# ACSC/STAT 3740, Predictive Analytics

WINTER 2025
Toby Kenney

Homework Sheet 3

Model Solutions

## Standard Questions

1. *An insurance company has collected the following data on life expectancy in the file HW3Q1.*

| *Variable* | *Meaning* |
|---|---|
| *current.age* | *The individual's current age.* |
| *sex* | *The individual's current sex.* |
| *BMI* | *The individual's BMI* |
| *cigarettes.per.day* | *The average number of cigarettes the individual smokes each day* |
| *daily.exercise* | *The average number of minutes per day spent doing physical exercise* |
| *health.index* | *An index measuring overall health* |
| *survival.five.year* | *Whether the individual survives 5 years* |

*Fit a generalised linear model, with a binomial response variable (and a logistic link function), to predict the probability of dying within 5 years. Use this model to predict the probability of dying for the individuals in the file HW3Q1test .*

We fit the following logistic regression model:

| | Est. | S.d. | $p$-value |
|---|---|---|---|
| (Intercept) | 5.3354193 | 1.0696473 | $6.10 \times 10^{-7}$ |
| current.age | $-0.0352538$ | 0.0085759 | $3.94 \times 10^{-5}$ |
| sexmale | 0.3261624 | 0.2881374 | 0.258 |
| BMI | $-0.0009522$ | 0.0234914 | 0.968 |
| cigarettes.per.day | $-0.0132777$ | 0.0442093 | 0.764 |
| daily.exercise | 0.0303735 | 0.3743734 | 0.935 |
| health.index | $-0.0075536$ | 0.0074890 | 0.313 |

It makes the following predictions for 5-year survival:

| Individual | prediction | Individual | prediction |
|---|---|---|---|
| 1006 | 0.9422064 | 1011 | 0.9646362 |
| 1007 | 0.9875033 | 1012 | 0.9832101 |
| 1008 | 0.9547466 | 1013 | 0.9175382 |
| 1009 | 0.9371595 | 1014 | 0.9889852 |
| 1010 | 0.9566812 | 1015 | 0.9183298 |

The following code was used to fit this model.

```
HW3Q1<-read.table("HW3Q1.txt",stringsAsFactors=TRUE)
HW3Q1_glm<-glm(survival.five.year~.,data=HW3Q1,family=binomial(link="logit"))
summary(HW3Q1_glm)
HW3Q1_test<-read.table("HW3Q1_test.txt",stringsAsFactors=TRUE)
predict(HW3Q1_glm,newdata=HW3Q1_test,type="response")
```

2. *A company is analysing data on the effect of maintainance on productivity in the file HW3Q2.*

| *Variable* | *Meaning* |
|---|---|
| *machine.age* | *The age of the machine.* |
| *machine.operators* | *The number of workers operating the machine.* |
| *machine.preemptive.maintainance* | *The amount spent on pre-emptive maintainance of the machine over the past year.* |
| *machine.corrective.maintainance* | *The amount spent on corrective maintainance of the machine over the past year.* |
| *machine.power* | *The power consumed by the machine.* |
| *machine.output* | *The number of parts produced by the machine.* |
| *machine.defect.rate* | *The proportion of part output by the machine that are defective.* |

*Fit a random forest to predict the machine defect rate from the other predictors. Use this model to predict defect rates for the machines in the file HW3Q2test .*

[Random forest has some randomness, so results may vary.]

The tuning selects `mtry=1`, using cross validation — that is, for each split, one variable is chosen at random. The model gives the following variable importances:

| Variable | Importance |
|---|---|
| machine.power | 100.00 |
| machine.preemptive.maintainance | 98.67 |
| machine.output | 88.31 |
| machine.age | 82.79 |
| machine.corrective.maintainance | 68.68 |
| machine.operators | 0.00 |

and the following predictions:

| Observation | Prediction | Observation | Prediction |
|---|---|---|---|
| 511 | 1.208576 | 516 | 1.148972 |
| 512 | 1.179932 | 517 | 1.854719 |
| 513 | 1.917122 | 518 | 1.655178 |
| 514 | 1.661475 | 519 | 1.180423 |
| 515 | 1.690702 | 520 | 3.015765 |

The following code was used to fit this model.

```
HW3Q2<-read.table("HW3Q2.txt",stringsAsFactors=TRUE)
HW3Q2_test<-read.table("HW3Q2_test.txt",stringsAsFactors=TRUE)
library(caret)
HW3Q2_rf<-train(machine.defect.rate~.,data=HW3Q2,method="rf",
                  trControl=trainControl(method="repeatedcv",number=10,repeats=2),
                  tuneGrid=expand.grid(mtry=seq_len(6)),ntree=500,varImp=TRUE)
varImp(HW3Q2_rf)
predict(HW3Q2_rf,newdata=HW3Q2_test)
```

3. *The file* `HW3Q3.txt` *contains measurements of the total annual rainfall in a certain city over the last century*

   *(a) Fit a quadratic model to estimate log annual rainfall as a function of time.*

   We use the following code:

```
HW3Q3<-read.table("HW3Q3.txt")

library(dplyr)


trend<-lm(log(rainfall)~year+I(year^2),data=HW3Q3)

summary(trend)
```

which gives the model:

| Coefficient | Estimate | Std. Error | $p$-value |
|---|---|---|---|
| (Intercept) | $-670.9$ | 441.2 | 0.132 |
| year | 0.6850 | 0.4469 | 0.129 |
| I(year$^2$) | $-0.0001740$ | 0.0001132 | 0.128 |

*(b) Use AIC to fit the best ARMA model to the residuals of the quadratic model.*

We use the following code:

```
library(forecast)
rain.resid.arma<-auto.arima(trend$residuals,ic="aic",max.d=0)

summary(rain.resid.arma)
```

It selects an ARMA(3,4) model with the following coefficients:

| Coefficient | Estimate | Std. Error |
|---|---|---|
| ar1 | 0.2421 | 0.1835 |
| ar2 | 0.6400 | 0.1792 |
| ar3 | −0.3802 | 0.1355 |
| ma1 | −0.7164 | 0.1868 |
| ma2 | 0.3830 | 0.1934 |
| ma3 | 0.0992 | 0.1629 |
| ma4 | −0.4078 | 0.1480 |

*(c) Fit a GARCH model to model the variance.*

Using the order (3,4) found in the previous part, we use the following code to fit a GARCH model

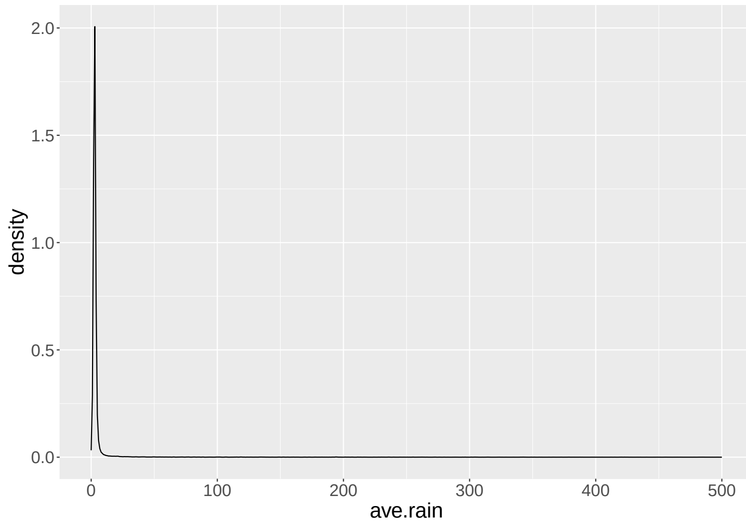
```
library(rugarch)
GARCH_model<-ugarchspec(mean.model=list(armaOrder=c(3,4)), distribution="norm")
GARCH_rain<-ugarchfit(GARCH_model,trend$residuals,solver="hybrid")
## The default solver fails to converge.
GARCH_rain
```

It fits the following model:

| Parameter | Estimate | Std. Error | $p$-value |
|---|---|---|---|
| mu | 0.00000 | 0.032613 | 1.000000 |
| ar1 | 0.22899 | 0.212551 | 0.281318 |
| ar2 | 0.66716 | 0.271391 | 0.013960 |
| ar3 | −0.36869 | 0.161440 | 0.022386 |
| ma1 | −0.71657 | 0.223655 | 0.001356 |
| ma2 | 0.36647 | 0.232203 | 0.114515 |
| ma3 | 0.10722 | 0.180269 | 0.551982 |
| ma4 | −0.43354 | 0.199150 | 0.029486 |
| omega | 0.14756 | 0.384908 | 0.701446 |
| alpha1 | 0.00000 | 0.059414 | 1.000000 |
| beta1 | 0.32819 | 1.606430 | 0.838119 |

*(d) Based on this model, what is the probability that average annual rainfall will exceed 2500 in the decade from 2090 to 2099? [You can use the* `ugarchboot` *function to run a simulation to estimate this.]*

```
GARCH_Bootstraps<-ugarchboot(GARCH_rain,
                             method="full",
                             n.ahead=75,
                             n.bootfit=400, # 400 parameter estimates
                             n.bootpred=400, # 400 bootstraps
                             rseed=seq_len(800)) #Need to explicitly set seed
### rseed needs to be a vector of length n.bootfit+n.bootpred

### This may take a few minutes to run. To make it run faster, you
### could reduce n.bootfit to about 100.  You could also use
### 'method="partial"' to used fixed parameter estimates from
### part (b).


### Calculate Distribution of average annual rainfall over the decade.
GARCH_boot_2090s<-GARCH_Bootstraps@fseries[,66:75]
trend_2090s<-predict(trend,newdata=list("year"=2090:2099))
ave.rain<-rowMeans(exp(GARCH_boot_2090s+
                   rep(1,dim(GARCH_boot_2090s)[1])%*%t(trend_2090s)))
### Remember to add the trend.
### Also remember that we log-transformed rainfall, so we need to exponentiate.
### Parameter estimates do not converge for some simulations
### So use dim(GARCH_boot_2090s)[1] instead of 160000

library(ggplot2)

ggplot(data.frame("ave.rain"=ave.rain),mapping=aes(x=ave.rain))+geom_density()
+largertextsize


mean(ave.rain>2500)
### probability of average rain exceeding 2500.
```




In my bootstap, the probability of this event is 0.1052381. It depends a lot on the bootstrapped predictors, so using the partial method is likely to underestimate this probability. In addition a number of the bootstraps fail, so choosing `bootfit` too small could also lead to volatile estimation.

4. *The file `HW3Q4.txt` contains the following data about school performances in standardised tests for Grade 8:*

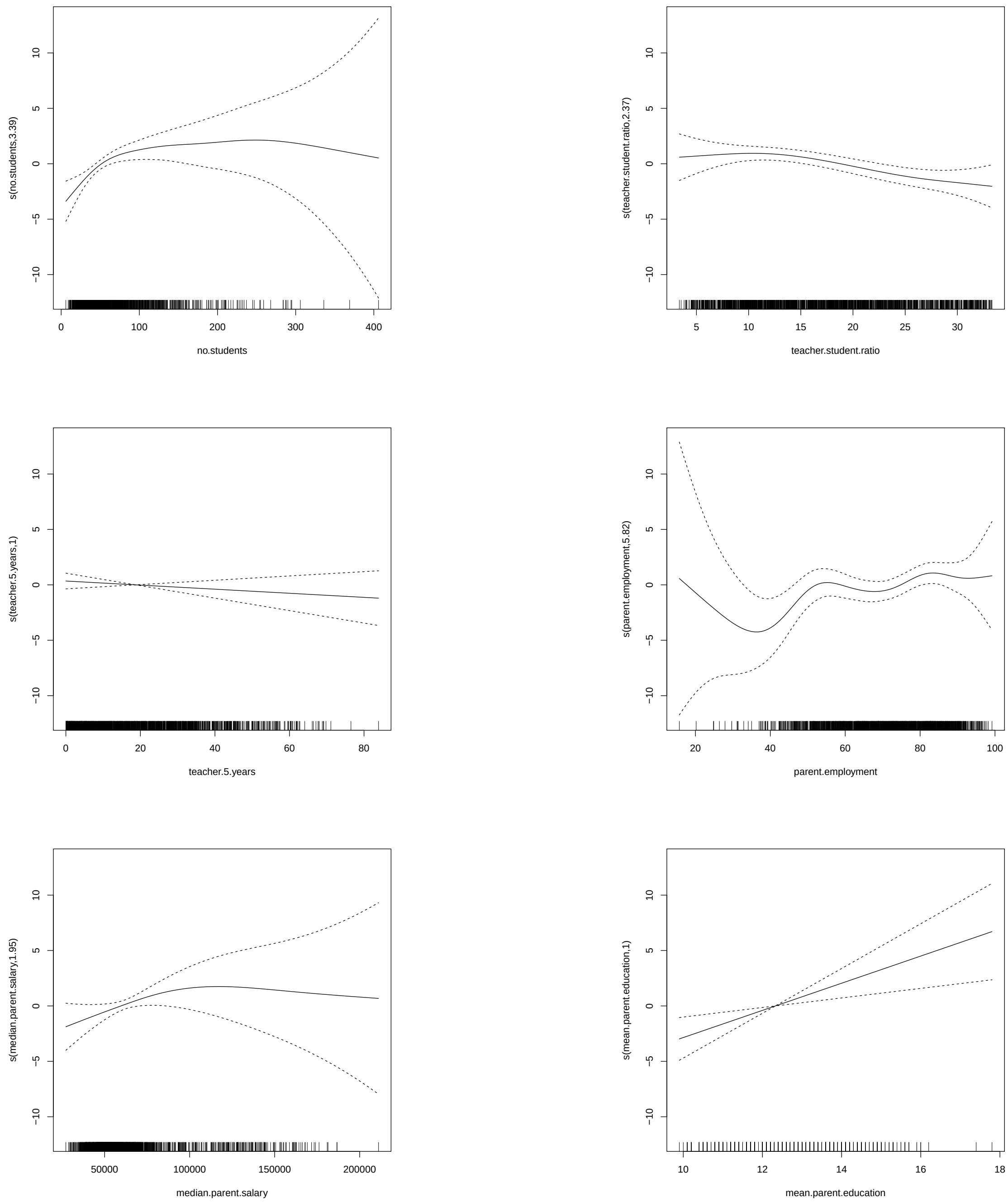
| *Variable* | *Meaning* |
|---|---|
| *no.students* | *The number of students in Grade 8 attending the school.* |
| *teacher.student.ratio* | *The average number of students per teacher in a class at the school.* |
| *funding* | *The schools source of funding — government, independent or private.* |
| *specialist.teacher* | *Whether the school employs teachers with specialist knowledge for each subject.* |
| *teacher.5.years* | *The percentage of teachers at the school with at least 5 years of experience.* |
| *parent.employment* | *The percentage of parents of children at the school who are employed.* |
| *median.parent.salary* | *The median salary of parents of children at the school* |
| *mean.parent.education* | *The average number of years of full-time education of parents of children at the school.* |
| *average.score.mathematics* | *The average score of children in Grade 8 at the school on the standardised mathematics test.* |
| *average.score.english* | *The average score of children in Grade 8 at the school on the standardised English test.* |

*Fit generalised additive models with Gaussian response and identity link function to predict `average.score.mathematics` and `average.score.english` from the other predictors.*

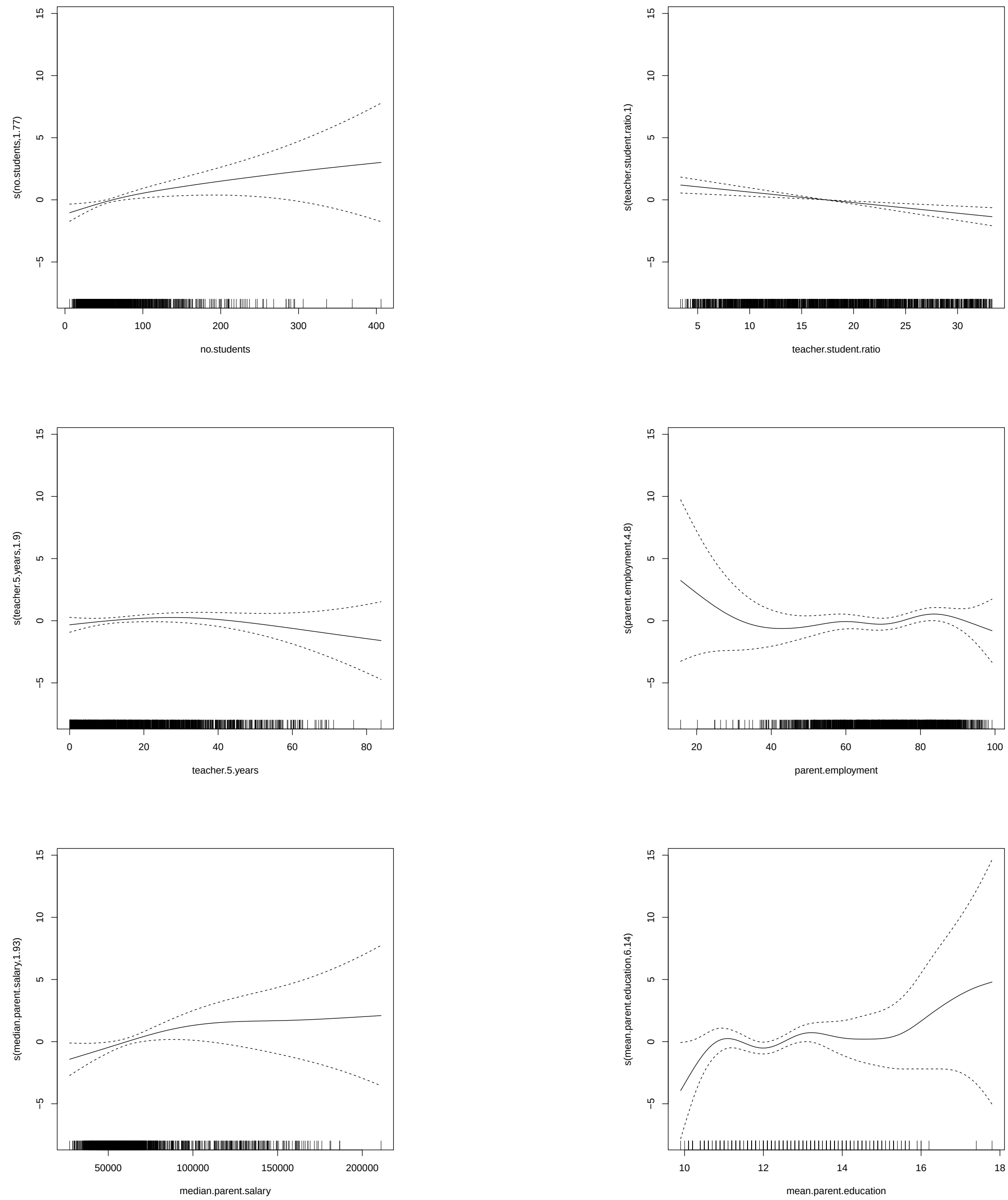
We use the following code:

```
HW3Q4<-read.table("HW3Q4.txt")
HW3Q4_test<-read.table("HW3Q4_test.txt")

library(mgcv)

### GAM does not allow the use of .
predictors<-"s(no.students)+s(teacher.student.ratio)+s(teacher.5.years)+
    s(parent.employment)+s(median.parent.salary)+s(mean.parent.education)+
    funding+specialist.teacher"


GAM_model_maths<-gam(as.formula(paste("average.score.mathematics",
                                      predictors,sep="~")),
                     data=HW3Q4)
GAM_model_english<-gam(as.formula(paste("average.score.english",
                                        predictors,sep="~")),
                       data=HW3Q4)


summary(GAM_model_maths)
summary(GAM_model_english)

for(i in seq_len(6)){
   plot(GAM_model_maths,select=i)
}


for(i in seq_len(6)){
   plot(GAM_model_english,select=i)
}
```

It produces the following smooth curves for the predictors' effects on average mathematics score:




and following smooth curves for the predictors' effects on average english score:




The models predict the following average scores for the test data:

| School no. | Predicted Mathematics Score | Predicted English Score |
|---|---|---|
| 1639 | 74.33654 | 78.97500 |
| 1640 | 68.87436 | 73.52036 |
| 1641 | 69.70518 | 74.85969 |
| 1642 | 77.61418 | 79.56861 |
| 1643 | 71.81126 | 76.88420 |
| 1644 | 81.96649 | 83.16504 |
| 1645 | 69.76377 | 75.22856 |
| 1646 | 70.03958 | 75.44406 |
| 1647 | 68.62463 | 76.54513 |
| 1648 | 74.79599 | 76.86610 |

5. *A company has collected the following data on employee training effectiveness in the file HW3Q5.*

| *Variable* | *Meaning* |
|---|---|
| *training.type* | *The type of training.* |
| *compulsory* | *Whether the training was compulsory for the employee.* |
| *employee.experience* | *The number of years of experience of the employee.* |
| *employee.salary* | *The employee's annual salary.* |
| *employee.gender* | *The employee's gender.* |
| *work.type* | *The type of work.* |
| *training.time* | *The amount of time spent on the training.* |
| *productivity.before* | *The employee's productivity rating before the training.* |
| *productivity.after* | *The employee's productivity rating after the training.* |

*Fit a linear model, using LASSO for variable selection and regularisation to predict sales from the other predictors. Use this model to predict sales for the scenarios in the file HW3Q5test.*

Lasso using one standard error on the cross-validation to select $\lambda$ selects $\lambda = 0.2231302$, while using the minimum for cross-validation gives $\lambda = 0.003027555$. These values of $\lambda$ give the following models:

| Coefficient | $\lambda_{1se}$ | $\lambda_{min}$ |
|---|---|---|
| (Intercept) | 0.1024965 | 0.1048495 |
| training.typeCourse | 0 | 0.07343168 |
| training.typeInteractive | 0 | −0.4134413 |
| training.typePassive | 0 | −0.4717399 |
| compulsory | 0 | −0.02457364 |
| employee.experience | 0 | 0.0005848728 |
| employee.salary | 0 | $4.874131 \times 10^{-7}$ |
| employee.gendermale | 0 | −0.01383078 |
| work.typecustomer service | 0 | 0.04246316 |
| work.typefinancial | 0 | 0.03341932 |
| work.typeIT | 0 | −0.08780233 |
| work.typemaintainance | 0 | 0.03475568 |
| training.time | 0.01829942 | 0.02110295 |
| productivity.before | 1.00027941 | 1.001852 |

and the following predictions:

| | $\lambda_{1se}$ | $\lambda_{min}$ |
|---|---|---|
| 804 | 157.64290 | 158.02254 |
| 805 | 149.34784 | 149.02260 |
| 806 | 249.97969 | 250.55525 |
| 807 | 70.82408 | 70.44031 |
| 808 | 158.65593 | 158.35815 |
| 809 | 291.29481 | 291.40193 |
| 810 | 64.96097 | 65.24300 |
| 811 | 198.47620 | 198.35855 |
| 812 | 290.93313 | 290.86321 |
| 813 | 35.42151 | 35.08816 |
| 814 | 164.25384 | 164.20132 |
| 815 | 236.20321 | 236.17607 |
| 816 | 221.98411 | 222.66893 |
| 817 | 212.74677 | 213.36404 |
| 818 | 285.50605 | 286.03936 |
| 819 | 256.19964 | 256.24233 |
| 820 | 236.37032 | 236.19666 |
| 821 | 179.66363 | 179.57840 |
| 822 | 145.96885 | 145.76334 |
| 823 | 605.77717 | 606.35474 |
| 824 | 293.39357 | 293.41553 |

Here is the code used to fit these models and make the predictions:

```
HW3Q5<-read.table("HW3Q5.txt",stringsAsFactors=TRUE)
library(glmnet)
HW3Q5_LASSO<-cv.glmnet(model.matrix(productivity.after~.,data=HW3Q5),
                       HW3Q5$productivity.after,
                       nfolds=10)
HW3Q5_LASSO$index
### The smallest lambda is chosen. This suggests the range is wrong.

HW3Q5_LASSO<-cv.glmnet(model.matrix(productivity.after~.,data=HW3Q5),
                         HW3Q5$productivity.after,nfolds=10,
                         lambda=exp(-seq_len(100)/10))

HW3Q5_LASSO$index ## looks OK now.

index.1se<-HW3Q5_LASSO$index["1se",1]
index.min<-HW3Q5_LASSO$index["min",1]

HW3Q5_LASSO$lambda[index.1se]

HW3Q5_LASSO$glmnet.fit$a0[index.1se]
HW3Q5_LASSO$glmnet.fit$beta[,index.1se]

HW3Q5_LASSO$lambda[index.min]

HW3Q5_LASSO$glmnet.fit$a0[index.min]
HW3Q5_LASSO$glmnet.fit$beta[,index.min]


HW3Q5_test<-read.table("HW3Q5_test.txt",stringsAsFactors=TRUE)
summary(HW3Q5_test) ## check all levels exist for factor variables.

HW3Q5_test$productivity.after<-1 ### Model matrix doesn't work with NAs

### Estimated values
model.matrix(productivity.after~.,data=HW3Q5_test)%*%
    HW3Q5_LASSO$glmnet.fit$beta[,index.1se]+
    HW3Q5_LASSO$glmnet.fit$a0[index.1se]


model.matrix(productivity.after~.,data=HW3Q5_test)%*%
    HW3Q5_LASSO$glmnet.fit$beta[,index.min]+
    HW3Q5_LASSO$glmnet.fit$a0[index.min]
```