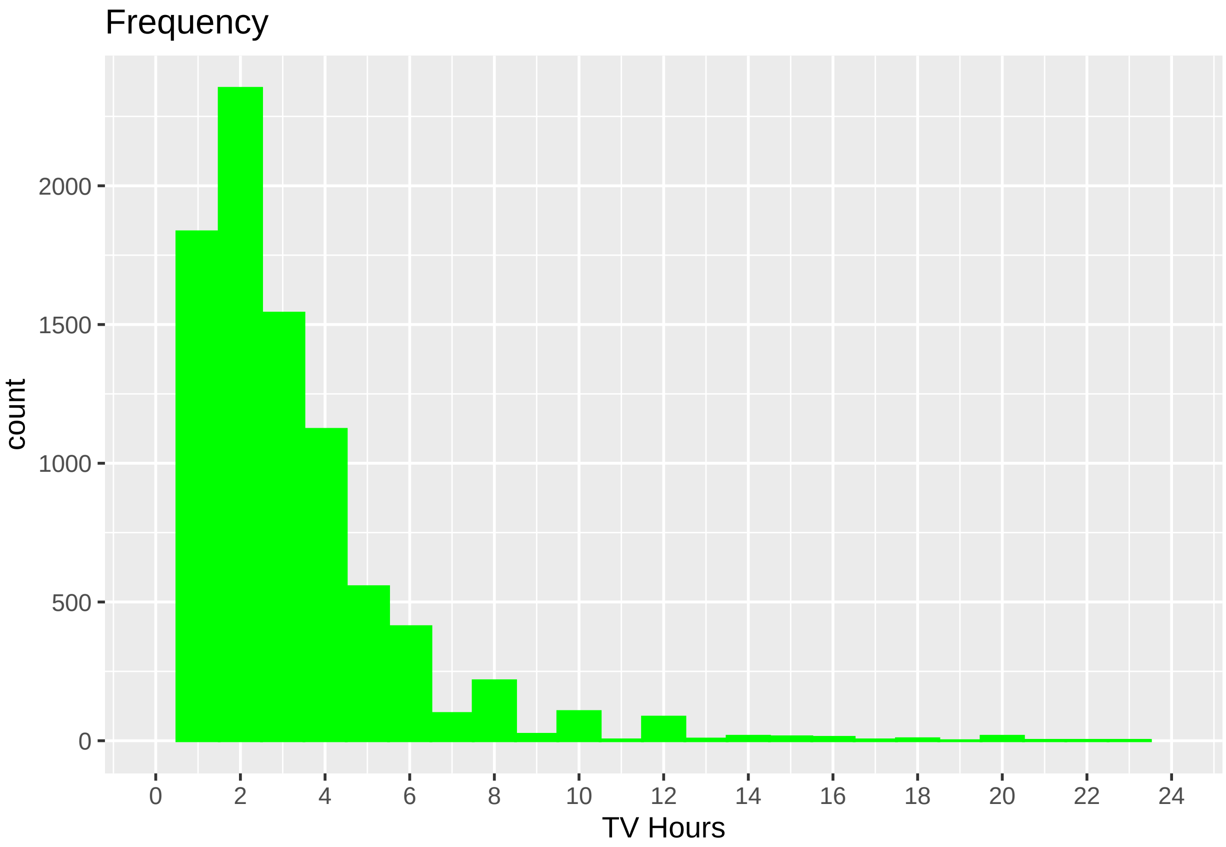


3.3 Transformations of the TV hours watched metric

One limitation of this study is that television watching is estimated for an average day for the respondent. We see many instances of 24 and 0 hours of television watching. We did not consider these outliers in the context of this study and what we know about television watching from the meter.

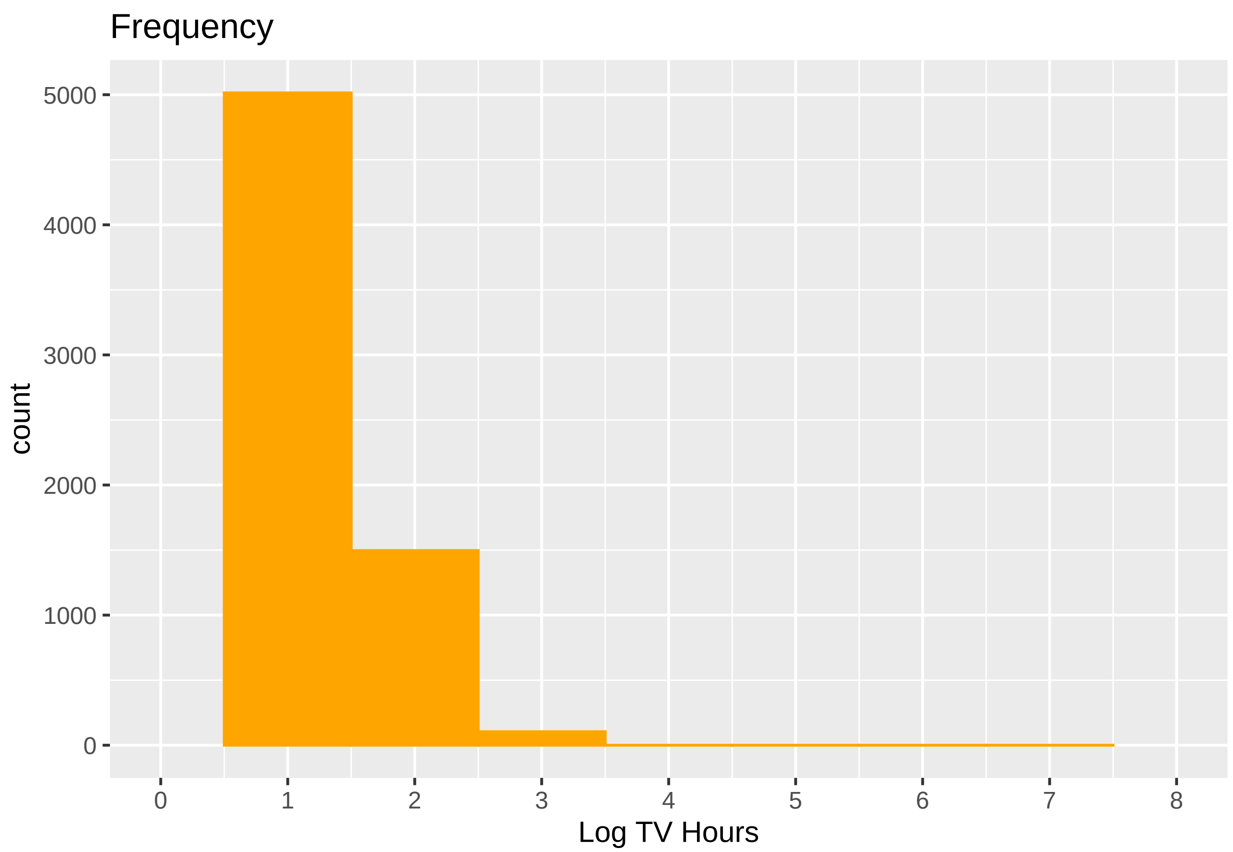
The plot below show the distribution of TV hours.

```
ggplot(gss_cat, aes(gss_cat$tvhours )) +
  geom_histogram(binwidth = 1, color = 'green', fill = 'green') +
  ggtitle("Frequency") +
  scale_x_continuous(name = "TV Hours",
                     breaks = seq(0, 24, 2),
                     limits = c(0, 24))
```



Since the distribution of TV hours is not normal, we also created a log transformation of TV hours to use in our models, see plot below.

```
ggplot(gss_cat, aes(log(gss_cat$tvhours) )) +
  geom_histogram(binwidth = 1, color = 'orange', fill = 'orange') +
  ggtitle("Frequency") +
  scale_x_continuous(name = "Log TV Hours",
                     breaks = seq(0, 8, 1),
                     limits =c (0, 8))
```



But, since we are also interested in looking at effects of “happy” and “pray” on “tvhours” we also created a binomial (watched any tv/did not watch tv). Because the GSS also asks about an approximation of TV hours watched in an average day, and there is little meaningful difference between 12 hours and 13 hours we concluded that creating a multinomial variable of “light”, “medium”, “heavy” TV watching.

4 Basic General Linear Model (GLM)

The first section of our analysis looks at using a general linear model on the log tranformation of TV hours using our in horizon variables compared to a model with TV hours. (Note we also compared models with and without the log transformation and found improved fit with the log transformation, as expected.)

The following code is used to run GLM.

```
#GLM models
lm_out0 <- glm(data = gss_TV_happy, tvhours ~ gender + race + age  + HH_income )
lm_out1 <- glm(data = gss_TV_happy, tvhours ~ gender + race + age  + HH_income + happy + pray)
lm_out2 <- glm(data = gss_TV_happy, tvhours ~ happy  + pray)

#log transformation
lm_out3 <- glm(data = gss_TV_happy, logTVhours ~ gender + race + age +HH_income )
lm_out4 <- glm(data = gss_TV_happy, logTVhours ~ gender + race + age +HH_income + happy + pray)
```

CompareGLM() is a function from the rcompanion package that shows similarities and differences between outputs of different models. The model with the lowest AIC value was the last model shown below, using the log transformation, the standard predictor variables, and “happy,” and “pray.”

```r
compareGLM(lm_out0, lm_out1, lm_out2, lm_out3, lm_out4)
```

```
## $Models
##   Formula
## 1 "tvhours ~ gender + race + age + HH_income"
## 2 "tvhours ~ gender + race + age + HH_income + happy + pray"
## 3 "tvhours ~ happy + pray"
## 4 "logTVhours ~ gender + race + age + HH_income"
## 5 "logTVhours ~ gender + race + age + HH_income + happy + pray"
##
## $Fit.criteria
##   Rank Df.res   AIC  AICc   BIC McFadden Cox.and.Snell Nagelkerke    p.value
## 1   18   8247 38940 38940 39080 0.038300       0.17100    0.17230  0.000e+00
## 2   25   8240 38900 38900 39080 0.039720       0.17670    0.17800  0.000e+00
## 3    8   8360 40310 40310 40380 0.004012       0.01921    0.01936 1.076e-251
## 4   18   8247 15650 15650 15790 0.083900       0.15890    0.18200  6.800e-97
## 5   25   8240 15590 15590 15780 0.088070       0.16610    0.19030  1.225e-97
```

The coefficient output for the last model shows signifigance for all variables at the same level, except “pray,” $\alpha = 0.05$.

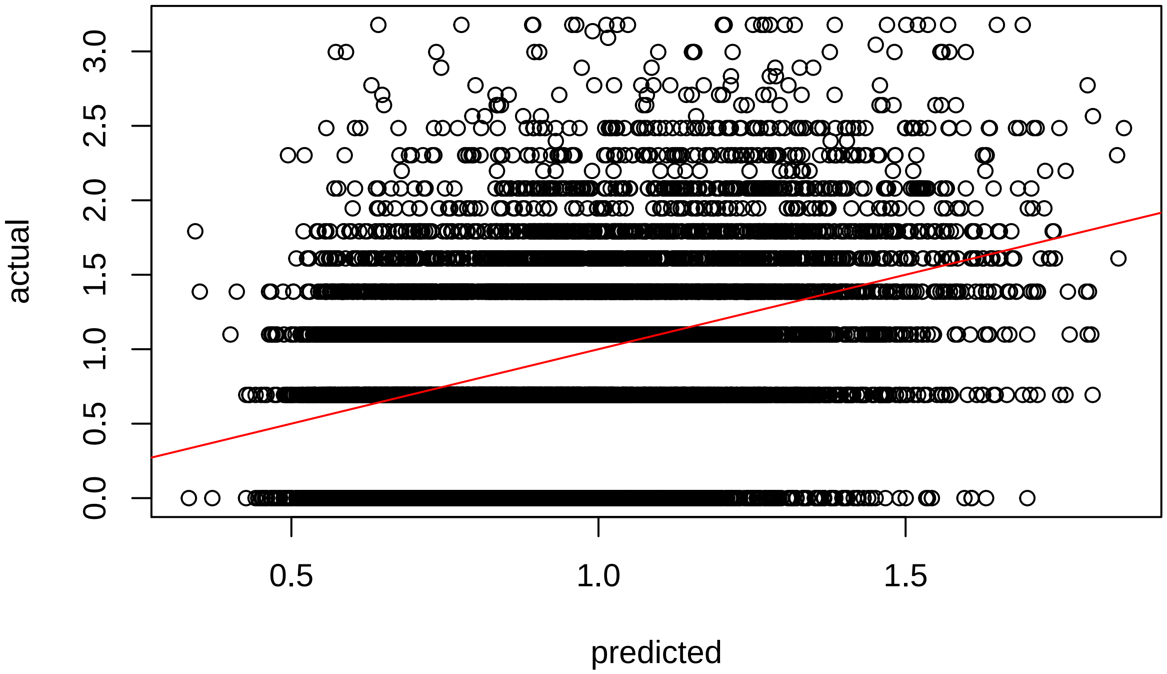
```r
summary(lm_out4)
```

```
##
## Call:
## glm(formula = logTVhours ~ gender + race + age + HH_income +
##     happy + pray, data = gss_TV_happy)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.69824  -0.48765   0.00945   0.41101   2.53647
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.1973169  0.0684876  17.482  < 2e-16 ***
## genderMale                0.0113832  0.0142063   0.801  0.42299
## raceOther                -0.3363435  0.0279754 -12.023  < 2e-16 ***
## raceWhite                -0.3199635  0.0195408 -16.374  < 2e-16 ***
## age                       0.0089509  0.0004116  21.745  < 2e-16 ***
## HH_income$10000 - 14999  -0.0592255  0.0653266  -0.907  0.36464
## HH_income$15000 - 19999  -0.0957427  0.0672593  -1.423  0.15463
## HH_income$20000 - 24999  -0.1777741  0.0650602  -2.732  0.00630 **
## HH_income$25000 or more  -0.4091229  0.0600287  -6.815 1.01e-11 ***
## HH_income$3000 to 3999   -0.0122981  0.0971126  -0.127  0.89923
## HH_income$4000 to 4999   -0.2235565  0.1146429  -1.950  0.05121 .
## HH_income$5000 to 5999   -0.0950172  0.1032217  -0.921  0.35733
## HH_income$6000 to 6999   -0.0127645  0.1019306  -0.125  0.90035
## HH_income$7000 to 7999   -0.0615937  0.0949360  -0.649  0.51649
## HH_income$8000 to 9999   -0.0266560  0.0781446  -0.341  0.73303
## HH_incomeDon't know      -0.2114259  0.0680994  -3.105  0.00191 **
## HH_incomeLt $1000        -0.0277561  0.0835490  -0.332  0.73974
## HH_incomeRefused         -0.5039230  0.0644315  -7.821 5.89e-15 ***
## happyPretty happy        -0.0884632  0.0216297  -4.090 4.36e-05 ***
## happyVery happy          -0.1616887  0.0235888  -6.854 7.67e-12 ***
## prayNever                 0.0150966  0.0283338   0.533  0.59418
## prayOnce a day           -0.0043542  0.0248433  -0.175  0.86088
## prayOnce a week           0.0605600  0.0343488   1.763  0.07792 .
## praySeveral times a day  -0.0398027  0.0254121  -1.566  0.11732
## praySeveral times a week  0.0448064  0.0299232   1.497  0.13433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3850842)
##
##     Null deviance: 3705.9  on 8264  degrees of freedom
## Residual deviance: 3173.1  on 8240  degrees of freedom
##   (103 observations deleted due to missingness)
## AIC: 15595
##
## Number of Fisher Scoring iterations: 2
```

4.1 Plot of Residuals of Prediction of TV Hours

While the last model was our best fitting GLM model, the residual plot below shows that there is ample room for improvement on model fit.

```r
plot(predict(lm_out4),y4,
     xlab = "predicted",ylab = "actual" )
abline(a = 0,b = 1, col = "red")
```



5 Logistic Regression

A brief story about recoding variables…

Logistic regression is a probabalistic binary outcome model commonly used in social sciences. We selected this model because we were curious whether there was a difference between people who did and did not watch television on an average day.

The following code is used to run logitic regression.

```
#simple logistic regression
logit <- glm(tvhours_YN ~ age, data=gssv_log_reg,family="binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = tvhours_YN ~ age, family = "binomial", data = gssv_log_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5640  -0.4570  -0.3732  -0.3079   2.6859
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.282845   0.110707  -11.59   <2e-16 ***
## age         -0.026399   0.002466  -10.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4952.0  on 9047  degrees of freedom
## Residual deviance: 4828.7  on 9046  degrees of freedom
##   (4746 observations deleted due to missingness)
## AIC: 4832.7
##
## Number of Fisher Scoring iterations: 5
```

```
TVwatching_by_age <- predict(logit,type="response")

#logistic regression with age and race
logit_age_race <- glm(tvhours_YN ~ age + race, data=gssv_log_reg,family="binomial")
summary(logit_age_race)
```

```
##
## Call:
## glm(formula = tvhours_YN ~ age + race, family = "binomial", data = gssv_log_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5962  -0.4546  -0.3686  -0.3000   2.8301
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.758553   0.154199 -11.404  < 2e-16 ***
## age         -0.027503   0.002494 -11.027  < 2e-16 ***
## raceOther    0.530749   0.167757   3.164  0.00156 **
## raceWhite    0.616294   0.128445   4.798  1.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4952.0  on 9047  degrees of freedom
## Residual deviance: 4802.4  on 9044  degrees of freedom
##   (4746 observations deleted due to missingness)
## AIC: 4810.4
##
## Number of Fisher Scoring iterations: 5
```

```
TVwatching_by_age_race <- predict(logit_age_race,type="response")

#logistic regression with age, race, and labor status
logit_age_race_labor <- glm(tvhours_YN ~ age + race + labor, data=gssv_log_reg,family="binomial")
summary(logit_age_race_labor)
```

```
##
## Call:
```

```
## glm(formula = tvhours_YN ~ age + race + labor, family = "binomial",
##     data = gssv_log_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6608  -0.4534  -0.3806  -0.2807   2.8594
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.415522   0.220684 -10.946  < 2e-16 ***
## age                   -0.021501   0.003071  -7.002 2.52e-12 ***
## raceOther              0.525376   0.168053   3.126 0.001771 **
## raceWhite              0.615816   0.128724   4.784 1.72e-06 ***
## laborNo answer         0.675866   1.067317   0.633 0.526578
## laborOther             0.022004   0.330342   0.067 0.946893
## laborRetired           0.004227   0.217267   0.019 0.984479
## laborSchool            0.776013   0.221095   3.510 0.000448 ***
## laborTemp not working  0.166819   0.331853   0.503 0.615184
## laborUnempl, laid off  0.568940   0.220559   2.580 0.009893 **
## laborWorking fulltime  0.485903   0.152778   3.180 0.001471 **
## laborWorking parttime  0.500855   0.179627   2.788 0.005298 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4952.0  on 9047  degrees of freedom
## Residual deviance: 4778.8  on 9036  degrees of freedom
##   (4746 observations deleted due to missingness)
## AIC: 4802.8
##
## Number of Fisher Scoring iterations: 6
```

```
TVwatching_by_age_race_labor <- predict(logit_age_race_labor,type="response")

#logistic regression for NAs as zero
logit2 <- glm(tvhours_NA_0_YN ~ age + race , data=gssv_log_reg,family="binomial")
summary(logit2)
```

```
##
## Call:
## glm(formula = tvhours_NA_0_YN ~ age + race, family = "binomial",
##     data = gssv_log_reg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0699  -0.9983  -0.9492   1.3506   1.5263
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.311615   0.064462  -4.834 1.34e-06 ***
## age         -0.005508   0.001035  -5.323 1.02e-07 ***
## raceOther    0.152452   0.071395   2.135  0.03273 *
## raceWhite    0.138934   0.049676   2.797  0.00516 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18193  on 13623  degrees of freedom
## Residual deviance: 18157  on 13620  degrees of freedom
##   (170 observations deleted due to missingness)
## AIC: 18165
##
## Number of Fisher Scoring iterations: 4
```

6 Random Forest Machine Learning

This is a subsection.

```
gssv$age<-as.numeric(gssv$age)

#create dataset with tvhours not = NA
gss_TV <-gssv[complete.cases(gssv[ , "tvhours"]),]

#Subset to just people who also have a "Happy" answer and "Pray" answer, we only lose a couple hundred
gss_TV<-as_tibble(gss_TV)
gss_TV_happy<-subset(gss_TV, gss_TV$happy != "No answer" & gss_TV$happy !="Don't know")

gss_TV_happy<- subset(gss_TV_happy, gss_TV_happy$pray !="Don't know" & gss_TV_happy$pray !="No answer")

#read the file, please change the file path.
TheData <- gss_TV_happy
TheData = TheData[,c("tvhours_NA_0_cat","age","gender","race","HH_income","happy","pray")]

#Ramdomly reorder rows of the dataset.
TheData <- TheData[sample(nrow(TheData)),]

#treat these variables as factor
TheData$tvhours_NA_0_cat<-as.factor(TheData$tvhours_NA_0_cat)
TheData$happy <- as.factor(TheData$happy)
TheData$pray <- as.factor(TheData$pray)
TheData$gender <- as.factor(TheData$gender)
TheData$race <- as.factor(TheData$race)
TheData$HH_income <- as.factor(TheData$HH_income)

TheData <- TheData[complete.cases(TheData),]

# select which variable you want to use to train the decision tree.
baseFormula <- tvhours_NA_0_cat ~ age + gender + race + HH_income
happyFormula <- tvhours_NA_0_cat ~ age + gender + race + HH_income + happy
prayFormula <- tvhours_NA_0_cat ~ age + gender + race + HH_income + happy + pray

#create two arrays to store precisions and recalls
numrow  = as.integer((nrow(TheData))/7)
```

```
TheResultData <- data.frame()
#Use 7 fold cross validation, devide the dataset into 7 part, each time use 6 part to train and 1 part
for(i in 0:6){
  testrow <- (numrow * i) : (numrow * (i+1) -1)

  if(i == 6){
    testrow <- (numrow * i) : nrow(TheData)
  }
  TestingData <- TheData[testrow,]
  TrainingData <- TheData[-testrow,]
  # Train
  fit <- randomForest(baseFormula,
                      data=TrainingData,
                      importance=TRUE,replace=FALSE, nodesize = 10 ,
                      proximity=TRUE,
                      ntree=500)

  happyFit <- randomForest(happyFormula,
                           data=TrainingData,
                           importance=TRUE,replace=FALSE, nodesize = 10 ,
                           proximity=TRUE,
                           ntree=500)

  prayFit <- randomForest(prayFormula,
                          data=TrainingData,
                          importance=TRUE,replace=FALSE, nodesize = 10 ,
                          proximity=TRUE,
                          ntree=500)

  # Predict
  prediction <- predict(fit,TestingData, norm.votes=TRUE, type = "cl")
  happyPrediction <- predict(happyFit,TestingData, norm.votes=TRUE, type = "cl")
  prayPrediction <- predict(prayFit,TestingData, norm.votes=TRUE, type = "cl")

  predictionMatrix <- as.data.frame(prediction)
  happyPredictionMatrix <- as.data.frame(happyPrediction)
  prayPredictionMatrix <- as.data.frame(prayPrediction)

  TestingData <- cbind(TestingData, predictionMatrix, happyPredictionMatrix, prayPredictionMatrix)
  TheResultData <- rbind(TheResultData, TestingData)
}

fit
```

```
##
## Call:
##  randomForest(formula = baseFormula, data = TrainingData, importance = TRUE,      replace = FALSE, n
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 51.8%
## Confusion matrix:
```

```
##        heavy light medium none class.error
## heavy    123   350    414    1   0.8614865
## light     79  2636    797    5   0.2504976
## medium   154  1583    942    4   0.6489005
## none      11   459    121    0   1.0000000
```

```
happyFit
```

```
##
## Call:
##  randomForest(formula = happyFormula, data = TrainingData, importance = TRUE,      replace = FALSE, r
##                 Type of random forest: classification
##                       Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 51.19%
## Confusion matrix:
##        heavy light medium none class.error
## heavy    118   353    417    0   0.8671171
## light     62  2702    753    0   0.2317316
## medium   126  1626    925    6   0.6552367
## none      11   446    131    3   0.9949239
```

```
prayFit
```

```
##
## Call:
##  randomForest(formula = prayFormula, data = TrainingData, importance = TRUE,      replace = FALSE, no
##                 Type of random forest: classification
##                       Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 51.41%
## Confusion matrix:
##        heavy light medium none class.error
## heavy    105   349    430    4   0.8817568
## light     68  2684    761    4   0.2368496
## medium   136  1603    937    7   0.6507641
## none      12   451    123    5   0.9915398
```

```
importance(fit, type = 1)
```

```
##            MeanDecreaseAccuracy
## age                  89.230214
## gender                8.725872
## race                 48.021420
## HH_income            70.040010
```

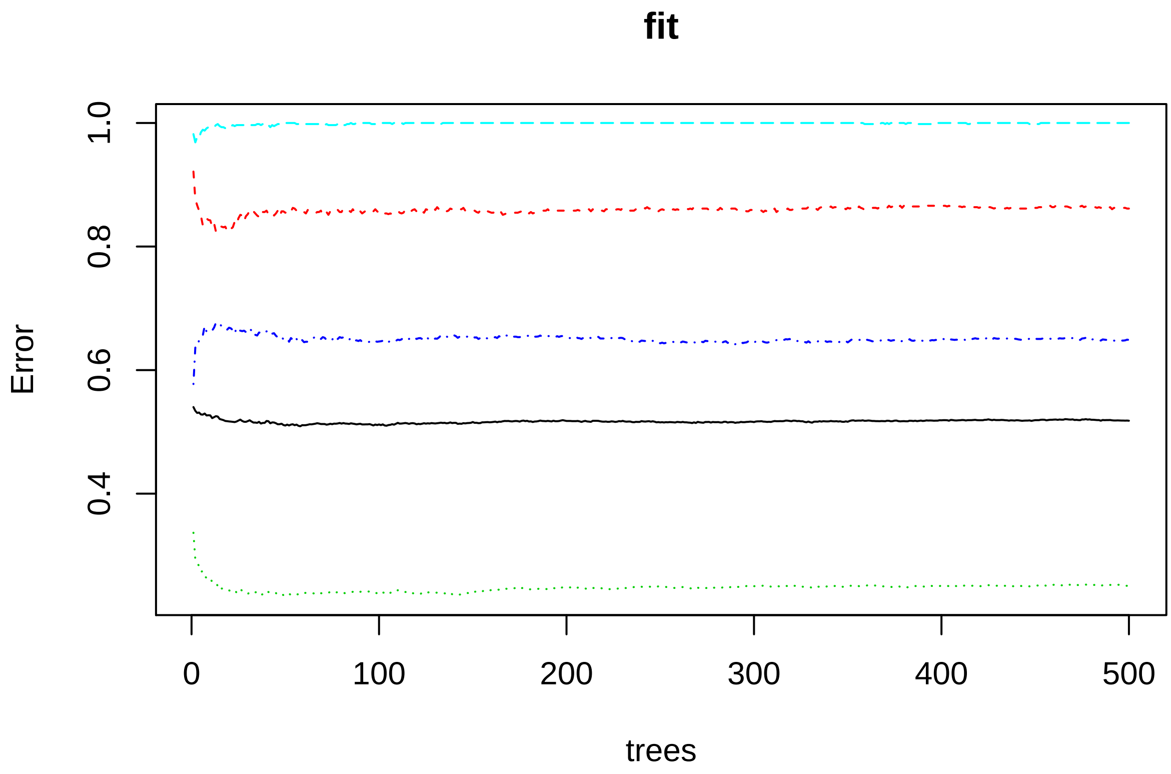
```
importance(happyFit, type = 1)
```

```
##            MeanDecreaseAccuracy
## age                    84.292932
## gender                  6.442704
## race                   45.870621
## HH_income              59.647002
## happy                   8.037837
```

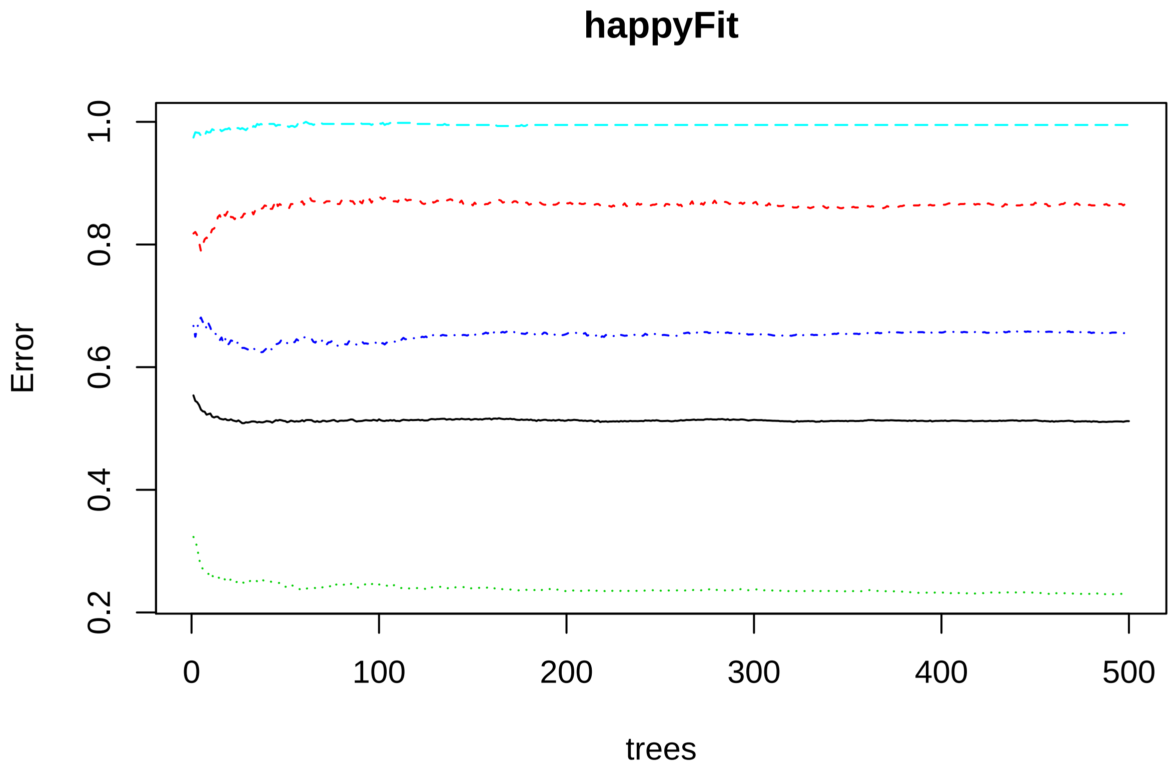
```
importance(prayFit, type = 1)
```

```
##           MeanDecreaseAccuracy
## age                   67.73469
## gender                 2.99326
## race                  38.45778
## HH_income             56.38370
## happy                  7.93659
## pray                   1.26690
```

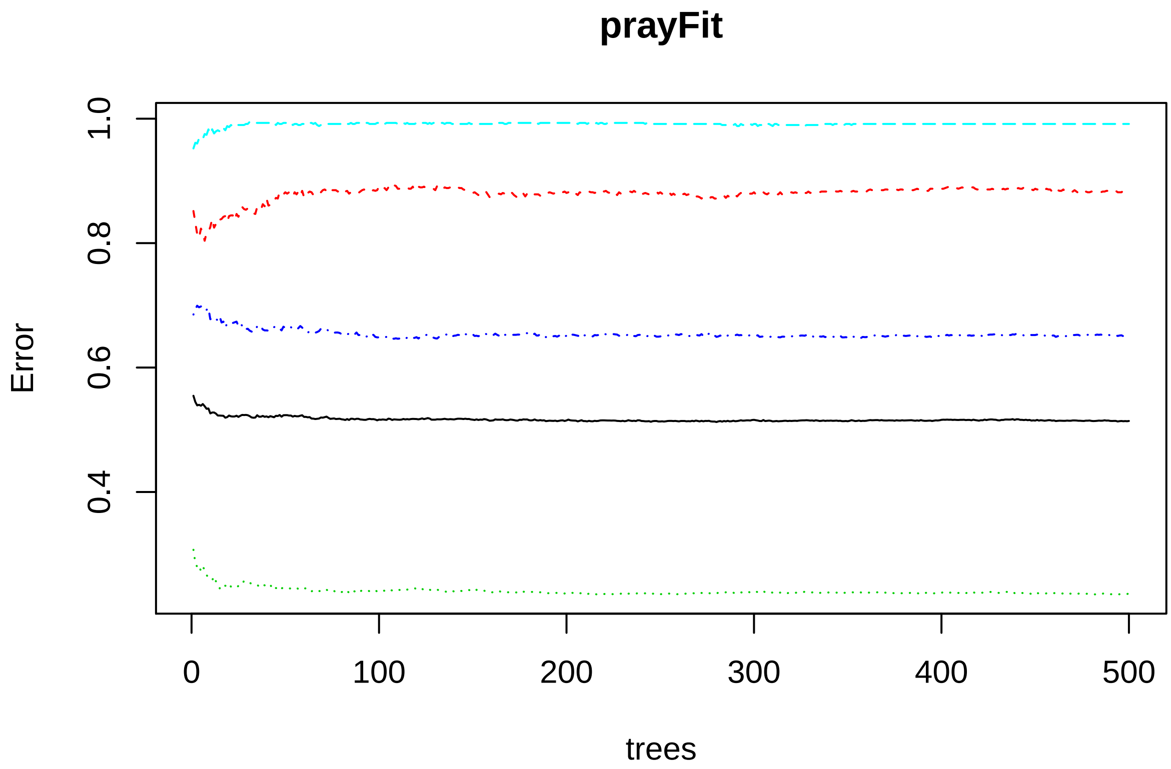
```
plot(fit)
```



```
plot(happyFit)
```



```
plot(prayFit)
```



7 Results

“Happy” and “pray” improved model fit in the GLM model and were important factors int he random forest model. The logistic regression model did not show much explanatory promise. Further research into ways to improve binomial and multinomial models fit could be fruitful. Certain categorical variables could have been recorded to a scale. This may improve model fit, and it would allow additional models to be tested.

8 References

"About the GSS NORC." 2016. http://gss.norc.org/About-The-GSS.

"GSS Data Explorer NORC at the University of Chicago." 2020. https://gssdataexplorer.norc.org/projects.

"GSS Data Explorer NORC at the University of Chicago Questionnaire 2014 Gss V1 English." 2020. https://gssdataexplorer.norc.org/documents/571/display.

Madrigal, Alexis. 2018. "How Much TV Do People Watch? - the Atlantic." *The Atlantic*. https://www.theatlantic.com/technology/archive/2018/05/when-did-tv-watching-peak/561464/.

Nielsen. 2020. "National Reference Supplement 2019-2020." https://answers.nielsen.com/portal/workspace/US+MEDIA+Client+Workspaces/National/Reference/National+Reference+Supplement/NRS+19-20+FINAL.pdf.