Practicum II

Anay Athawale

2023-07-18

Problem 1

Download the data set Census Income Data for Adults along with its explanation. There are two data sets (adult.data and adult.test). Note that the data file does not contain header names; you may wish to add those. The description of each column can be found in the data set explanation. Combine the two data sets into a single data set.

```
adult.data<-read.csv2("/Users/anayathawale/Downloads/adult/adult.data", header = FALSE, sep = ",", stri
adult.test<-read.csv2("/Users/anayathawale/Downloads/adult/adult.test", header = FALSE, sep = ",", stri

# Assigning column names
colnames(adult.data)<-c("age", "workclass", "fnlwgt", "education", "education-num", "marital-status", "
colnames(adult.test)<-c("age", "workclass", "fnlwgt", "education", "education-num", "marital-status", "

df<- rbind(adult.data,adult.test)
df<-na.omit(df)
df$income <- ifelse(grepl("^>", df$income), ">50K", "<50K")
```

Explore the combined data set as you see fit and that allows you to get a sense of the data and get comfortable with it.

```
##      age             workclass             fnlwgt          education
##  Length:48842       Length:48842       Min.   :  12285   Length:48842
##  Class :character   Class :character   1st Qu.: 117550   Class :character
##  Mode  :character   Mode  :character   Median : 178144   Mode  :character
##                                        Mean   : 189664
##                                        3rd Qu.: 237642
##                                        Max.   :1490400
##  education-num   marital-status      occupation        relationship
##  Min.   : 1.00   Length:48842       Length:48842       Length:48842
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.08
##  3rd Qu.:12.00
##  Max.   :16.00
##      race                sex             capital-gain    capital-loss
##  Length:48842       Length:48842       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1079   Mean   :  87.5
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##  hours-per-week  native.country        income
##  Min.   : 1.00   Length:48842       Length:48842
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.42
##  3rd Qu.:45.00
##  Max.   :99.00
```

```
## ’data.frame’:    48842 obs. of  15 variables:
##  $ age           : chr  "39" "50" "38" "53" ...
##  $ workclass     : chr  "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education     : chr  "Bachelors" "Bachelors" "HS-grad" "11th" ...
##  $ education-num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital-status: chr  "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse" ...
##  $ occupation    : chr  "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-cleaners" ...
##  $ relationship  : chr  "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ race          : chr  "White" "White" "White" "Black" ...
##  $ sex           : chr  "Male" "Male" "Male" "Male" ...
##  $ capital-gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours-per-week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<50K" "<50K" "<50K" "<50K" ...
##  - attr(*, "na.action")= ’omit’ Named int 32562
##   ..- attr(*, "names")= chr "32562"
```

```
##   age        workclass fnlwgt education education-num     marital-status
## 1  39        State-gov  77516 Bachelors            13      Never-married
## 2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
## 3  38          Private 215646   HS-grad             9           Divorced
## 4  53          Private 234721      11th             7 Married-civ-spouse
## 5  28          Private 338409 Bachelors            13 Married-civ-spouse
## 6  37          Private 284582   Masters            14 Married-civ-spouse
##          occupation  relationship  race    sex capital-gain capital-loss
## 1      Adm-clerical Not-in-family White   Male         2174            0
## 2   Exec-managerial       Husband White   Male            0            0
## 3 Handlers-cleaners Not-in-family White   Male            0            0
## 4 Handlers-cleaners       Husband Black   Male            0            0
## 5    Prof-specialty          Wife Black Female            0            0
## 6   Exec-managerial          Wife White Female            0            0
##   hours-per-week native.country income
## 1             40  United-States   <50K
## 2             13  United-States   <50K
## 3             40  United-States   <50K
## 4             40  United-States   <50K
## 5             40           Cuba   <50K
## 6             40  United-States   <50K
```

Split the combined data set 75/25% so you retain 25% for validation using random sampling without replacement. Use a fixed seed so you produce the same results each time you run the code. Going forward you will use the 75% data set for training and the 25% data set for validation and determine accuracy.

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
# Set seed for reproducibility
set.seed(123)

# Create indices for the train split
trainIndex <- createDataPartition(df$income, p = 0.75, list = FALSE)

# Create training and validation sets
train_df <- df[trainIndex,]
valid_df <- df[-trainIndex,]
```

Using the Naive Bayes Classification algorithm from the KlaR package, build a binary classifier that predicts whether an individual earns more than or less than US$50,000. Only use the features age, education, workclass, sex, race, and native-country. Ignore any other features in your model. You need to transform continuous variables into categorical variables by binning (use equal size bins from min to max).

```
library(klaR)
```

```
## Loading required package: MASS
```

```
# Ignoring all other features
valid_df <- valid_df[ ,c("age", "education", "workclass", "sex", "race","native.country","income")]
train_df <- train_df[, c("age", "education", "workclass", "sex", "race","native.country","income")]

# Converting variable age to numeric
train_df$age <- as.numeric(as.character(train_df$age))
valid_df$age <- as.numeric(as.character(valid_df$age))

# Convert 'age' into categorical variable
num_bins <- 4
labels <- paste("Bin", seq(1, num_bins))

# Create bins for 'age' in both training and validation sets
train_df$age <- cut(train_df$age, breaks=num_bins, labels=labels)
valid_df$age <- cut(valid_df$age, breaks=num_bins, labels=labels)
```

```r
# Convert all columns of train_df to factors
train_df[] <- lapply(train_df, as.factor)

# Convert all columns of valid_df to factors
valid_df[] <- lapply(valid_df, as.factor)

# Build the Naive Bayes model
nb_model <- NaiveBayes(income ~ ., data=train_df)

# Make predictions
predictions <- predict(nb_model, valid_df)
```

```
## Warning in FUN(X[[i]], ...): NAs produced for new level(s)
## ('Holand-Netherlands') of newdata$native.country

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 808

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1110

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2196

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2585

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 4223

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 4453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5640

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5682

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5790

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 6721

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 6942

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 7592
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 7594
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 8070
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 8354
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 8603
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 9339
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 10009
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 11072
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 11137
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 11343
```

Build a confusion matrix for the classifier from (4) and comment on it, e.g., explain what it means.

```r
# Evaluate model
confusionMatrix(predictions$class, valid_df$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <50K >50K
##       <50K 8620 1786
##       >50K  668 1135
##
##                Accuracy : 0.799
##                  95% CI : (0.7918, 0.8061)
##     No Information Rate : 0.7608
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3645
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
##               Sensitivity : 0.9281
##               Specificity : 0.3886
##            Pos Pred Value : 0.8284
##            Neg Pred Value : 0.6295
##                Prevalence : 0.7608
##            Detection Rate : 0.7060
##      Detection Prevalence : 0.8523
##         Balanced Accuracy : 0.6583
##
##          'Positive' Class : <50K
##
```

1. Confusion Matrix: The confusion matrix shows that out of all the predictions made by your model: 8620 were True Positives, i.e., the model correctly predicted them as '<50K'. 1135 were True Negatives, i.e., the model correctly predicted them as '>50K'. 668 were False Negatives, i.e., the model incorrectly predicted them as '<50K', when they were actually '>50K'. 1786 were False Positives, i.e., the model incorrectly predicted them as '>50K', when they were actually '<50K'.

2. Comment: The model demonstrates good sensitivity (0.9281), implying it is successful at predicting individuals who earn '<50K'. However, it performs poorly on specificity (0.3886), indicating the model struggles to correctly identify individuals who earn '>50K'. This could be a problem if the goal of the prediction is to find individuals who earn more than 50K. In such a case, the model would lead to many false positives, i.e., people being incorrectly identified as earning '>50K'.

3. The overall accuracy of the model is 0.799, indicating it made correct predictions for approximately 80% of the data points.

4. The Kappa statistic of 0.3645 is relatively low, suggesting that the model's performance isn't significantly better than random guessing.

Create a full logistic regression model of the same features as in (4) (i.e., do not eliminate any features regardless of p-value). Be sure to either use some encoding for categorical features or convert them to factor variables and ensure that the glm function does the dummy coding.

```r
# Factorizing the variables in train and validation dataframes
for (v in names(train_df)) {
    train_df[[v]] <- as.factor(train_df[[v]])
    valid_df[[v]] <- factor(valid_df[[v]], levels = levels(train_df[[v]]))
}


# Fit logistic regression model
logistic_model <- glm(income ~ ., data=train_df, family=binomial(link="logit"))

# Print summary of the model
summary(logistic_model)
```

```
##
## Call:
## glm(formula = income ~ ., family = binomial(link = "logit"),
##     data = train_df)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2948  -0.6492  -0.4011  -0.1263   3.2335
##
## Coefficients:
##                                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                              -6.053606   0.264104 -22.921  < 2e-16
## ageBin 2                                  1.374525   0.033152  41.462  < 2e-16
## ageBin 3                                  1.335116   0.044937  29.711  < 2e-16
## ageBin 4                                  0.494022   0.135878   3.636 0.000277
## education11th                            -0.064689   0.179603  -0.360 0.718717
## education12th                             0.316774   0.222399   1.424 0.154345
## education1st-4th                         -0.467713   0.394959  -1.184 0.236331
## education5th-6th                         -0.158791   0.277443  -0.572 0.567095
## education7th-8th                         -0.263137   0.202594  -1.299 0.193999
## education9th                             -0.207152   0.227442  -0.911 0.362405
## educationAssoc-acdm                       1.655108   0.146223  11.319  < 2e-16
## educationAssoc-voc                        1.628097   0.142307  11.441  < 2e-16
## educationBachelors                        2.375025   0.130996  18.131  < 2e-16
## educationDoctorate                        3.481772   0.172656  20.166  < 2e-16
## educationHS-grad                          0.932903   0.130114   7.170 7.51e-13
## educationMasters                          2.762558   0.137487  20.093  < 2e-16
## educationPreschool                       -1.155505   1.024122  -1.128 0.259198
## educationProf-school                      3.550558   0.161327  22.008  < 2e-16
## educationSome-college                     1.306678   0.131140   9.964  < 2e-16
## workclassFederal-gov                      1.170784   0.110114  10.632  < 2e-16
## workclassLocal-gov                        0.699753   0.098430   7.109 1.17e-12
## workclassNever-worked                    -7.899740  70.321745  -0.112 0.910556
## workclassPrivate                          0.743788   0.085148   8.735  < 2e-16
## workclassSelf-emp-inc                     1.455376   0.105443  13.803  < 2e-16
## workclassSelf-emp-not-inc                 0.348515   0.095944   3.632 0.000281
## workclassState-gov                        0.323946   0.109155   2.968 0.003000
## workclassWithout-pay                     -0.579502   1.057232  -0.548 0.583602
## sexMale                                   1.289080   0.035943  35.864  < 2e-16
## raceAsian-Pac-Islander                    0.685390   0.216753   3.162 0.001566
## raceBlack                                 0.020814   0.190553   0.109 0.913022
## raceOther                                -0.024667   0.291365  -0.085 0.932532
## raceWhite                                 0.581602   0.182319   3.190 0.001423
## native.countryCambodia                    0.725431   0.604261   1.201 0.229935
## native.countryCanada                      0.711700   0.237655   2.995 0.002747
## native.countryChina                      -0.282074   0.304589  -0.926 0.354405
## native.countryColumbia                   -1.446679   0.584406  -2.475 0.013306
## native.countryCuba                        0.067161   0.292362   0.230 0.818311
## native.countryDominican-Republic         -0.910415   0.566075  -1.608 0.107771
## native.countryEcuador                     0.096791   0.632560   0.153 0.878387
## native.countryEl-Salvador                -0.232385   0.385025  -0.604 0.546137
## native.countryEngland                     0.718018   0.276835   2.594 0.009496
## native.countryFrance                      0.363746   0.536617   0.678 0.497866
## native.countryGermany                     0.378149   0.240773   1.571 0.116284
## native.countryGreece                      0.969950   0.414290   2.341 0.019220
## native.countryGuatemala                  -0.915564   0.749101  -1.222 0.221625
## native.countryHaiti                      -0.004499   0.496585  -0.009 0.992771
## native.countryHonduras                    0.278567   0.868791   0.321 0.748485
## native.countryHong                        0.365071   0.664178   0.550 0.582554
## native.countryHungary                     0.531500   0.598663   0.888 0.374642
## native.countryIndia                       0.008503   0.271849   0.031 0.975047
## native.countryIran                        0.232262   0.385033   0.603 0.546358
## native.countryIreland                     1.019161   0.470727   2.165 0.030382
## native.countryItaly                       0.776407   0.296501   2.619 0.008830
## native.countryJamaica                     0.537644   0.374374   1.436 0.150970
## native.countryJapan                       0.191103   0.335102   0.570 0.568485
## native.countryLaos                       -0.439097   0.958547  -0.458 0.646891
## native.countryMexico                     -0.559294   0.218256  -2.563 0.010390
## native.countryNicaragua                  -1.623386   1.044333  -1.554 0.120072
## native.countryOutlying-US(Guam-USVI-etc) -1.296472   1.058938  -1.224 0.220834
## native.countryPeru                       -0.437894   0.598198  -0.732 0.464155
## native.countryPhilippines                 0.211436   0.223673   0.945 0.344510
## native.countryPoland                     -0.116450   0.379349  -0.307 0.758863
## native.countryPortugal                    0.699132   0.393166   1.778 0.075369
## native.countryPuerto-Rico                -0.065760   0.317358  -0.207 0.835845
## native.countryScotland                   -0.213090   0.803605  -0.265 0.790881
## native.countrySouth                      -0.363875   0.345911  -1.052 0.292830
## native.countryTaiwan                      0.094945   0.399799   0.237 0.812283
## native.countryThailand                   -1.221679   0.787921  -1.551 0.121019
## native.countryTrinadad&Tobago            -1.384482   1.155677  -1.198 0.230924
## native.countryUnited-States               0.361032   0.111423   3.240 0.001194
## native.countryVietnam                    -1.112933   0.510498  -2.180 0.029251
## native.countryYugoslavia                  0.246595   0.635931   0.388 0.698186
##
## (Intercept)                              ***
## ageBin 2                                 ***
## ageBin 3                                 ***
## ageBin 4                                 ***
## education11th
## education12th
## education1st-4th
## education5th-6th
## education7th-8th
## education9th
## educationAssoc-acdm                      ***
## educationAssoc-voc                       ***
## educationBachelors                       ***
## educationDoctorate                       ***
## educationHS-grad                         ***
## educationMasters                         ***
## educationPreschool
## educationProf-school                     ***
## educationSome-college                    ***
## workclassFederal-gov                     ***
## workclassLocal-gov                       ***
## workclassNever-worked
## workclassPrivate                         ***
## workclassSelf-emp-inc                    ***
## workclassSelf-emp-not-inc                ***
## workclassState-gov                       **
## workclassWithout-pay
## sexMale                                  ***
## raceAsian-Pac-Islander                   **
## raceBlack
## raceOther
## raceWhite                                **
## native.countryCambodia
## native.countryCanada                     **
## native.countryChina
## native.countryColumbia                   *
## native.countryCuba
## native.countryDominican-Republic
## native.countryEcuador
## native.countryEl-Salvador
## native.countryEngland                    **
## native.countryFrance
## native.countryGermany
## native.countryGreece                     *
## native.countryGuatemala
## native.countryHaiti
## native.countryHonduras
## native.countryHong
## native.countryHungary
## native.countryIndia
## native.countryIran
## native.countryIreland                    *
## native.countryItaly                      **
## native.countryJamaica
## native.countryJapan
## native.countryLaos
## native.countryMexico                     *
## native.countryNicaragua
## native.countryOutlying-US(Guam-USVI-etc)
## native.countryPeru
## native.countryPhilippines
## native.countryPoland
## native.countryPortugal                   .
## native.countryPuerto-Rico
## native.countryScotland
## native.countrySouth
## native.countryTaiwan
## native.countryThailand
## native.countryTrinadad&Tobago
## native.countryUnited-States              **
## native.countryVietnam                    *
## native.countryYugoslavia
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 40316  on 36632  degrees of freedom
## Residual deviance: 30853  on 36561  degrees of freedom
## AIC: 30997
##
## Number of Fisher Scoring iterations: 10
```

Build a confusion matrix for the classifier from (6) and comment on it, e.g., explain what it means.

```
# Predict outcomes on the validation data
probs <- predict(logistic_model, newdata = valid_df, type = "response")

# Convert probabilities into classes
predicted_class <- ifelse(probs > 0.5, ">50K", "<50K")

# Convert the predicted classes to a factor
predicted_class <- as.factor(predicted_class)

# Build confusion matrix
confusion_matrix <- confusionMatrix(predicted_class, valid_df$income)

# Print confusion matrix
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <50K >50K
##       <50K 8734 1890
##       >50K  553 1031
##
##                Accuracy : 0.7999
##                  95% CI : (0.7927, 0.807)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.348
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9405
##             Specificity : 0.3530
##          Pos Pred Value : 0.8221
##          Neg Pred Value : 0.6509
##              Prevalence : 0.7607
##          Detection Rate : 0.7154
##    Detection Prevalence : 0.8702
##       Balanced Accuracy : 0.6467
##
##        'Positive' Class : <50K
##
```

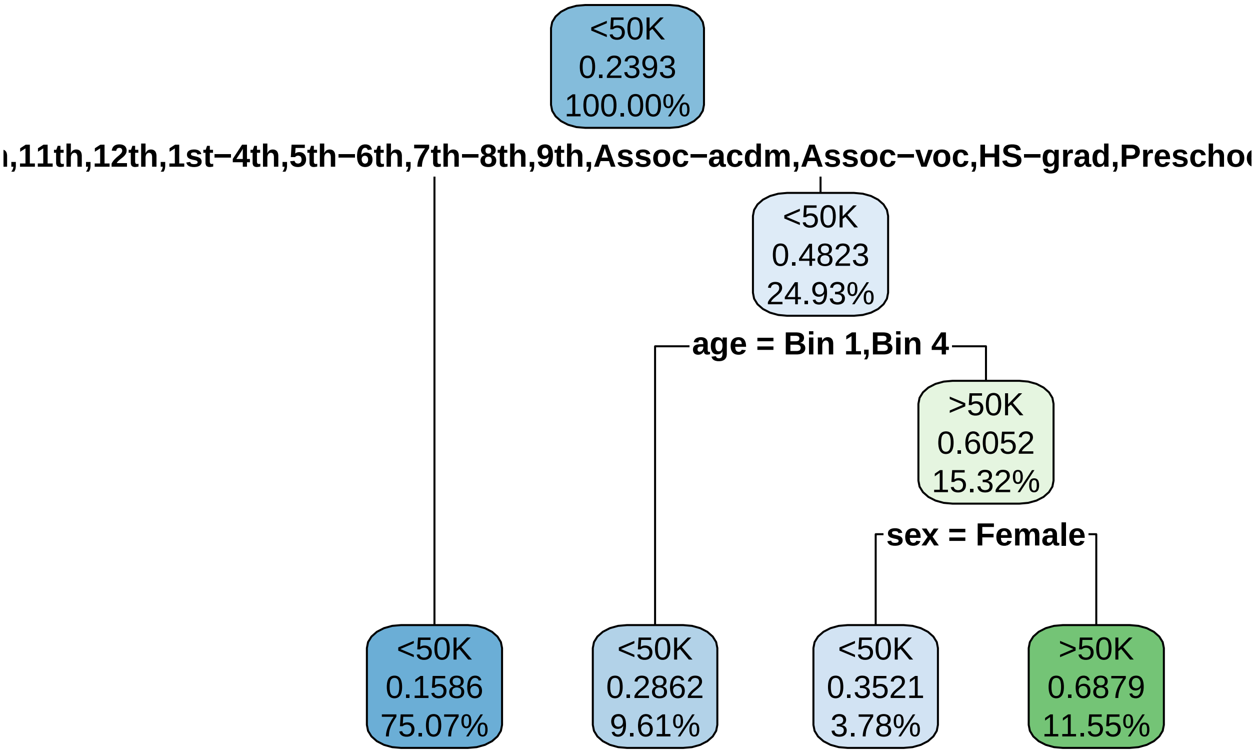
This matrix gives us a picture of how our logistic regression model is performing: 1. Out of all the "<50K" predictions, 8734 were correct, and 1890 were incorrect. This means the model correctly predicted "<50K" 82.21% of the time (the Positive Predictive Value or Precision). 2. Out of all the ">50K" predictions, 1031 were correct, and 553 were incorrect. This means the model correctly predicted ">50K" 65.09% of the time (the Negative Predictive Value). 3. The model correctly predicted the "<50K" class 94.05% of the time when it was indeed "<50K". This is the Sensitivity or True Positive Rate. 4. The model correctly predicted the “>50K” class 35.3% of the time when it was indeed “>50K”. This is the Specificity or True Negative Rate. 5. Overall, the model was correct 79.99% of the time (Accuracy). 6. While the model does a good job of identifying those earning “<50K”, its ability to accurately identify those earning “>50K” (as indicated by the low Specificity) is less reliable. Furthermore, the lower value of Kappa (0.348) than the Naive Bayes model shows that it is less consistent. Also, the P-Value of Mcnemar’s Test being less than 0.05 indicates a significant difference in the misclassification rate of both classes.

Create a Decision Tree model from rpart package, build a classifier that predicts whether an individual earns more than or less than US$50,000. Use the same features as (4). Make sure to transform categorical variables.

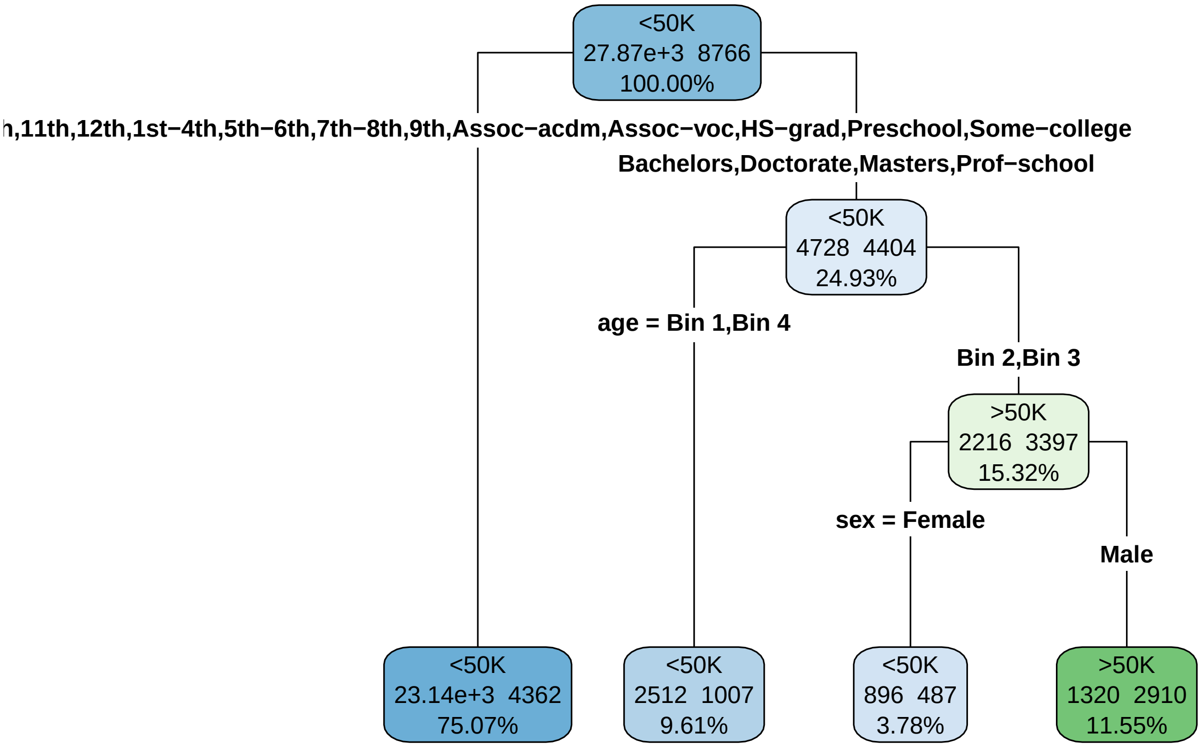
```r
library(rpart)
library(rpart.plot)

# Build the decision tree model
dt_model <- rpart(income ~ ., data=train_df, method="class")

# Print the resulting tree
printcp(dt_model)
```

```
##
## Classification tree:
## rpart(formula = income ~ ., data = train_df, method = "class")
##
## Variables actually used in tree construction:
## [1] age       education sex
##
## Root node error: 8766/36633 = 0.23929
##
## n= 36633
##
##         CP nsplit rel error  xerror      xstd
## 1 0.067363      0   1.00000 1.00000 0.0093155
## 2 0.046658      2   0.86527 0.86527 0.0088470
## 3 0.010000      3   0.81862 0.81862 0.0086656
```

```r
# Plotting the decision tree with default settings
rpart.plot(dt_model, digits=4)
```



```r
# Enhanced decision tree plot with detailed node information
rpart.plot(dt_model, digits = 4, fallen.leaves = TRUE,
           type = 4, extra = 101)
```



Build a confusion matrix for the classifier from (8) and comment on it, e.g., explain what it means.

```r
# Load the caret package for confusionMatrix function
library(caret)

# Make predictions
predictions <- predict(dt_model, valid_df, type="class")

# Evaluate model
confusionMatrix(predictions, valid_df$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <50K >50K
##       <50K 8807 2009
##       >50K  481  912
##
##                Accuracy : 0.7961
##                  95% CI : (0.7888, 0.8032)
##     No Information Rate : 0.7608
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.3173
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9482
##             Specificity : 0.3122
##          Pos Pred Value : 0.8143
##          Neg Pred Value : 0.6547
##              Prevalence : 0.7608
##          Detection Rate : 0.7214
##    Detection Prevalence : 0.8859
##       Balanced Accuracy : 0.6302
##
##        'Positive' Class : <50K
##
```

1. The Decision Tree classifier correctly predicted the income category for 79.61% (accuracy) of individuals in the validation set. It has a high sensitivity (94.82%), indicating it is proficient at correctly identifying individuals who earn less than $50K.
2. However, its specificity is quite low (31.22%), demonstrating that it's not as effective at correctly identifying individuals who earn more than $50K.
3. The Kappa value of 0.3173 suggests that the classifier is significantly better than a random guess.

Build a function called predictEarningsClass() that predicts whether an individual makes more or less than US$50,000 and that combines the three predictive models from (4), (6), and (8) into a simple ensemble and uses majority vote to determine the final prediction using the individual predictions.

```r
predictEarningsClass <- function(input_df, nb_model, logistic_model, dt_model) {

  # Naive Bayes Predictions
  nb_predictions <- as.character(predict(nb_model, input_df)$class)

  # Logistic Regression Predictions
  lr_prob <- predict(logistic_model, newdata = input_df, type = "response")
  lr_predictions <- ifelse(lr_prob > 0.5, ">50K", "<50K")

  # Decision Tree Predictions
  dt_predictions <- as.character(predict(dt_model, input_df, type = "class"))

  # Combine all predictions into a dataframe
  pred_df <- data.frame(NaiveBayes = nb_predictions, LogisticReg = lr_predictions, DecisionTree = dt_pre

  # Use majority voting to determine the final prediction
  final_prediction <- apply(pred_df, 1, function(x) {
    if(sum(x == "<50K") > sum(x == ">50K")) {
      return("<50K")
    } else {
      return(">50K")
    }
  })

  return(final_prediction)
}
```

Using the ensemble model from (10), predict whether a 29-year-old white female adult who is a local government worker, never married and with a Bacherlor's degree who immigrated from France earns more or less than US$50,000.

```r
individual <- data.frame(
  age = factor("Bin 2", levels = levels(train_df$age)),
  education = factor("Bachelors", levels = levels(train_df$education)),
  workclass = factor("Local-gov", levels = levels(train_df$workclass)),
  sex = factor("Female", levels = levels(train_df$sex)),
  race = factor("White", levels = levels(train_df$race)),
  native.country = factor("France", levels = levels(train_df$`native.country`))
)

prediction <- predictEarningsClass(individual, nb_model, logistic_model, dt_model)
print(prediction)
```

```
##      1
## "<50K"
```

Based on the majority voting of the three models (Naive Bayes, Logistic Regression, and Decision Tree), the function predictEarningsClass() predicts that a 29-year-old white female adult, who is a local government worker, never married, has a Bachelor's degree, and who immigrated from France is likely to earn less than US$50,000.

Compare the decision tree of the rpart package with C5.0 package which is used in the book? Which one do you prefer and why?

1. Both C5.0 and rpart are decision tree-based algorithms available in R, used for performing supervised machine learning tasks. The key objective of these algorithms is to use certain metrics to establish the best split rule for the data at every node, to achieve the purest possible class segregation.

2. C5.0 operates on the concept of entropy computation for making split decisions, while rpart utilizes the Gini coefficient as its metric. The ultimate output is a decision tree, or a set of rules, which is then utilized for class prediction on new, unseen data.

3. C5.0 is appreciated for its computational efficiency and robustness, whereas rpart, despite being relatively easy to compute, can sometimes present computational challenges.

4. The selection between these two algorithms generally depends on the specifics of the data in hand, as well as the computational constraints associated with the problem. However, it's important to remember that different algorithms have different strengths, and choosing the best one should be based on a combination of empirical results and problem requirements.

Problem 2

Download the Energy Efficiency dataset from UCI repository Links to an external site.. Load the dataset into an R dataframe and call it energy.df.

```r
# Load the library
library(readxl)

# Read the excel file
energy.df <- read_excel('/Users/anayathawale/Downloads/energydata.xlsx')


colnames(energy.df) <- c("Relative Compactness", "Surface Area", "Wall Area", "Roof Area",
                         "Overall Height", "Orientation", "Glazing Area",
                         "Glazing Area Distribution", "Heating Load", "Cooling Load")

# This will print the first 6 rows of your dataframe
head(energy.df)
```

```
## # A tibble: 6 x 10
##   Relative Com~1 Surfa~2 Wall ~3 Roof ~4 Overa~5 Orien~6 Glazi~7 Glazi~8 Heati~9
##            <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1           0.98    514.     294    110.       7       2       0       0    15.6
## 2           0.98    514.     294    110.       7       3       0       0    15.6
## 3           0.98    514.     294    110.       7       4       0       0    15.6
## 4           0.98    514.     294    110.       7       5       0       0    15.6
## 5           0.9     564.    318.    122.       7       2       0       0    20.8
## 6           0.9     564.    318.    122.       7       3       0       0    21.5
## # ... with 1 more variable: ‘Cooling Load‘ <dbl>, and abbreviated variable
## #   names 1: ‘Relative Compactness‘, 2: ‘Surface Area‘, 3: ‘Wall Area‘,
## #   4: ‘Roof Area‘, 5: ‘Overall Height‘, 6: Orientation, 7: ‘Glazing Area‘,
## #   8: ‘Glazing Area Distribution‘, 9: ‘Heating Load‘
```

The heating load is the amount of heat energy that would need to be added to a space to maintain the temperature in an acceptable range. The cooling load is the amount of heat energy that would need to be removed from a space (cooling) to maintain the temperature in an acceptable range. In this exercise we are focusing on predicting the heating load as a function of building parameters. Please remove the Cooling Load column for the rest of this exercise.

```r
library(dplyr)
```

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

## The following object is masked from 'package:MASS':
##
##     select
## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```r
# Remove 'Cooling Load' column using subset
energy.df <- energy.df[,-10]
```

Are there outliers in any one of the features in the data set? How do you identify outliers? Remove them but create a second data set with outliers removed called energy.no.df. Keep the original data set energy.df.

```r
# Create a copy of the original dataframe
energy.no.df <- energy.df

# Initialize the counter
outlier_count <- 0

# Loop through each column in the dataframe
for(col in colnames(energy.df)){
  if(is.numeric(energy.df[[col]])){  # Check if the column is numeric
    # Calculate z-scores
    z_scores <- abs(scale(energy.df[[col]]))

    # Identify outliers
    outliers <- z_scores > 3

    # Add the number of outliers in this column to the counter
    outlier_count <- outlier_count + sum(outliers)

    # Replace outliers with NA
    energy.no.df[[col]][outliers] <- NA
  }
}

# Remove rows with NA values
energy.no.df <- na.omit(energy.no.df)

# Print the total number of outliers
print(paste("Total number of outliers: ", outlier_count))
```

```
## [1] "Total number of outliers:  0"
```

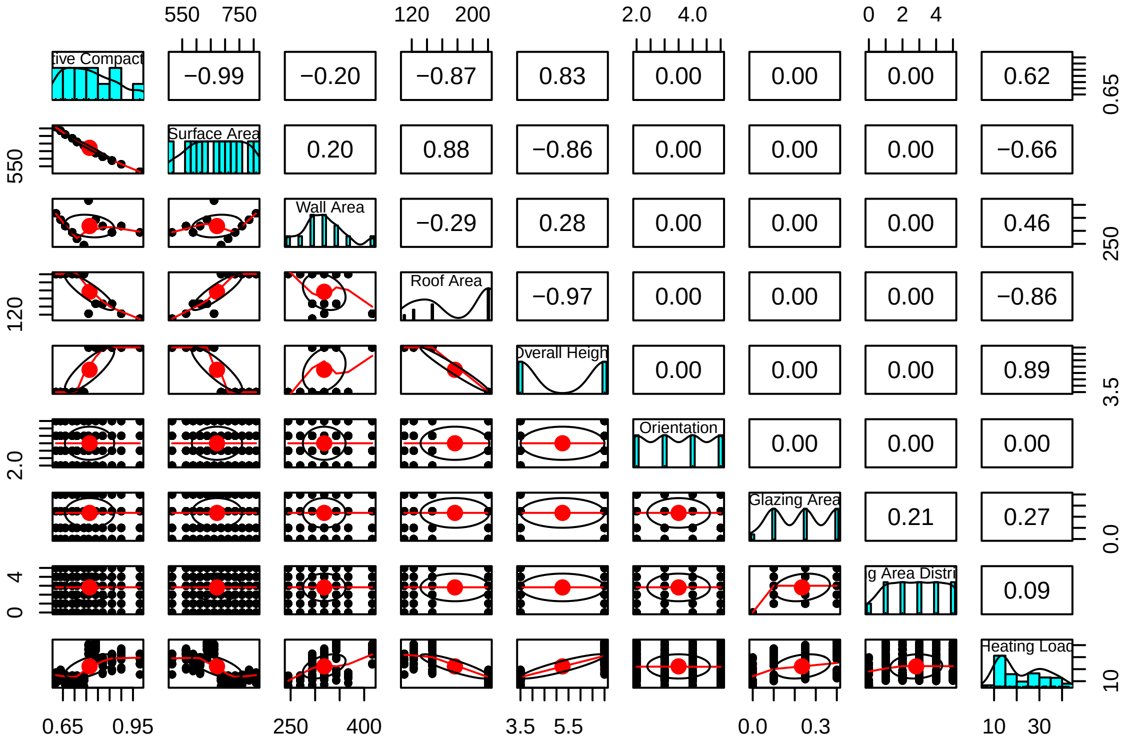
1. In this code, we calculate the Z-scores using the scale() function. Then, any data point that is three standard deviations away from the mean (having a Z-score greater than 3 or less than -3) is considered an outlier. The total number of outliers across all numeric columns is 0.

Using pairs.panel, what are the distributions of each of the features in the data set with outliers removed (energy.no.df)? Are they reasonably normal so you can apply a statistical learner such as regression? Can you normalize features through a log, inverse, or square-root transform? State which features should be transformed and then transform as needed and build a new data set, energy.tx.

```
library(psych)
```

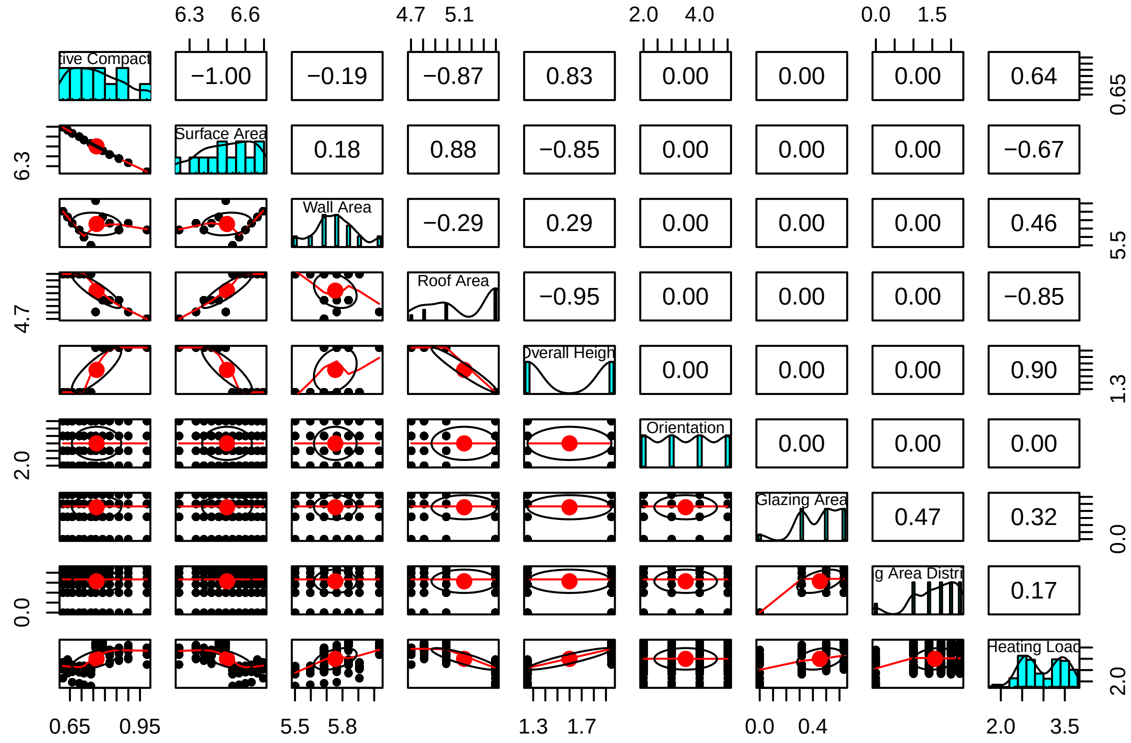
```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
# Generate a scatterplot matrix
pairs.panels(energy.no.df)
```



```
# Copy the dataframe with colnames
col_names <- c("Relative Compactness", "Surface Area",  "Roof Area", "Overall Height",  "Orientation",
energy.tx <- energy.no.df[,col_names]

# Transforming variables as needed
```

```r
energy.tx <- energy.no.df
energy.tx$`Surface Area` <- log(energy.no.df$`Surface Area`)
energy.tx$`Wall Area` <- log(energy.no.df$`Wall Area`)
energy.tx$`Roof Area` <- log(energy.no.df$`Roof Area`)
energy.tx$`Overall Height` <- log(energy.no.df$`Overall Height`)
energy.tx$`Glazing Area` <- sqrt(energy.no.df$`Glazing Area`)
energy.tx$`Glazing Area Distribution` <- sqrt(energy.no.df$`Glazing Area Distribution`)
energy.tx$`Heating Load` <- log(energy.no.df$`Heating Load`)

# Checking the distribution of selected columns
pairs.panels(energy.tx)
```



What are the correlations to the response variable (Heating Load) for energy.no.df? Are there collinearities? Build a full correlation matrix.

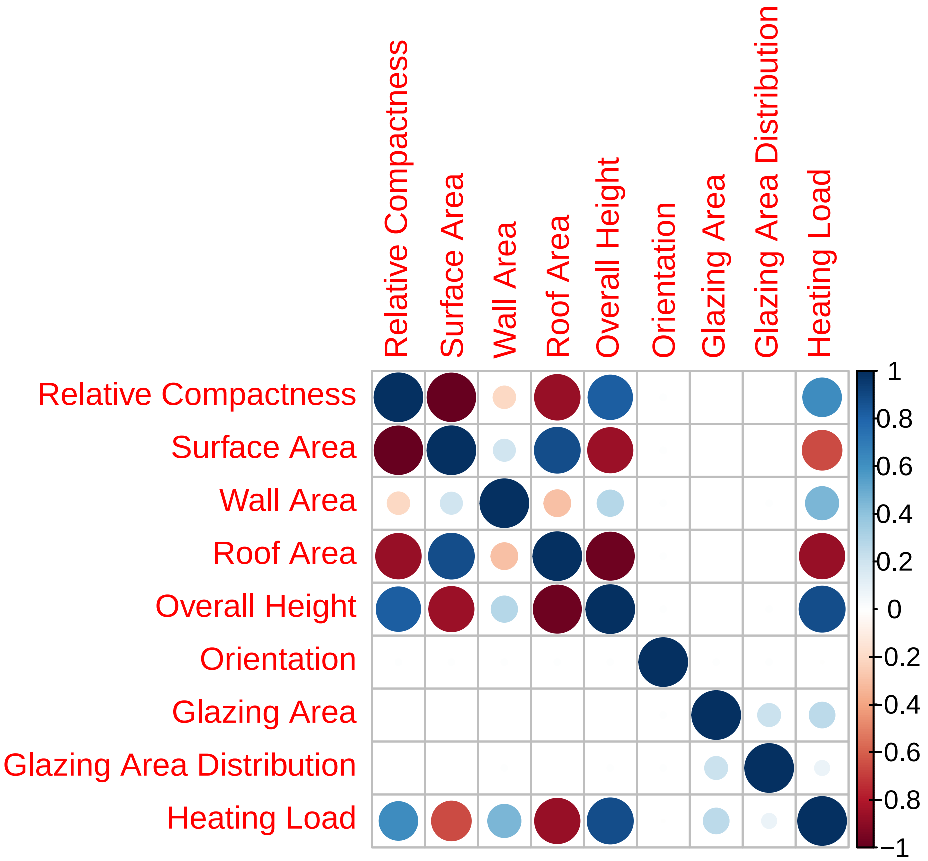
```r
# Calculate correlation matrix
cor_matrix <- cor(energy.no.df, use="complete.obs", method="pearson")

# Print correlation matrix
print(cor_matrix)
```

```
##                           Relative Compactness  Surface Area     Wall Area
## Relative Compactness              1.000000e+00 -9.919015e-01 -2.037817e-01
## Surface Area                     -9.919015e-01  1.000000e+00  1.955016e-01
## Wall Area                        -2.037817e-01  1.955016e-01  1.000000e+00
## Roof Area                        -8.688234e-01  8.807195e-01 -2.923165e-01
## Overall Height                    8.277473e-01 -8.581477e-01  2.809757e-01
## Orientation                       0.000000e+00  0.000000e+00  0.000000e+00
## Glazing Area                     -4.558152e-17 -8.841667e-17 -8.766699e-18
## Glazing Area Distribution         5.735014e-17  5.641703e-17  0.000000e+00
## Heating Load                      6.222719e-01 -6.581199e-01  4.556714e-01
##                               Roof Area Overall Height  Orientation
## Relative Compactness      -8.688234e-01   8.277473e-01  0.000000000
## Surface Area               8.807195e-01  -8.581477e-01  0.000000000
## Wall Area                 -2.923165e-01   2.809757e-01  0.000000000
## Roof Area                  1.000000e+00  -9.725122e-01  0.000000000
## Overall Height            -9.725122e-01   1.000000e+00  0.000000000
## Orientation                0.000000e+00   0.000000e+00  1.000000000
## Glazing Area              -5.427147e-17  -1.861418e-18  0.000000000
## Glazing Area Distribution -8.675350e-17   0.000000e+00  0.000000000
## Heating Load              -8.618281e-01   8.894305e-01 -0.002586763
##                            Glazing Area Glazing Area Distribution Heating Load
## Relative Compactness      -4.558152e-17              5.735014e-17  0.622271936
## Surface Area              -8.841667e-17              5.641703e-17 -0.658119917
## Wall Area                 -8.766699e-18              0.000000e+00  0.455671365
## Roof Area                 -5.427147e-17             -8.675350e-17 -0.861828052
## Overall Height            -1.861418e-18              0.000000e+00  0.889430464
## Orientation                0.000000e+00              0.000000e+00 -0.002586763
## Glazing Area               1.000000e+00              2.129642e-01  0.269841685
## Glazing Area Distribution  2.129642e-01              1.000000e+00  0.087368460
## Heating Load               2.698417e-01              8.736846e-02  1.000000000
```

```
# For better visualization
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor_matrix, method="circle")
```



From this correlation matrix, it's clear that: 1. "Relative Compactness" and "Surface Area" are highly negatively correlated (-0.9919), which indicates high multicollinearity. Similarly, "Roof Area" and "Overall Height" are also highly negatively correlated (-0.9725). 2. "Heating Load" is highly positively correlated with "Relative Compactness" (0.6222) and "Overall Height" (0.8894), and highly negatively correlated with "Surface Area" (-0.6581) and "Roof Area" (-0.8618). 3. There is no correlation between "Orientation" and any other variables. 4. "Glazing Area" shows very low correlation with the other variables. 5. "Glazing Area Distribution" has a weak positive correlation with "Heating Load" (0.0874), indicating that as the Glazing Area Distribution increases, the Heating Load also tends to increase slightly.

Split each of the three data sets, energy.no.df, energy.df, and energy.tx into 70%/30% subsets, so you retain 30% for testing using random sampling without replacement. Call the data sets, energy.training and energy.testing, energy.no.training and energy.no.testing, and energy.tx.training and energy.tx.testing.

```r
# Load necessary library
library(caTools)

# Set a random seed for reproducibility
set.seed(123)

# Split for energy.df
split <- sample.split(energy.df$'Heating Load', SplitRatio = 0.7)
energy.training <- subset(energy.df, split==TRUE)
energy.testing <- subset(energy.df, split==FALSE)

# Split for energy.no.df
split <- sample.split(energy.no.df$'Heating Load', SplitRatio = 0.7)
energy.no.training <- subset(energy.no.df, split==TRUE)
energy.no.testing <- subset(energy.no.df, split==FALSE)

# Split for energy.tx
split <- sample.split(energy.tx$'Heating Load', SplitRatio = 0.7)
energy.tx.training <- subset(energy.tx, split==TRUE)
energy.tx.testing <- subset(energy.tx, split==FALSE)
```

Build three Multiple Regression models for energy.training, energy.no.training, and energy.tx.training using backward elimination based on p-value for predicting Heating Load.

```r
set.seed(123)
library(caret)

# Model for energy.training
lr_model1 <- lm(`Heating Load` ~ ., data = energy.training)
lr_model1 <- step(lr_model1, direction = "backward", trace = FALSE)
prediction_1 <- predict(lr_model1, energy.testing)
ad_r_sq_1 <- summary(lr_model1)$adj.r.squared
rmse_test1 <- RMSE(prediction_1, energy.testing$`Heating Load`)

# Model for energy.no.training
lr_model2 <- lm(`Heating Load` ~ ., data = energy.no.training)
lr_model2 <- step(lr_model2, direction = "backward", trace = FALSE)
prediction_2 <- predict(lr_model2, energy.no.testing)
ad_r_sq_2 <- summary(lr_model2)$adj.r.squared
rmse_test2 <- RMSE(prediction_2, energy.no.testing$`Heating Load`)

# Model for energy.tx.training
# Remove rows with missing or invalid values
energy.tx.training <- na.omit(energy.tx.training)
energy.tx.testing <- na.omit(energy.tx.testing)
lr_model3 <- lm(`Heating Load` ~ ., data = energy.tx.training)
lr_model3 <- step(lr_model3, direction = "backward", trace = FALSE)
prediction_train3 <- predict(lr_model3, energy.tx.training)
prediction_test3 <- predict(lr_model3, energy.tx.testing)

# check for missing or invalid values in predicted values
if (anyNA(prediction_train3) || anyNA(prediction_test3) || !is.numeric(prediction_train3) || !is.numeri
  stop("Invalid predicted values")
}

# adjusted r squared value
ad_r_sq_train3 <- 1 - ((1 - summary(lr_model3)$r.squared) * (nrow(energy.tx.training) - 1) / (nrow(ener
ad_r_sq_test3 <-  1 - ((1 - summary(lr_model3)$r.squared) * (nrow(energy.tx.testing) - 1) / (nrow(energ
# calculate RMSE values
rmse_train3 <- RMSE(prediction_train3, energy.tx.training$`Heating Load`)
rmse_test3 <- RMSE(prediction_test3, energy.tx.testing$`Heating Load`)

# Print results
print(paste("Adjusted R-squared for Model 1: ", ad_r_sq_1))
```

```
## [1] "Adjusted R-squared for Model 1:  0.920340277931368"
```

```r
print(paste("Root Mean Squared Error for Model 1: ", rmse_test1))
```

```
## [1] "Root Mean Squared Error for Model 1:  3.10205803284087"
```

```r
print(paste("Adjusted R-squared for Model 2: ", ad_r_sq_2))
```

```
## [1] "Adjusted R-squared for Model 2:  0.918430674839147"
```

```r
print(paste("Root Mean Squared Error for Model 2: ", rmse_test2))
```

```
## [1] "Root Mean Squared Error for Model 2:  3.10566838202208"
```

```r
print(paste("Adjusted R-squared for Model 3 (Train): ", ad_r_sq_train3))
```

```
## [1] "Adjusted R-squared for Model 3 (Train):  0.949864133568808"
```

```r
print(paste("Adjusted R-squared for Model 3 (Test): ", ad_r_sq_test3))
```

```
## [1] "Adjusted R-squared for Model 3 (Test):  0.948698517845236"
```

```r
print(paste("Root Mean Squared Error for Model 3 (Train): ", rmse_train3))
```

```
## [1] "Root Mean Squared Error for Model 3 (Train):  0.105491083510637"
```

```r
print(paste("Root Mean Squared Error for Model 3 (Test): ", rmse_test3))
```

```
## [1] "Root Mean Squared Error for Model 3 (Test):  0.0960028006949282"
```

Build three Regression Tree models using the rpart package for predicting Heating Load: one with energy.training, one with energy.no.training, and one with energy.tx.training.

```r
# Load the necessary library
library(rpart)

# Set a seed for reproducibility
set.seed(123)

# Build the model for energy.training
tree_model1 <- rpart(`Heating Load` ~ ., data = energy.training, method = "anova")
print(summary(tree_model1))
```

```
## Call:
## rpart(formula = ‘Heating Load‘ ~ ., data = energy.training, method = "anova")
##   n= 537
##
##           CP nsplit  rel error     xerror        xstd
## 1 0.79706652      0 1.00000000 1.00227616 0.038230409
## 2 0.09025439      1 0.20293348 0.20383102 0.014316982
## 3 0.02709443      2 0.11267909 0.11387273 0.008846634
## 4 0.01256719      3 0.08558466 0.08675909 0.006688519
## 5 0.01220655      4 0.07301747 0.08017820 0.006383008
## 6 0.01000000      5 0.06081092 0.06688265 0.004803697
##
## Variable importance
## Relative Compactness         Surface Area       Overall Height
##                   24                   24                   21
##            Roof Area            Wall Area         Glazing Area
##                   21                    9                    1
##
## Node number 1: 537 observations,    complexity param=0.7970665
##   mean=21.98939, MSE=102.5617
##   left son=2 (278 obs) right son=3 (259 obs)
##   Primary splits:
##       Overall Height       < 5.25   to the left,  improve=0.7970665, (0 missing)
##       Relative Compactness < 0.75   to the left,  improve=0.7970665, (0 missing)
##       Roof Area            < 183.75 to the right, improve=0.7970665, (0 missing)
##       Surface Area         < 673.75 to the right, improve=0.7970665, (0 missing)
##       Wall Area            < 281.75 to the left,  improve=0.2070272, (0 missing)
##   Surrogate splits:
##       Relative Compactness < 0.75   to the left,  agree=1.000, adj=1.00, (0 split)
##       Surface Area         < 673.75 to the right, agree=1.000, adj=1.00, (0 split)
##       Roof Area            < 183.75 to the right, agree=1.000, adj=1.00, (0 split)
##       Wall Area            < 281.75 to the left,  agree=0.657, adj=0.29, (0 split)
##
## Node number 2: 278 observations,    complexity param=0.01256719
##   mean=13.26234, MSE=6.869463
##   left son=4 (108 obs) right son=5 (170 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.3624351, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3114269, (0 missing)
##       Relative Compactness      < 0.65   to the right, improve=0.2702899, (0 missing)
##       Surface Area              < 771.75 to the left,  improve=0.2702899, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.2702899, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.68, adj=0.176, (0 split)
##
## Node number 3: 259 observations,    complexity param=0.09025439
##   mean=31.35664, MSE=35.77984
##   left son=6 (168 obs) right son=7 (91 obs)
##   Primary splits:
##       Relative Compactness      < 0.805  to the right, improve=0.5364011, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.5364011, (0 missing)
##       Surface Area              < 624.75 to the left,  improve=0.5364011, (0 missing)
##       Glazing Area              < 0.175  to the left,  improve=0.2833498, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.1744826, (0 missing)
##   Surrogate splits:
##       Surface Area < 624.75 to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 330.75 to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 108 observations
##   mean=11.28269, MSE=5.362688
##
## Node number 5: 170 observations
##   mean=14.52, MSE=3.75526
##
## Node number 6: 168 observations,    complexity param=0.02709443
##   mean=28.13238, MSE=18.02454
##   left son=12 (69 obs) right son=13 (99 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.4927946, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3433437, (0 missing)
##       Surface Area              < 600.25 to the right, improve=0.1588406, (0 missing)
##       Relative Compactness      < 0.84   to the left,  improve=0.1588406, (0 missing)
##       Roof Area                 < 134.75 to the right, improve=0.1129024, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.649, adj=0.145, (0 split)
##
## Node number 7: 91 observations,    complexity param=0.01220655
##   mean=37.30912, MSE=13.93448
##   left son=14 (31 obs) right son=15 (60 obs)
##   Primary splits:
##       Glazing Area         < 0.175  to the left,  improve=0.5301760, (0 missing)
##       Surface Area         < 649.25 to the right, improve=0.0828404, (0 missing)
##       Relative Compactness < 0.775  to the left,  improve=0.0828404, (0 missing)
##       Wall Area            < 379.75 to the right, improve=0.0828404, (0 missing)
##       Roof Area            < 134.75 to the left,  improve=0.0828404, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.714, adj=0.161, (0 split)
##
## Node number 12: 69 observations
##   mean=24.56246, MSE=11.18535
##
## Node number 13: 99 observations
##   mean=30.62051, MSE=7.718092
##
## Node number 14: 31 observations
##   mean=33.52774, MSE=11.35595
##
## Node number 15: 60 observations
##   mean=39.26283, MSE=4.062004
##
## n= 537
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 537 55075.6200 21.98939
##    2) Overall Height< 5.25 278  1909.7110 13.26234
##      4) Glazing Area< 0.175 108   579.1703 11.28269 *
##      5) Glazing Area>=0.175 170   638.3942 14.52000 *
##    3) Overall Height>=5.25 259  9266.9770 31.35664  
##      6) Relative Compactness>=0.805 168  3028.1230 28.13238  
##       12) Glazing Area< 0.175 69   771.7891 24.56246 *
##       13) Glazing Area>=0.175 99   764.0911 30.62051 *
##      7) Relative Compactness< 0.805 91  1268.0380 37.30912  
##       14) Glazing Area< 0.175 31   352.0343 33.52774 *
##       15) Glazing Area>=0.175 60   243.7202 39.26283 *
```

```r
# Build the model for energy.no.training
tree_model2 <- rpart(`Heating Load` ~ ., data = energy.no.training, method = "anova")
print(summary(tree_model2))
```

```
## Call:
## rpart(formula = `Heating Load` ~ ., data = energy.no.training, 
##     method = "anova")
##   n= 537 
## 
##           CP nsplit  rel error     xerror        xstd
## 1 0.79642974      0 1.00000000 1.00471784 0.037444212
## 2 0.08124859      1 0.20357026 0.20464162 0.014582271
## 3 0.02884783      2 0.12232167 0.12346669 0.009344940
## 4 0.01415732      3 0.09347385 0.09494496 0.006903003
## 5 0.01232923      4 0.07931653 0.08160377 0.006439624
## 6 0.01000000      5 0.06698729 0.07515352 0.006143695
## 
## Variable importance
## Relative Compactness         Surface Area       Overall Height 
##                   23                   23                   21 
##            Roof Area            Wall Area         Glazing Area 
##                   21                   10                    1 
##          Orientation 
##                    1 
## 
## Node number 1: 537 observations,    complexity param=0.7964297
##   mean=22.1421, MSE=100.2455 
##   left son=2 (267 obs) right son=3 (270 obs)
##   Primary splits:
##       Relative Compactness < 0.75   to the left,  improve=0.7964297, (0 missing)
##       Overall Height       < 5.25   to the left,  improve=0.7964297, (0 missing)
##       Surface Area         < 673.75 to the right, improve=0.7964297, (0 missing)
##       Roof Area            < 183.75 to the right, improve=0.7964297, (0 missing)
##       Wall Area            < 281.75 to the left,  improve=0.2223079, (0 missing)
##   Surrogate splits:
##       Surface Area   < 673.75 to the right, agree=1.000, adj=1.000, (0 split)
##       Roof Area      < 183.75 to the right, agree=1.000, adj=1.000, (0 split)
##       Overall Height < 5.25   to the left,  agree=1.000, adj=1.000, (0 split)
##       Wall Area      < 281.75 to the left,  agree=0.678, adj=0.352, (0 split)
##       Orientation    < 3.5    to the left,  agree=0.518, adj=0.030, (0 split)
## 
## Node number 2: 267 observations,    complexity param=0.01415732
##   mean=13.1568, MSE=7.002004 
##   left son=4 (102 obs) right son=5 (165 obs)
##   Primary splits:
```

```
##       Glazing Area              < 0.175  to the left,  improve=0.4076491, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3588154, (0 missing)
##       Relative Compactness      < 0.65   to the right, improve=0.3095923, (0 missing)
##       Surface Area              < 771.75 to the left,  improve=0.3095923, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.3095923, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.685, adj=0.176, (0 split)
##
## Node number 3: 270 observations,    complexity param=0.08124859
##   mean=31.02756, MSE=33.66304
##   left son=6 (186 obs) right son=7 (84 obs)
##   Primary splits:
##       Relative Compactness      < 0.805  to the right, improve=0.4812134, (0 missing)
##       Surface Area              < 624.75 to the left,  improve=0.4812134, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.4812134, (0 missing)
##       Glazing Area              < 0.175  to the left,  improve=0.2513790, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.1833111, (0 missing)
##   Surrogate splits:
##       Surface Area < 624.75 to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 330.75 to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 102 observations
##   mean=11.008, MSE=5.165217
##
## Node number 5: 165 observations
##   mean=14.48515, MSE=3.518598
##
## Node number 6: 186 observations,    complexity param=0.02884783
##   mean=28.3228, MSE=18.75478
##   left son=12 (73 obs) right son=13 (113 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.44517130, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.33790500, (0 missing)
##       Surface Area              < 600.25 to the right, improve=0.15382430, (0 missing)
##       Relative Compactness      < 0.84   to the left,  improve=0.15382430, (0 missing)
##       Roof Area                 < 134.75 to the right, improve=0.07795348, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.667, adj=0.151, (0 split)
##
## Node number 7: 84 observations,    complexity param=0.01232923
##   mean=37.01667, MSE=14.60563
##   left son=14 (32 obs) right son=15 (52 obs)
##   Primary splits:
##       Glazing Area         < 0.175  to the left,  improve=0.54097290, (0 missing)
##       Surface Area         < 649.25 to the right, improve=0.09269237, (0 missing)
##       Relative Compactness < 0.775  to the left,  improve=0.09269237, (0 missing)
##       Roof Area            < 134.75 to the left,  improve=0.09269237, (0 missing)
##       Wall Area            < 379.75 to the right, improve=0.09269237, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.679, adj=0.156, (0 split)
##
## Node number 12: 73 observations
##   mean=24.72781, MSE=12.00401
##
```

```
## Node number 13: 113 observations
##   mean=30.64522, MSE=9.373152
##
## Node number 14: 32 observations
##   mean=33.43344, MSE=10.48016
##
## Node number 15: 52 observations
##   mean=39.22173, MSE=4.380822
##
## n= 537
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 537 53831.8200 22.14210
##    2) Relative Compactness< 0.75 267  1869.5350 13.15680
##      4) Glazing Area< 0.175 102   526.8521 11.00800 *
##      5) Glazing Area>=0.175 165   580.5687 14.48515 *
##    3) Relative Compactness>=0.75 270  9089.0220 31.02756
##      6) Relative Compactness>=0.805 186  3488.3900 28.32280
##       12) Glazing Area< 0.175 73   876.2926 24.72781 *
##       13) Glazing Area>=0.175 113  1059.1660 30.64522 *
##      7) Relative Compactness< 0.805 84  1226.8730 37.01667
##       14) Glazing Area< 0.175 32   335.3651 33.43344 *
##       15) Glazing Area>=0.175 52   227.8027 39.22173 *
```

```
# Build the model for energy.tx.training
# Make sure to handle missing values as rpart() does not handle them automatically
energy.tx.training <- na.omit(energy.tx.training)
tree_model3 <- rpart(`Heating Load` ~ ., data = energy.tx.training, method = "anova")
print(summary(tree_model3))
```

```
## Call:
## rpart(formula = ‘Heating Load‘ ~ ., data = energy.tx.training,
##     method = "anova")
##   n= 537
##
##           CP nsplit  rel error     xerror        xstd
## 1 0.81214111      0 1.00000000 1.00099705 0.038907023
## 2 0.04277249      1 0.18785889 0.18986262 0.015602827
## 3 0.03630434      2 0.14508641 0.16266787 0.012927902
## 4 0.02119471      3 0.10878207 0.11116889 0.008015748
## 5 0.02047113      4 0.08758736 0.09787175 0.007603611
## 6 0.01313718      5 0.06711623 0.07537215 0.005414681
## 7 0.01000000      6 0.05397906 0.05595452 0.003889504
##
## Variable importance
##      Relative Compactness              Surface Area            Overall Height
##                        23                        23                        21
##                 Roof Area                 Wall Area              Glazing Area
##                        21                         8                         2
## Glazing Area Distribution               Orientation
##                         2                         1
##
```

```
## Node number 1: 537 observations,    complexity param=0.8121411
##   mean=2.992702, MSE=0.2257549
##   left son=2 (270 obs) right son=3 (267 obs)
##   Primary splits:
##       Relative Compactness < 0.75      to the left,  improve=0.8121411, (0 missing)
##       Roof Area            < 5.193165  to the right, improve=0.8121411, (0 missing)
##       Overall Height       < 1.599337  to the left,  improve=0.8121411, (0 missing)
##       Surface Area         < 6.512694  to the right, improve=0.8121411, (0 missing)
##       Wall Area            < 5.640074  to the left,  improve=0.2229821, (0 missing)
##   Surrogate splits:
##       Surface Area   < 6.512694  to the right, agree=1.000, adj=1.000, (0 split)
##       Roof Area      < 5.193165  to the right, agree=1.000, adj=1.000, (0 split)
##       Overall Height < 1.599337  to the left,  agree=1.000, adj=1.000, (0 split)
##       Wall Area      < 5.640074  to the left,  agree=0.654, adj=0.303, (0 split)
##       Orientation    < 3.5       to the right, agree=0.523, adj=0.041, (0 split)
##
## Node number 2: 270 observations,    complexity param=0.04277249
##   mean=2.566899, MSE=0.0448872
##   left son=4 (18 obs) right son=5 (252 obs)
##   Primary splits:
##       Glazing Area              < 0.1581139 to the left,  improve=0.4278482, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.4278482, (0 missing)
##       Relative Compactness      < 0.65      to the right, improve=0.2163006, (0 missing)
##       Surface Area              < 6.648535  to the left,  improve=0.2163006, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.2163006, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5       to the left,  agree=1, adj=1, (0 split)
##
## Node number 3: 267 observations,    complexity param=0.03630434
##   mean=3.423288, MSE=0.0399051
##   left son=6 (178 obs) right son=7 (89 obs)
##   Primary splits:
##       Relative Compactness      < 0.805     to the right, improve=0.4130764, (0 missing)
##       Surface Area              < 6.437159  to the left,  improve=0.4130764, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.4130764, (0 missing)
##       Glazing Area              < 0.1581139 to the left,  improve=0.2781648, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.2781648, (0 missing)
##   Surrogate splits:
##       Surface Area < 6.437159  to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 5.800676  to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 18 observations
##   mean=2.048374, MSE=0.04165259
##
## Node number 5: 252 observations,    complexity param=0.02119471
##   mean=2.603937, MSE=0.02454156
##   left son=10 (168 obs) right son=11 (84 obs)
##   Primary splits:
##       Relative Compactness      < 0.65      to the right, improve=0.415466700, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.415466700, (0 missing)
##       Surface Area              < 6.648535  to the left,  improve=0.415466700, (0 missing)
##       Glazing Area              < 0.5662278 to the left,  improve=0.305767300, (0 missing)
##       Glazing Area Distribution < 1.573132  to the right, improve=0.007198365, (0 missing)
##   Surrogate splits:
```

```
##       Surface Area < 6.648535  to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 5.800676  to the left,  agree=1, adj=1, (0 split)
##
## Node number 6: 178 observations,    complexity param=0.02047113
##   mean=3.332503, MSE=0.0278924
##   left son=12 (12 obs) right son=13 (166 obs)
##   Primary splits:
##       Glazing Area              < 0.1581139 to the left,  improve=0.49985870, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.49985870, (0 missing)
##       Relative Compactness      < 0.84      to the left,  improve=0.08599162, (0 missing)
##       Surface Area              < 6.397138  to the right, improve=0.08599162, (0 missing)
##       Roof Area                 < 4.899272  to the right, improve=0.04170109, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5       to the left,  agree=1, adj=1, (0 split)
##
## Node number 7: 89 observations
##   mean=3.604858, MSE=0.01447892
##
## Node number 10: 168 observations,    complexity param=0.01313718
##   mean=2.532536, MSE=0.01407968
##   left son=20 (112 obs) right son=21 (56 obs)
##   Primary splits:
##       Glazing Area              < 0.5662278 to the left,  improve=0.673304200, (0 missing)
##       Surface Area              < 6.616265  to the left,  improve=0.026341050, (0 missing)
##       Relative Compactness      < 0.675     to the right, improve=0.026341050, (0 missing)
##       Wall Area                 < 5.723601  to the left,  improve=0.026341050, (0 missing)
##       Glazing Area Distribution < 1.573132  to the right, improve=0.006295481, (0 missing)
##
## Node number 11: 84 observations
##   mean=2.746739, MSE=0.01487671
##
## Node number 12: 12 observations
##   mean=2.893336, MSE=0.01272934
##
## Node number 13: 166 observations
##   mean=3.36425, MSE=0.01403839
##
## Node number 20: 112 observations
##   mean=2.463688, MSE=0.00588785
##
## Node number 21: 56 observations
##   mean=2.67023, MSE=0.00202362
##
## n= 537
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 537 121.2304000 2.992702
##    2) Relative Compactness< 0.75 270  12.1195400 2.566899
##      4) Glazing Area< 0.1581139 18   0.7497467 2.048374 *
##      5) Glazing Area>=0.1581139 252   6.1844720 2.603937
##       10) Relative Compactness>=0.65 168   2.3653870 2.532536
##         20) Glazing Area< 0.5662278 112   0.6594392 2.463688 *
##         21) Glazing Area>=0.5662278 56   0.1133227 2.670230 *
##       11) Relative Compactness< 0.65 84   1.2496430 2.746739 *
##    3) Relative Compactness>=0.75 267  10.6546600 3.423288
##      6) Relative Compactness>=0.805 178   4.9648480 3.332503
##       12) Glazing Area< 0.1581139 12   0.1527521 2.893336 *
##       13) Glazing Area>=0.1581139 166   2.3303730 3.364250 *
##      7) Relative Compactness< 0.805 89   1.2886240 3.604858 *
```

Provide an analysis of all 6 models (using their respective testing data sets), including Adjusted R-Squared and RMSE. Which of these models is the best? Why?

```
# Multiple regression models

# Predict on testing set
predict1 <- predict(lr_model1, newdata=energy.testing)
predict2 <- predict(lr_model2, newdata=energy.no.testing)
predict3 <- predict(lr_model3, newdata=energy.tx.testing)

# Compute RMSE
rmse1 <- RMSE(predict1, energy.testing$`Heating Load`)
rmse2 <- RMSE(predict2, energy.no.testing$`Heating Load`)
rmse3 <- RMSE(predict3, energy.tx.testing$`Heating Load`)

# Compute Adjusted R-Squared
adj_r_squared1 <- summary(lr_model1)$adj.r.squared
adj_r_squared2 <- summary(lr_model2)$adj.r.squared
adj_r_squared3 <- summary(lr_model3)$adj.r.squared

# Regression Tree models

# Predict on testing set
predict4 <- predict(tree_model1, newdata=energy.testing)
predict5 <- predict(tree_model2, newdata=energy.no.testing)
predict6 <- predict(tree_model3, newdata=energy.tx.testing)

# Compute RMSE
rmse4 <- RMSE(predict4, energy.testing$`Heating Load`)
rmse5 <- RMSE(predict5, energy.no.testing$`Heating Load`)
rmse6 <- RMSE(predict6, energy.tx.testing$`Heating Load`)

# Compute R-Squared for the tree models from the summary function
r_squared4 <- summary(tree_model1)$rsq[1]
```

```
## Call:
## rpart(formula = ‘Heating Load‘ ~ ., data = energy.training, method = "anova")
##   n= 537
##
##           CP nsplit  rel error     xerror        xstd
## 1 0.79706652      0 1.00000000 1.00227616 0.038230409
## 2 0.09025439      1 0.20293348 0.20383102 0.014316982
## 3 0.02709443      2 0.11267909 0.11387273 0.008846634
## 4 0.01256719      3 0.08558466 0.08675909 0.006688519
## 5 0.01220655      4 0.07301747 0.08017820 0.006383008
## 6 0.01000000      5 0.06081092 0.06688265 0.004803697
##
## Variable importance
## Relative Compactness         Surface Area       Overall Height
##                   24                   24                   21
##            Roof Area            Wall Area         Glazing Area
##                   21                    9                    1
##
## Node number 1: 537 observations,    complexity param=0.7970665
##   mean=21.98939, MSE=102.5617
##   left son=2 (278 obs) right son=3 (259 obs)
##   Primary splits:
##       Overall Height       < 5.25   to the left,  improve=0.7970665, (0 missing)
##       Relative Compactness < 0.75   to the left,  improve=0.7970665, (0 missing)
##       Roof Area            < 183.75 to the right, improve=0.7970665, (0 missing)
##       Surface Area         < 673.75 to the right, improve=0.7970665, (0 missing)
##       Wall Area            < 281.75 to the left,  improve=0.2070272, (0 missing)
##   Surrogate splits:
##       Relative Compactness < 0.75   to the left,  agree=1.000, adj=1.00, (0 split)
##       Surface Area         < 673.75 to the right, agree=1.000, adj=1.00, (0 split)
##       Roof Area            < 183.75 to the right, agree=1.000, adj=1.00, (0 split)
##       Wall Area            < 281.75 to the left,  agree=0.657, adj=0.29, (0 split)
##
## Node number 2: 278 observations,    complexity param=0.01256719
##   mean=13.26234, MSE=6.869463
##   left son=4 (108 obs) right son=5 (170 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.3624351, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3114269, (0 missing)
##       Relative Compactness      < 0.65   to the right, improve=0.2702899, (0 missing)
##       Surface Area              < 771.75 to the left,  improve=0.2702899, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.2702899, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.68, adj=0.176, (0 split)
##
## Node number 3: 259 observations,    complexity param=0.09025439
##   mean=31.35664, MSE=35.77984
##   left son=6 (168 obs) right son=7 (91 obs)
##   Primary splits:
##       Relative Compactness      < 0.805  to the right, improve=0.5364011, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.5364011, (0 missing)
##       Surface Area              < 624.75 to the left,  improve=0.5364011, (0 missing)
##       Glazing Area              < 0.175  to the left,  improve=0.2833498, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.1744826, (0 missing)
##   Surrogate splits:
##       Surface Area < 624.75 to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 330.75 to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 108 observations
##   mean=11.28269, MSE=5.362688
##
```

```
## Node number 5: 170 observations
##   mean=14.52, MSE=3.75526
##
## Node number 6: 168 observations,    complexity param=0.02709443
##   mean=28.13238, MSE=18.02454
##   left son=12 (69 obs) right son=13 (99 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.4927946, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3433437, (0 missing)
##       Surface Area              < 600.25 to the right, improve=0.1588406, (0 missing)
##       Relative Compactness      < 0.84   to the left,  improve=0.1588406, (0 missing)
##       Roof Area                 < 134.75 to the right, improve=0.1129024, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.649, adj=0.145, (0 split)
##
## Node number 7: 91 observations,    complexity param=0.01220655
##   mean=37.30912, MSE=13.93448
##   left son=14 (31 obs) right son=15 (60 obs)
##   Primary splits:
##       Glazing Area         < 0.175  to the left,  improve=0.5301760, (0 missing)
##       Surface Area         < 649.25 to the right, improve=0.0828404, (0 missing)
##       Relative Compactness < 0.775  to the left,  improve=0.0828404, (0 missing)
##       Wall Area            < 379.75 to the right, improve=0.0828404, (0 missing)
##       Roof Area            < 134.75 to the left,  improve=0.0828404, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.714, adj=0.161, (0 split)
##
## Node number 12: 69 observations
##   mean=24.56246, MSE=11.18535
##
## Node number 13: 99 observations
##   mean=30.62051, MSE=7.718092
##
## Node number 14: 31 observations
##   mean=33.52774, MSE=11.35595
##
## Node number 15: 60 observations
##   mean=39.26283, MSE=4.062004
```

```
r_squared5 <- summary(tree_model2)$rsq[1]
```

```
## Call:
## rpart(formula = ‘Heating Load‘ ~ ., data = energy.no.training,
##     method = "anova")
##   n= 537
##
##           CP nsplit  rel error     xerror        xstd
## 1 0.79642974      0 1.00000000 1.00471784 0.037444212
## 2 0.08124859      1 0.20357026 0.20464162 0.014582271
## 3 0.02884783      2 0.12232167 0.12346669 0.009344940
## 4 0.01415732      3 0.09347385 0.09494496 0.006903003
## 5 0.01232923      4 0.07931653 0.08160377 0.006439624
## 6 0.01000000      5 0.06698729 0.07515352 0.006143695
##
```

```
## Variable importance
## Relative Compactness         Surface Area       Overall Height
##                   23                   23                   21
##            Roof Area            Wall Area         Glazing Area
##                   21                   10                    1
##          Orientation
##                    1
##
## Node number 1: 537 observations,    complexity param=0.7964297
##   mean=22.1421, MSE=100.2455
##   left son=2 (267 obs) right son=3 (270 obs)
##   Primary splits:
##       Relative Compactness < 0.75   to the left,  improve=0.7964297, (0 missing)
##       Overall Height       < 5.25   to the left,  improve=0.7964297, (0 missing)
##       Surface Area         < 673.75 to the right, improve=0.7964297, (0 missing)
##       Roof Area            < 183.75 to the right, improve=0.7964297, (0 missing)
##       Wall Area            < 281.75 to the left,  improve=0.2223079, (0 missing)
##   Surrogate splits:
##       Surface Area   < 673.75 to the right, agree=1.000, adj=1.000, (0 split)
##       Roof Area      < 183.75 to the right, agree=1.000, adj=1.000, (0 split)
##       Overall Height < 5.25   to the left,  agree=1.000, adj=1.000, (0 split)
##       Wall Area      < 281.75 to the left,  agree=0.678, adj=0.352, (0 split)
##       Orientation    < 3.5    to the left,  agree=0.518, adj=0.030, (0 split)
##
## Node number 2: 267 observations,    complexity param=0.01415732
##   mean=13.1568, MSE=7.002004
##   left son=4 (102 obs) right son=5 (165 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.4076491, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.3588154, (0 missing)
##       Relative Compactness      < 0.65   to the right, improve=0.3095923, (0 missing)
##       Surface Area              < 771.75 to the left,  improve=0.3095923, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.3095923, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.685, adj=0.176, (0 split)
##
## Node number 3: 270 observations,    complexity param=0.08124859
##   mean=31.02756, MSE=33.66304
##   left son=6 (186 obs) right son=7 (84 obs)
##   Primary splits:
##       Relative Compactness      < 0.805  to the right, improve=0.4812134, (0 missing)
##       Surface Area              < 624.75 to the left,  improve=0.4812134, (0 missing)
##       Wall Area                 < 330.75 to the left,  improve=0.4812134, (0 missing)
##       Glazing Area              < 0.175  to the left,  improve=0.2513790, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.1833111, (0 missing)
##   Surrogate splits:
##       Surface Area < 624.75 to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 330.75 to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 102 observations
##   mean=11.008, MSE=5.165217
##
## Node number 5: 165 observations
##   mean=14.48515, MSE=3.518598
```

```
##
## Node number 6: 186 observations,    complexity param=0.02884783
##   mean=28.3228, MSE=18.75478
##   left son=12 (73 obs) right son=13 (113 obs)
##   Primary splits:
##       Glazing Area              < 0.175  to the left,  improve=0.44517130, (0 missing)
##       Glazing Area Distribution < 0.5    to the left,  improve=0.33790500, (0 missing)
##       Surface Area              < 600.25 to the right, improve=0.15382430, (0 missing)
##       Relative Compactness      < 0.84   to the left,  improve=0.15382430, (0 missing)
##       Roof Area                 < 134.75 to the right, improve=0.07795348, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.667, adj=0.151, (0 split)
##
## Node number 7: 84 observations,    complexity param=0.01232923
##   mean=37.01667, MSE=14.60563
##   left son=14 (32 obs) right son=15 (52 obs)
##   Primary splits:
##       Glazing Area         < 0.175  to the left,  improve=0.54097290, (0 missing)
##       Surface Area         < 649.25 to the right, improve=0.09269237, (0 missing)
##       Relative Compactness < 0.775  to the left,  improve=0.09269237, (0 missing)
##       Roof Area            < 134.75 to the left,  improve=0.09269237, (0 missing)
##       Wall Area            < 379.75 to the right, improve=0.09269237, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5    to the left,  agree=0.679, adj=0.156, (0 split)
##
## Node number 12: 73 observations
##   mean=24.72781, MSE=12.00401
##
## Node number 13: 113 observations
##   mean=30.64522, MSE=9.373152
##
## Node number 14: 32 observations
##   mean=33.43344, MSE=10.48016
##
## Node number 15: 52 observations
##   mean=39.22173, MSE=4.380822
```

```
r_squared6 <- summary(tree_model3)$rsq[1]
```

```
## Call:
## rpart(formula = ‘Heating Load‘ ~ ., data = energy.tx.training,
##     method = "anova")
##   n= 537
##
##           CP nsplit  rel error     xerror        xstd
## 1 0.81214111      0 1.00000000 1.00099705 0.038907023
## 2 0.04277249      1 0.18785889 0.18986262 0.015602827
## 3 0.03630434      2 0.14508641 0.16266787 0.012927902
## 4 0.02119471      3 0.10878207 0.11116889 0.008015748
## 5 0.02047113      4 0.08758736 0.09787175 0.007603611
## 6 0.01313718      5 0.06711623 0.07537215 0.005414681
## 7 0.01000000      6 0.05397906 0.05595452 0.003889504
##
## Variable importance
##      Relative Compactness              Surface Area            Overall Height
##                        23                        23                        21
##                 Roof Area                 Wall Area              Glazing Area
##                        21                         8                         2
## Glazing Area Distribution               Orientation
##                         2                         1
##
## Node number 1: 537 observations,    complexity param=0.8121411
##   mean=2.992702, MSE=0.2257549
##   left son=2 (270 obs) right son=3 (267 obs)
##   Primary splits:
##       Relative Compactness < 0.75      to the left,  improve=0.8121411, (0 missing)
##       Roof Area            < 5.193165  to the right, improve=0.8121411, (0 missing)
##       Overall Height       < 1.599337  to the left,  improve=0.8121411, (0 missing)
##       Surface Area         < 6.512694  to the right, improve=0.8121411, (0 missing)
##       Wall Area            < 5.640074  to the left,  improve=0.2229821, (0 missing)
##   Surrogate splits:
##       Surface Area   < 6.512694  to the right, agree=1.000, adj=1.000, (0 split)
##       Roof Area      < 5.193165  to the right, agree=1.000, adj=1.000, (0 split)
##       Overall Height < 1.599337  to the left,  agree=1.000, adj=1.000, (0 split)
##       Wall Area      < 5.640074  to the left,  agree=0.654, adj=0.303, (0 split)
##       Orientation    < 3.5       to the right, agree=0.523, adj=0.041, (0 split)
##
## Node number 2: 270 observations,    complexity param=0.04277249
##   mean=2.566899, MSE=0.0448872
##   left son=4 (18 obs) right son=5 (252 obs)
##   Primary splits:
##       Glazing Area              < 0.1581139 to the left,  improve=0.4278482, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.4278482, (0 missing)
##       Relative Compactness      < 0.65      to the right, improve=0.2163006, (0 missing)
##       Surface Area              < 6.648535  to the left,  improve=0.2163006, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.2163006, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5       to the left,  agree=1, adj=1, (0 split)
##
## Node number 3: 267 observations,    complexity param=0.03630434
##   mean=3.423288, MSE=0.0399051
##   left son=6 (178 obs) right son=7 (89 obs)
##   Primary splits:
##       Relative Compactness      < 0.805     to the right, improve=0.4130764, (0 missing)
##       Surface Area              < 6.437159  to the left,  improve=0.4130764, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.4130764, (0 missing)
##       Glazing Area              < 0.1581139 to the left,  improve=0.2781648, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.2781648, (0 missing)
##   Surrogate splits:
##       Surface Area < 6.437159  to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 5.800676  to the left,  agree=1, adj=1, (0 split)
##
## Node number 4: 18 observations
##   mean=2.048374, MSE=0.04165259
##
## Node number 5: 252 observations,    complexity param=0.02119471
##   mean=2.603937, MSE=0.02454156
##   left son=10 (168 obs) right son=11 (84 obs)
##   Primary splits:
##       Relative Compactness      < 0.65      to the right, improve=0.415466700, (0 missing)
##       Wall Area                 < 5.800676  to the left,  improve=0.415466700, (0 missing)
##       Surface Area              < 6.648535  to the left,  improve=0.415466700, (0 missing)
##       Glazing Area              < 0.5662278 to the left,  improve=0.305767300, (0 missing)
##       Glazing Area Distribution < 1.573132  to the right, improve=0.007198365, (0 missing)
##   Surrogate splits:
##       Surface Area < 6.648535  to the left,  agree=1, adj=1, (0 split)
##       Wall Area    < 5.800676  to the left,  agree=1, adj=1, (0 split)
##
## Node number 6: 178 observations,    complexity param=0.02047113
##   mean=3.332503, MSE=0.0278924
##   left son=12 (12 obs) right son=13 (166 obs)
##   Primary splits:
##       Glazing Area              < 0.1581139 to the left,  improve=0.49985870, (0 missing)
##       Glazing Area Distribution < 0.5       to the left,  improve=0.49985870, (0 missing)
##       Relative Compactness      < 0.84      to the left,  improve=0.08599162, (0 missing)
##       Surface Area              < 6.397138  to the right, improve=0.08599162, (0 missing)
##       Roof Area                 < 4.899272  to the right, improve=0.04170109, (0 missing)
##   Surrogate splits:
##       Glazing Area Distribution < 0.5       to the left,  agree=1, adj=1, (0 split)
##
## Node number 7: 89 observations
##   mean=3.604858, MSE=0.01447892
##
## Node number 10: 168 observations,    complexity param=0.01313718
##   mean=2.532536, MSE=0.01407968
##   left son=20 (112 obs) right son=21 (56 obs)
##   Primary splits:
##       Glazing Area              < 0.5662278 to the left,  improve=0.673304200, (0 missing)
##       Surface Area              < 6.616265  to the left,  improve=0.026341050, (0 missing)
##       Relative Compactness      < 0.675     to the right, improve=0.026341050, (0 missing)
##       Wall Area                 < 5.723601  to the left,  improve=0.026341050, (0 missing)
##       Glazing Area Distribution < 1.573132  to the right, improve=0.006295481, (0 missing)
##
## Node number 11: 84 observations
##   mean=2.746739, MSE=0.01487671
##
## Node number 12: 12 observations
##   mean=2.893336, MSE=0.01272934
##
## Node number 13: 166 observations
##   mean=3.36425, MSE=0.01403839
##
## Node number 20: 112 observations
##   mean=2.463688, MSE=0.00588785
##
## Node number 21: 56 observations
##   mean=2.67023, MSE=0.00202362
```

```r
# Display results
results <- data.frame(Model=c("Multiple Regression - energy.training", "Multiple Regression - energy.no
print(results)
```

```
##                                      Model R_Squared      RMSE
## 1    Multiple Regression - energy.training 0.9203403 3.1020580
## 2 Multiple Regression - energy.no.training 0.9184307 3.1056684
## 3 Multiple Regression - energy.tx.training 0.9500537 0.0960028
## 4        Regression Tree - energy.training 0.9203403 2.9233058
## 5     Regression Tree - energy.no.training 0.9184307 2.7379038
## 6     Regression Tree - energy.tx.training 0.9500537 0.1139632
```

1. From your results, it appears that the multiple regression model trained on the energy.tx.training dataset has the highest Adjusted R-Squared and lowest RMSE, thus indicating it's the best performing model among the six. This means that this model explains the variation in the heating load the best and has the smallest average prediction error.

For each of the predictions, calculate the 95% prediction interval for the Heating Load. (Exclude Regression Trees)

```r
# 95% prediction interval for energy.training
pred_interval1 <- predict(lr_model1, newdata = energy.testing, interval = "prediction", level = 0.95)
pred_interval1
```

```
##           fit         lwr      upr
## 1   22.677000 16.98208532 28.37191
## 2   22.677000 16.98208532 28.37191
## 3   25.132195 19.46835537 30.79604
## 4   25.132195 19.46835537 30.79604
## 5   24.071911 18.39762712 29.74620
## 6   26.062136 20.39432746 31.72995
## 7   27.351400 21.68288067 33.01992
## 8   31.691174 25.98851698 37.39383
## 9   31.691174 25.98851698 37.39383
## 10   5.750708  0.07675971 11.42466
## 11   7.039972  1.37223545 12.70771
## 12   7.039972  1.37223545 12.70771
## 13   7.628275  1.97008762 13.28646
## 14   8.917539  3.25890666 14.57617
## 15  24.840840 19.16276401 30.51892
## 16  27.296036 21.64916258 32.94291
## 17  26.235752 20.57733117 31.89417
## 18  26.235752 20.57733117 31.89417
## 19  29.515241 23.86308711 35.16740
## 20  29.515241 23.86308711 35.16740
## 21  29.515241 23.86308711 35.16740
## 22  33.855014 28.17017515 39.53985
## 23  33.855014 28.17017515 39.53985
## 24   7.914548  2.25758533 13.57151
## 25   9.792115  4.15089312 15.43334
## 26  11.081379  5.43981572 16.72294
## 27  11.081379  5.43981572 16.72294
## 28  11.669682  6.02177370 17.31759
## 29  25.096562 19.42299109 30.77013
## 30  27.551758 21.90940440 33.19411
## 31  26.491473 20.83762228 32.14532
## 32  26.491473 20.83762228 32.14532
## 33  28.481699 22.83470160 34.12870
## 34  29.770963 24.12342219 35.41850
## 35  29.770963 24.12342219 35.41850
## 36  34.110736 28.43046410 39.79101
## 37   9.459534  3.81335281 15.10572
## 38  10.047837  4.41122590 15.68445
## 39  11.337101  5.70021354 16.97399
## 40  11.925404  6.28232310 17.56848
## 41  11.925404  6.28232310 17.56848
## 42  11.925404  6.28232310 17.56848
## 43  25.352283 19.67862076 31.02595
## 44  27.807479 22.16502338 33.44993
## 45  27.807479 22.16502338 33.44993
## 46  34.366457 28.68616110 40.04675
## 47   9.715255  4.06890138 15.36161
## 48   9.715255  4.06890138 15.36161
## 49  10.303558  4.66693120 15.94019
## 50  11.592822  5.95598417 17.22966
## 51  11.592822  5.95598417 17.22966
## 52  12.769428  7.10800792 18.43085
## 53  12.769428  7.10800792 18.43085
## 54  25.608005 19.92965325 31.28636
## 55  25.608005 19.92965325 31.28636
## 56  27.002916 21.34436443 32.66147
## 57  28.993141 23.34138547 34.64490
## 58  34.622179 28.93726622 40.30709
## 59   8.681713  3.02403679 14.33939
## 60   8.681713  3.02403679 14.33939
## 61   9.970977  4.31983061 15.62212
## 62   9.970977  4.31983061 15.62212
## 63  11.848544  6.20712748 17.48996
## 64  11.848544  6.20712748 17.48996
## 65  13.025149  7.35948352 18.69081
## 66  13.025149  7.35948352 18.69081
## 67  28.318922 22.66240518 33.97544
## 68  28.318922 22.66240518 33.97544
## 69  27.258638 21.59082694 32.92645
## 70  27.258638 21.59082694 32.92645
## 71  29.248863 23.58781061 34.90992
## 72  30.538127 24.87672767 36.19953
## 73  34.877900 29.18379063 40.57201
## 74  34.877900 29.18379063 40.57201
## 75  34.877900 29.18379063 40.57201
## 76  10.226698  4.56615225 15.88724
## 77  10.815001  5.16447090 16.46553
## 78  10.815001  5.16447090 16.46553
## 79  12.104265  6.45365472 17.75488
## 80  12.692568  7.03624872 18.34889
## 81  30.158215 24.51322083 35.80321
## 82  30.158215 24.51322083 35.80321
## 83  30.158215 24.51322083 35.80321
## 84  29.097931 23.43968994 34.75617
## 85  29.097931 23.43968994 34.75617
## 86  31.088156 25.43740209 36.73891
## 87  31.088156 25.43740209 36.73891
## 88  31.088156 25.43740209 36.73891
## 89  10.776727  5.12191659 16.43154
## 90  10.776727  5.12191659 16.43154
## 91  12.065991  6.41757750 17.71441
## 92  12.065991  6.41757750 17.71441
## 93  12.654294  7.01479562 18.29379
## 94  12.654294  7.01479562 18.29379
## 95  13.943558  8.30377756 19.58334
## 96  14.531861  8.88523994 20.17848
## 97  15.120164  9.45420523 20.78612
## 98  15.120164  9.45420523 20.78612
## 99  27.958741 22.28892435 33.62856
## 100 27.958741 22.28892435 33.62856
## 101 29.353652 23.70194007 35.00536
## 102 31.343878 25.69963455 36.98812
## 103 32.633142 26.98841426 38.27787
## 104 36.972915 31.29779815 42.64803
## 105 11.032449  5.38397645 16.68092
## 106 12.321713  6.67970981 17.96372
## 107 14.199280  8.56614218 19.83242
## 108 14.199280  8.56614218 19.83242
## 109 14.787583  9.14775343 20.42741
## 110 14.787583  9.14775343 20.42741
## 111 15.375885  9.71685191 21.03492
## 112 15.375885  9.71685191 21.03492
## 113 28.214462 22.54650721 33.88242
## 114 30.669658 25.03301062 36.30631
## 115 29.609374 23.95957730 35.25917
## 116 29.609374 23.95957730 35.25917
## 117 31.599599 25.95724798 37.24195
## 118 32.888863 27.24609279 38.53163
## 119 32.888863 27.24609279 38.53163
## 120 37.228636 31.55544658 42.90183
## 121 11.288170  5.64142048 16.93492
## 122 13.165737  7.53476657 18.79671
## 123 15.043304  9.40564472 20.68096
## 124 15.631607  9.97489229 21.28832
## 125 15.631607  9.97489229 21.28832
## 126 28.470184 22.79948728 34.14088
## 127 29.865095 24.21259694 35.51759
## 128 31.855320 26.21023773 37.50040
## 129 31.855320 26.21023773 37.50040
## 130 31.855320 26.21023773 37.50040
## 131 37.484358 31.80849646 43.16022
## 132 11.543892  5.89424448 17.19354
## 133 11.543892  5.89424448 17.19354
## 134 12.833156  7.19010701 18.47620
## 135 14.710723  9.07698265 20.34446
## 136 14.710723  9.07698265 20.34446
## 137 15.299025  9.65890845 20.93914
## 138 15.887328 10.22832070 21.54634
## 139 28.725905 23.04787123 34.40394
## 140 31.181101 25.53429841 36.82790
## 141 30.120817 24.46100561 35.78063
## 142 30.120817 24.46100561 35.78063
## 143 32.111042 26.45861052 37.76347
## 144 32.111042 26.45861052 37.76347
## 145 33.400306 27.74758637 39.05303
## 146 13.088877  7.43837446 18.73938
## 147 13.677180  8.03621685 19.31814
## 148 13.677180  8.03621685 19.31814
## 149 14.966444  9.32545997 20.60743
## 150 14.966444  9.32545997 20.60743
## 151 16.143050 10.47714275 21.80896
## 152 30.565198 24.87657719 36.25382
## 153 30.565198 24.87657719 36.25382
## 154 33.020394 27.36305548 38.67773
## 155 33.020394 27.36305548 38.67773
## 156 31.960110 26.28786198 37.63236
## 157 33.950335 28.28616602 39.61450
## 158 33.950335 28.28616602 39.61450
## 159 35.239599 29.57494162 40.90426
## 160 35.239599 29.57494162 40.90426
## 161 35.239599 29.57494162 40.90426
## 162 39.579372 33.88680237 45.27194
## 163 13.638906  7.97204839 19.30576
## 164 15.516473  9.86446098 21.16849
## 165 15.516473  9.86446098 21.16849
## 166 16.805737 11.15350287 22.45797
## 167 17.394040 11.73448783 23.05359
## 168 17.394040 11.73448783 23.05359
## 169 17.982343 12.30300639 23.66168
## 170 17.982343 12.30300639 23.66168
## 171 33.276116 27.62720793 38.92502
## 172 33.276116 27.62720793 38.92502
## 173 33.276116 27.62720793 38.92502
## 174 32.215831 26.55205066 37.87961
## 175 34.206057 28.55034052 39.86177
## 176 35.495321 29.83918049 41.15146
## 177 35.495321 29.83918049 41.15146
## 178 39.835094 34.15097984 45.51921
## 179 15.183892  9.53183237 20.83595
## 180 15.183892  9.53183237 20.83595
## 181 17.061459 11.41781450 22.70510
## 182 17.061459 11.41781450 22.70510
## 183 17.649762 11.99894496 23.30058
## 184 17.649762 11.99894496 23.30058
## 185 17.649762 11.99894496 23.30058
## 186 18.238064 12.56758913 23.90854
## 187 18.238064 12.56758913 23.90854
## 188 18.238064 12.56758913 23.90854
## 189 31.076641 25.40022312 36.75306
## 190 33.531837 27.88674867 39.17693
## 191 33.531837 27.88674867 39.17693
## 192 32.471553 26.81163983 38.13147
## 193 34.461778 28.80990892 40.11365
## 194 34.461778 28.80990892 40.11365
## 195 34.461778 28.80990892 40.11365
## 196 35.751042 30.09881375 41.40327
## 197 35.751042 30.09881375 41.40327
## 198 40.090815 34.41057425 45.77106
## 199 14.150349  8.49544997 19.80525
## 200 14.150349  8.49544997 19.80525
## 201 15.439613  9.79130053 21.08793
## 202 15.439613  9.79130053 21.08793
## 203 16.027916 10.38833849 21.66749
## 204 17.317180 11.67751048 22.95685
## 205 17.317180 11.67751048 22.95685
## 206 17.317180 11.67751048 22.95685
## 207 17.905483 12.25879263 23.55217
## 208 33.787558 28.14166836 39.43345
## 209 32.727274 27.06662007 38.38793
## 210 32.727274 27.06662007 38.38793
## 211 32.727274 27.06662007 38.38793
## 212 32.727274 27.06662007 38.38793
## 213 34.717499 29.06486182 40.37014
## 214 15.695335 10.04615023 21.34452
## 215 15.695335 10.04615023 21.34452
## 216 18.161204 12.51402073 23.80839
## 217 18.749507 13.08296467 24.41605
## 218 31.588084 25.90549792 37.27067
## 219 31.588084 25.90549792 37.27067
## 220 31.588084 25.90549792 37.27067
## 221 34.043280 28.39196895 39.69459
## 222 34.043280 28.39196895 39.69459
## 223 32.982996 27.31699319 38.64900
## 224 32.982996 27.31699319 38.64900
## 225 34.973221 29.31520109 40.63124
## 226 36.262485 30.60423648 41.92073
## 227 36.262485 30.60423648 41.92073
## 228 40.602258 34.91598750 46.28853
## 229 14.661792  9.00041035 20.32317
## 230 15.951056 10.29638360 21.60573
## 231 17.828623 12.18302779 23.47422
```

```
# 95% prediction interval for energy.no.training
pred_interval2 <- predict(lr_model2, newdata = energy.no.testing, interval = "prediction", level = 0.95
pred_interval2
```

```
##           fit       lwr      upr
## 1   22.934095 17.2362945 28.63189
## 2   22.934095 17.2362945 28.63189
## 3   25.187008 19.5207016 30.85331
## 4   23.829569 18.1523995 29.50674
## 5   25.783990 20.1130245 31.45496
## 6   27.063762 21.3917710 32.73575
## 7   31.655393 25.9501361 37.36065
## 8   31.655393 25.9501361 37.36065
## 9    5.807722  0.1306559 11.48479
## 10   8.972387  3.3107116 14.63406
## 11   9.577509  3.9084335 15.24658
## 12   9.577509  3.9084335 15.24658
## 13  10.182631  4.4931950 15.87207
## 14  10.182631  4.4931950 15.87207
## 15  27.317519 21.6686018 32.96644
## 16  25.960080 20.3004497 31.61971
## 17  27.914501 22.2610751 33.56793
## 18  27.914501 22.2610751 33.56793
## 19  29.194273 23.5398354 34.84871
## 20  33.785904 28.0979136 39.47390
## 21   7.938233  2.2785144 13.59795
## 22   7.938233  2.2785144 13.59795
## 23   9.218005  3.5650150 14.87099
## 24   9.823126  4.1792326 15.46702
## 25  11.102898  5.4587040 16.74709
## 26  11.102898  5.4587040 16.74709
## 27  11.708020  6.0564556 17.35958
## 28  11.708020  6.0564556 17.35958
## 29  26.133315 20.4783155 31.78832
## 30  26.133315 20.4783155 31.78832
## 31  28.087737 22.4388500 33.73662
## 32  29.367508 23.7175578 35.01746
## 33  33.959140 28.2754375 39.64284
## 34  33.959140 28.2754375 39.64284
## 35   8.111468  2.4564438 13.76649
## 36   9.391240  3.7428982 15.03958
## 37   9.391240  3.7428982 15.03958
## 38   9.996362  4.3571160 15.63561
## 39  11.276133  5.6365354 16.91573
## 40  11.276133  5.6365354 16.91573
## 41  11.881255  6.2342737 17.52824
## 42  11.881255  6.2342737 17.52824
## 43  12.486377  6.8190010 18.15375
## 44  25.411076 19.7355033 31.08665
## 45  25.411076 19.7355033 31.08665
## 46  26.306551 20.6517238 31.96138
## 47  26.306551 20.6517238 31.96138
## 48  28.260972 22.6121623 33.90978
## 49  29.540743 23.8908184 35.19067
## 50  29.540743 23.8908184 35.19067
## 51  34.132375 28.4485261 39.81622
## 52  34.132375 28.4485261 39.81622
## 53   8.284704  2.6299156 13.93949
## 54   8.284704  2.6299156 13.93949
## 55   9.564475  3.9163186 15.21263
## 56  12.054490  6.4076278 17.70135
## 57  12.054490  6.4076278 17.70135
## 58  27.837224 22.1887517 33.48570
## 59  26.479786 20.8206740 32.13890
## 60  26.479786 20.8206740 32.13890
## 61  28.434207 22.7810119 34.08740
## 62  29.713979 24.0596173 35.36834
## 63  34.305610 28.6171797 39.99404
## 64  34.305610 28.6171797 39.99404
## 65   8.457939  2.7989292 14.11695
## 66   8.457939  2.7989292 14.11695
## 67  10.342832  4.6994722 15.98619
## 68  11.622604  5.9787884 17.26642
## 69  12.227726  6.5765177 17.87893
## 70  12.832847  7.1612457 18.50445
## 71  12.832847  7.1612457 18.50445
## 72  28.010460 22.3532125 33.66771
## 73  28.607442 22.9454092 34.26948
## 74  29.887214 24.2239648 35.55046
## 75  10.516067  4.8639548 16.16818
## 76  10.516067  4.8639548 16.16818
## 77  12.400961  6.7409536 18.06097
## 78  12.400961  6.7409536 18.06097
## 79  12.400961  6.7409536 18.06097
## 80  12.400961  6.7409536 18.06097
## 81  13.006083  7.3257059 18.68646
## 82  28.000519 22.3221345 33.67890
## 83  30.253432 24.6070066 35.89986
## 84  28.895994 23.2390873 34.55290
## 85  30.850415 25.1998317 36.50100
## 86  30.850415 25.1998317 36.50100
## 87  30.850415 25.1998317 36.50100
## 88  32.130187 26.4786944 37.78168
## 89  32.130187 26.4786944 37.78168
## 90  12.153918  6.5033750 17.80446
## 91  12.759040  7.1176865 18.40039
## 92  12.759040  7.1176865 18.40039
## 93  14.038812  8.3972608 19.68036
## 94  14.038812  8.3972608 19.68036
## 95  14.643933  8.9950988 20.29277
## 96  15.249055  9.5799295 20.91818
## 97  15.249055  9.5799295 20.91818
## 98  15.249055  9.5799295 20.91818
## 99  28.173755 22.5014809 33.84603
## 100 28.173755 22.5014809 33.84603
## 101 30.426667 24.7862559 36.06708
## 102 30.426667 24.7862559 36.06708
## 103 31.023650 25.3790055 36.66830
## 104 32.303422 26.6578154 37.94903
## 105 36.895053 31.2156506 42.57446
## 106 11.047382  5.3960950 16.69867
## 107 12.327153  6.6826569 17.97165
## 108 12.327153  6.6826569 17.97165
## 109 12.327153  6.6826569 17.97165
## 110 14.212047  8.5764932 19.84760
## 111 14.212047  8.5764932 19.84760
## 112 15.422291  9.7591182 21.08546
## 113 28.346990 22.6763849 34.01759
## 114 28.346990 22.6763849 34.01759
## 115 30.599903 24.9610374 36.23877
## 116 32.476657 26.8324728 38.12084
## 117 37.068289 31.3901274 42.74645
## 118 11.220617  5.5709623 16.87027
## 119 12.500389  6.8574744 18.14330
## 120 13.105511  7.4717837 18.73924
## 121 13.105511  7.4717837 18.73924
## 122 14.990404  9.3490678 20.63174
## 123 15.595526  9.9338572 21.25719
## 124 15.595526  9.9338572 21.25719
## 125 28.520225 22.8468426 34.19361
## 126 28.520225 22.8468426 34.19361
## 127 28.520225 22.8468426 34.19361
## 128 30.773138 25.1313477 36.41493
## 129 29.415699 23.7634951 35.06790
## 130 29.415699 23.7634951 35.06790
## 131 31.370121 25.7239563 37.01629
## 132 32.649892 27.0026631 38.29712
## 133 32.649892 27.0026631 38.29712
## 134 37.241524 31.5601640 42.92288
## 135 11.393852  5.7413668 17.04634
## 136 12.673624  7.0278238 18.31942
## 137 14.558517  8.9215399 20.19549
## 138 14.558517  8.9215399 20.19549
## 139 15.768761 10.1041428 21.43338
## 140 28.693460 23.0128605 34.37406
## 141 28.693460 23.0128605 34.37406
## 142 30.946373 25.2971936 36.59555
## 143 29.588935 23.9293797 35.24849
## 144 32.823128 27.1683936 38.47786
## 145 32.823128 27.1683936 38.47786
## 146 37.414759 31.7257678 43.10375
## 147 37.414759 31.7257678 43.10375
## 148 11.567088  5.9073153 17.22686
## 149 12.846859  7.1937120 18.50001
## 150 13.451981  7.8079901 19.09597
## 151 13.451981  7.8079901 19.09597
## 152 14.731753  9.0873576 20.37615
## 153 15.336874  9.6851701 20.98858
## 154 15.941996 10.2699821 21.61401
## 155 30.936433 25.2466203 36.62625
## 156 30.936433 25.2466203 36.62625
## 157 30.936433 25.2466203 36.62625
## 158 33.189346 27.5317203 38.84697
## 159 31.831907 26.1640587 37.49976
## 160 35.066100 29.4038737 40.72833
## 161 35.066100 29.4038737 40.72833
## 162 39.657732 33.9619602 45.35350
## 163 13.810060  8.1413585 19.47876
## 164 13.810060  8.1413585 19.47876
## 165 15.089832  9.4280538 20.75161
## 166 17.579847 11.9200563 23.23964
## 167 17.579847 11.9200563 23.23964
## 168 17.579847 11.9200563 23.23964
## 169 31.109668 25.4273418 36.79199
## 170 33.959564 28.3054633 39.61366
## 171 35.239335 29.5843777 40.89429
## 172 15.263067  9.6087179 20.91742
## 173 15.868189 10.2231058 21.51327
## 174 17.147960 11.5027315 22.79319
## 175 17.753082 12.1006567 23.40551
## 176 17.753082 12.1006567 23.40551
## 177 18.358204 12.6855826 24.03083
## 178 31.282903 25.6036309 36.96218
## 179 34.132799 28.4815652 39.78403
## 180 34.132799 28.4815652 39.78403
## 181 35.412571 29.7604275 41.06471
## 182 35.412571 29.7604275 41.06471
## 183 35.412571 29.7604275 41.06471
## 184 40.004202 34.3181109 45.69029
## 185 14.156531  8.4983229 19.81474
## 186 15.436302  9.7849276 21.08768
## 187 15.436302  9.7849276 21.08768
## 188 16.041424 10.3993131 21.68354
## 189 17.321196 11.6788870 22.96350
## 190 17.321196 11.6788870 22.96350
## 191 17.926318 12.2768012 23.57583
## 192 17.926318 12.2768012 23.57583
## 193 18.531439 12.8617094 24.20117
## 194 33.709051 28.0602379 39.35787
## 195 33.709051 28.0602379 39.35787
## 196 32.351613 26.6926357 38.01059
## 197 34.306034 28.6532056 39.95886
## 198 34.306034 28.6532056 39.95886
## 199 35.585806 29.9320165 41.23960
## 200 35.585806 29.9320165 41.23960
## 201 40.177437 34.4895379 45.86534
## 202 40.177437 34.4895379 45.86534
## 203 40.177437 34.4895379 45.86534
## 204 14.329766  8.6701246 19.98941
## 205 14.329766  8.6701246 19.98941
## 206 15.609538  9.9566760 21.26240
## 207 16.214659 10.5710517 21.85827
## 208 16.214659 10.5710517 21.85827
## 209 17.494431 11.8505739 23.13829
## 210 17.494431 11.8505739 23.13829
## 211 17.494431 11.8505739 23.13829
## 212 18.099553 12.4484829 23.75062
## 213 18.099553 12.4484829 23.75062
## 214 18.099553 12.4484829 23.75062
## 215 18.704675 13.0333893 24.37596
## 216 18.704675 13.0333893 24.37596
## 217 31.629374 25.9428940 37.31585
## 218 33.882287 28.2274871 39.53709
## 219 33.882287 28.2274871 39.53709
## 220 33.882287 28.2274871 39.53709
## 221 32.524848 26.8599207 38.18978
## 222 34.479270 28.8203882 40.13815
## 223 34.479270 28.8203882 40.13815
## 224 35.759041 30.0991485 41.41893
## 225 40.350673 34.6565349 46.04481
## 226 14.503001  8.8374737 20.16853
## 227 16.387895 10.7383251 22.03746
## 228 16.387895 10.7383251 22.03746
## 229 17.667666 12.0177959 23.31754
## 230 18.272788 12.6157053 23.92987
## 231 18.877910 13.2006259 24.55519
```

```
# 95% prediction interval for energy.tx.training
pred_interval3 <- predict(lr_model3, newdata = energy.tx.testing, interval = "prediction", level = 0.95
pred_interval3
```

```
##          fit      lwr      upr
## 1   2.907143 2.694847 3.119438
## 2   2.985829 2.774657 3.197002
## 3   2.984291 2.772628 3.195954
## 4   2.984291 2.772628 3.195954
## 5   3.202317 2.989624 3.415010
## 6   2.028112 1.815755 2.240469
## 7   2.028112 1.815755 2.240469
## 8   2.051658 1.840083 2.263232
## 9   2.149701 1.938711 2.360691
## 10  2.163208 1.951602 2.374814
## 11  2.163208 1.951602 2.374814
## 12  3.187675 2.976945 3.398405
## 13  3.266362 3.056747 3.475977
## 14  3.266362 3.056747 3.475977
## 15  3.264823 3.054734 3.474913
## 16  3.344762 3.134998 3.554526
## 17  3.482849 3.271664 3.694035
## 18  3.482849 3.271664 3.694035
## 19  2.308645 2.097877 2.519412
## 20  2.332190 2.122206 2.542175
## 21  2.332190 2.122206 2.542175
## 22  2.443740 2.233673 2.653808
## 23  2.531305 2.321657 2.740952
## 24  2.613528 2.403164 2.823893
## 25  2.613528 2.403164 2.823893
## 26  3.195394 2.984761 3.406027
## 27  3.195394 2.984761 3.406027
## 28  3.274081 3.064551 3.483611
## 29  3.261283 3.051128 3.471439
## 30  3.261283 3.051128 3.471439
## 31  3.261283 3.051128 3.471439
## 32  3.272542 3.062529 3.482556
## 33  3.272542 3.062529 3.482556
## 34  3.272542 3.062529 3.482556
## 35  3.352481 3.142786 3.562177
## 36  2.316364 2.105665 2.527063
## 37  2.316364 2.105665 2.527063
## 38  2.339909 2.130000 2.549819
## 39  2.437953 2.228579 2.647327
## 40  2.437953 2.228579 2.647327
## 41  2.539024 2.329455 2.748593
## 42  3.201317 2.990590 3.412044
## 43  3.267206 3.056943 3.477469
## 44  3.267206 3.056943 3.477469
## 45  3.278465 3.068341 3.488590
## 46  3.278465 3.068341 3.488590
## 47  3.278465 3.068341 3.488590
## 48  3.358404 3.148592 3.568216
## 49  3.496491 3.285216 3.707766
## 50  3.496491 3.285216 3.707766
## 51  2.322287 2.111472 2.533101
## 52  2.322287 2.111472 2.533101
## 53  2.345832 2.135811 2.555853
## 54  2.443876 2.234391 2.653360
## 55  2.443876 2.234391 2.653360
## 56  2.457382 2.247286 2.667478
## 57  2.544947 2.335269 2.754625
## 58  2.627170 2.416771 2.837569
## 59  3.206310 2.995390 3.417231
## 60  3.284997 3.075160 3.494835
## 61  3.284997 3.075160 3.494835
## 62  3.272199 3.061731 3.482667
## 63  3.272199 3.061731 3.482667
## 64  3.272199 3.061731 3.482667
## 65  3.283459 3.073127 3.493790
## 66  3.363397 3.153373 3.573422
## 67  3.363397 3.153373 3.573422
## 68  3.501485 3.289984 3.712986
## 69  3.501485 3.289984 3.712986
## 70  3.501485 3.289984 3.712986
## 71  2.327280 2.116254 2.538306
## 72  2.327280 2.116254 2.538306
## 73  2.350825 2.140597 2.561054
## 74  2.350825 2.140597 2.561054
## 75  2.462376 2.252075 2.672676
## 76  2.462376 2.252075 2.672676
## 77  2.549940 2.340056 2.759824
## 78  2.549940 2.340056 2.759824
## 79  3.289396 3.079294 3.499498
## 80  3.276599 3.065864 3.487333
## 81  3.287858 3.077257 3.498458
## 82  3.367797 3.157499 3.578095
## 83  3.505884 3.294098 3.717670
## 84  2.331679 2.120381 2.542977
## 85  2.355225 2.144727 2.565723
## 86  2.453268 2.243307 2.663229
## 87  2.466775 2.256207 2.677342
## 88  2.636563 2.425688 2.847437
## 89  2.636563 2.425688 2.847437
## 90  3.339874 3.129003 3.550744
## 91  3.339874 3.129003 3.550744
## 92  3.418561 3.208813 3.628309
## 93  3.418561 3.208813 3.628309
## 94  3.418561 3.208813 3.628309
## 95  3.405763 3.195404 3.616122
## 96  3.417022 3.206824 3.627221
## 97  3.417022 3.206824 3.627221
## 98  3.496961 3.287080 3.706842
## 99  2.460843 2.249988 2.671698
## 100 2.460843 2.249988 2.671698
## 101 2.460843 2.249988 2.671698
## 102 2.582432 2.372847 2.792018
## 103 2.582432 2.372847 2.792018
## 104 2.595939 2.385736 2.806142
## 105 2.595939 2.385736 2.806142
## 106 2.683503 2.473695 2.893312
## 107 2.765727 2.555181 2.976273
## 108 3.347593 3.137010 3.558175
## 109 3.426280 3.216808 3.635751
## 110 3.413482 3.203394 3.623570
## 111 3.424741 3.214810 3.634672
## 112 3.504680 3.295059 3.714300
## 113 3.642767 3.431717 3.853817
## 114 3.642767 3.431717 3.853817
## 115 2.468562 2.257967 2.679157
## 116 2.492108 2.282291 2.701925
## 117 2.590151 2.380835 2.799467
## 118 2.773446 2.563167 2.983724
## 119 3.353516 3.142986 3.564046
## 120 3.353516 3.142986 3.564046
## 121 3.419405 3.209356 3.629453
## 122 3.510603 3.301013 3.720193
## 123 3.510603 3.301013 3.720193
## 124 2.474485 2.263921 2.685049
## 125 2.498031 2.288250 2.707812
## 126 2.498031 2.288250 2.707812
## 127 2.609581 2.399689 2.819473
## 128 2.697145 2.487646 2.906644
## 129 3.358509 3.147909 3.569108
## 130 3.358509 3.147909 3.569108
## 131 3.437196 3.227689 3.646702
## 132 3.424398 3.214269 3.634527
## 133 3.435657 3.225679 3.645635
## 134 2.479479 2.268827 2.690130
## 135 2.479479 2.268827 2.690130
## 136 2.503024 2.293159 2.712889
## 137 2.702139 2.492557 2.911720
## 138 2.784362 2.574037 2.994687
## 139 3.362908 3.152161 3.573655
## 140 3.441595 3.231933 3.651257
## 141 3.441595 3.231933 3.651257
## 142 3.440057 3.229918 3.650195
## 143 3.519995 3.310153 3.729838
## 144 3.658083 3.446766 3.869399
## 145 3.658083 3.446766 3.869399
## 146 3.658083 3.446766 3.869399
## 147 2.483878 2.273062 2.694693
## 148 2.507423 2.297398 2.717449
## 149 2.507423 2.297398 2.717449
## 150 2.507423 2.297398 2.717449
## 151 2.618973 2.408842 2.829105
## 152 2.618973 2.408842 2.829105
## 153 2.706538 2.496798 2.916278
## 154 2.706538 2.496798 2.916278
## 155 2.788761 2.578277 2.999246
## 156 2.788761 2.578277 2.999246
## 157 3.528259 3.318056 3.738462
## 158 3.528259 3.318056 3.738462
## 159 3.528259 3.318056 3.738462
## 160 3.515462 3.304655 3.726268
## 161 3.526721 3.316085 3.737356
## 162 3.606660 3.396335 3.816984
## 163 3.606660 3.396335 3.816984
## 164 3.744747 3.533034 3.956460
## 165 2.594088 2.383573 2.804602
## 166 2.692131 2.482090 2.902172
## 167 2.705638 2.494979 2.916296
## 168 2.793202 2.582919 3.003485
## 169 2.875426 2.664391 3.086460
## 170 2.875426 2.664391 3.086460
## 171 2.875426 2.664391 3.086460
## 172 3.535978 3.326190 3.745766
## 173 3.535978 3.326190 3.745766
## 174 3.535978 3.326190 3.745766
## 175 3.523181 3.312783 3.733578
## 176 3.534440 3.324210 3.744670
## 177 3.534440 3.324210 3.744670
## 178 3.614379 3.404453 3.824304
## 179 2.578261 2.367383 2.789139
## 180 2.578261 2.367383 2.789139
## 181 2.601807 2.391697 2.811916
## 182 2.713357 2.503107 2.923606
## 183 2.713357 2.503107 2.923606
## 184 2.800921 2.591047 3.010795
## 185 2.883144 2.672515 3.093774
## 186 3.463214 3.252469 3.673960
## 187 3.463214 3.252469 3.673960
## 188 3.541901 3.332262 3.751540
## 189 3.541901 3.332262 3.751540
## 190 3.529103 3.318851 3.739356
## 191 3.540363 3.330275 3.750450
## 192 3.620301 3.410513 3.830090
## 193 3.620301 3.410513 3.830090
## 194 3.758389 3.547166 3.969611
## 195 2.584184 2.373443 2.794925
## 196 2.584184 2.373443 2.794925
## 197 2.584184 2.373443 2.794925
## 198 2.607729 2.397762 2.817697
## 199 2.705773 2.496282 2.915263
## 200 2.705773 2.496282 2.915263
## 201 2.705773 2.496282 2.915263
## 202 2.806844 2.597114 3.016574
## 203 2.806844 2.597114 3.016574
## 204 2.806844 2.597114 3.016574
## 205 2.889067 2.678581 3.099554
## 206 3.545356 3.335274 3.755438
## 207 3.625295 3.415507 3.835082
## 208 3.763382 3.552145 3.974619
## 209 3.763382 3.552145 3.974619
## 210 3.763382 3.552145 3.974619
## 211 2.612723 2.402761 2.822685
## 212 2.612723 2.402761 2.822685
## 213 2.710766 2.501282 2.920250
## 214 2.724273 2.514177 2.934369
## 215 2.724273 2.514177 2.934369
## 216 2.724273 2.514177 2.934369
## 217 2.811837 2.602115 3.021559
## 218 2.811837 2.602115 3.021559
## 219 2.894061 2.683580 3.104541
## 220 3.472607 3.261812 3.683401
## 221 3.551294 3.341589 3.760998
## 222 3.538496 3.328173 3.748818
## 223 3.629694 3.419821 3.839567
## 224 3.629694 3.419821 3.839567
## 225 3.767781 3.556446 3.979116
## 226 2.593576 2.382752 2.804401
## 227 2.715165 2.505601 2.924730
## 228 2.728672 2.518497 2.938847
## 229 2.816236 2.606434 3.026038
## 230 2.816236 2.606434 3.026038
## 231 2.898460 2.687898 3.109021
```