ACSC/STAT 3740, Predictive Analytics

WINTER 2025

Toby Kenney

Homework Sheet 4

Model Solutions

Note: All data sets in this homework are simulated.

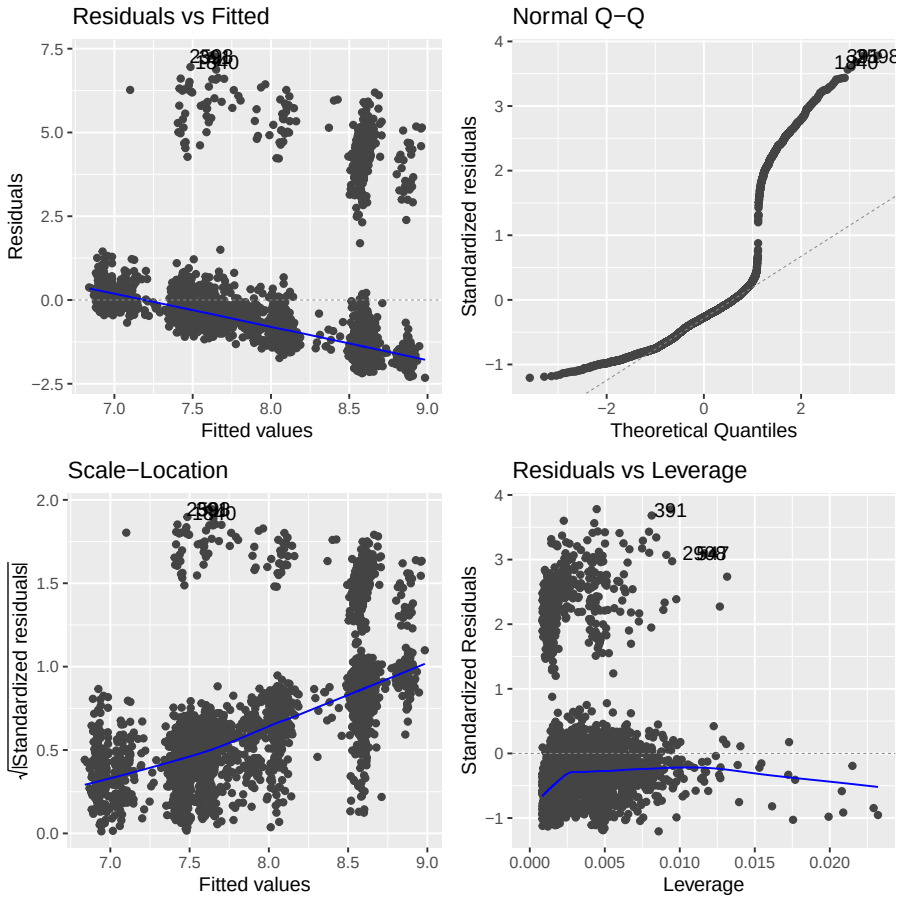
Standard Questions

1. *A home insurance company has collected the following data about fire damage in the file* `HW4Q1`.

Variable	*Meaning*
material	*The main material used to build the house.*
living.area	*The living area of the house.*
recent.rain	*The amount of rainfall in the week preceding the fire.*
fire.alarm	*Whether the home is equipped with fire alarms.*
sprinkler	*Whether the home is equipped with sprinklers.*
occupied	*Whether the home was occupied at the time of the fire.*
fire.station.distance	*The distance from the home to the nearest fire station.*
damage	*The total cost to repair the house.*

They have used the code in the the file `HW4Q1_code` *to fit a linear regression model to predict the treatment outcome for each patient. 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 standard diagnostics.



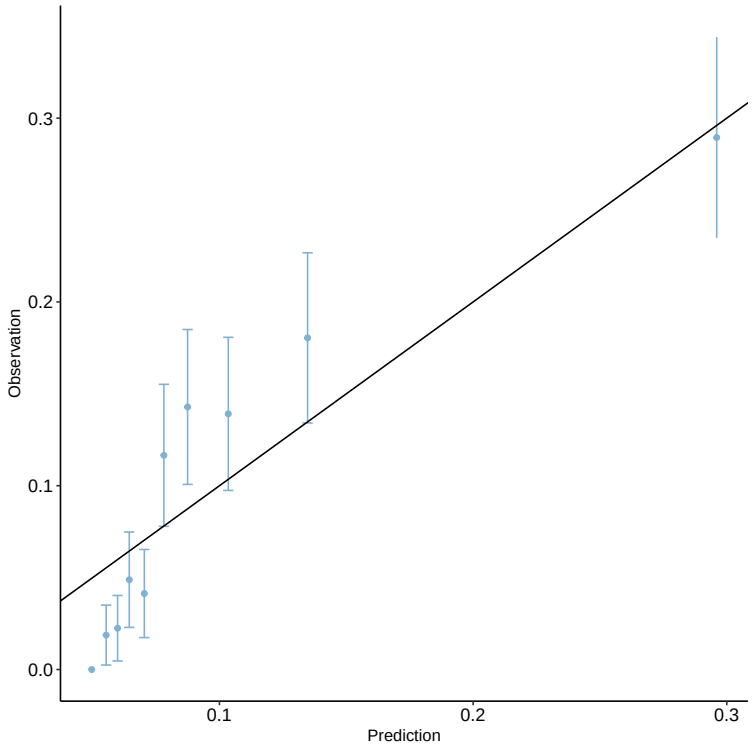
We immediately see that the points are in two clusters, indicating that the response is bimodal, and clearly therefore not normal. Given the bimodal distribution of the response, one approach would be to fit a two-part model, where we first estimate the probability of damage being high or low, then estimate the conditional mean given that the damage is high or low.

2. *A company has collected the following data about customer service in the file* `HW4Q2`.

Variable	*Meaning*
age	*The age of the customer.*
gender	*The gender of the customer.*
previous.customer	*Whether the customer bought anything within the past 12 months.*
amount.spent	*The amount the customer spent.*
agent.experience	*The number of years of experience the customer service agent had.*
agent.gender	*The agent's gender.*
time	*The time spent with the customer (minutes)*
rating	*The customer's rating of the agent.*

They have used the code in the file `HW4Q2_code.R` *to predict the rating given in each case. Assess the model assumptions and predictive performance of the model. How might the model be improved?*

We start with a calibration plot.



We see that the model is not well calibrated, underestimating the small probabilities and overestimating the large probabilities. This is a calibration plot on training data, so there is danger of overfitting. However, overfitting would not usually cause the calibration plot to look like this. Therefore, the plot shows that the model is miscalibrated. Generally, either transforming a predictor or including interaction terms can improve the calibration. Another possibility is to change the link function. How-

ever, some choices of link function can cause difficulty converging.

We also note that the probabilities are small, suggesting that the cut-off used for rating might be too large. Using a lower cut-off would result in a more balanced dataset, which might allow better fitting and potentially a better logistic regression, though this is much less reliable than adding transformations or interactions.

```
library(predtools)
calibration_plot(data.frame(obs=Rating.Data$rating>6,
                            pred=predict(glm.model,type="response")),
                 obs="obs",pred="pred")
```

3. *A health researcher is studying the effect of access to a family doctor on long-term health outcomes. She has collected the following data in the file* `HW4Q3`*.*

Variable	*Meaning*
population	*The population of the region*
family.doctors	*Number family doctors in the region.*
ave.travel	*The average time an inhabitant must travel to attend an appointment.*
over.sixty	*The proportion of the population over 60.*
ave.income	*The average income in the region.*
cancer.deaths	*The number of cancer deaths.*
heart.deaths	*The number of deaths caused by heart problems.*

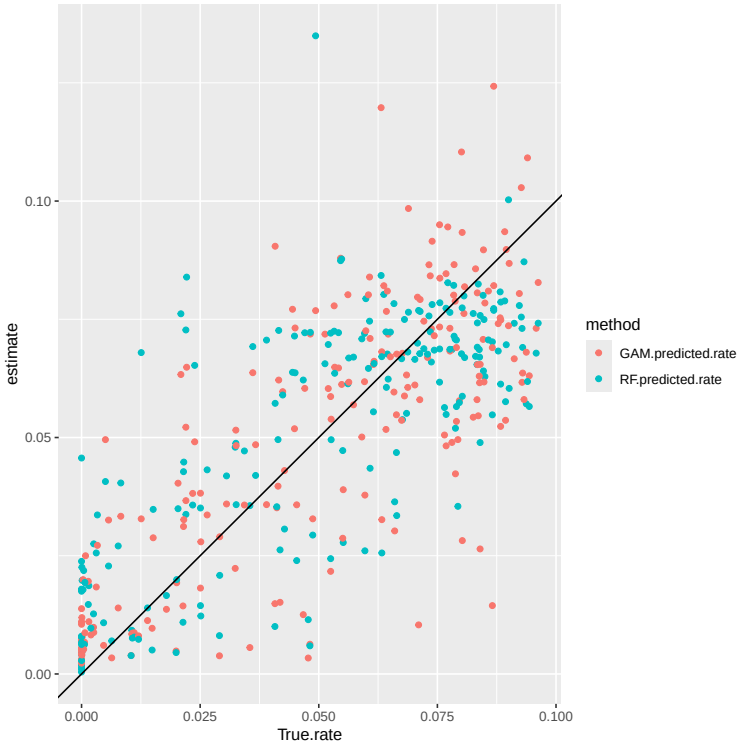
They have used the code in the file `HW4Q3code` *to fit two models to predict heart death rates. The first model is a GAM. The second is a random forest model. Determine which model is better for predicting the heart death rates. [You may need to modify the code for this.]*

We use cross-validation to assess the models. As the data set is relatively small, a single training-test split would not produce very accurate error estimates.

There are three different metrics we can use to assess performance. The results are given in the following table:

	GAM	Random Forest
MSE of predicted number of deaths	132024.2	130867.7
MSE of predicted rate of deaths	0.0004003498	0.0003983952
Poisson log-likelihood	1045010	1044977

We see that Random forest has slightly lower MSE both on the rate scale, and on the total deaths scale. On the other hand, the GAM has slightly higher log-likelihood. Given the small differences, there is not much difference between the methods, and the difference could be due to randomness.



The reason that log-likelihood and MSE give different results could be because the log-likelihood places more emphasis on relative errors, whereas MSE places more emphasis on absolute errors.

The following code was used to perform this cross-validation.

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

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

nfold<-10 # 10 folds

Folds<-createFolds(Health.Data$heart.deaths/Health.Data$population,k=nfold)
### Use per-capita deaths to stratify. Create 10 folds for cross-validation

Predictions<-matrix(0,dim(Health.Data)[1],2)
###Create an empty matrix to store the predictions.

for(i in seq_len(nfold)){
    test<-Folds[[i]]
    training.data<-Health.Data[-test,] # This will fail if any fold is empty.
    test.data<-Health.Data[test,]

    Health.GAM<-gam(heart.deaths~s(per.capita.doctors)+s(ave.travel)+s(over.sixty)+s(ave.income),
                    data=training.data%>%mutate(per.capita.doctors=family.doctors/population),
                    offset=log(population),family=poisson(link="log"))

    Predictions[test,1]<-predict(Health.GAM,newdata=test.data%>%
                                                    mutate(per.capita.doctors=family.doctors/population),
                                 type="response")

    Health.RF<-train(heart.deaths~.-cancer.deaths,data=training.data,method="rf",
                     trControl=trainControl(method="repeatedcv",repeats=2,number=10),
                     tuneGrid=expand.grid(mtry=seq_len(5)),ntree=100)

    Predictions[test,2]<-predict(Health.RF,newdata=test.data)
}

Results<-data.frame(
    GAM.predicted.rate=Predictions[,1],
    GAM.predicted.deaths=Predictions[,1]*Health.Data$population,
    RF.predicted.rate=Predictions[,2]/Health.Data$population,
    RF.predicted.deaths=Predictions[,2],
    True.rate=Health.Data$heart.deaths/Health.Data$population,
    True.deaths=Health.Data$heart.deaths)

### Plot fitted versus true value for both methods.
library(ggplot2)
library(dplyr)
library(tidyr)

ggplot(Results%>%pivot_longer(cols=c(1,3),
                              names_to="method",
                              values_to="estimate"),
       mapping=aes(x=True.rate,y=estimate,colour=method))+
    geom_point()+
    geom_abline(mapping=aes(slope=1,intercept=0))

Results.summary<-data.frame(GAM=rep(NA,3),RF=rep(NA,3))
rownames(Results.summary)<-c("MSE.rate","MSE.deaths","log.likelihood")


      Results.summary["MSE.rate","GAM"]<-mean((Results$GAM.predicted.rate-Results$True.rate)^2)
    Results.summary["MSE.deaths","GAM"]<-mean((Results$GAM.predicted.deaths-Results$True.deaths)^2)
Results.summary["log.likelihood","GAM"]<-sum(log(Results$GAM.predicted.deaths)*Results$True.deaths-Results

      Results.summary["MSE.rate","RF"]<-mean((Results$RF.predicted.rate-Results$True.rate)^2)
    Results.summary["MSE.deaths","RF"]<-mean((Results$RF.predicted.deaths-Results$True.deaths)^2)
Results.summary["log.likelihood","RF"]<-sum(log(Results$RF.predicted.deaths)*Results$True.deaths-Results$
```

4. *A doctor is studying the effect of antibiotic usage on obesity. He has collected the following data in the file* `HW4Q4`.

Variable	*Meaning*
patient.BMI.before	*The patient's BMI before the start of treatment.*
patient.age	*The patient's age*
patient.sex	*The patient's sex.*
antibiotic.dosage	*The total dosage of antibiotics prescribed.*
patient.BMI.after	*The patient's BMI 6 months after the start of treatment.*

He has used the code in the file `HW4Q4_code` *to fit two GAM models to predict patient BMI afterwards, with different choices for nonlinear terms. Determine which model is better for predicting patient BMI afterwards.*

We perform cross-validation for this. As the data set is relatively large, a single training-test split might also give reasonable results. Since one model is on the original scale, and one has a log-transformed predictor, we could compare on either scale. Since the first model produces negative predictions, we replace the negative predictions by 1. We get the following MSEs:

	Model 1	Model 2
MSE	67.15723	80.43901
log scale MSE	0.2289252	0.1997187

We see that the first model is better at prediction on the original scale, while the second model is better at prediction on the log-scale (which measures something similar to relative error). [We can alternatively measure the relative MSEs as 42.39% and 33.19% for Models 1 and 2 respectively.]

The following code was used to perform this cross-validation.

```
Obesity.Data<-read.table("HW4Q4.txt")

library(mgcv)
library(caret)

nfold<-10 # 10 folds

Folds<-createFolds(Obesity.Data$patient.BMI.after,k=nfold)
### Create 10 stratified folds for cross-validation

Predictions<-matrix(0,dim(Obesity.Data)[1],2)
###Create an empty matrix to store the predictions.

for(i in seq_len(nfold)){
    test<-Folds[[i]]
    training.data<-Obesity.Data[-test,] # This will fail if any fold is empty.
    test.data<-Obesity.Data[test,]



    BMI.GAM<-gam(patient.BMI.after~patient.BMI.before+s(patient.age)+s(antibiotic.dosage)+patient.sex,dat


    Predictions[test,1]<-predict(BMI.GAM,newdata=test.data)

    BMI.GAM2<-gam(log(patient.BMI.after)~log(patient.BMI.before)+s(patient.age)+antibiotic.dosage+patient

    Predictions[test,2]<-predict(BMI.GAM2,newdata=test.data)

}

MSE1<-mean((Predictions[,1]-Obesity.Data$patient.BMI.after)^2)
MSE2<-mean((exp(Predictions[,2])-Obesity.Data$patient.BMI.after)^2)

### Can also compare MSE on a log scale predictions, since model 2 uses
### a log-transformed predictor.

MSE1.log<-mean((log(pmax(Predictions[,1],1))-log(Obesity.Data$patient.BMI.after))^2)
### Model one has some negative predictions, which we replace by 1.

MSE2.log<-mean((Predictions[,2]-log(Obesity.Data$patient.BMI.after))^2)
```