ACSC/STAT 3740, Predictive Analytics

WINTER 2023

Toby Kenney

Homework Sheet 4

Model Solutions

Note: All data sets in this homework are simulated.

Standard Questions

1. *The file* `HW4Q1.txt` *contains data on the relation between economic policy and child poverty rates. The data set contains the following variables:*

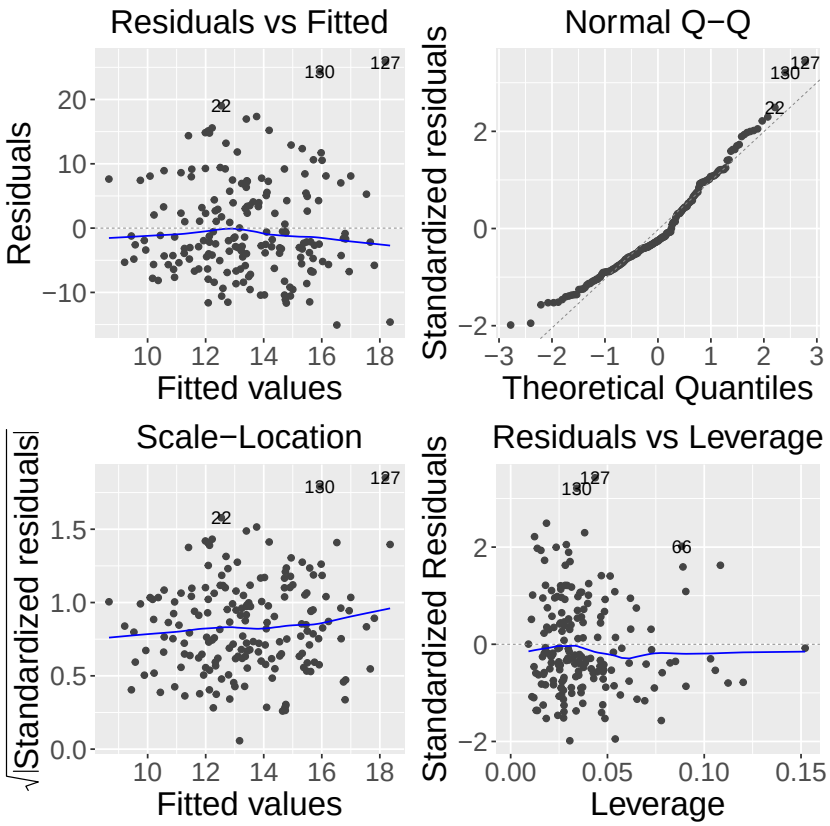
Variable	*Meaning*
base.tax	*The lowest rate of income tax*
top.tax	*The highest marginal rate of income tax*
gdp	*The per.capita gdp*
free.health	*Whether the country has government-provided healthcare*
free.school.years	*Number of years of government-funded education*
free.higher.edu	*Whether the government funds higher education.*
child.poverty	*The percentage of children living in poverty*

A data analyst uses the following code to fit a linear regression model to the data.

```
HW4Q1<-read.table("HW4Q1.txt")
HW4Q1_linear<-lm(child.poverty~.,data=HW4Q1)
```

Use appropriate diagnostics to assess how appropriate the assumptions of the linear regression model are. What changes would you suggest making to the model to better model the data?

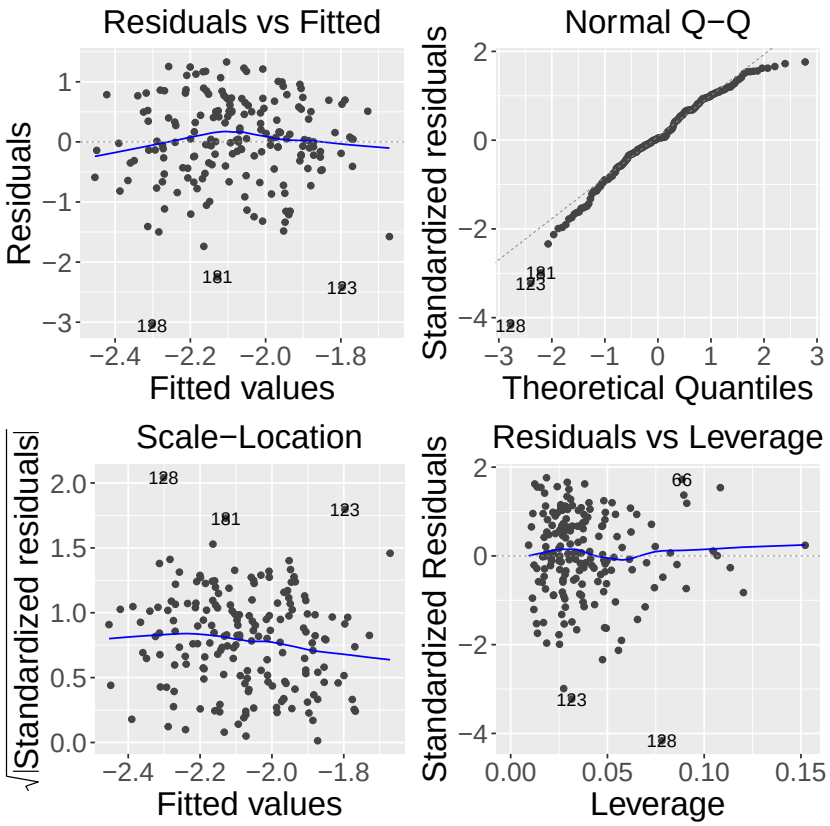
We start by making plots of residuals vs. fitted values; Q-Q plots of residuals; Scale vs. location; and Cook's distance vs. leverage



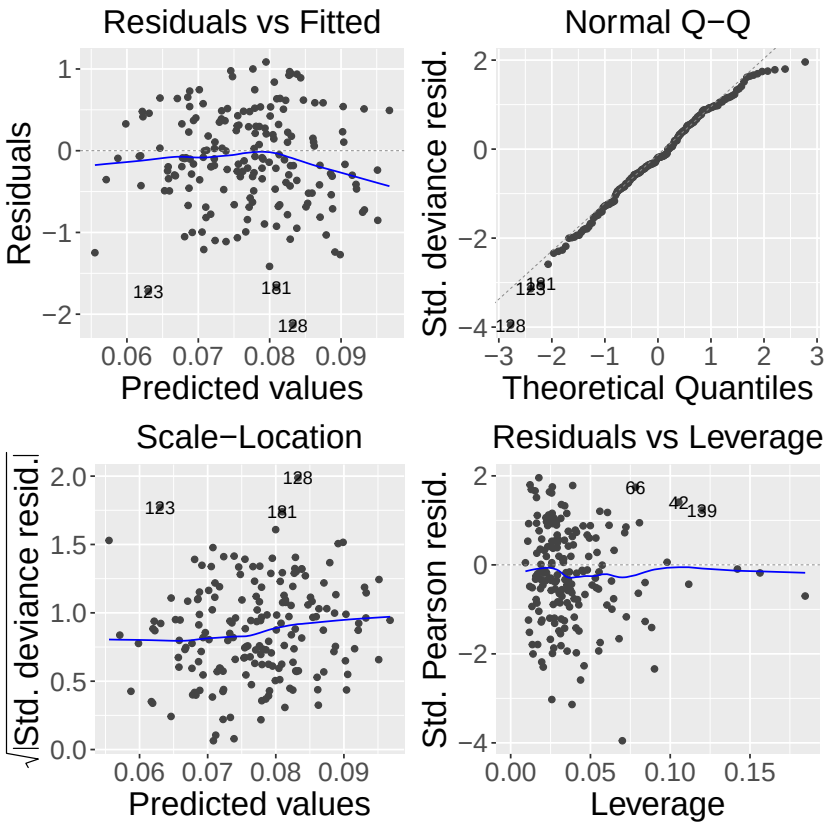
We note the following:

- Residuals seem to be independent of fitted values.
- There are two slight outliers.
- Residuals are positively skewed, so not normal.
- The outliers do not have a huge influence on the slope of the line.

The first thing to do it to remove the outliers, and either transform the response variable, or use a generalised linear model. Since the response is a percentage, a logistic transformation may be possible. Sometimes a transformation can affect heteroskedacity and normality of residuals. We will refit the model with the logistic transformation and outliers removed, then make the usual diagnostic plots.



This model also does not appear to suffer from non-linear effects or heteroskedasticity, but the residuals are now negatively skewed, with a few outliers. We try a generalised linear model with a gamma response.



This time, the diagnostic plots indicate that the model fits the data fairly well. We should also compare cross-validated or test predictions from these models to confirm that they fit better.

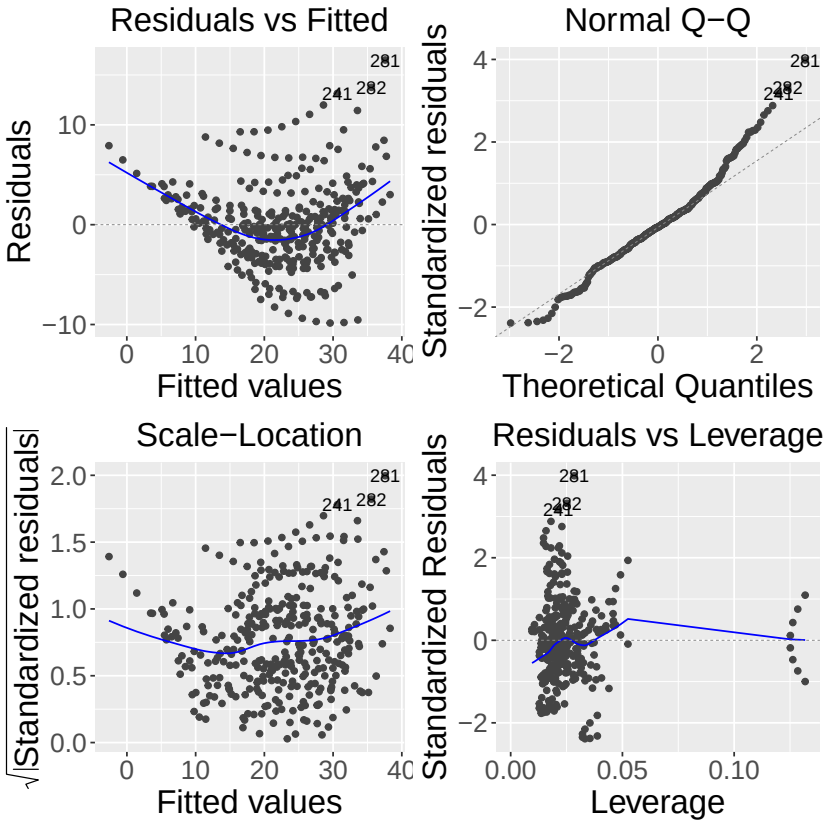
2. *A data scientist at a car manufacturing company is analysing data about engine efficiency in the file* `HW4Q2.txt`.

Variable	*Meaning*
cylinder.number	*The number of cylinders*
fuel.type	*Regular, premium, diesel or electric*
vehicle.weight	*The weight of the vehicle.*
vehicle.speed	*The speed at which the vehicle is being driven*
vehicle.make	*The manufacturer of the vehicle*
mpg	*The vehicles miles per gallon*

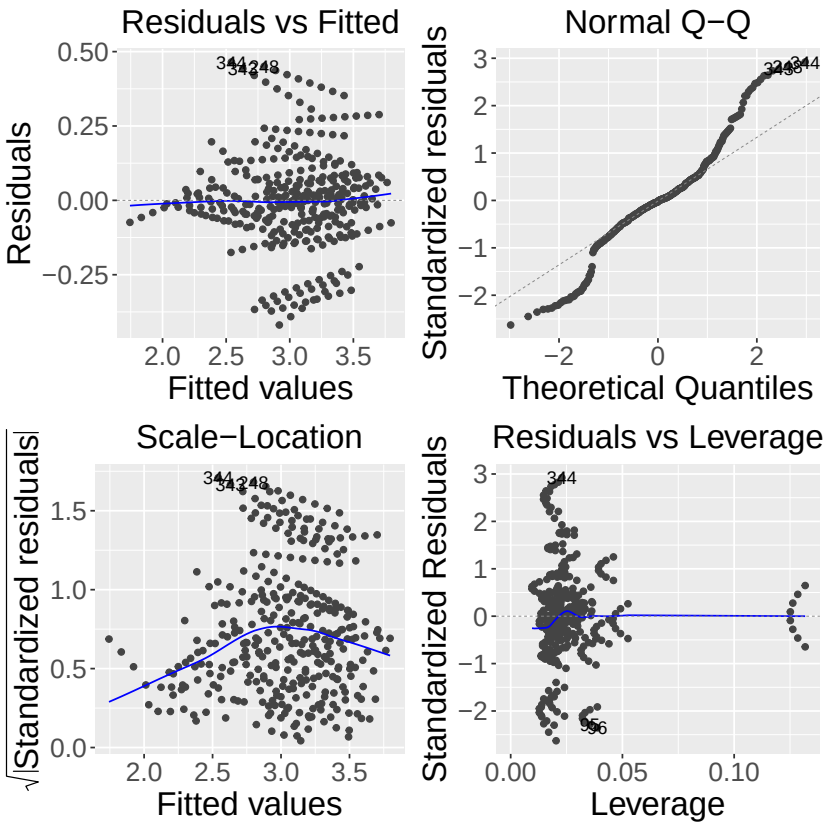
He has fitted a linear model to predict mpg, using the code in the file `HW4Q2_linear.R`. *Perform diagnostics to test which of the assumptions of this model are reasonable. What changes would you suggest making to the model to better model the data?*

We first plot the usual diagnostic plots of residuals vs. fitted values; Q-Q

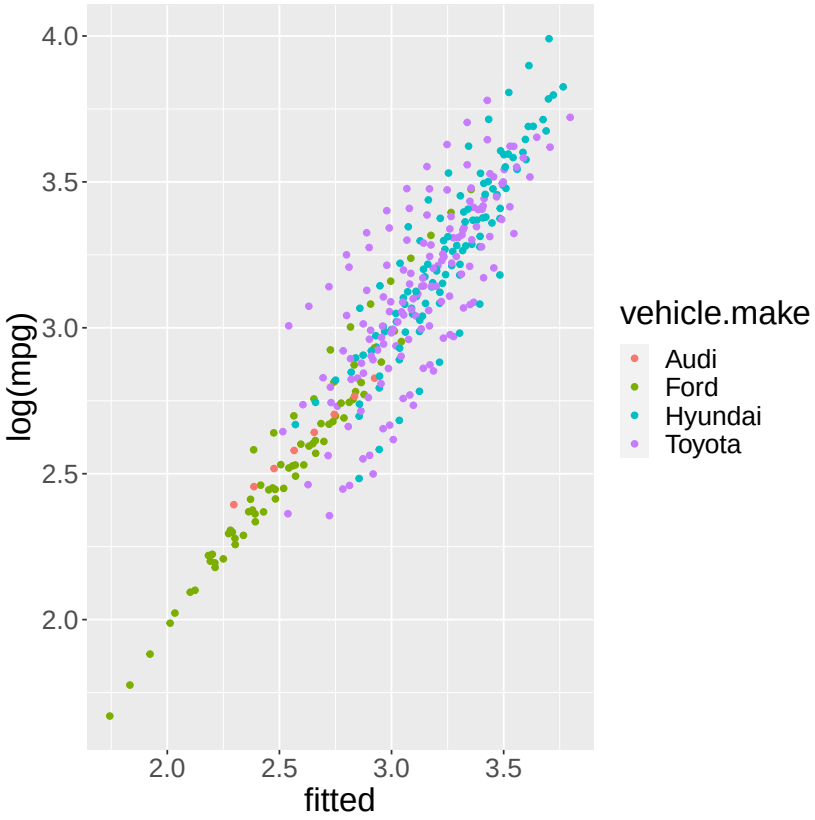
plots of residuals; Scale vs. location; and Cook's distance vs. leverage



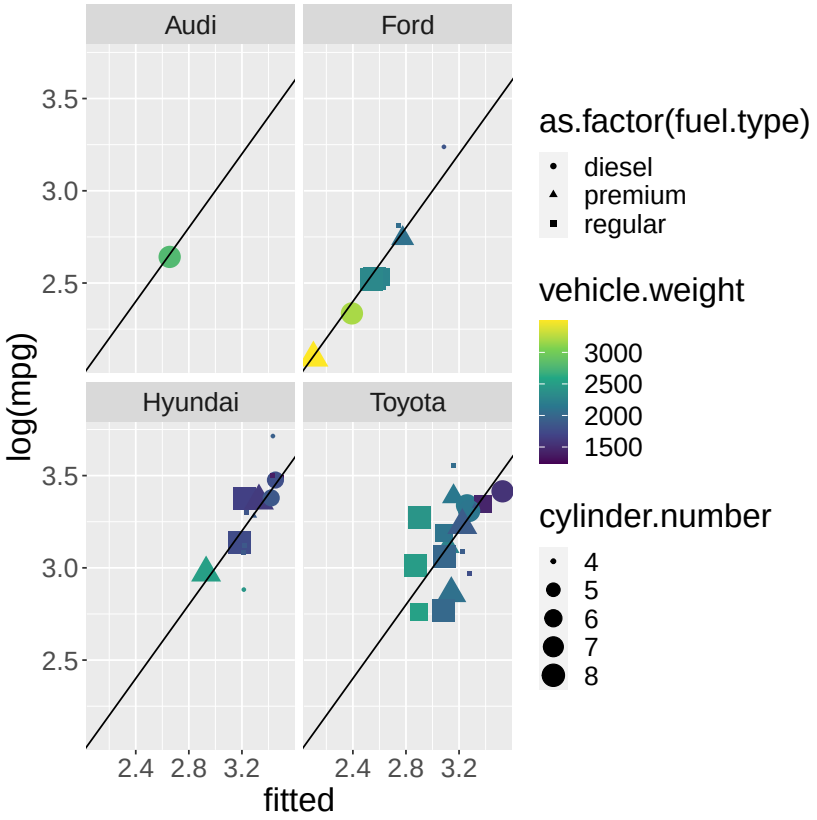
We see a slightly strange pattern in the residuals, with several separate curves. This type of pattern is often caused by discrete (or rounded) response variables. In this case, we see that `mpg` is rounded to the nearest whole number. We also see a very nonlinear pattern in the residuals, suggesting either a transformation of the response variable or the inclusion of non-linear transformations of predictors are appropriate. It seems that the variance of `mpg` increases with fitted value, which suggests a log-transformation of `mpg`. There are also a number of high leverage points. After fitting a model with log-transformed `mpg`, we get the following diagnostic plots:



The mean residuals are closer to linear, but there is still some heteroskedasticity, and the residuals are very non-normal. To help interpret these results, we colour the plots of fitted values against true values by vehicle make.



We see that the variance of the residuals is different for different makes of vehicle. We also see that the data points are in very regular patterns. This might suggest that some predictors behave differently for different makes, which could be modelled using interaction terms. To assess this, we include a plot that shows more of the predictors. Since the patterns for different vehicle speeds seem very regular, we have selected a single speed for each vehicle to make the plot easier to follow.



We see that the variance of the residuals is different for different vehicle makes, but the residuals do not show any relation to other predictors, which would be expected if interaction terms could resolve the issue.

3. *A scientist is reviewing data about the relation between the strength of a material and the production technique, in the file* `HW4Q3.txt`.

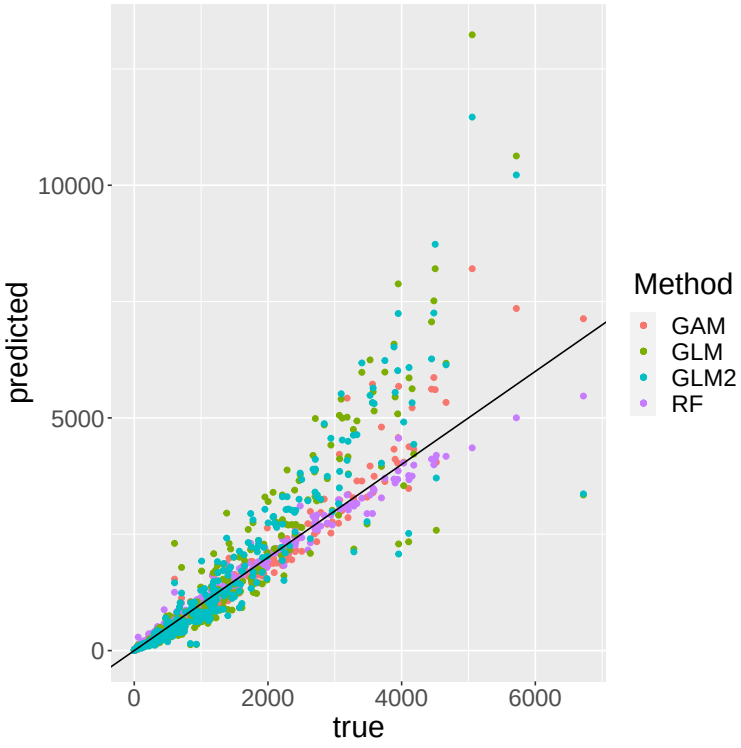
Variable	*Meaning*
carbon.proportion	*The proportion of carbon in the mixture*
titanium.proportion	*The proportion of titanium in the mixture*
production.temp	*The temperature used to produce the material*
production.pressure	*The pressure used to produce the material*
cooling.time	*The time period over which the mixture is allowed to cool*
tensile.strength	*The strength of the eventual material*

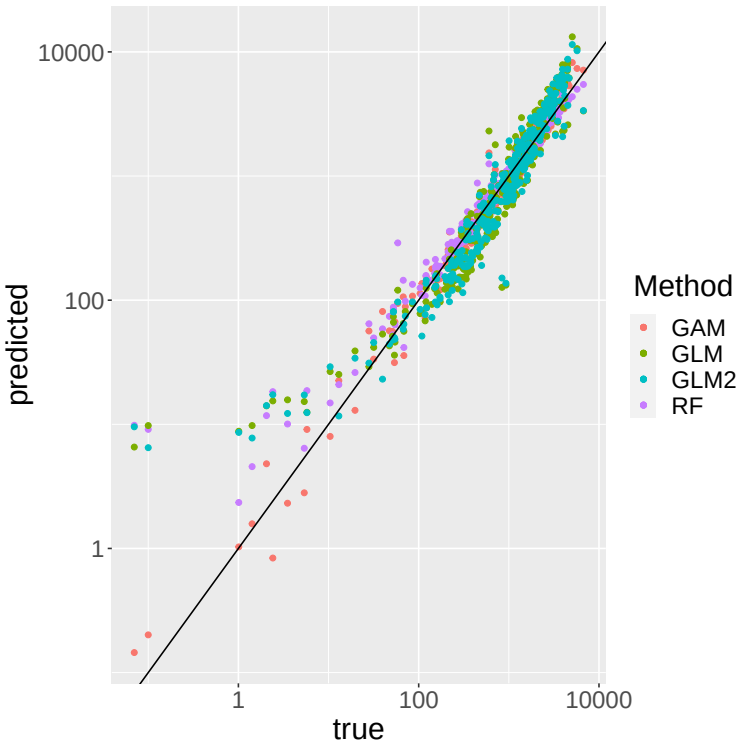
She has fitted a generalised additive model, a random forest model and a generalised linear model including a number of interaction terms and polynomial terms, to predict the total damage, using the code in the file `HW4Q3_models.R`. *Assess which of these models is better at predicting the data. [You may need to modify the code provided to do this.]*

The simplest approach is to divide the data into a training and test data set, and compare predictive performance on the test data. Since some of the models are fitted on a log-transformed scale, and some are fitted on the original scale, we should compare performance on both scales.

```
HW4Q3<-read.table("HW4Q3.txt")

library(mgcv)
library(caret)
library(dplyr)



n<-423

train.index<-createDataPartition(HW4Q3$tensile.strength,  p = 0.75,  list = TRUE)[[1]]

HW4Q3.train<-HW4Q3[train.index,]
HW4Q3.test<-HW4Q3[train.index,]


### Creating stratified folds makes unequal fold sizes



### Fit a smooth function on cooling.time , production.temp and
### production.pressure, but not on carbon.proportion and
### titanium.proportion. The dataset is fairly small, so fitting a
### model with too many degrees of freedom can be inaccurate.

GAM.Model.train<-gam(log(tensile.strength)~s(cooling.time)+
                           carbon.proportion+
                           titanium.proportion+
                           s(production.temp)+
                           s(production.pressure),
                     data=HW4Q3.train)

### Random forest is fairly straightforward. On my computer, 500 trees
### does not take long, because it is a small data set. If it is
### slower on your computer, you can try reducing ntree, though I
### doubt that will be necessary.

RF.Model.train<-train(HW4Q3.train[,-6],
                      HW4Q3.train[,6],
                       trControl=trainControl(method="repeatedcv",number=10,repeats=2),
                       tuneGrid=expand.grid(mtry=seq_len(5)),ntree=500)

### Include quadratic terms for the predictors where we fitted smooth
### functions in the GAM above.

GLM.Formula<-log(tensile.strength)~cooling.time+I(cooling.time^2)+
     carbon.proportion+
     titanium.proportion+
     production.temp+I(production.temp^2)+
     production.pressure+I(production.pressure^2)


GLM.Model.train<-lm(GLM.Formula,data=HW4Q3.train)


GLM.Formula2<-log(tensile.strength)~cooling.time+I(cooling.time^2)+
     carbon.proportion+
     titanium.proportion+
     I(carbon.proportion*titanium.proportion)+
     production.temp+I(production.temp^2)+
     production.pressure+I(production.pressure^2)


GLM.Model2.train<-lm(GLM.Formula2,data=HW4Q3.train)


### get test errors:
```

This gives the following predictions





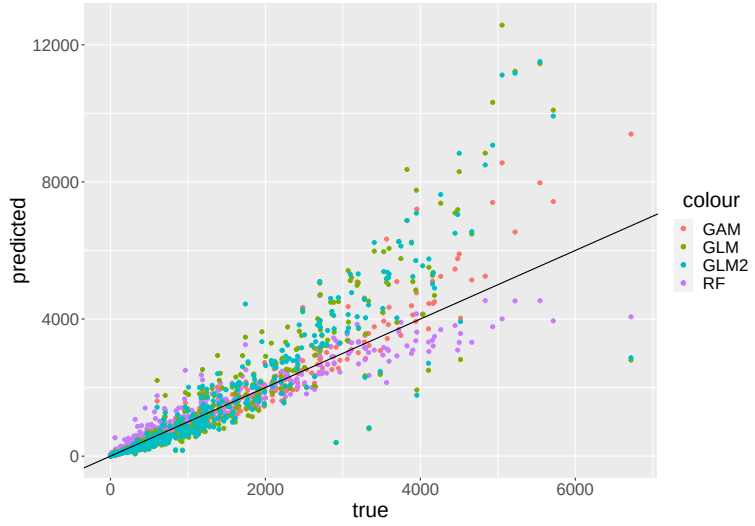
And the following MSEs.

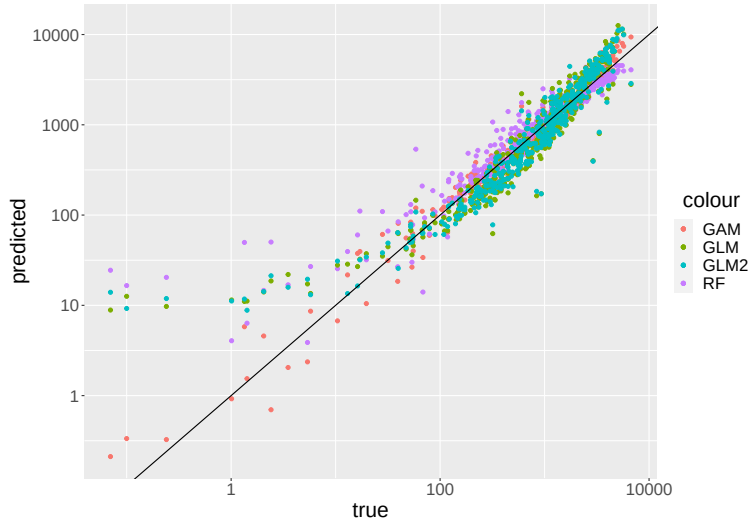
Method	MSE	log-transformed MSE
GAM	137413.68	0.04555893
RF	32393.24	0.21250892
GLM	905671.43	0.35964580
GLM2	729002.94	0.33281756

A better approach is to find the cross-validated predictions.

```
### Use cross-validation for a better estimate of prediction errors

n<-423

nFold<-9        #Make 9 folds of size 47

Folds<-createFolds(as.factor(rep(1,n)),k=nFold)
### Creating stratified folds makes unequal fold sizes

predicted.values<-as.data.frame(matrix(0,n,5)) # prepare matrix for answers


colnames(predicted.values)<-c("GAM","RF","GLM","GLM2","true")

predicted.values$true<-HW4Q3$tensile.strength

for(i in seq_len(nFold)){
    train.data<-HW4Q3[-Folds[[i]],]
    test.data<-HW4Q3[Folds[[i]],]

### Fit a smooth function on cooling.time , production.temp and
### production.pressure, but not on carbon.proportion and
### titanium.proportion. The dataset is fairly small, so fitting a
### model with too many degrees of freedom can be inaccurate.

    GAM.Model<-gam(log(tensile.strength)~s(cooling.time)+
                       carbon.proportion+
                       titanium.proportion+
                       s(production.temp)+
                       s(production.pressure),
                   data=train.data)

### Random forest is fairly straightforward. On my computer, 500 trees
### does not take long, because it is a small data set. If it is
### slower on your computer, you can try reducing ntree, though I
### doubt that will be necessary.

    RF.Model<-train(train.data[,-6],
                    train.data[,6],
                    trControl=trainControl(method="repeatedcv",number=10,repeats=2),
                    tuneGrid=expand.grid(mtry=seq_len(5)),ntree=500)

### Include quadratic terms for the predictors where we fitted smooth
### functions in the GAM above.

    GLM.Formula<-log(tensile.strength)~cooling.time+I(cooling.time^2)+
        carbon.proportion+
        titanium.proportion+
        production.temp+I(production.temp^2)+
        production.pressure+I(production.pressure^2)


    GLM.Model<-lm(GLM.Formula,data=train.data)


    GLM.Formula2<-log(tensile.strength)~cooling.time+I(cooling.time^2)+
        carbon.proportion+
        titanium.proportion+
        I(carbon.proportion*titanium.proportion)+
        production.temp+I(production.temp^2)+
        production.pressure+I(production.pressure^2)


    GLM.Model2<-lm(GLM.Formula2,data=train.data)

    predicted.values$GAM  [Folds[[i]]]<-exp(predict(GAM.Model,newdata=test.data))
    predicted.values$RF   [Folds[[i]]]<-predict(RF.Model,newdata=test.data)
    predicted.values$GLM  [Folds[[i]]]<-exp(predict(GLM.Model,newdata=test.data))
    predicted.values$GLM2 [Folds[[i]]]<-exp(predict(GLM.Model2,newdata=test.data))
```

This gives the following predictions





Method	MSE	log-transformed MSE
GAM	223674.8	0.06675256
RF	170575.3	0.40338575
GLM	1098512.3	0.39341624
GLM2	941347.3	0.37182761

In both cases, random forest performs better on the original scale, while the GAM performs better on the log-transformed scale.

4. *The file* `HW4Q4.txt` *contains data from an insurance company about the probability that a settlement offer is accepted. The data set contains the*

following variables:

Variable	*Meaning*
accident.year	*The year of the accident*
number.affected	*The number of individuals affected by the accident*
property.damage	*The estimated amount of property damage.*
injury.loss	*The direct loss due to injury.*
injured.sex	*The sex of the injured party.*
injured.age	*The age of the injured party.*
injured.salary	*The salary of the injured individual.*
settlement.amount	*The amount of settlement offered.*
settlement.accepted	*Whether the settlement was accepted.*

A data analyst uses the following code to fit a decision tree to the data:

```
Reaction_data<-read.table("HW4Q4.txt")

library(rpart)

Reaction_dt<-rpart(formula=reaction.time~.,
                   data=Reaction_data,
                   control=rpart.control(minbucket=10, # smallest size of node
                                         maxdepth=10)) # largest depth of tree.
```

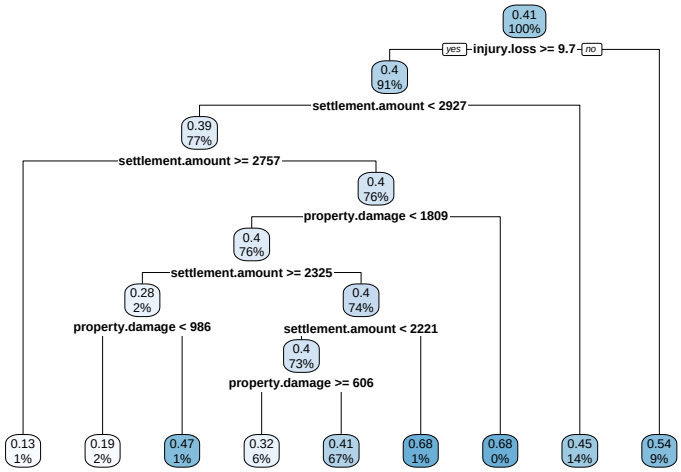
and uses the following code to select variables using stepwise regression with AIC:

```
Reaction_Null_model<-lm(reaction.time~1,data=Reaction_data)

Reaction_Full_model<-lm(reaction.time~.,data=Reaction_data)


library(MASS)
Reaction_Forward<-stepAIC(Reaction_Null_model,
                          direction="forward",
                          scope=list(lower=Reaction_Null_model,
                                     upper=Reaction_Full_model))
```

The code is in the files `HW4_Q4_Decision_tree.R` *and* `HW4_Q4_Stepwise_AIC.R` *respectively.*

Based on the results of these analyses, how could he try to adjust the models to better fit the data?

We first examine the fitted decision tree:

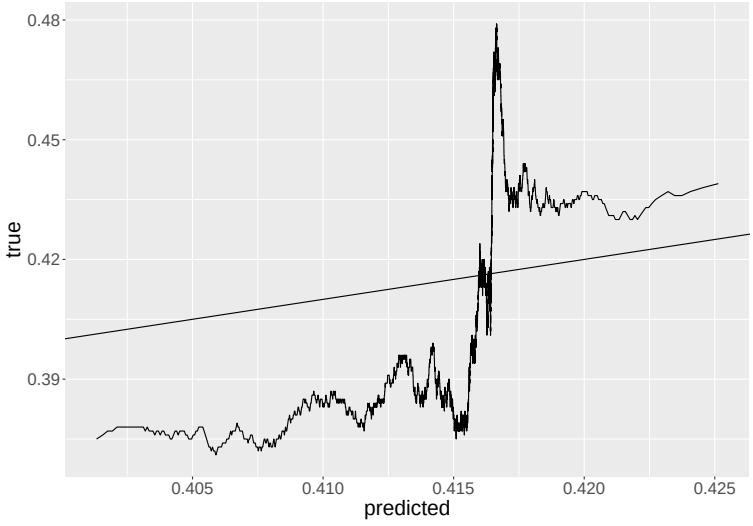


We see that the variables `injury.loss`, `settlement.amount` and `property.damage` are the most important. This is also found by the forward selection method, which selects only `injury.loss` and `settlement.amount`.

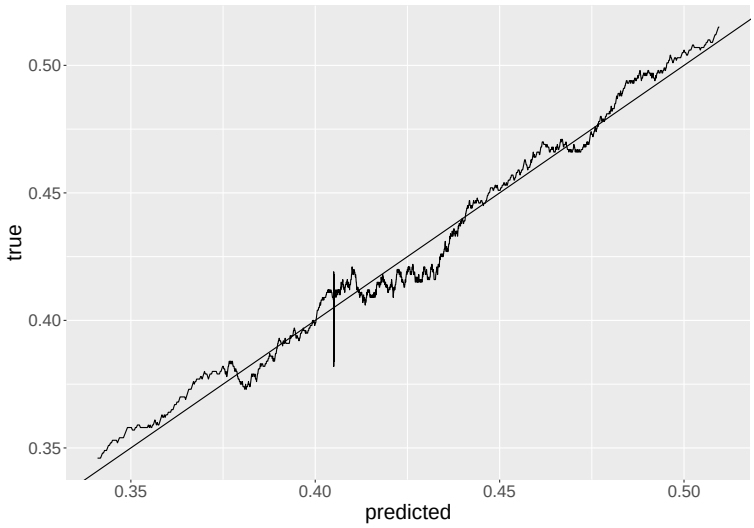
The decision tree could certainly be better tuned, possibly by using cross-validation to select the complexity parameter.

The structure of the tree, with different variables interacting suggests that interaction terms may be helpful. We also plot the moving averages of the predicted and true probabilities of acceptance for the two methods.

Forward selection:



Decision tree:



We see that for both methods, the predicted probability of acceptance does not vary very much between points. The decision tree is fairly well callibrated (but note that these are traning predictions, so there could still be overfitting, leading to miscallibration). The forward selection method shows clear signs of miscallibration, indicating that the probability is likely to be a non-linear function of the predictors. We should therefore add interaction terms or higher order terms (possibly a GAM would be appropriate).

Random forest is often a good approach to improve accuracy for tree-based methods. It should at least be compared for this dataset.