Capstone Project Pre-Processing and Feature Engineering

Steve Uzupis

2024-07-07

*Packages Load*

# load packages  
  
library(readr) # load csv files  
#library(readxl) # load excel files  
library(dplyr) # data manipulation

##   
## 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

#library(lubridate) # date & time manipulation  
library(ggplot2) # data visualization  
#library(tidyr) # collection of statistical packages, packages loaded individually  
#library(corrplot) # to visualize correlations  
#library(leaps) # for subset selection  
library(caret) # test for correlation and create dummy variables

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice

#library(car) # for VIF  
#library(scales) # for visualizing plots in %  
#library(forcats) # ordering data frames  
#library(codebookr) # adding appendix to r code  
#library(gtsummary) # creating tables  
#library(cardx) # to include statistic results  
#library(moments) # to calculate skewness and kurtosis  
#library(VIM) # to run K- Nearest Neighbour  
#library(pROC) # to analyse and display reciever operating characteristics(ROC) curvees  
#library(randomForest) # to impletement Random Forest algorithm  
#library(mice) # Imputing missing values using mice

*Data load*

# load data  
sdoh\_data <- read\_csv("data/sdoh\_data.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,  
## e.g.:  
## dat <- vroom(...)  
## problems(dat)

## Rows: 3229 Columns: 682  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (15): COUNTYFIPS, STATEFIPS, STATE, COUNTY, REGION, CAF\_ADJ\_COUNTY\_1, C...  
## dbl (664): YEAR, TERRITORY, ACS\_TOT\_POP\_WT, ACS\_TOT\_POP\_US\_ABOVE1, ACS\_TOT\_P...  
## lgl (3): CAF\_ADJ\_COUNTY\_12, CAF\_ADJ\_COUNTY\_13, CAF\_ADJ\_COUNTY\_14  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

dim(sdoh\_data)

## [1] 3229 682

# remove unwanted features, create calculated feature, convert fips\_code to data type matching chr\_data  
sdoh\_data <- sdoh\_data %>%   
 select("COUNTYFIPS",   
 "STATE",   
 "COUNTY",   
 "REGION",   
 "ACS\_TOT\_POP\_WT",   
 "ACS\_AVG\_HH\_SIZE",   
 "ACS\_PCT\_MALE",   
 "ACS\_PCT\_AIAN",   
 "ACS\_PCT\_ASIAN",   
 "ACS\_PCT\_BLACK",   
 "ACS\_PCT\_HISPANIC",   
 "ACS\_PCT\_OTHER\_RACE",   
 "ACS\_PCT\_WHITE",   
 "ACS\_PCT\_CHILD\_1FAM",   
 "ACS\_PCT\_CHILDREN\_GRANDPARENT",   
 "ACS\_PCT\_GRANDP\_RESPS\_NO\_P",   
 "ACS\_PCT\_GRANDP\_RESPS\_P",  
 "ACS\_PCT\_HH\_OTHER\_COMP",   
 "ACS\_PCT\_HH\_INTERNET",  
 "ACS\_PCT\_EMPLOYED",  
 "ACS\_PCT\_HH\_INC\_99999",  
 "ACS\_PCT\_MEDICARE\_ONLY",  
 "AHRF\_CLIN\_NURSE\_SPEC\_RATE",   
 "AHRF\_DENTISTS\_RATE",  
 "AHRF\_PHYSICIAN\_ASSIST\_RATE",  
 "AMFAR\_MHFAC\_RATE",  
 "CEN\_POPDENSITY\_COUNTY",  
 "NEPHTN\_HEATIND\_90",  
 "SAIPE\_MEDIAN\_HH\_INCOME",  
 "POS\_MEDIAN\_DIST\_ED",  
 "POS\_MEDIAN\_DIST\_PED\_ICU",  
 "POS\_MEDIAN\_DIST\_CLINIC",   
 "POS\_MEDIAN\_DIST\_ALC",   
 ) %>%   
 mutate(pct\_grandparents\_as\_guardians = ACS\_PCT\_CHILDREN\_GRANDPARENT \* ((ACS\_PCT\_GRANDP\_RESPS\_P + ACS\_PCT\_GRANDP\_RESPS\_NO\_P)/100)) %>%   
 select(-ACS\_PCT\_GRANDP\_RESPS\_P, -ACS\_PCT\_GRANDP\_RESPS\_NO\_P, -ACS\_PCT\_CHILDREN\_GRANDPARENT) %>%   
 rename("fips\_code" = "COUNTYFIPS",  
 "state" = "STATE",  
 "county" = "COUNTY",  
 "region" = "REGION",  
 "weighted\_population" = "ACS\_TOT\_POP\_WT",  
 "average\_hh\_size" = "ACS\_AVG\_HH\_SIZE",  
 "pct\_male" = "ACS\_PCT\_MALE",  
 "pct\_native\_american" = "ACS\_PCT\_AIAN",  
 "pct\_asian" = "ACS\_PCT\_ASIAN",  
 "pct\_black" = "ACS\_PCT\_BLACK",  
 "pct\_hispanic" = "ACS\_PCT\_HISPANIC",  
 "pct\_other\_race" = "ACS\_PCT\_OTHER\_RACE",  
 "pct\_white" = "ACS\_PCT\_WHITE",  
 "pct\_single\_parent" = "ACS\_PCT\_CHILD\_1FAM",  
 "pct\_hh\_other\_computer" = "ACS\_PCT\_HH\_OTHER\_COMP",  
 "pct\_hh\_internet" = "ACS\_PCT\_HH\_INTERNET",  
 "pct\_employed" = "ACS\_PCT\_EMPLOYED",  
 "pct\_hh\_inc\_99999" = "ACS\_PCT\_HH\_INC\_99999", # renamed by mg  
 "pct\_w\_medicare" = "ACS\_PCT\_MEDICARE\_ONLY",  
 "clinical\_nurse\_pt" = "AHRF\_CLIN\_NURSE\_SPEC\_RATE",  
 "dentist\_pt" = "AHRF\_DENTISTS\_RATE",  
 "pa\_pt" = "AHRF\_PHYSICIAN\_ASSIST\_RATE",  
 "mental\_health\_faciliy\_pt" = "AMFAR\_MHFAC\_RATE",  
 "population\_density" = "CEN\_POPDENSITY\_COUNTY",  
 "days\_over\_90\_f" = "NEPHTN\_HEATIND\_90",  
 "median\_hh\_income" = "SAIPE\_MEDIAN\_HH\_INCOME",  
 "median\_er\_dist" = "POS\_MEDIAN\_DIST\_ED",  
 "median\_pediatric\_icu\_dist" = "POS\_MEDIAN\_DIST\_PED\_ICU",  
 "median\_health\_clinic\_dist" = "POS\_MEDIAN\_DIST\_CLINIC",  
 "median\_drug\_alcohol\_care\_dist" = "POS\_MEDIAN\_DIST\_ALC"  
   
 ) %>%   
 mutate(fips\_code = as.numeric(fips\_code))  
  
chr\_data <- read\_csv("data/chr\_data.csv", skip = 1)

## Rows: 3194 Columns: 720  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): state, county  
## dbl (572): statecode, countycode, fipscode, year, county\_ranked, v001\_rawval...  
## lgl (146): v002\_numerator, v002\_denominator, v036\_numerator, v036\_denominato...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

dim(chr\_data)

## [1] 3194 720

# remove unwanted features  
# convert principal care providers from per 100,000 people to per 1,000 people to match other data  
  
chr\_data <- chr\_data %>%  
 select("fipscode",  
 "v002\_rawvalue",  
 "v009\_rawvalue",  
 "v011\_rawvalue",  
 "v070\_rawvalue",   
 "v049\_rawvalue",  
 "v085\_rawvalue",  
 "v168\_rawvalue",   
 "v069\_rawvalue",  
 "v044\_rawvalue",   
 "v140\_rawvalue",  
 "v125\_rawvalue",  
 "v124\_rawvalue",  
 "v136\_other\_data\_1",  
 "v136\_other\_data\_2",  
 "v137\_rawvalue",  
 "v147\_rawvalue",  
 "v139\_rawvalue",  
 "v177\_rawvalue",  
 "v153\_rawvalue",  
 "v053\_rawvalue",   
 "v058\_rawvalue",   
 "v004\_rawvalue",  
 ) %>%   
 mutate(pcp\_pt = v004\_rawvalue/100) %>%   
 select(-v004\_rawvalue) %>%   
 rename("fips\_code" = "fipscode",  
 "pct\_poor\_to\_fair\_health" = "v002\_rawvalue",  
 "pct\_adult\_smokers" = "v009\_rawvalue",  
 "pct\_obese\_adults" = "v011\_rawvalue",  
 "pct\_no\_exercise" = "v070\_rawvalue",  
 "pct\_binge\_drinkers" = "v049\_rawvalue",  
 "pct\_under\_65\_no\_health\_insurance" = "v085\_rawvalue",  
 "pct\_highschool\_diploma" = "v168\_rawvalue",  
 "pct\_some\_college" = "v069\_rawvalue",  
 "inequality\_ratio" = "v044\_rawvalue",  
 "social\_clubs\_per\_10k" = "v140\_rawvalue",  
 "air\_polution\_metric" = "v125\_rawvalue",  
 "water\_quality" = "v124\_rawvalue", # renamed by mg  
 "pct\_high\_housing\_costs" = "v136\_other\_data\_1",  
 "pct\_overcrowded\_hh" = "v136\_other\_data\_2",  
 "pct\_food\_insecurities" = "v139\_rawvalue",  
 "pct\_voters" = "v177\_rawvalue",  
 "pct\_home\_owner" = "v153\_rawvalue",  
 "pct\_65\_plus" = "v053\_rawvalue",  
 "pct\_rural\_population" = "v058\_rawvalue",  
 "life\_expectancy\_years" = "v147\_rawvalue",  
 "pct\_30\_min\_plus\_commute" = "v137\_rawvalue")  
  
# full data sets are extremely large, initial dimension reduction was performed previously

*Combine datasets*

# Create and clean the qol\_data dataset  
qol\_data <- sdoh\_data %>%  
 inner\_join(chr\_data, by = "fips\_code") %>%  
 mutate(response = ifelse(pct\_poor\_to\_fair\_health >= 0.154, "worse", "better")) %>%  
 mutate(response = as.factor(response)) %>%  
 select(-pct\_poor\_to\_fair\_health) %>% # keep until analysis has been performed  
 mutate\_at(vars(state, county, region), as.factor) #%>% # convert characters to factors  
# rename("response" = "pct\_poor\_to\_fair\_health")  
  
# clean local encviroment  
rm(chr\_data)  
rm(sdoh\_data)

eliminate variables with no predictive value: fipscode, county and state, pct\_poor\_to\_fair\_health:

qol\_data <- qol\_data %>%   
 select(  
 -c(  
 fips\_code,  
 county,  
 state,  
 life\_expectancy\_years  
 )  
 )

calcSplitRatio <- function(p = NA, df) {  
 ## @p = the number of parameters. by default, if none are provided, the number of columns (predictors) in the dataset are used  
 ## @df = the dataframe that will be used for the analysis  
   
 ## If the number of parameters isn't supplied, set it to the number of features minus 1 for the target  
 if(is.na(p)) {  
 p <- ncol(df) -1 ## COMMENT HERE  
 }  
   
 ## Calculate the ideal number of testing set  
 test\_N <- (1/sqrt(p))\*nrow(df)  
 ## Turn that into a testing proportion  
 test\_prop <- round((1/sqrt(p))\*nrow(df)/nrow(df), 2)  
 ## And find the training proportion  
 train\_prop <- 1-test\_prop  
   
 ## Tell us the results!  
 print(paste0("The ideal split ratio is ", train\_prop, ":", test\_prop, " training:testing"))  
   
 ## Return the size of the training set  
 return(train\_prop)  
}  
  
calcSplitRatio(df = qol\_data)

## [1] "The ideal split ratio is 0.86:0.14 training:testing"

## [1] 0.86

# Find columns with missing data  
colSums(is.na(qol\_data))

## region weighted\_population   
## 0 1   
## average\_hh\_size pct\_male   
## 1 1   
## pct\_native\_american pct\_asian   
## 1 1   
## pct\_black pct\_hispanic   
## 1 1   
## pct\_other\_race pct\_white   
## 1 1   
## pct\_single\_parent pct\_hh\_other\_computer   
## 2 1   
## pct\_hh\_internet pct\_employed   
## 1 1   
## pct\_hh\_inc\_99999 pct\_w\_medicare   
## 1 1   
## clinical\_nurse\_pt dentist\_pt   
## 0 0   
## pa\_pt mental\_health\_faciliy\_pt   
## 0 0   
## population\_density days\_over\_90\_f   
## 1 34   
## median\_hh\_income median\_er\_dist   
## 2 1   
## median\_pediatric\_icu\_dist median\_health\_clinic\_dist   
## 13 1   
## median\_drug\_alcohol\_care\_dist pct\_grandparents\_as\_guardians   
## 1 40   
## pct\_adult\_smokers pct\_obese\_adults   
## 2 2   
## pct\_no\_exercise pct\_binge\_drinkers   
## 2 2   
## pct\_under\_65\_no\_health\_insurance pct\_highschool\_diploma   
## 1 0   
## pct\_some\_college inequality\_ratio   
## 0 7   
## social\_clubs\_per\_10k air\_polution\_metric   
## 0 27   
## water\_quality pct\_high\_housing\_costs   
## 43 0   
## pct\_overcrowded\_hh pct\_30\_min\_plus\_commute   
## 0 0   
## pct\_food\_insecurities pct\_voters   
## 0 30   
## pct\_home\_owner pct\_65\_plus   
## 0 0   
## pct\_rural\_population pcp\_pt   
## 7 147   
## response   
## 2

# sum of NAs in each column  
na\_counts <- colSums(is.na(qol\_data))  
  
# combine column names and NA counts into a dataframe  
na\_counts\_df <- data.frame(variable\_name = names(na\_counts), na\_count = na\_counts)  
  
# Sort the dataframe by NA\_Count in descending order  
na\_counts\_df <- na\_counts\_df[order(-na\_counts\_df$na\_count), ]  
  
# View the sorted dataframe  
print(na\_counts\_df)

## variable\_name na\_count  
## pcp\_pt pcp\_pt 147  
## water\_quality water\_quality 43  
## pct\_grandparents\_as\_guardians pct\_grandparents\_as\_guardians 40  
## days\_over\_90\_f days\_over\_90\_f 34  
## pct\_voters pct\_voters 30  
## air\_polution\_metric air\_polution\_metric 27  
## median\_pediatric\_icu\_dist median\_pediatric\_icu\_dist 13  
## inequality\_ratio inequality\_ratio 7  
## pct\_rural\_population pct\_rural\_population 7  
## pct\_single\_parent pct\_single\_parent 2  
## median\_hh\_income median\_hh\_income 2  
## pct\_adult\_smokers pct\_adult\_smokers 2  
## pct\_obese\_adults pct\_obese\_adults 2  
## pct\_no\_exercise pct\_no\_exercise 2  
## pct\_binge\_drinkers pct\_binge\_drinkers 2  
## response response 2  
## weighted\_population weighted\_population 1  
## average\_hh\_size average\_hh\_size 1  
## pct\_male pct\_male 1  
## pct\_native\_american pct\_native\_american 1  
## pct\_asian pct\_asian 1  
## pct\_black pct\_black 1  
## pct\_hispanic pct\_hispanic 1  
## pct\_other\_race pct\_other\_race 1  
## pct\_white pct\_white 1  
## pct\_hh\_other\_computer pct\_hh\_other\_computer 1  
## pct\_hh\_internet pct\_hh\_internet 1  
## pct\_employed pct\_employed 1  
## pct\_hh\_inc\_99999 pct\_hh\_inc\_99999 1  
## pct\_w\_medicare pct\_w\_medicare 1  
## population\_density population\_density 1  
## median\_er\_dist median\_er\_dist 1  
## median\_health\_clinic\_dist median\_health\_clinic\_dist 1  
## median\_drug\_alcohol\_care\_dist median\_drug\_alcohol\_care\_dist 1  
## pct\_under\_65\_no\_health\_insurance pct\_under\_65\_no\_health\_insurance 1  
## region region 0  
## clinical\_nurse\_pt clinical\_nurse\_pt 0  
## dentist\_pt dentist\_pt 0  
## pa\_pt pa\_pt 0  
## mental\_health\_faciliy\_pt mental\_health\_faciliy\_pt 0  
## pct\_highschool\_diploma pct\_highschool\_diploma 0  
## pct\_some\_college pct\_some\_college 0  
## social\_clubs\_per\_10k social\_clubs\_per\_10k 0  
## pct\_high\_housing\_costs pct\_high\_housing\_costs 0  
## pct\_overcrowded\_hh pct\_overcrowded\_hh 0  
## pct\_30\_min\_plus\_commute pct\_30\_min\_plus\_commute 0  
## pct\_food\_insecurities pct\_food\_insecurities 0  
## pct\_home\_owner pct\_home\_owner 0  
## pct\_65\_plus pct\_65\_plus 0

rm(na\_counts\_df)

# Delete rows with missing values in the "response" column  
qol\_data <- qol\_data[!is.na(qol\_data$response), ]  
  
# delete column with large number of missing values 'pcp\_pt'  
qol\_data <- qol\_data[, !names(qol\_data) %in% "pcp\_pt"]

# Splitting the data into train and test sets  
qol\_train <- qol\_data[sample(nrow(qol\_data), 0.86\*nrow(qol\_data)), ]  
qol\_test <- qol\_data[!(rownames(qol\_data) %in% rownames(qol\_train)), ]

# Selecting only numeric columns for imputation  
numeric\_cols <- sapply(qol\_train, is.numeric)  
qol\_train\_numeric <- qol\_train[, numeric\_cols]  
qol\_test\_numeric <- qol\_test[, numeric\_cols]

# Imputing missing values in the training set  
qol\_train\_imputed <- mice::mice(  
 qol\_train\_numeric,   
 m = 1,   
 method = "cart", # Using CART for imputation  
 maxit = 5 # Maximum iterations for imputation  
 )

##   
## iter imp variable  
## 1 1 pct\_single\_parent days\_over\_90\_f median\_hh\_income median\_pediatric\_icu\_dist pct\_grandparents\_as\_guardians pct\_under\_65\_no\_health\_insurance inequality\_ratio air\_polution\_metric water\_quality pct\_voters pct\_rural\_population  
## 2 1 pct\_single\_parent days\_over\_90\_f median\_hh\_income median\_pediatric\_icu\_dist pct\_grandparents\_as\_guardians pct\_under\_65\_no\_health\_insurance inequality\_ratio air\_polution\_metric water\_quality pct\_voters pct\_rural\_population  
## 3 1 pct\_single\_parent days\_over\_90\_f median\_hh\_income median\_pediatric\_icu\_dist pct\_grandparents\_as\_guardians pct\_under\_65\_no\_health\_insurance inequality\_ratio air\_polution\_metric water\_quality pct\_voters pct\_rural\_population  
## 4 1 pct\_single\_parent days\_over\_90\_f median\_hh\_income median\_pediatric\_icu\_dist pct\_grandparents\_as\_guardians pct\_under\_65\_no\_health\_insurance inequality\_ratio air\_polution\_metric water\_quality pct\_voters pct\_rural\_population  
## 5 1 pct\_single\_parent days\_over\_90\_f median\_hh\_income median\_pediatric\_icu\_dist pct\_grandparents\_as\_guardians pct\_under\_65\_no\_health\_insurance inequality\_ratio air\_polution\_metric water\_quality pct\_voters pct\_rural\_population

# Imputing missing values in the test set using the trained model  
qol\_test\_imputed <- mice::mice(  
 qol\_test\_numeric,   
 m = 1,   
 method = "cart",  
 maxit = 5,  
 use.all = FALSE # Using only the training data for imputation  
 )

##   
## iter imp variable  
## 1 1 pct\_grandparents\_as\_guardians inequality\_ratio water\_quality  
## 2 1 pct\_grandparents\_as\_guardians inequality\_ratio water\_quality  
## 3 1 pct\_grandparents\_as\_guardians inequality\_ratio water\_quality  
## 4 1 pct\_grandparents\_as\_guardians inequality\_ratio water\_quality  
## 5 1 pct\_grandparents\_as\_guardians inequality\_ratio water\_quality

# Combining the imputed numeric columns back with the original data  
qol\_train <- cbind(  
 qol\_train[, !numeric\_cols],  
 mice::complete(qol\_train\_imputed)  
 )  
  
qol\_test <- cbind(  
 qol\_test[, !numeric\_cols],  
 mice::complete(qol\_test\_imputed)  
 )  
  
# confirm code was ran  
  
# remove numeric list to clean environment, no further use  
rm(numeric\_cols)  
rm(qol\_train\_imputed)  
rm(qol\_test\_imputed)  
rm(qol\_train\_numeric)  
rm(qol\_test\_numeric)

# Create dummy variables for 'region' in training data  
dummies\_train <- dummyVars(  
 ~ region,  
 data = qol\_train,   
 fullRank = FALSE  
 )  
  
qol\_train\_encoded <- predict(  
 dummies\_train,  
 newdata = qol\_train  
 )  
  
qol\_train <- cbind(  
 qol\_train,  
 qol\_train\_encoded) %>%  
 select(-region) # Remove the original 'region' column  
  
# Create dummy variables for 'region' in test data  
dummies\_test <- dummyVars(  
 ~ region,  
 data = qol\_test,  
 fullRank = FALSE  
 )  
  
qol\_test\_encoded <- predict(  
 dummies\_test,  
 newdata = qol\_test  
 )  
  
qol\_test <- cbind(  
 qol\_test,  
 qol\_test\_encoded) %>%  
 select(-region) # Remove the original 'region' column  
  
# remove objects for cleaner enviroment  
rm(dummies\_train)  
rm(qol\_train\_encoded)  
rm(dummies\_test)  
rm(qol\_test\_encoded)

#Encode response variable as 1(worse) and 0(better)   
  
#encode response variable in training set  
qol\_train <- qol\_train %>%  
 mutate(response = ifelse(response == "better", 0, 1))  
  
# Convert response variable to factor   
qol\_train$response <- as.factor(qol\_train$response)

# confirm response distribution is similiar between train and test  
table(qol\_train$response)

##   
## 0 1   
## 1337 1363

#encode response variable in testing set  
qol\_test <- qol\_test %>%  
 mutate(response = ifelse(response == "better", 0, 1))  
table(qol\_test$response)

##   
## 0 1   
## 240 200

# Convert response variable to factor   
qol\_test$response <- as.factor(qol\_test$response)

# Count missing values in the train set  
total\_missing <- sum(is.na(qol\_train))  
print(paste("Total missing values in dataframe:", total\_missing))

## [1] "Total missing values in dataframe: 0"

# Count missing values in the test set  
total\_missing <- sum(is.na(qol\_train))  
print(paste("Total missing values in dataframe:", total\_missing))

## [1] "Total missing values in dataframe: 0"

To determine if there are unwanted characters and whitespace and make corrections: We print a sample of the data for visual inspection. We use regular expressions to detect unwanted characters in text columns. We check for leading, trailing, or excessive whitespace in text columns.

# Visual inspection  
#print(head(qol\_train))  
  
# Detect unwanted characters using regular expressions  
unwanted\_chars <- function(x) {  
 if (is.character(x)) {  
 return(grepl("[^[:alnum:][:space:]]", x))  
 } else {  
 return(FALSE)  
 }  
}  
  
# Apply the function to each column qol\_train  
unwanted\_chars\_results <- sapply(qol\_train, function(col) sum(unwanted\_chars(col)))  
print("Number of unwanted characters in each column:")

## [1] "Number of unwanted characters in each column:"

#print(unwanted\_chars\_results)  
  
# Apply the function to each column qol\_test  
unwanted\_chars\_results <- sapply(qol\_train, function(col) sum(unwanted\_chars(col)))  
print("Number of unwanted characters in each column:")

## [1] "Number of unwanted characters in each column:"

#print(unwanted\_chars\_results)  
  
  
  
# Detect leading, trailing, or excessive whitespace  
whitespace\_check <- function(x) {  
 if (is.character(x)) {  
 return(grepl("^\\s+|\\s+$|\\s{2,}", x))  
 } else {  
 return(FALSE)  
 }  
}  
  
# Apply the function to each column of training and test sets  
whitespace\_results <- sapply(qol\_train, function(col) sum(whitespace\_check(col)))  
print("Number of whitespace issues in each column:")

## [1] "Number of whitespace issues in each column:"

#print(whitespace\_results)  
  
  
whitespace\_results <- sapply(qol\_test, function(col) sum(whitespace\_check(col)))  
print("Number of whitespace issues in each column:")

## [1] "Number of whitespace issues in each column:"

#print(whitespace\_results)

names(qol\_train)

## [1] "response" "weighted\_population"   
## [3] "average\_hh\_size" "pct\_male"   
## [5] "pct\_native\_american" "pct\_asian"   
## [7] "pct\_black" "pct\_hispanic"   
## [9] "pct\_other\_race" "pct\_white"   
## [11] "pct\_single\_parent" "pct\_hh\_other\_computer"   
## [13] "pct\_hh\_internet" "pct\_employed"   
## [15] "pct\_hh\_inc\_99999" "pct\_w\_medicare"   
## [17] "clinical\_nurse\_pt" "dentist\_pt"   
## [19] "pa\_pt" "mental\_health\_faciliy\_pt"   
## [21] "population\_density" "days\_over\_90\_f"   
## [23] "median\_hh\_income" "median\_er\_dist"   
## [25] "median\_pediatric\_icu\_dist" "median\_health\_clinic\_dist"   
## [27] "median\_drug\_alcohol\_care\_dist" "pct\_grandparents\_as\_guardians"   
## [29] "pct\_adult\_smokers" "pct\_obese\_adults"   
## [31] "pct\_no\_exercise" "pct\_binge\_drinkers"   
## [33] "pct\_under\_65\_no\_health\_insurance" "pct\_highschool\_diploma"   
## [35] "pct\_some\_college" "inequality\_ratio"   
## [37] "social\_clubs\_per\_10k" "air\_polution\_metric"   
## [39] "water\_quality" "pct\_high\_housing\_costs"   
## [41] "pct\_overcrowded\_hh" "pct\_30\_min\_plus\_commute"   
## [43] "pct\_food\_insecurities" "pct\_voters"   
## [45] "pct\_home\_owner" "pct\_65\_plus"   
## [47] "pct\_rural\_population" "region.Midwest"   
## [49] "region.Northeast" "region.South"   
## [51] "region.West"

# exclude response, classification variables  
exclude\_cols <- c(1, 48:51)   
  
# scale, exclude the specified columns  
training\_combined <- scale(qol\_train[, -exclude\_cols])  
  
# get mean and sd used for scaling  
train\_means <- attr(training\_combined, "scaled:center")  
train\_sds <- attr(training\_combined, "scaled:scale")  
  
# scale training set  
qol\_train[, -exclude\_cols] <- training\_combined  
  
# scale test set predictors using training set's statistics  
qol\_test[, -exclude\_cols] <- scale(qol\_test[, -exclude\_cols],  
 center = train\_means,  
 scale = train\_sds)  
  
# response variable is continuous, scale separately  
response\_mean <- mean(qol\_train[, 28], na.rm = TRUE)  
response\_sd <- sd(qol\_train[, 28], na.rm = TRUE)  
  
qol\_train[, 28] <- scale(qol\_train[, 28], center = response\_mean, scale = response\_sd)  
qol\_test[, 28] <- scale(qol\_test[, 28], center = response\_mean, scale = response\_sd)  
  
# Print the scaled data to verify  
print(head(qol\_train))

## response weighted\_population average\_hh\_size pct\_male pct\_native\_american  
## 1 0 0.1632088 0.09694753 -0.9969929 -0.18966904  
## 2 0 -0.3386387 -0.59169943 0.3761602 -0.15529624  
## 3 1 -0.0262330 0.13319211 -0.5850470 -0.20367278  
## 4 0 -0.3098685 -0.66418858 0.2267288 -0.06618156  
## 5 0 -0.2381854 -1.06287893 -0.4113835 0.42904145  
## 6 0 -0.3411104 -1.60654758 -0.6012017 0.17697421  
## pct\_asian pct\_black pct\_hispanic pct\_other\_race pct\_white  
## 1 2.4638560 0.1458029 -0.0764847 -0.1523387 -0.3835841  
## 2 -0.4573134 -0.5345774 -0.3129713 -0.2805579 0.7112262  
## 3 -0.1293886 -0.3298429 -0.5007273 -0.4190346 0.5484444  
## 4 -0.2085429 -0.3705141 0.2309479 -0.1882400 0.5713631  
## 5 -0.3480052 -0.5766272 -0.5014439 -0.4267278 0.5084835  
## 6 -0.1557734 -0.6248812 -0.3609852 0.1117929 0.4244482  
## pct\_single\_parent pct\_hh\_other\_computer pct\_hh\_internet pct\_employed  
## 1 -0.6419884 0.4120815 1.33196432 0.4668344  
## 2 -0.8896633 0.4955363 -0.40852159 0.7957516  
## 3 -0.1494008 -0.1065308 0.63552343 -0.4893203  
## 4 -0.9752907 -0.2317130 1.21116572 0.1685141  
## 5 0.2686081 -0.3926617 -0.07694177 -0.7111482  
## 6 -0.1871505 1.0379928 0.53074914 -0.1565785  
## pct\_hh\_inc\_99999 pct\_w\_medicare clinical\_nurse\_pt dentist\_pt pa\_pt  
## 1 -0.75123922 -0.9510003 0.5809795 5.28524712 1.32155459  
## 2 0.37387337 -0.4645260 -0.4389455 -1.35044215 0.21069424  
## 3 -0.08810803 -0.4600629 -0.4389455 0.02954633 -0.35897774  
## 4 0.55291879 -0.1342590 -0.4389455 0.55805255 0.43856303  
## 5 -0.27599520 -0.4377476 1.6009044 0.05890778 0.09675984  
## 6 -0.55229986 1.4367408 -0.4389455 -1.35044215 1.97667736  
## mental\_health\_faciliy\_pt population\_density days\_over\_90\_f median\_hh\_income  
## 1 -0.3510728 0.05081419 0.4405421 1.781826234  
## 2 3.9113744 -0.14410701 0.3617575 -0.000006471  
## 3 -0.5429362 -0.03338839 -0.3210424 -0.140481744  
## 4 0.4388834 -0.14052131 -0.5836577 0.370688267  
## 5 1.5888796 -0.13936704 -1.4502883 -0.614392829  
## 6 11.3253560 -0.14506890 -1.2139345 0.403456475  
## median\_er\_dist median\_pediatric\_icu\_dist median\_health\_clinic\_dist  
## 1 -0.2765695 -0.94875858 -0.3269992  
## 2 0.8317848 0.52786585 -0.4504813  
## 3 -0.3806053 -0.75203002 -0.4526476  
## 4 -0.4921207 0.07562222 -0.4407327  
## 5 0.4292414 0.74032238 -0.3020862  
## 6 1.6572708 0.26564705 0.7475114  
## median\_drug\_alcohol\_care\_dist pct\_grandparents\_as\_guardians pct\_adult\_smokers  
## 1 -0.85749028 -1.06760158 -1.7977831  
## 2 -0.17898886 1.85971670 -0.5627026  
## 3 -0.16025723 0.03218664 0.1783456  
## 4 -0.98611417 -1.20977585 -0.3897914  
## 5 -0.05161375 -0.12654128 0.4994665  
## 6 1.64921865 0.45347637 -0.3403882  
## pct\_obese\_adults pct\_no\_exercise pct\_binge\_drinkers  
## 1 -1.95767511 -1.8135757 -0.5310763  
## 2 0.03068185 -0.6494795 0.2100544  
## 3 -0.14221875 0.2429943 -0.7707792  
## 4 0.09551958 -0.5912747 1.4085832  
## 5 -0.29350678 -0.4166602 0.5472435  
## 6 -1.15800981 -0.9987084 0.4050914  
## pct\_under\_65\_no\_health\_insurance pct\_highschool\_diploma pct\_some\_college  
## 1 -0.6079747 0.9737627 2.23615773  
## 2 -0.3923713 0.9374262 0.33495155  
## 3 -0.9531477 0.2879585 1.01200390  
## 4 0.1764046 0.4348255 0.40250098  
## 5 0.2728549 0.5455355 0.01154107  
## 6 -0.4033967 0.9275999 0.23030403  
## inequality\_ratio social\_clubs\_per\_10k air\_polution\_metric water\_quality  
## 1 1.1864361 -0.2970930 0.5711042 -0.7099236  
## 2 -1.2384855 -1.9229575 0.1054247 1.4080805  
## 3 0.5630010 -0.4743653 0.5128943 -0.7099236  
## 4 0.3282529 1.2686924 -1.9901334 -0.7099236  
## 5 1.1318035 -0.9467526 -1.7572936 -0.7099236  
## 6 0.9448580 0.6697934 -0.2438350 -0.7099236  
## pct\_high\_housing\_costs pct\_overcrowded\_hh pct\_30\_min\_plus\_commute  
## 1 1.2569517 -0.137351167 0.1248455  
## 2 -0.5098248 -0.441055868 -0.2521081  
## 3 0.9053099 -0.416414299 0.3054692  
## 4 -0.8782268 0.002631423 -2.1525829  
## 5 0.3864986 -0.600019778 -0.5976490  
## 6 0.5038439 -0.305405318 -1.1866391  
## pct\_food\_insecurities pct\_voters pct\_home\_owner pct\_65\_plus  
## 1 -0.6064346 1.1984845 -1.0496540 -0.9045949  
## 2 -0.8485223 0.8603756 0.5768380 0.8783136  
## 3 0.2812202 -0.4469179 -1.4005657 -1.2003605  
## 4 -0.2836510 0.1045918 0.2764742 0.1167710  
## 5 0.6846997 0.4857614 0.6138870 1.1536579  
## 6 -0.2836510 1.0221731 0.2620917 1.8275098  
## pct\_rural\_population region.Midwest region.Northeast region.South region.West  
## 1 -0.9654379 0 0 1 0  
## 2 1.3135982 1 0 0 0  
## 3 -0.6470157 0 0 1 0  
## 4 -1.0794770 1 0 0 0  
## 5 1.0705688 0 1 0 0  
## 6 1.3135982 0 0 0 1

print(head(qol\_test))

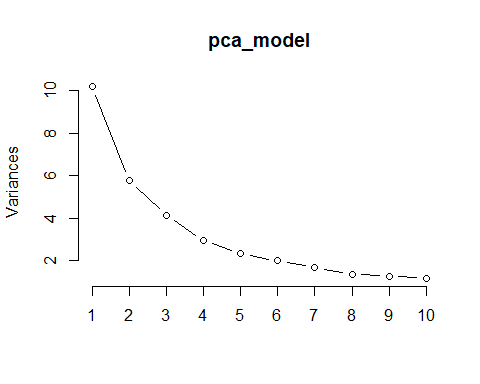
## response weighted\_population average\_hh\_size pct\_male pct\_native\_american  
## 1 0 -0.34036334 0.1331921 2.90033872 -0.2291341  
## 2 1 -0.05525811 0.4956379 -0.22156529 -0.2202226  
## 3 1 -0.24721347 0.9668174 -0.47196380 -0.1883960  
## 4 1 -0.26678464 0.4956379 -0.43157694 -0.2354995  
## 5 0 0.13560453 1.7641981 -0.09232735 -0.1591154  
## 6 1 -0.31407875 1.7641981 0.23884488 -0.2227688  
## pct\_asian pct\_black pct\_hispanic pct\_other\_race pct\_white  
## 1 -0.15577337 -0.34087235 0.61505939 -0.5421251 0.6653887  
## 2 -0.06531135 -0.02584317 -0.10156661 -0.3010729 0.2269945  
## 3 -0.19346588 -0.46081782 0.74691857 -0.2087551 0.5466814  
## 4 -0.31031266 0.48909513 -0.04710303 -0.2805579 -0.1003320  
## 5 -0.31408191 -0.52561595 0.23238111 -0.2010620 0.5931066  
## 6 -0.27638940 -0.56421909 3.89290669 0.7298095 0.1235664  
## pct\_single\_parent pct\_hh\_other\_computer pct\_hh\_internet pct\_employed  
## 1 -1.88312501 -0.25555728 0.7181102 1.8398725  
## 2 0.05776216 -0.63706515 0.8956102 0.2985512  
## 3 0.70595218 -0.32709001 0.3852978 -0.2254216  
## 4 0.26308372 -0.18998562 0.1782145 0.9066655  
## 5 -1.30859294 0.23921073 1.2000720 0.1914618  
## 6 -0.52229425 0.01269044 -0.4233133 1.4574106  
## pct\_hh\_inc\_99999 pct\_w\_medicare clinical\_nurse\_pt dentist\_pt pa\_pt  
## 1 1.6072974 -0.6430487 -0.4389455 -1.3504422 -0.9001661  
## 2 -0.4660928 0.2093972 -0.4389455 -0.4402370 -0.8431989  
## 3 0.5882858 0.5619795 -0.4389455 -0.4402370 -0.8147153  
## 4 -0.3423083 1.0172124 -0.4389455 -0.5870443 -0.6438137  
## 5 -0.6385069 -0.1655005 -0.0989705 -0.3227912 -0.4159449  
## 6 0.5241831 -0.7367731 -0.4389455 -1.0274661 -0.3020105  
## mental\_health\_faciliy\_pt population\_density days\_over\_90\_f median\_hh\_income  
## 1 -0.6684765 -0.14515440 0.4405421 0.2578590  
## 2 -0.5251711 -0.01178332 1.6485726 0.2796109  
## 3 -0.2646158 -0.12975351 1.6223111 0.1481872  
## 4 -0.1580250 -0.13063524 1.3859573 -0.3790511  
## 5 -0.3487041 -0.06404055 1.4910034 2.1947618  
## 6 -0.6684765 -0.14006174 0.3880190 0.3263424  
## median\_er\_dist median\_pediatric\_icu\_dist median\_health\_clinic\_dist  
## 1 1.35672321 -0.3191241 2.82937579  
## 2 -0.03653943 -0.6187290 -0.23709560  
## 3 -0.29424882 0.1264158 -0.22084796  
## 4 0.03417766 -0.2273346 0.08352451  
## 5 -0.07733775 -0.4557769 -0.05403885  
## 6 0.08041576 0.7106713 -0.29017123  
## median\_drug\_alcohol\_care\_dist pct\_grandparents\_as\_guardians pct\_adult\_smokers  
## 1 0.156931783 -0.6026266 -0.63680747  
## 2 -0.444145551 0.2557236 0.05483757  
## 3 -0.210208251 -0.2772674 0.10424079  
## 4 0.006246191 0.1841983 0.40066009  
## 5 -0.645198427 -0.3388767 -0.56270264  
## 6 1.475638843 -0.7954520 -0.53800103  
## pct\_obese\_adults pct\_no\_exercise pct\_binge\_drinkers  
## 1 -0.46640739 -0.61067627 0.85126201  
## 2 -0.03415587 -0.12563617 0.70058102  
## 3 0.07390700 0.08778148 0.54054762  
## 4 0.33325791 0.08778148 0.56863720  
## 5 -0.74737087 -0.82409392 0.93299275  
## 6 0.72228427 1.11606651 0.06754682  
## pct\_under\_65\_no\_health\_insurance pct\_highschool\_diploma pct\_some\_college  
## 1 1.0316168 -0.09022319 -0.14913178  
## 2 0.8685416 0.16801377 -0.15796112  
## 3 1.7664747 -0.18709447 -0.67750764  
## 4 1.6499761 -0.89695981 0.03295319  
## 5 0.9262391 0.28800007 0.50339096  
## 6 2.9929552 -2.48753182 -0.86071440  
## inequality\_ratio social\_clubs\_per\_10k air\_polution\_metric water\_quality  
## 1 0.280498276 0.47548889 -1.4662439 -0.7099236  
## 2 0.007227404 -0.27533634 0.3964744 1.4080805  
## 3 0.236829500 0.17282000 0.3964744 1.4080805  
## 4 1.095691261 -0.08260047 1.1532037 1.4080805  
## 5 -0.307702425 -0.47173221 0.9203639 1.4080805  
## 6 -1.189543730 0.22813808 -0.9423543 1.4080805  
## pct\_high\_housing\_costs pct\_overcrowded\_hh pct\_30\_min\_plus\_commute  
## 1 -1.16546213 -0.1629153 1.22429377  
## 2 -0.85895590 0.1141291 0.05416673  
## 3 1.21469891 0.5243050 0.56462483  
## 4 -0.60941719 0.3783456 0.39970760  
## 5 -0.06613839 0.1149688 1.66407306  
## 6 -0.95278692 2.8890481 -0.95889627  
## pct\_food\_insecurities pct\_voters pct\_home\_owner pct\_65\_plus  
## 1 -0.06846200 1.06563077 0.48075941 -0.71182300  
## 2 1.22267228 -0.70457496 0.46005860 -0.83532652  
## 3 0.92678734 -0.63210416 0.09261919 0.02674746  
## 4 0.95368597 0.07669215 0.73091053 0.04355223  
## 5 -0.09536063 0.99886461 1.10116948 -0.90057469  
## 6 -0.49884009 -1.69423539 -0.54320174 -1.05076464  
## pct\_rural\_population region.Midwest region.Northeast region.South region.West  
## 1 1.31359817 0 0 1 0  
## 2 -0.75372105 0 0 1 0  
## 3 -0.27398220 0 0 1 0  
## 4 0.44360937 0 0 1 0  
## 5 -0.08681535 0 0 1 0  
## 6 0.03854028 0 0 1 0

# PCA

qol\_train\_standardized <- qol\_train %>%  
 select(-response) %>%  
 scale() # standardize the data (exclude the response variable)  
  
pca\_model <- prcomp(  
 qol\_train\_standardized,  
 center = TRUE,  
 scale. = TRUE) # apply PCA  
  
summary(pca\_model) # print summary of PCA model

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 3.1947 2.4016 2.0323 1.71294 1.52746 1.40723 1.29562  
## Proportion of Variance 0.2041 0.1153 0.0826 0.05868 0.04666 0.03961 0.03357  
## Cumulative Proportion 0.2041 0.3195 0.4021 0.46076 0.50743 0.54703 0.58060  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Standard deviation 1.16064 1.11792 1.07509 1.02978 1.0099 0.96843 0.95772  
## Proportion of Variance 0.02694 0.02499 0.02312 0.02121 0.0204 0.01876 0.01834  
## Cumulative Proportion 0.60755 0.63254 0.65566 0.67687 0.6973 0.71602 0.73436  
## PC15 PC16 PC17 PC18 PC19 PC20 PC21  
## Standard deviation 0.9407 0.90876 0.87542 0.84717 0.84083 0.80144 0.79731  
## Proportion of Variance 0.0177 0.01652 0.01533 0.01435 0.01414 0.01285 0.01271  
## Cumulative Proportion 0.7521 0.76858 0.78391 0.79826 0.81240 0.82525 0.83796  
## PC22 PC23 PC24 PC25 PC26 PC27 PC28  
## Standard deviation 0.75954 0.73981 0.70434 0.68262 0.6707 0.65902 0.64741  
## Proportion of Variance 0.01154 0.01095 0.00992 0.00932 0.0090 0.00869 0.00838  
## Cumulative Proportion 0.84950 0.86045 0.87037 0.87969 0.8887 0.89737 0.90576  
## PC29 PC30 PC31 PC32 PC33 PC34 PC35  
## Standard deviation 0.64153 0.61126 0.59094 0.58528 0.57621 0.54136 0.53284  
## Proportion of Variance 0.00823 0.00747 0.00698 0.00685 0.00664 0.00586 0.00568  
## Cumulative Proportion 0.91399 0.92146 0.92844 0.93529 0.94193 0.94780 0.95347  
## PC36 PC37 PC38 PC39 PC40 PC41 PC42  
## Standard deviation 0.5197 0.50783 0.49227 0.48242 0.46031 0.4471 0.43872  
## Proportion of Variance 0.0054 0.00516 0.00485 0.00465 0.00424 0.0040 0.00385  
## Cumulative Proportion 0.9589 0.96403 0.96888 0.97353 0.97777 0.9818 0.98562  
## PC43 PC44 PC45 PC46 PC47 PC48 PC49  
## Standard deviation 0.39140 0.38146 0.37603 0.35129 0.31117 0.22583 0.08732  
## Proportion of Variance 0.00306 0.00291 0.00283 0.00247 0.00194 0.00102 0.00015  
## Cumulative Proportion 0.98868 0.99159 0.99442 0.99689 0.99883 0.99985 1.00000  
## PC50  
## Standard deviation 1.738e-15  
## Proportion of Variance 0.000e+00  
## Cumulative Proportion 1.000e+00

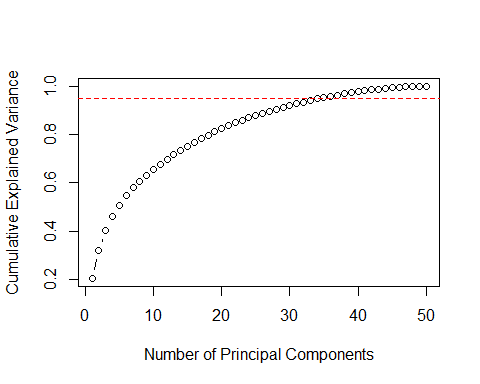
plot(pca\_model,  
 type = "l") # plot explained variance



pca\_components <- predict(  
 pca\_model,  
 qol\_train\_standardized  
 ) # extract the principal components  
  
head(pca\_components)

## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## 1 -3.5542084 5.3028381 -1.4664131 -1.16469742 -0.5028603 0.4263102 1.4985923  
## 2 -1.6203446 -2.3267345 0.9718123 -0.02227473 -0.4641936 0.6769108 0.1197962  
## 3 0.3213530 1.3440502 -1.6538349 -0.18938282 0.2081043 0.5117809 -0.5577245  
## 4 -2.8954401 -0.7211789 0.2615312 -0.60272801 2.3250150 -1.5582193 0.2006907  
## 5 -0.8293505 -1.2671521 0.6996353 -3.21634149 -1.1542298 -0.5726351 -1.8650291  
## 6 -2.0669251 -2.1481946 3.7793444 -4.77291083 -2.8060991 -1.2987680 1.7910707  
## PC8 PC9 PC10 PC11 PC12 PC13  
## 1 -1.27689675 0.5842305 -0.5574088 -0.39829058 -2.7032432 0.8317901  
## 2 0.03660643 1.2459835 0.3753102 -0.74378625 1.4122173 -0.6339413  
## 3 -0.42825330 0.9568689 0.3499697 -0.01419796 -1.2039756 1.1635286  
## 4 -0.49329382 -0.7526181 0.3348718 0.34497803 -0.4105079 0.3780644  
## 5 1.20401082 0.4352828 -1.1913110 0.96570808 -0.5630903 -0.9447676  
## 6 0.12650349 3.0117755 1.6816338 -1.13444144 0.3375801 -1.8451375  
## PC14 PC15 PC16 PC17 PC18 PC19  
## 1 0.7503587 0.43194291 -0.0722205 -0.04218681 -1.5614465 -1.7413619  
## 2 -1.8575569 0.83876440 -1.1490238 -2.48756209 0.3386236 -0.9424160  
## 3 0.2980153 0.61300908 -0.4896908 0.99660675 0.5299998 0.6166035  
## 4 0.4353880 -0.03184139 -0.3331133 -0.38569762 0.6372862 -0.4184895  
## 5 -1.0654047 -1.97495458 -1.3029705 0.36540463 -0.2692003 0.4381684  
## 6 -4.7015592 0.95377211 -4.5348413 -3.26095463 3.2056798 -4.1045753  
## PC20 PC21 PC22 PC23 PC24 PC25  
## 1 0.27947362 -0.3873300 0.22690098 0.2085404 -2.3479199 0.4538065  
## 2 -1.03377804 1.4649121 -0.03747908 1.5876224 0.9077865 -0.6384534  
## 3 0.04154892 0.2604089 0.41788012 0.5764926 0.1337463 0.6043390  
## 4 0.92956421 0.4672772 -0.14584694 0.2099929 -0.1044012 -0.4814866  
## 5 0.07573086 1.7915131 -0.32556479 0.4272328 0.0798187 -0.6673821  
## 6 0.87553050 1.3038569 -0.32161527 -0.1684187 1.1890272 1.8402874  
## PC26 PC27 PC28 PC29 PC30 PC31  
## 1 -0.6578633 -1.1566904 -0.3432983 -0.7857331 0.4338726 -0.006149088  
## 2 -1.0382067 -0.5711852 -0.1147816 1.1960490 -1.8381874 -0.748911186  
## 3 -0.7439811 0.4256431 -0.1417125 -0.4477650 0.1224037 0.292264560  
## 4 0.8301202 0.3099234 -0.9206465 0.5228020 0.3688398 0.796641475  
## 5 -0.2885516 -0.3843727 -1.2179200 -0.4664415 0.2333683 0.078457048  
## 6 -0.9781080 0.9046557 0.6192406 1.0558032 -0.4473620 -0.554108283  
## PC32 PC33 PC34 PC35 PC36 PC37  
## 1 -1.3032281 0.323341758 0.2201082 -0.50906682 0.04196137 0.39364032  
## 2 -0.6097664 0.084384272 -0.3666394 -0.24519135 0.27101762 -0.78480403  
## 3 0.3915439 -0.112005584 -0.6054932 -0.42457234 0.07078715 0.19800904  
## 4 0.1295881 -0.340045542 0.9227244 -0.66529784 -0.02687637 0.22607968  
## 5 -0.7188668 0.303022856 0.7723867 0.00666464 0.26846096 -0.01263467  
## 6 0.3849461 0.002135051 0.6974959 0.19503480 1.05463166 -0.19434243  
## PC38 PC39 PC40 PC41 PC42 PC43  
## 1 0.026662226 -0.20475202 0.04157369 0.2407893 0.35813824 -0.34102718  
## 2 -0.233051769 0.18324018 0.25237938 0.7958147 -0.15336306 0.45285948  
## 3 0.398685451 -0.11485008 -0.20730170 0.3801761 -0.10486197 -0.15673689  
## 4 -0.386222136 0.44522386 0.27785879 -1.0904725 0.32751004 0.18235394  
## 5 -0.486862768 -0.05720323 -0.23780023 -0.5141368 0.03373246 -0.06443455  
## 6 -0.001818079 0.19909056 -0.45604659 0.5923943 0.37097823 0.23409645  
## PC44 PC45 PC46 PC47 PC48 PC49  
## 1 0.37064352 0.008140640 0.23556070 0.231344183 -0.34856124 -0.070787858  
## 2 -0.06521299 0.022279707 0.18252382 0.373276470 0.02201107 -0.015451043  
## 3 0.03145367 0.314524643 0.13795661 -0.006466904 0.12069951 -0.018381136  
## 4 0.07498862 -0.452692526 -0.01116502 -0.181493389 0.01776896 -0.074468547  
## 5 -0.40869985 0.008277048 0.19506877 0.094140600 -0.28366275 -0.015831183  
## 6 -0.27418610 -0.486437164 -0.10910387 -0.130983065 -0.15005308 -0.004370987  
## PC50  
## 1 1.637579e-15  
## 2 1.859624e-15  
## 3 -2.775558e-17  
## 4 -2.775558e-17  
## 5 -1.137979e-15  
## 6 -2.442491e-15

# Calculate cumulative explained variance  
explained\_variance <- summary(pca\_model)$importance[2, ]  
cumulative\_explained\_variance <- cumsum(explained\_variance)  
  
# Plot cumulative explained variance  
plot(cumulative\_explained\_variance, type = "b", xlab = "Number of Principal Components", ylab = "Cumulative Explained Variance")  
abline(h = 0.95, col = "red", lty = 2) # line for 95% explained variance



# Find the number of components needed to explain 95% of the variance  
num\_components\_95 <- which(cumulative\_explained\_variance >= 0.95)[1]  
num\_components\_95

## PC35   
## 35

# 35 features predict 95% of response variability

# extract the features (rotation matrix)  
features <- pca\_model$rotation  
  
# subset the loadings for the first 35 principal components  
top\_features <- features[, 1:35]  
  
# find the most important variables for each principal component  
important\_vars <- apply(top\_features, 2, function(x) names(x[order(abs(x), decreasing = TRUE)]))  
  
# convert to a data frame  
important\_vars\_df <- as.data.frame(important\_vars)  
head(important\_vars\_df)

## PC1 PC2 PC3  
## 1 pct\_no\_exercise pct\_rural\_population median\_pediatric\_icu\_dist  
## 2 pct\_food\_insecurities pct\_home\_owner median\_drug\_alcohol\_care\_dist  
## 3 pct\_highschool\_diploma pct\_high\_housing\_costs pct\_overcrowded\_hh  
## 4 pct\_some\_college pct\_asian region.West  
## 5 pct\_obese\_adults weighted\_population median\_er\_dist  
## 6 pct\_adult\_smokers pct\_65\_plus pct\_native\_american  
## PC4 PC5 PC6  
## 1 average\_hh\_size region.Midwest pct\_hispanic  
## 2 pct\_hispanic pct\_30\_min\_plus\_commute median\_health\_clinic\_dist  
## 3 days\_over\_90\_f pct\_w\_medicare pct\_other\_race  
## 4 pct\_single\_parent pct\_65\_plus pct\_30\_min\_plus\_commute  
## 5 pa\_pt pct\_voters pct\_native\_american  
## 6 inequality\_ratio pct\_obese\_adults median\_er\_dist  
## PC7 PC8 PC9  
## 1 days\_over\_90\_f population\_density mental\_health\_faciliy\_pt  
## 2 pct\_black region.Northeast pct\_male  
## 3 social\_clubs\_per\_10k pct\_male pct\_native\_american  
## 4 median\_drug\_alcohol\_care\_dist median\_er\_dist population\_density  
## 5 median\_er\_dist region.West region.Midwest  
## 6 region.Northeast pct\_asian pct\_overcrowded\_hh  
## PC10 PC11 PC12  
## 1 water\_quality pct\_hh\_other\_computer water\_quality  
## 2 region.Northeast clinical\_nurse\_pt pa\_pt  
## 3 region.West region.Northeast region.South  
## 4 pa\_pt pct\_black pct\_single\_parent  
## 5 region.Midwest pct\_w\_medicare pct\_hh\_other\_computer  
## 6 region.South weighted\_population population\_density  
## PC13 PC14 PC15  
## 1 pct\_hh\_inc\_99999 pct\_hh\_other\_computer clinical\_nurse\_pt  
## 2 pct\_hh\_other\_computer pct\_male pa\_pt  
## 3 inequality\_ratio mental\_health\_faciliy\_pt pct\_hh\_inc\_99999  
## 4 clinical\_nurse\_pt clinical\_nurse\_pt median\_health\_clinic\_dist  
## 5 region.Northeast inequality\_ratio dentist\_pt  
## 6 pa\_pt dentist\_pt region.Northeast  
## PC16 PC17  
## 1 pct\_hh\_other\_computer mental\_health\_faciliy\_pt  
## 2 mental\_health\_faciliy\_pt water\_quality  
## 3 pct\_hh\_inc\_99999 pct\_male  
## 4 pct\_male median\_health\_clinic\_dist  
## 5 pct\_w\_medicare population\_density  
## 6 median\_health\_clinic\_dist pct\_native\_american  
## PC18 PC19 PC20  
## 1 median\_health\_clinic\_dist mental\_health\_faciliy\_pt social\_clubs\_per\_10k  
## 2 population\_density dentist\_pt pct\_asian  
## 3 pa\_pt median\_health\_clinic\_dist pa\_pt  
## 4 mental\_health\_faciliy\_pt pa\_pt population\_density  
## 5 median\_drug\_alcohol\_care\_dist population\_density water\_quality  
## 6 pct\_asian pct\_hh\_inc\_99999 weighted\_population  
## PC21 PC22  
## 1 social\_clubs\_per\_10k pct\_employed  
## 2 pct\_binge\_drinkers population\_density  
## 3 median\_health\_clinic\_dist weighted\_population  
## 4 pct\_other\_race pct\_hh\_internet  
## 5 water\_quality pct\_w\_medicare  
## 6 days\_over\_90\_f pct\_grandparents\_as\_guardians  
## PC23 PC24  
## 1 pct\_grandparents\_as\_guardians weighted\_population  
## 2 air\_polution\_metric dentist\_pt  
## 3 median\_pediatric\_icu\_dist pa\_pt  
## 4 region.West pct\_grandparents\_as\_guardians  
## 5 pa\_pt pct\_w\_medicare  
## 6 social\_clubs\_per\_10k population\_density  
## PC25 PC26  
## 1 pct\_high\_housing\_costs pct\_other\_race  
## 2 median\_pediatric\_icu\_dist air\_polution\_metric  
## 3 social\_clubs\_per\_10k inequality\_ratio  
## 4 pct\_30\_min\_plus\_commute pct\_hh\_inc\_99999  
## 5 pct\_hh\_inc\_99999 pct\_highschool\_diploma  
## 6 region.Northeast pct\_under\_65\_no\_health\_insurance  
## PC27 PC28  
## 1 pct\_w\_medicare air\_polution\_metric  
## 2 pct\_employed pct\_employed  
## 3 dentist\_pt pct\_grandparents\_as\_guardians  
## 4 pct\_grandparents\_as\_guardians region.West  
## 5 pct\_highschool\_diploma pct\_other\_race  
## 6 median\_pediatric\_icu\_dist pct\_single\_parent  
## PC29 PC30  
## 1 pct\_asian pct\_single\_parent  
## 2 dentist\_pt pct\_grandparents\_as\_guardians  
## 3 weighted\_population pct\_high\_housing\_costs  
## 4 median\_drug\_alcohol\_care\_dist region.South  
## 5 pa\_pt pct\_native\_american  
## 6 pct\_overcrowded\_hh region.Midwest  
## PC31 PC32  
## 1 pct\_employed pct\_binge\_drinkers  
## 2 pct\_under\_65\_no\_health\_insurance pct\_adult\_smokers  
## 3 pct\_single\_parent pct\_other\_race  
## 4 pct\_binge\_drinkers pct\_w\_medicare  
## 5 pct\_hh\_inc\_99999 pct\_65\_plus  
## 6 pct\_high\_housing\_costs pct\_hh\_internet  
## PC33 PC34  
## 1 median\_pediatric\_icu\_dist pct\_overcrowded\_hh  
## 2 inequality\_ratio inequality\_ratio  
## 3 pct\_hh\_inc\_99999 pct\_under\_65\_no\_health\_insurance  
## 4 pct\_home\_owner pct\_hh\_internet  
## 5 pct\_under\_65\_no\_health\_insurance pct\_some\_college  
## 6 pct\_high\_housing\_costs median\_drug\_alcohol\_care\_dist  
## PC35  
## 1 median\_er\_dist  
## 2 pct\_voters  
## 3 median\_drug\_alcohol\_care\_dist  
## 4 pct\_binge\_drinkers  
## 5 pct\_home\_owner  
## 6 pct\_overcrowded\_hh

# most important variables  
important\_vars\_list <- unique(unlist(important\_vars\_df))  
  
important\_vars\_list

## [1] "pct\_no\_exercise" "pct\_food\_insecurities"   
## [3] "pct\_highschool\_diploma" "pct\_some\_college"   
## [5] "pct\_obese\_adults" "pct\_adult\_smokers"   
## [7] "pct\_hh\_internet" "pct\_voters"   
## [9] "median\_hh\_income" "pct\_grandparents\_as\_guardians"   
## [11] "pct\_binge\_drinkers" "region.South"   
## [13] "pct\_single\_parent" "pct\_employed"   
## [15] "days\_over\_90\_f" "inequality\_ratio"   
## [17] "pct\_white" "pct\_black"   
## [19] "pct\_under\_65\_no\_health\_insurance" "pct\_hh\_inc\_99999"   
## [21] "region.Midwest" "air\_polution\_metric"   
## [23] "pct\_overcrowded\_hh" "average\_hh\_size"   
## [25] "dentist\_pt" "pa\_pt"   
## [27] "pct\_w\_medicare" "region.Northeast"   
## [29] "pct\_asian" "region.West"   
## [31] "pct\_home\_owner" "pct\_native\_american"   
## [33] "social\_clubs\_per\_10k" "pct\_hispanic"   
## [35] "pct\_rural\_population" "pct\_65\_plus"   
## [37] "pct\_30\_min\_plus\_commute" "pct\_high\_housing\_costs"   
## [39] "weighted\_population" "pct\_other\_race"   
## [41] "median\_pediatric\_icu\_dist" "median\_drug\_alcohol\_care\_dist"   
## [43] "pct\_hh\_other\_computer" "clinical\_nurse\_pt"   
## [45] "median\_health\_clinic\_dist" "population\_density"   
## [47] "median\_er\_dist" "pct\_male"   
## [49] "water\_quality" "mental\_health\_faciliy\_pt"

# Prepare data for cross-validation  
#train\_control <- trainControl(method = "cv", number = 10)  
#tune\_grid <- expand.grid(.ncomp = 1:ncol(pca\_components))  
  
# Train a model using cross-validation to find the optimal number of components  
#cv\_model <- train(  
# response ~ .,  
# data = cbind(qol\_train$response, pca\_components),  
# method = "lm",  
# trControl = train\_control,  
# tuneGrid = tune\_grid  
#)  
  
# Get the optimal number of components  
#optimal\_components <- cv\_model$bestTune$.ncomp  
#optimal\_components

*Subset regression viability check*

Build a simple OLS model

# --- OLS Model ---  
  
# Fit the OLS model  
ols\_model <- lm(response ~ ., data = qol\_train)

## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor  
## response will be ignored

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

# Predict on the test set  
predictions\_ols <- predict(ols\_model, newdata = qol\_test)  
  
# Evaluate the OLS model  
confusion\_matrix\_ols <- table(qol\_test$response, predictions\_ols > 0.5)  
accuracy\_ols <- sum(diag(confusion\_matrix\_ols)) / sum(confusion\_matrix\_ols)  
print(paste("Accuracy (OLS):", accuracy\_ols))

## [1] "Accuracy (OLS): 0.545454545454545"

# Calculate feature importance for OLS  
feature\_importance\_ols <- abs(summary(ols\_model)$coefficients[, "Estimate"])

## Warning in Ops.factor(r, 2): '^' not meaningful for factors

importance\_df\_ols <- data.frame(Feature = names(feature\_importance\_ols), Importance = feature\_importance\_ols)  
importance\_df\_ols <- importance\_df\_ols[order(-importance\_df\_ols$Importance), ]  
  
# --- Random Forest Model ---  
  
# Train a random forest model  
rf\_model <- randomForest::randomForest(as.factor(response) ~ ., data = qol\_train, ntree = 500, na.action = na.omit)  
  
# Predict on the test set  
predictions\_rf <- predict(rf\_model, qol\_test, type = "prob")[, 2]  
  
# Evaluate the Random Forest model  
confusion\_matrix\_rf <- table(qol\_test$response, predictions\_rf > 0.5)  
accuracy\_rf <- sum(diag(confusion\_matrix\_rf)) / sum(confusion\_matrix\_rf)  
print(paste("Accuracy (RF):", accuracy\_rf))

## [1] "Accuracy (RF): 0.997727272727273"

# --- ROC Curve (Both Models) ---  
  
# Function to plot ROC curves  
plot\_roc <- function(predictions, model\_name) {  
 roc\_obj <- roc(qol\_test$response, predictions)  
 auc\_value <- auc(roc\_obj)  
   
 roc\_df <- data.frame(  
 tpr = roc\_obj$sensitivities,  
 fpr = 1 - roc\_obj$specificities  
 )  
   
 ggplot(roc\_df, aes(x = fpr, y = tpr)) +  
 geom\_line() +  
 geom\_abline(linetype = "dashed") +  
 xlab("False Positive Rate") +  
 ylab("True Positive Rate") +  
 ggtitle(paste("ROC Curve (", model\_name, ", AUC =", round(auc\_value, 2), ")"))  
}

library(pROC)

## Warning: package 'pROC' was built under R version 4.3.2

## Type 'citation("pROC")' for a citation.

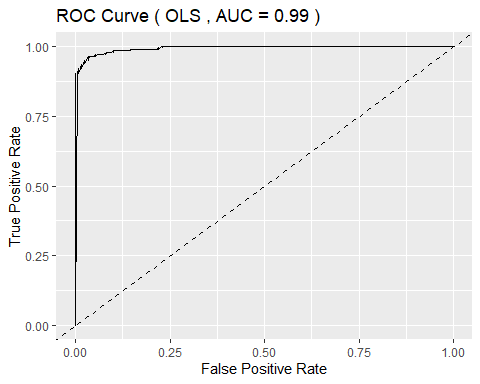
##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

# Plot ROC curves for both models  
plot\_roc(predictions\_ols, "OLS")

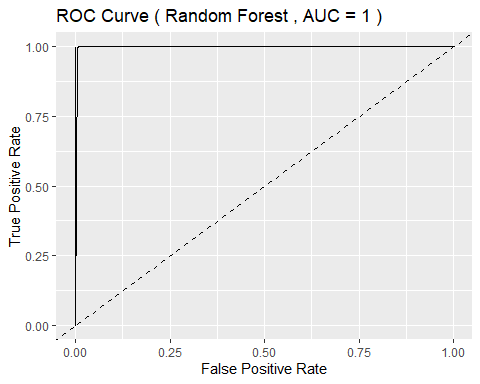
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

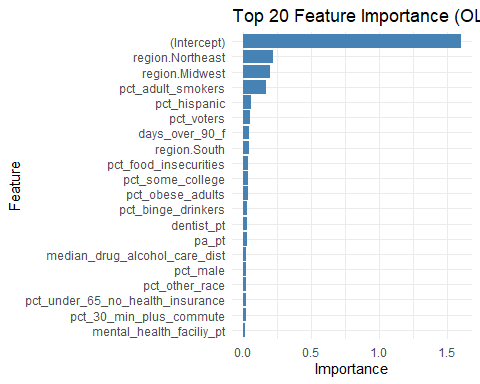


plot\_roc(predictions\_rf, "Random Forest")

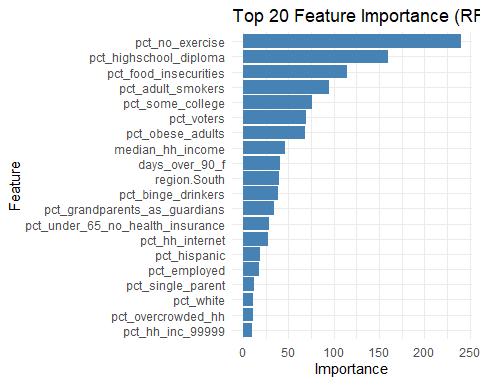
## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases



# --- Feature Importance Visualization (OLS) ---  
  
# Select top 20 features  
top\_20\_features\_ols <- head(importance\_df\_ols, 20)  
  
# Plot top 20 feature importance for OLS  
ggplot(top\_20\_features\_ols, aes(x = reorder(Feature, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 labs(title = "Top 20 Feature Importance (OLS)", x = "Feature", y = "Importance") +  
 theme\_minimal()



# --- Feature Importance Visualization (RF) ---  
  
# Calculate feature importance for RF  
feature\_importance\_rf <- randomForest::importance(rf\_model)  
importance\_df\_rf <- data.frame(Feature = rownames(feature\_importance\_rf), Importance = feature\_importance\_rf[, 1])  
importance\_df\_rf <- importance\_df\_rf[order(-importance\_df\_rf$Importance), ]  
  
# Select top 20 features  
top\_20\_features\_rf <- head(importance\_df\_rf, 20)  
  
# Plot top 20 feature importance for RF  
ggplot(top\_20\_features\_rf, aes(x = reorder(Feature, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 labs(title = "Top 20 Feature Importance (RF)", x = "Feature", y = "Importance") +  
 theme\_minimal()



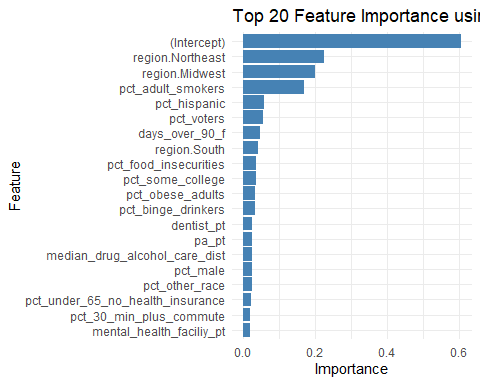
build initial model (OLS)

# Convert response variable to numeric  
qol\_train$response <- as.numeric(as.character(qol\_train$response))  
# Fit the OLS model  
ols\_model <- lm(  
 response ~ .,  
 data = qol\_train  
 )  
# Predict on the test set  
predictions <- predict(  
 ols\_model,  
 newdata = qol\_test  
 )  
  
# Convert predictions to binary outcomes  
predicted\_classes <- ifelse(predictions > 0.5, 1, 0)  
  
# Evaluate the model  
confusion\_matrix <- table(  
 qol\_test$response,  
 predicted\_classes  
 )  
  
accuracy\_ols <- sum(diag(confusion\_matrix)) / sum(confusion\_matrix)  
# Extract coefficients  
coefficients <- summary(ols\_model)$coefficients  
  
# Calculate importance (absolute value of coefficients)  
feature\_importance <- abs(coefficients[, "Estimate"])  
  
# Create a data frame for better visualization  
importance\_df <- data.frame(  
 Feature = rownames(coefficients),  
 Importance = feature\_importance  
 )  
  
# Sort by importance  
importance\_df <- importance\_df[order(-importance\_df$Importance), ]  
  
print(importance\_df)

## Feature Importance  
## (Intercept) (Intercept) 6.065686e-01  
## region.Northeast region.Northeast 2.247162e-01  
## region.Midwest region.Midwest 2.005283e-01  
## pct\_adult\_smokers pct\_adult\_smokers 1.695130e-01  
## pct\_hispanic pct\_hispanic 5.788575e-02  
## pct\_voters pct\_voters 5.504464e-02  
## days\_over\_90\_f days\_over\_90\_f 4.660694e-02  
## region.South region.South 4.197146e-02  
## pct\_food\_insecurities pct\_food\_insecurities 3.745090e-02  
## pct\_some\_college pct\_some\_college 3.702241e-02  
## pct\_obese\_adults pct\_obese\_adults 3.467994e-02  
## pct\_binge\_drinkers pct\_binge\_drinkers 3.235476e-02  
## dentist\_pt dentist\_pt 2.639298e-02  
## pa\_pt pa\_pt 2.620384e-02  
## median\_drug\_alcohol\_care\_dist median\_drug\_alcohol\_care\_dist 2.522178e-02  
## pct\_male pct\_male 2.496785e-02  
## pct\_other\_race pct\_other\_race 2.426572e-02  
## pct\_under\_65\_no\_health\_insurance pct\_under\_65\_no\_health\_insurance 2.101870e-02  
## pct\_30\_min\_plus\_commute pct\_30\_min\_plus\_commute 1.999765e-02  
## mental\_health\_faciliy\_pt mental\_health\_faciliy\_pt 1.869019e-02  
## pct\_no\_exercise pct\_no\_exercise 1.848760e-02  
## pct\_hh\_internet pct\_hh\_internet 1.825548e-02  
## pct\_high\_housing\_costs pct\_high\_housing\_costs 1.734561e-02  
## water\_quality water\_quality 1.716508e-02  
## pct\_employed pct\_employed 1.626491e-02  
## median\_hh\_income median\_hh\_income 1.537700e-02  
## pct\_hh\_inc\_99999 pct\_hh\_inc\_99999 1.338018e-02  
## air\_polution\_metric air\_polution\_metric 1.293266e-02  
## pct\_home\_owner pct\_home\_owner 1.202414e-02  
## pct\_native\_american pct\_native\_american 1.126768e-02  
## pct\_overcrowded\_hh pct\_overcrowded\_hh 1.042646e-02  
## pct\_rural\_population pct\_rural\_population 9.885690e-03  
## average\_hh\_size average\_hh\_size 9.628259e-03  
## pct\_65\_plus pct\_65\_plus 9.467883e-03  
## median\_pediatric\_icu\_dist median\_pediatric\_icu\_dist 9.413730e-03  
## pct\_grandparents\_as\_guardians pct\_grandparents\_as\_guardians 8.605350e-03  
## pct\_highschool\_diploma pct\_highschool\_diploma 6.251225e-03  
## pct\_asian pct\_asian 6.149321e-03  
## median\_health\_clinic\_dist median\_health\_clinic\_dist 5.405437e-03  
## pct\_white pct\_white 5.326776e-03  
## median\_er\_dist median\_er\_dist 5.246382e-03  
## pct\_w\_medicare pct\_w\_medicare 3.991952e-03  
## social\_clubs\_per\_10k social\_clubs\_per\_10k 3.571101e-03  
## pct\_black pct\_black 2.564668e-03  
## clinical\_nurse\_pt clinical\_nurse\_pt 2.291458e-03  
## weighted\_population weighted\_population 2.271588e-03  
## inequality\_ratio inequality\_ratio 2.040122e-03  
## pct\_single\_parent pct\_single\_parent 9.056815e-04  
## population\_density population\_density 7.783507e-04  
## pct\_hh\_other\_computer pct\_hh\_other\_computer 9.625559e-05

visualize feature importance

# Select the top 20 important features  
top\_20\_features\_ols <- importance\_df %>%  
 arrange(desc(Importance)) %>%  
 head(20)  
  
# Create the bar plot for the top 20 features  
ggplot(top\_20\_features\_ols, aes(x = reorder(Feature, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() + # Flip coordinates for better readability  
 labs(title = "Top 20 Feature Importance using ols", x = "Feature", y = "Importance") +  
 theme\_minimal()



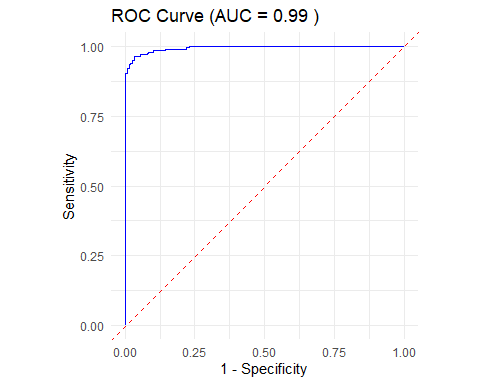
Build ROC curve

# Load necessary libraries  
library(ggplot2)  
library(pROC)  
  
# Calculate ROC curve  
roc\_ols <- roc(qol\_test$response, predictions)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

# Create a data frame for ggplot  
roc\_data <- data.frame(  
 specificity = rev(roc\_ols$specificities),  
 sensitivity = rev(roc\_ols$sensitivities)  
)  
  
# Plot ROC curve using ggplot2  
ggplot(roc\_data, aes(x = 1-specificity, y = sensitivity)) +  
 geom\_line(color = "blue") +  
 geom\_abline(linetype = "dashed", color = "red") +  
 labs(title = paste("ROC Curve (AUC =", round(auc(roc\_ols), 2), ")"),  
 x = "1 - Specificity",  
 y = "Sensitivity") +  
 theme\_minimal() +  
 coord\_fixed(ratio = 1)

 ### Build random forest model

# Train a random forest model for classification  
model\_rf <- randomForest::randomForest(  
 as.factor(response) ~ .,  
 data = qol\_train,  
 ntree = 500  
 )  
  
# Predict the response variable for the testing set  
predictions2 <- predict(  
 model\_rf,  
 qol\_test,  
 type = "response"  
 )  
  
# Calculate the accuracy of the random forest model  
accuracy <- mean(qol\_test$response == predictions2)  
print(accuracy)

## [1] 0.9977273

# Calculate the ROC AUC score  
roc\_obj <- roc(  
 qol\_test$response,  
 as.numeric(predictions2)  
 )

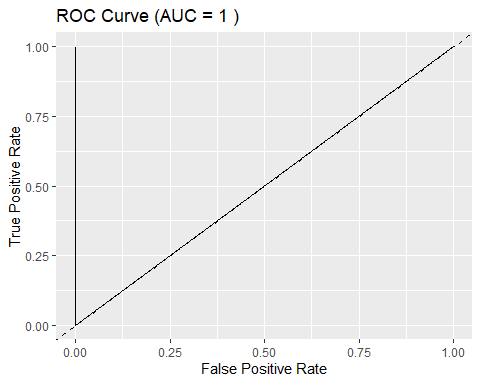
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

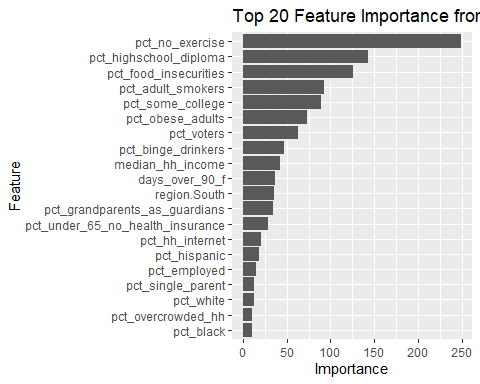
auc\_rf <- auc(roc\_obj)  
print(auc\_rf)

## Area under the curve: 0.9975

# Plot the ROC curve using ggplot2  
roc\_df <- data.frame(  
 tpr = roc\_obj$sensitivities,  
 fpr = 1 - roc\_obj$specificities  
)  
  
ggplot(roc\_df, aes(x = fpr, y = tpr)) +  
 geom\_line() +  
 geom\_abline(linetype = "dashed") +  
 xlab("False Positive Rate") +  
 ylab("True Positive Rate") +  
 ggtitle(paste("ROC Curve (AUC =", round(auc\_rf, 2), ")"))



# Calculate feature importance  
importance\_rf <- randomForest::importance(model\_rf)  
importance\_df <- data.frame(  
 Feature = rownames(importance\_rf),  
 Importance = importance\_rf[, 1])  
  
# Select top 20 important features  
top\_20\_importance <- importance\_df[order(-importance\_df$Importance), ][1:20, ]  
  
# Plot top 20 feature importance using ggplot2  
ggplot(top\_20\_importance, aes(x = reorder(Feature, Importance), y = Importance)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 xlab("Feature") +  
 ylab("Importance") +  
 ggtitle("Top 20 Feature Importance from Random Forest Model")



*Boxcox Analysis*

qol\_data <- qol\_data %>%  
 mutate(response = ifelse(response == "better", 1, 2)) # convert to 1 and 2 for boxcox analysis, log doesn't work on 0

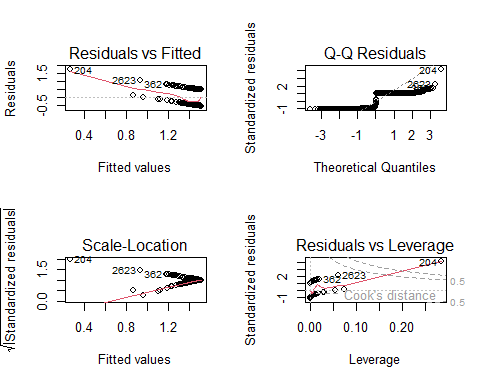
# Initialize the results data frame  
transform\_output <- data.frame(  
 Predictor = character(),  
 Optimal\_Lambda = numeric(),  
 Original\_R\_squared = numeric(),  
 Original\_Adj\_R\_squared = numeric(),  
 Transformed\_R\_squared = numeric(),  
 Transformed\_Adj\_R\_squared = numeric(),  
 stringsAsFactors = FALSE  
)

*Repeat the following:*

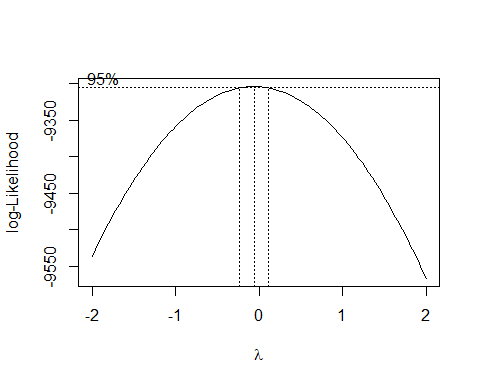
# Fit the initial linear model  
lm\_weighted\_pop <- lm(  
 response ~ weighted\_population,  
 qol\_data  
 )  
summary(lm\_weighted\_pop)

##   
## Call:  
## lm(formula = response ~ weighted\_population, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5093 -0.5052 -0.3151 0.4934 1.7315   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.509e+00 9.321e-03 161.928 < 2e-16 \*\*\*  
## weighted\_population -1.236e-07 2.678e-08 -4.615 4.09e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4985 on 3138 degrees of freedom  
## Multiple R-squared: 0.006742, Adjusted R-squared: 0.006425   
## F-statistic: 21.3 on 1 and 3138 DF, p-value: 4.085e-06

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_weighted\_pop)



original\_r\_squared <- summary(lm\_weighted\_pop)$r.squared  
original\_adj\_r\_squared <- summary(lm\_weighted\_pop)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_cox\_weighted\_pop <- MASS::boxcox(  
 object = lm\_weighted\_pop  
 )



# identify the optimal lambda  
optimal\_lambda <- box\_cox\_weighted\_pop$x[which.max(box\_cox\_weighted\_pop$y)]  
print(optimal\_lambda)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_weighted\_pop = weighted\_population^optimal\_lambda)  
  
  
# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_weighted\_pop, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_weighted\_pop, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.6904 -0.4902 -0.3883 0.4976 0.6529   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.00059 0.09969 10.037 < 2e-16 \*\*\*  
## transformed\_weighted\_pop 0.92059 0.18432 4.994 6.22e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4982 on 3138 degrees of freedom  
## Multiple R-squared: 0.007886, Adjusted R-squared: 0.00757   
## F-statistic: 24.94 on 1 and 3138 DF, p-value: 6.224e-07

transformed\_r\_squared <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared)

## [1] 0.007886352

print(transformed\_adj\_r\_squared)

## [1] 0.007570191

# Store the results in the results data frame  
results <- rbind(transform\_output, data.frame(  
 Predictor = "weighted\_population",  
 Optimal\_Lambda = optimal\_lambda,  
 Original\_R\_squared = original\_r\_squared,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared,  
 Transformed\_R\_squared = transformed\_r\_squared,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared  
))  
  
# Print the results  
print(results)

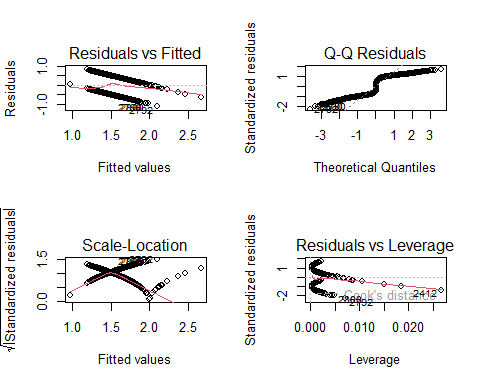
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 0.006741906 0.00642538  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 0.007886352 0.007570191

*repeat*

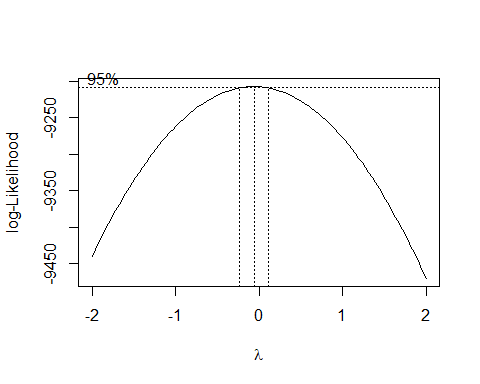
# Fit the initial linear model  
lm\_hh\_size <- lm(  
 response ~ average\_hh\_size,  
 qol\_data  
 )  
summary(lm\_hh\_size)

##   
## Call:  
## lm(formula = response ~ average\_hh\_size, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.0931 -0.4485 -0.2485 0.4904 0.8103   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.32390 0.07942 4.078 4.65e-05 \*\*\*  
## average\_hh\_size 0.47053 0.03168 14.853 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4834 on 3138 degrees of freedom  
## Multiple R-squared: 0.06568, Adjusted R-squared: 0.06538   
## F-statistic: 220.6 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_hh\_size)



original\_r\_squared2 <- summary(lm\_hh\_size)$r.squared  
original\_adj\_r\_squared2 <- summary(lm\_hh\_size)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_cox\_hh\_size <- MASS::boxcox(  
 object = lm\_hh\_size  
 )



# identify the optimal lambda  
optimal\_lambda2 <- box\_cox\_hh\_size$x[which.max(box\_cox\_hh\_size$y)]  
print(optimal\_lambda2)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_hh\_size = average\_hh\_size^optimal\_lambda2)  
  
# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_hh\_size, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_hh\_size, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.0249 -0.4503 -0.1971 0.4817 0.8886   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.663 1.372 16.52 <2e-16 \*\*\*  
## transformed\_hh\_size -22.363 1.449 -15.43 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4822 on 3138 degrees of freedom  
## Multiple R-squared: 0.07052, Adjusted R-squared: 0.07023   
## F-statistic: 238.1 on 1 and 3138 DF, p-value: < 2.2e-16

transformed\_r\_squared2 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared2 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared2)

## [1] 0.0705244

print(transformed\_adj\_r\_squared2)

## [1] 0.0702282

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "average\_hh\_size",  
 Optimal\_Lambda = optimal\_lambda2,  
 Original\_R\_squared = original\_r\_squared2,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared2,  
 Transformed\_R\_squared = transformed\_r\_squared2,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared2  
))  
  
results <- temp\_results  
# Print the results  
print(results)

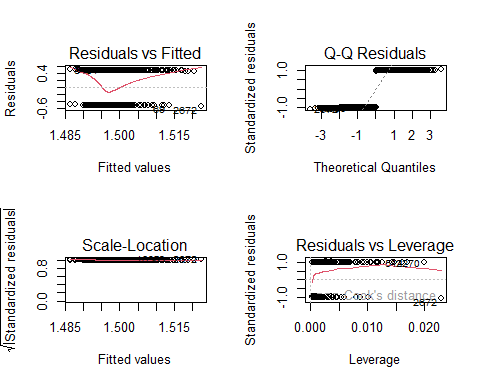
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 0.006741906 0.00642538  
## 2 average\_hh\_size -0.06060606 0.065681275 0.06538353  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 0.007886352 0.007570191  
## 2 0.070524399 0.070228199

*repeat*

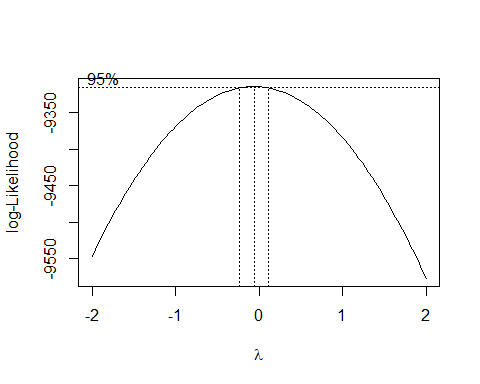
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_male,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_male, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5224 -0.4961 -0.4916 0.5043 0.5136   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.434307 0.180861 7.930 3.01e-15 \*\*\*  
## pct\_male 0.001241 0.003606 0.344 0.731   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5001 on 3138 degrees of freedom  
## Multiple R-squared: 3.777e-05, Adjusted R-squared: -0.0002809   
## F-statistic: 0.1185 on 1 and 3138 DF, p-value: 0.7307

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared3 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared3 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda3 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda3)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_male = pct\_male^optimal\_lambda3)  
  
# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_male, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_male, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5053 -0.4968 -0.4892 0.5029 0.5198   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.6725 3.1560 0.213 0.831  
## transformed\_male 1.0445 4.0006 0.261 0.794  
##   
## Residual standard error: 0.5001 on 3138 degrees of freedom  
## Multiple R-squared: 2.172e-05, Adjusted R-squared: -0.0002969   
## F-statistic: 0.06817 on 1 and 3138 DF, p-value: 0.794

transformed\_r\_squared3 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared3 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared3)

## [1] 2.172349e-05

print(transformed\_adj\_r\_squared3)

## [1] -0.0002969439

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_male",  
 Optimal\_Lambda = optimal\_lambda3,  
 Original\_R\_squared = original\_r\_squared3,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared3,  
 Transformed\_R\_squared = transformed\_r\_squared3,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared3  
))  
  
results <- temp\_results  
# Print the results  
print(results)

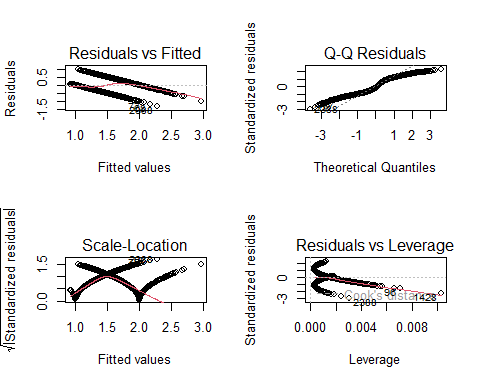
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439

*repeat*

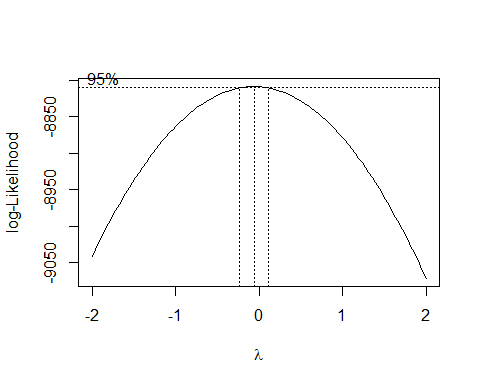
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_hh\_internet,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_hh\_internet, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.27132 -0.36472 -0.09233 0.39635 0.95214   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.0331461 0.0738832 54.59 <2e-16 \*\*\*  
## pct\_hh\_internet -0.0321033 0.0009301 -34.52 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4258 on 3138 degrees of freedom  
## Multiple R-squared: 0.2752, Adjusted R-squared: 0.275   
## F-statistic: 1191 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared4 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared4 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda4 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda4)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_internet = pct\_hh\_internet^optimal\_lambda4)

# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_internet, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_internet, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4925 -0.3705 -0.1414 0.4196 0.8852   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -35.946 1.138 -31.59 <2e-16 \*\*\*  
## transformed\_internet 48.777 1.482 32.91 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4312 on 3138 degrees of freedom  
## Multiple R-squared: 0.2566, Adjusted R-squared: 0.2563   
## F-statistic: 1083 on 1 and 3138 DF, p-value: < 2.2e-16

transformed\_r\_squared4 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared4 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared4)

## [1] 0.2565791

print(transformed\_adj\_r\_squared4)

## [1] 0.2563422

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_hh\_internet",  
 Optimal\_Lambda = optimal\_lambda4,  
 Original\_R\_squared = original\_r\_squared4,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared4,  
 Transformed\_R\_squared = transformed\_r\_squared4,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared4  
))  
  
results <- temp\_results  
# Print the results  
print(results)

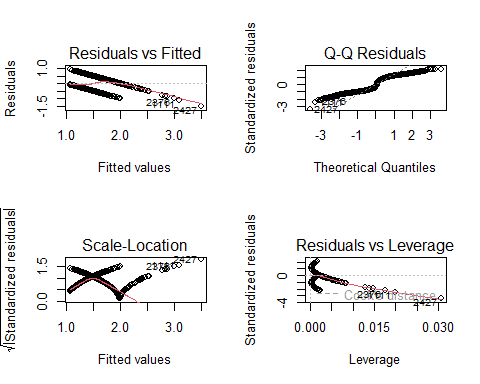
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## 4 pct\_hh\_internet -0.06060606 2.751841e-01 0.2749530727  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439  
## 4 2.565791e-01 0.2563421874

*repeat*

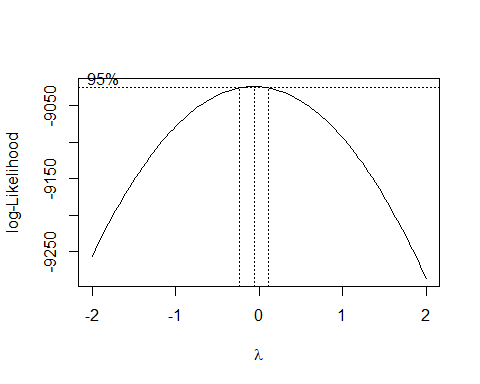
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_employed,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_employed, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.5005 -0.4138 -0.1441 0.4526 0.9171   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.038096 0.298929 30.23 <2e-16 \*\*\*  
## pct\_employed -0.079552 0.003152 -25.24 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.456 on 3138 degrees of freedom  
## Multiple R-squared: 0.1687, Adjusted R-squared: 0.1685   
## F-statistic: 637 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared5 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared5 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda5 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda5)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_employed = pct\_employed^optimal\_lambda5)

# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_employed, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_employed, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.7435 -0.4149 -0.1634 0.4575 0.8912   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -117.579 4.800 -24.49 <2e-16 \*\*\*  
## transformed\_employed 156.898 6.325 24.81 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4573 on 3138 degrees of freedom  
## Multiple R-squared: 0.1639, Adjusted R-squared: 0.1637   
## F-statistic: 615.3 on 1 and 3138 DF, p-value: < 2.2e-16

transformed\_r\_squared5 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared5 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared5)

## [1] 0.1639411

print(transformed\_adj\_r\_squared5)

## [1] 0.1636747

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_employed",  
 Optimal\_Lambda = optimal\_lambda5,  
 Original\_R\_squared = original\_r\_squared5,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared5,  
 Transformed\_R\_squared = transformed\_r\_squared5,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared5  
))  
  
results <- temp\_results  
# Print the results  
print(results)

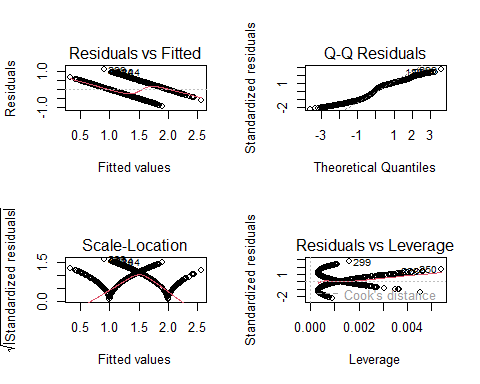
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## 4 pct\_hh\_internet -0.06060606 2.751841e-01 0.2749530727  
## 5 pct\_employed -0.06060606 1.687333e-01 0.1684683737  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439  
## 4 2.565791e-01 0.2563421874  
## 5 1.639411e-01 0.1636746853

*repeat*

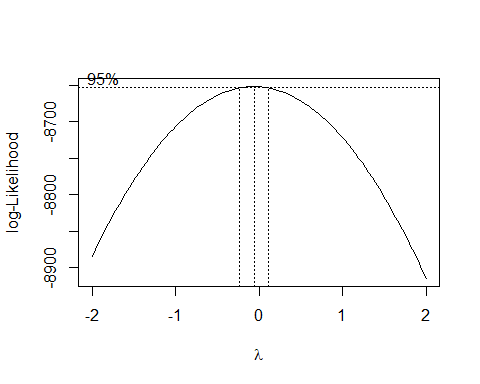
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_obese\_adults,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_obese\_adults, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.89865 -0.37026 0.04482 0.34668 1.09522   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.78102 0.05657 -13.80 <2e-16 \*\*\*  
## pct\_obese\_adults 6.29030 0.15497 40.59 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.405 on 3138 degrees of freedom  
## Multiple R-squared: 0.3443, Adjusted R-squared: 0.3441   
## F-statistic: 1648 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared5 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared5 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda5 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda5)

## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_obese = pct\_obese\_adults^optimal\_lambda5)

# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_obese, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_obese, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.8491 -0.3986 0.0897 0.3494 1.1086   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 35.4687 0.8801 40.3 <2e-16 \*\*\*  
## transformed\_obese -31.9251 0.8270 -38.6 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4118 on 3138 degrees of freedom  
## Multiple R-squared: 0.322, Adjusted R-squared: 0.3217   
## F-statistic: 1490 on 1 and 3138 DF, p-value: < 2.2e-16

transformed\_r\_squared5 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared5 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared5)

## [1] 0.3219628

print(transformed\_adj\_r\_squared5)

## [1] 0.3217467

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_obese\_adults",  
 Optimal\_Lambda = optimal\_lambda5,  
 Original\_R\_squared = original\_r\_squared5,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared5,  
 Transformed\_R\_squared = transformed\_r\_squared5,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared5  
))  
  
results <- temp\_results  
# Print the results  
print(results)

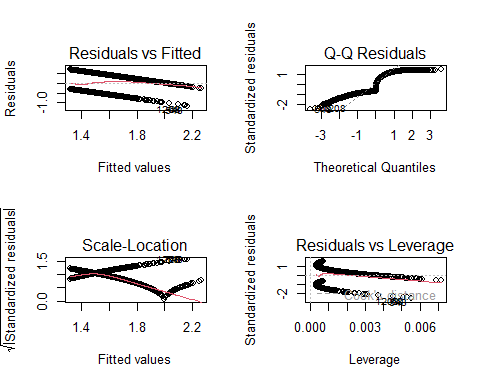
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## 4 pct\_hh\_internet -0.06060606 2.751841e-01 0.2749530727  
## 5 pct\_employed -0.06060606 1.687333e-01 0.1684683737  
## 6 pct\_obese\_adults -0.06060606 3.442706e-01 0.3440616529  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439  
## 4 2.565791e-01 0.2563421874  
## 5 1.639411e-01 0.1636746853  
## 6 3.219628e-01 0.3217467135

*repeat*

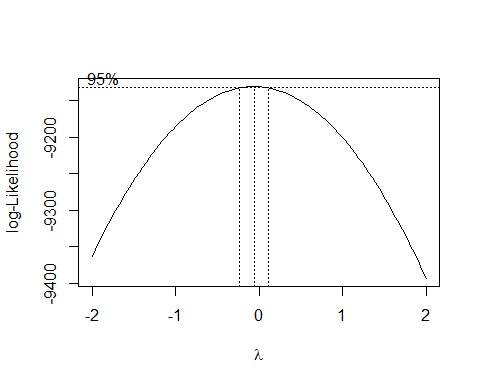
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_white,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_white, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.1677 -0.4013 -0.3295 0.4945 0.6814   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.2980989 0.0415222 55.35 <2e-16 \*\*\*  
## pct\_white -0.0098042 0.0004973 -19.71 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4718 on 3138 degrees of freedom  
## Multiple R-squared: 0.1102, Adjusted R-squared: 0.1099   
## F-statistic: 388.7 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared6 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared6 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda6 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda6)

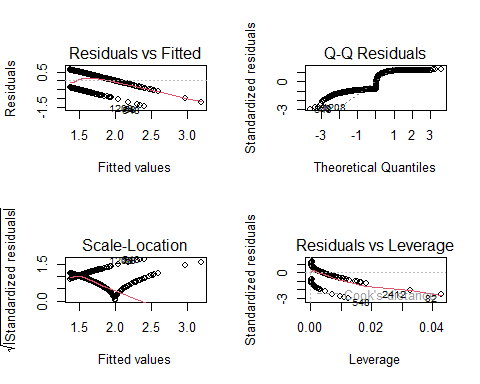
## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_white = pct\_white^optimal\_lambda6)

# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_white, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_white, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4049 -0.4259 -0.3916 0.5152 0.6167   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -6.4756 0.4646 -13.94 <2e-16 \*\*\*  
## transformed\_white 10.3884 0.6053 17.16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4782 on 3138 degrees of freedom  
## Multiple R-squared: 0.08581, Adjusted R-squared: 0.08552   
## F-statistic: 294.6 on 1 and 3138 DF, p-value: < 2.2e-16

par(mfrow = c(2, 2))  
plot(transformed\_lm)



transformed\_r\_squared6 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared6 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared6)

## [1] 0.08581412

print(transformed\_adj\_r\_squared6)

## [1] 0.08552279

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_white",  
 Optimal\_Lambda = optimal\_lambda6,  
 Original\_R\_squared = original\_r\_squared6,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared6,  
 Transformed\_R\_squared = transformed\_r\_squared6,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared6  
))  
  
results <- temp\_results  
# Print the results  
print(results)

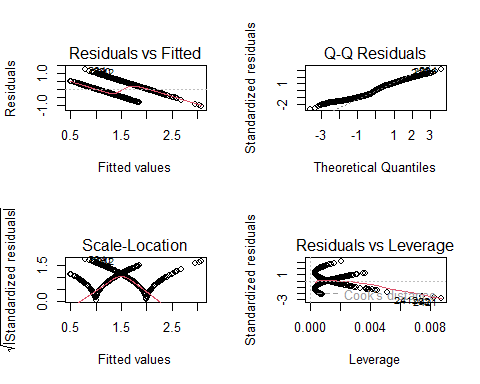
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## 4 pct\_hh\_internet -0.06060606 2.751841e-01 0.2749530727  
## 5 pct\_employed -0.06060606 1.687333e-01 0.1684683737  
## 6 pct\_obese\_adults -0.06060606 3.442706e-01 0.3440616529  
## 7 pct\_white -0.06060606 1.102106e-01 0.1099270372  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439  
## 4 2.565791e-01 0.2563421874  
## 5 1.639411e-01 0.1636746853  
## 6 3.219628e-01 0.3217467135  
## 7 8.581412e-02 0.0855227922

*repeat*

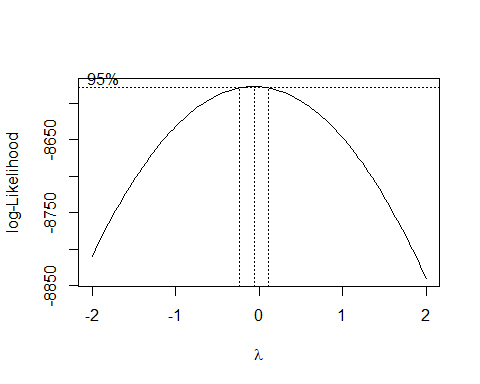
# Fit the initial linear model  
lm\_initial <- lm(  
 response ~ pct\_adult\_smokers,  
 qol\_data  
 )  
summary(lm\_initial)

##   
## Call:  
## lm(formula = response ~ pct\_adult\_smokers, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.0694 -0.3439 -0.0087 0.3200 1.2163   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.0005877 0.0352377 -0.017 0.987   
## pct\_adult\_smokers 7.4694609 0.1722499 43.364 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3955 on 3138 degrees of freedom  
## Multiple R-squared: 0.3747, Adjusted R-squared: 0.3745   
## F-statistic: 1880 on 1 and 3138 DF, p-value: < 2.2e-16

# Plot diagnostic plots for the initial model  
par(mfrow = c(2, 2))  
plot(lm\_initial)



original\_r\_squared7 <- summary(lm\_initial)$r.squared  
original\_adj\_r\_squared7 <- summary(lm\_initial)$adj.r.squared  
  
# Perform Box-Cox transformation on the response variable  
par(mfrow = c(1, 1))  
box\_initial\_lm <- MASS::boxcox(  
 object = lm\_initial  
 )



# identify the optimal lambda  
optimal\_lambda7 <- box\_initial\_lm$x[which.max(box\_initial\_lm$y)]  
print(optimal\_lambda6)

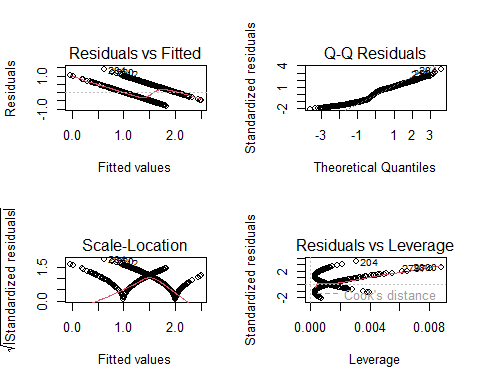
## [1] -0.06060606

# add transformed data to dataframe, use log of value if optimal lambda is 0  
qol\_data <- qol\_data %>%   
 mutate(transformed\_smokers = pct\_adult\_smokers^optimal\_lambda7)

# Fit a new linear model with the transformed predictor variable  
transformed\_lm <- lm(response ~ transformed\_smokers, qol\_data)  
summary(transformed\_lm)

##   
## Call:  
## lm(formula = response ~ transformed\_smokers, data = qol\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.8255 -0.3725 0.0537 0.3202 1.3791   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 24.2471 0.5515 43.97 <2e-16 \*\*\*  
## transformed\_smokers -20.6097 0.4995 -41.26 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4027 on 3138 degrees of freedom  
## Multiple R-squared: 0.3517, Adjusted R-squared: 0.3515   
## F-statistic: 1702 on 1 and 3138 DF, p-value: < 2.2e-16

par(mfrow = c(2, 2))  
plot(transformed\_lm)



transformed\_r\_squared7 <- summary(transformed\_lm)$r.squared  
transformed\_adj\_r\_squared7 <- summary(transformed\_lm)$adj.r.squared  
  
print(transformed\_r\_squared7)

## [1] 0.3516808

print(transformed\_adj\_r\_squared7)

## [1] 0.3514742

# Store the results in the results data frame  
# Store the results in the results data frame  
temp\_results <- rbind(results, data.frame(  
 Predictor = "pct\_adult\_smokers",  
 Optimal\_Lambda = optimal\_lambda7,  
 Original\_R\_squared = original\_r\_squared7,  
 Original\_Adj\_R\_squared = original\_adj\_r\_squared7,  
 Transformed\_R\_squared = transformed\_r\_squared7,  
 Transformed\_Adj\_R\_squared = transformed\_adj\_r\_squared6  
))  
  
results <- temp\_results  
# Print the results  
print(results)

## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## 1 weighted\_population -0.06060606 6.741906e-03 0.0064253797  
## 2 average\_hh\_size -0.06060606 6.568127e-02 0.0653835315  
## 3 pct\_male -0.06060606 3.776962e-05 -0.0002808927  
## 4 pct\_hh\_internet -0.06060606 2.751841e-01 0.2749530727  
## 5 pct\_employed -0.06060606 1.687333e-01 0.1684683737  
## 6 pct\_obese\_adults -0.06060606 3.442706e-01 0.3440616529  
## 7 pct\_white -0.06060606 1.102106e-01 0.1099270372  
## 8 pct\_adult\_smokers -0.06060606 3.747065e-01 0.3745072798  
## Transformed\_R\_squared Transformed\_Adj\_R\_squared  
## 1 7.886352e-03 0.0075701913  
## 2 7.052440e-02 0.0702281994  
## 3 2.172349e-05 -0.0002969439  
## 4 2.565791e-01 0.2563421874  
## 5 1.639411e-01 0.1636746853  
## 6 3.219628e-01 0.3217467135  
## 7 8.581412e-02 0.0855227922  
## 8 3.516808e-01 0.0855227922

# Convert the dataframe to a tibble for better printing  
boxcox\_tbl <- as\_tibble(results)  
  
# Print the tibble  
print(boxcox\_tbl)

## # A tibble: 8 × 6  
## Predictor Optimal\_Lambda Original\_R\_squared Original\_Adj\_R\_squared  
## <chr> <dbl> <dbl> <dbl>  
## 1 weighted\_population -0.0606 0.00674 0.00643   
## 2 average\_hh\_size -0.0606 0.0657 0.0654   
## 3 pct\_male -0.0606 0.0000378 -0.000281  
## 4 pct\_hh\_internet -0.0606 0.275 0.275   
## 5 pct\_employed -0.0606 0.169 0.168   
## 6 pct\_obese\_adults -0.0606 0.344 0.344   
## 7 pct\_white -0.0606 0.110 0.110   
## 8 pct\_adult\_smokers -0.0606 0.375 0.375   
## # ℹ 2 more variables: Transformed\_R\_squared <dbl>,  
## # Transformed\_Adj\_R\_squared <dbl>