ACSC/STAT 3740, Predictive Analytics

WINTER 2023
Toby Kenney

Homework Sheet 3

Model Solutions

Standard Questions

1. *A music streaming company is building a recommendation system to suggest songs to its readers. It has collected the following data in the file* `HW3Q1.txt`.

Variable	*Meaning*
genre	*The genre (type of music) of the song*
artist	*The identifier of the artist.*
rating	*The songs average user rating (scale 1–5)*
same.artist	*A measure of how much the user listens to songs by the artist. (scale 0–5)*
same.genre	*A measure of how much the user listens to songs from this genre (scale 0–5)*
friend.listen	*The number of the users "friends" that listen to the song*
friend.recommend	*The average of the recommendation scores for the song give by the user's friends u*
listen	*Whether the user listens to the recommended song.*

(a) Fit a logistic regression model to predict whether the user will listen to the recommended song.

```
Q1a_model<-glm(listen ~ ., data=HW3Q1, family=binomial(link="logit"))
summary(Q1a_model)
```

	Estimate	Std. Error
(Intercept)	−10.20877	0.69803
genreMetal	0.48174	0.63562
genrePop	0.30646	0.41686
genreRock	0.17207	0.42953
genreTechno	−0.09778	0.52648
artistB	0.34489	0.71172
artistC	0.03561	0.77868
artistD	0.37080	0.58422
artistE	0.53106	0.60287
artistF	0.15279	0.60332
artistG	−0.43709	1.24991
artistH	0.27937	0.63711
artistI	0.27306	0.65652
artistJ	0.06297	0.84363
rating	**0.51166**	**0.08058**
same.artist	**1.02820**	**0.05956**
same.genre	**0.49787**	**0.04006**
friend.listen	**0.09344**	**0.01135**
friend.recommend	**0.54638**	**0.14712**

(b) The predictor `friend.listen` *is skewed and heavy tailed. Try a log transformation and a sqare root transformation of this variable. Fit models including all combinations of these transformations.*

We fit a linear model:

```
Q1b_model_I<-glm(listen~.-friend.listen+log(friend.listen),data=HW3Q1,family=binomial(lin
Q1b_model_II<-glm(listen~.-friend.listen+sqrt(friend.listen),data=HW3Q1,family=binomial(l
Q1b_model_III<-glm(listen~.-friend.listen+log(friend.listen)+sqrt(friend.listen),data=HW3
Q1b_model_IV<-glm(listen~.+log(friend.listen),data=HW3Q1,family=binomial(link="logit"))
Q1b_model_V<-glm(listen~.+sqrt(friend.listen),data=HW3Q1,family=binomial(link="logit"))
Q1b_model_VI<-glm(listen~.+log(friend.listen)+sqrt(friend.listen),data=HW3Q1,family=binom
summary(Q1b_model_I)
summary(Q1b_model_VI)
summary(Q1b_model_II)
summary(Q1b_model_III)
summary(Q1b_model_IV)
summary(Q1b_model_V)
```

This gives us the following

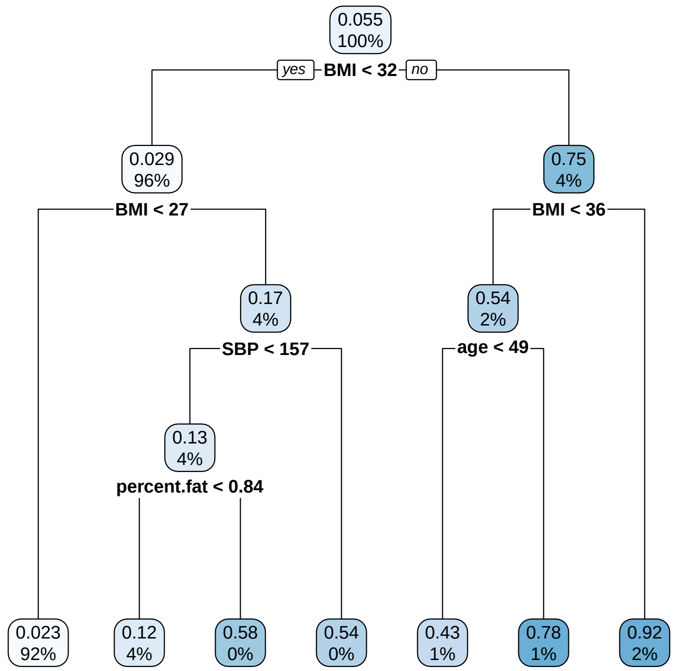
Parameter	log Est.	log S. E.	sqrt Est.	sqrt S. E.	log+sqrt Est.	log+sqrt S. E.	friend.listen+log Est.	friend.listen+log S. E.	friend.listen+sqrt Est.	friend.listen+sqrt S. E.	friend.lister Est.
(Intercept)	−10.52	0.71	−10.81	0.71	−10.80	0.72	−10.33	0.71	−10.63	0.77	−7.79
genreMetal	0.55	0.64	0.52	0.64	0.52	0.64	0.51	0.64	0.51	0.64	0.50
genrePop	0.32	0.42	0.31	0.42	0.31	0.42	0.30	0.42	0.30	0.42	0.29
genreRock	0.19	0.43	0.17	0.43	0.17	0.43	0.17	0.43	0.17	0.43	0.16
genreTechno	−0.05	0.53	−0.07	0.53	−0.07	0.53	−0.08	0.53	−0.08	0.53	−0.09
artistB	0.32	0.72	0.33	0.71	0.33	0.72	0.34	0.71	0.34	0.71	0.36
artistC	−0.02	0.79	0.01	0.78	0.00	0.78	0.01	0.78	0.01	0.78	0.02
artistD	0.40	0.59	0.37	0.59	0.37	0.59	0.37	0.59	0.37	0.59	0.39
artistE	0.51	0.61	0.52	0.60	0.52	0.61	0.52	0.60	0.52	0.60	0.55
artistF	0.14	0.61	0.14	0.61	0.14	0.61	0.15	0.61	0.15	0.60	0.18
artistG	−0.31	1.24	−0.38	1.24	−0.38	1.24	−0.37	1.24	−0.39	1.24	−0.29
artistH	0.29	0.64	0.28	0.64	0.28	0.64	0.29	0.64	0.28	0.64	0.31
artistI	0.28	0.66	0.28	0.66	0.28	0.66	0.28	0.66	0.28	0.66	0.31
artistJ	0.07	0.85	0.07	0.85	0.07	0.85	0.07	0.85	0.07	0.85	0.08
rating	0.52	0.08	0.51	0.08	0.51	0.08	0.51	0.08	0.51	0.08	0.52
same.artist	1.02	0.06	1.03	0.06	1.03	0.06	1.03	0.06	1.03	0.06	1.02
same.genre	0.49	0.04	0.50	0.04	0.50	0.04	0.50	0.04	0.50	0.04	0.50
friend.listen							0.06	0.02	0.03	0.05	0.34
friend.recommend	0.49	0.15	0.51	0.15	0.51	0.15	0.51	0.15	0.52	0.15	0.50
log(friend.listen)	0.69	0.08			0.03	0.33	0.30	0.18			2.20
sqrt(friend.listen)			0.56	0.07	0.54	0.26			0.39	0.29	−3.09

2. *The file* `HW3Q2.txt` *contains data from a study on the effect of exercise on the risk of heart disease in men. The variables included are*

Variable	*Meaning*
age	*The age of the patient*
ave.weekly.exercise	*The number of hours per week spent exercising.*
weekly.cals	*The number of calories consumed weekly.*
percent.fat	*The percentage of the patient's diet that consists of fats.*
percent.fibre	*The percentage of the patient's diet that consists of fibre.*
fam.hist	*Whether the patient has family history of heart disease.*
BMI	*The patient's BMI.*
SBP	*The patients systolic blood pressure.*
heart.5.year	*Whether the patient develops heart disease within the following 5 years.*

Fit a decision tree to predict whether an individual will develop heart disease in the next 5 years.

```
HW3Q2<-read.table("HW3Q2.txt")
Q2_dt<-rpart(heart.5.year~.,data=HW3Q2,control=rpart.control(minbucket=1,cp=0.0001,xval=1
Q2_dt$cptable[which(Q2_dt$cptable[,4]==min(Q2_dt$cptable[,4])),1]
## Find complexity parameter value that minimises error.
rpart.plot(prune(Q2_dt,cp=0.006546903))
```



3. *The file* `HW3Q3.txt` *contains daily new influenza infections counts in a particular country.*

(a) log-transform the counts and fit a seasonal trend using the function $\sin(2\pi t)$ *and* $\cos(2\pi t)$ *where* t *is the time in years.*

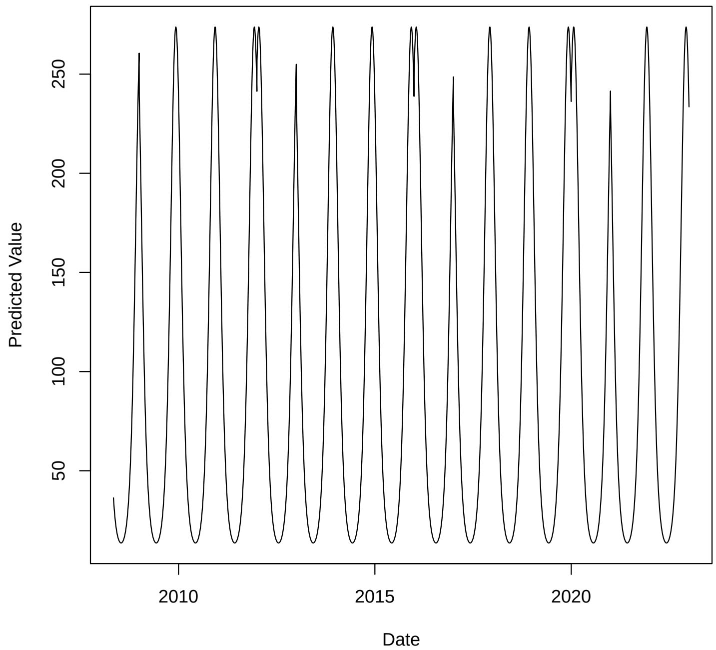
We fit a linear model:

```
HW3Q3<-read.table("HW3Q3.txt")
season_trend<-lm(log.new.cases~sin(2*pi*t)+cos(2*pi*t),
                 data=HW3Q3%>%
                   mutate(
                     t=as.numeric(ymd(paste(HW3Q3$year,HW3Q3$month,HW3Q3$day)))/(365+
                     log.new.cases=log(new.cases)))
summary(season_trend)
```

which gives the model:

	Estimate	Std. Error	p-value
Intercept	4.110561	0.007187	$< 2 \times 10^{-16}$
$\sin(2\pi t)$	-0.382762	0.010144	$< 2 \times 10^{-16}$
$\cos(2\pi t)$	1.452153	0.010183	$< 2 \times 10^{-16}$

and the predicted values



(b) After subtracting the seasonal trend, fit an ARMA model to the residuals, using AIC to determine the best choices for p and q.

```
flu_trend_arma<-auto.arima(season_trend$residuals,ic="aic",max.d=0)

summary(flu_trend_arma)
```

This selects an AR(5) model, and estimates the following parameters:

gives the model

	Coefficient	Std. Error
ar1	0.2447	0.0135
ar2	0.3685	0.0137
ar3	0.0154	0.0146
ar4	0.1832	0.0137
ar5	0.1334	0.0136

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

```
library(rugarch)
GARCH_model<-ugarchspec(mean.model=list(armaOrder=c(5,0)), distribution="norm")
GARCH_flu<-ugarchfit(GARCH_model,season_trend$residuals,solver="hybrid")
## The default solver fails to converge.
GARCH_flu
```

This selects an `sGARCH(1,1)` model for the variance, with the following parameters.

Parameter	Estimate	Std. Error	p-value
mu	0.000000	0.066147	1.000
ar1	0.292396	0.014237	0
ar2	0.426270	0.014590	0
ar3	−0.063533	0.016011	0.000072
ar4	0.191281	0.014487	0
ar5	0.115290	0.014102	0
omega	0.000565	0.000121	0.000003
alpha1	0.059222	0.005259	0
beta1	0.932089	0.005682	0

(d) Based on this model, what is the probability that there are fewer than 15000 flu cases in the first four months of 2023? [You can use the `ugarchboot` function to run a simulation to estimate this.]

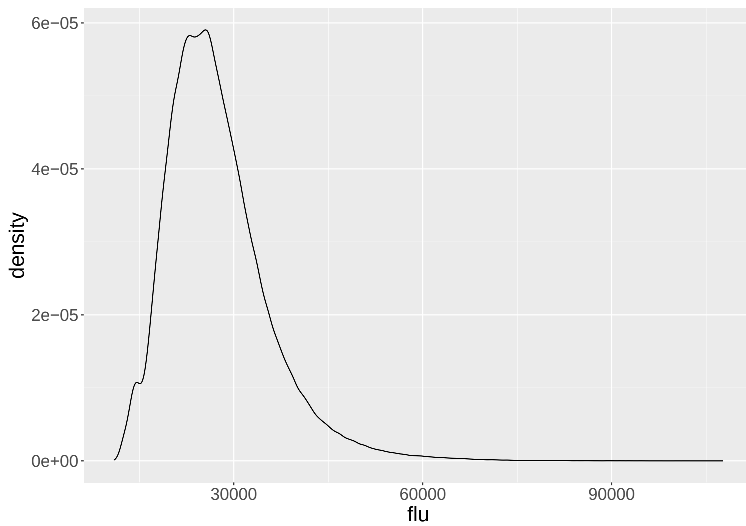
```
GARCH_Bootstraps<-ugarchboot(GARCH_flu,
                              method="full",
                              n.ahead=120,  # 4 months is 120 days in a non-leap year.
                              n.bootfit=1000, # 1000 parameter estimates
                              n.bootpred=1000, # 1000 bootstraps
                              rseed=seq_len(2000)) #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 Flu Distribution
Total.flu.dist<-rowSums(exp(GARCH_Bootstraps@fseries+rep(1,1000000)%*%t(
    predict(season_trend,newdata=list("t"=seq_len(120)/365)))))
### Remember to add the trend.
### The prediction was on log-transformed data, so we need to
### exponentiate to get the original data back.

ggplot(data.frame("flu"=Total.flu.dist),mapping=aes(x=flu))+geom_density()+largertextsize


mean(Total.flu.dist<14999.5)
### probability of less than 15000 cases.
### Since number of cases is an integer, I have used 14999.5 as the cut-off.
```



In my bootstap, the probability of this event is 0.022564.

4. *A reinsurance company has collected the following data on earthquakes in the file* `HW3Q4.txt`.

Variable	*Meaning*
magnitude	*The magnitude of the earthquake on the Ricter scale*
population	*The population of the affected city or region*
distance	*The distance of the epicentre from the affected area*
depth	*The depth of the epicentre*
year	*The year of the earthquake*
years.since.5	*The number of years since a magnitude 5 earthquake hit the same region*
country.gdp	*The annual per-capita gdp of the affected country*
damage	*The total damage caused by the earthquake*

Fit generalised linear models to predict the probability that an earthquake will cause damage, and for an earthquake which does cause damage, to predict the total damage, using a gamma response variable and a log-link function.

Use these models to predict the total damage for the earthquakes in the file `HW3Q4_test.txt`.

```
HW3Q4<-read.table("../HW3Q4.txt")
HW3Q4_test<-read.table("../HW3Q4_test.txt")
damage_prob<-glm(cause.damage~.,data=HW3Q4%>%mutate(cause.damage=(damage>0))%>%select(-c(
summary(damage_prob)
HW3Q4.damaging<-HW3Q4[HW3Q4$damage>0,]
damage_amount<-glm(damage~.,data=HW3Q4.damaging,family=Gamma(link="log"))

test.damage.probs<-predict(damage_prob,newdata=HW3Q4_test,type="response")
test.damage.vals<-predict(damage_amount,newdata=HW3Q4_test,type="response")
cbind(test.damage.probs,test.damage.vals,test.damage.probs*test.damage.vals)
```

No.	Prob Damage	Cond. Exp. Damage	Exp. Damage
404	0.8678214	2193661.44	1903706.42
405	0.2463193	651125.59	160384.77
406	0.9734117	792498.20	771427.02
407	0.9324269	2560302.84	2387295.13
408	0.9998725	22581537.56	22578659.48
409	0.8711961	953754.82	830907.45
410	0.9990740	633698.79	633111.99
411	0.9999309	22745015.29	22743443.89
412	0.9862094	1358243.33	1339512.32
413	0.9999335	1801972.54	1801852.63
414	0.9610747	876870.62	842738.15
415	0.9758311	647947.19	632287.05
416	0.9770732	304832.62	297843.80
417	0.5415518	143189.03	77544.27
418	0.9990411	2374097.70	2371821.22
419	0.4737955	197527.21	93587.51
420	0.3470020	272049.80	94401.82
421	0.2817239	370036.33	104248.08
422	0.9737238	1602758.87	1560644.49
423	0.9991449	1272932283.94	1271843836.63
424	0.9991372	5883380.66	5878304.44
425	0.9206701	3738815.26	3442215.42
426	0.8562643	197792.43	169362.59
427	0.8929791	45669408.83	40781829.02
428	0.9929486	168220.84	167034.65
429	0.2699715	160285.84	43272.61
430	0.5258784	8220484.82	4322975.20
431	0.2894384	4735716.58	1370698.03
432	0.5391881	112809.97	60825.80

No.	Prob Damage	Cond. Exp. Damage	Exp. Damage
433	0.6759868	19033.40	12866.32
434	0.9997568	6390293.92	6388739.50
435	0.7305791	1706045.03	1246400.76
436	0.9999728	10327042.14	10326760.74
437	0.9999863	7950513.00	7950404.19
438	0.9989259	3366851.69	3363235.42
439	0.9932973	4768525.06	4736562.97
440	0.9999800	295205.52	295199.62
441	0.9537350	2802661.96	2672996.94
442	0.6570247	154248628.99	101345154.28
443	0.8472959	93854.91	79522.88
444	0.2862934	176794.70	50615.16
445	0.9662820	1394173.40	1347164.69
446	0.9795238	956086.53	936509.52
447	0.9999669	789554255.88	789528148.06
448	0.4218416	149970.37	63263.74
449	0.9019580	691718.22	623900.76
450	0.9992449	1813749.04	1812379.44
451	0.3695368	195852.97	72374.87
452	0.3001554	74999757.46	22511584.92
453	0.2264533	1404392.28	318029.22
454	0.3970924	162371.99	64476.69
455	0.9302402	3050346.53	2837554.97
456	0.9999802	114308006.22	114305744.73
457	0.2688549	7451165.91	2003282.14
458	0.4344709	174625.38	75869.64
459	0.9998075	6377562.18	6376334.37
460	0.9010524	170814.67	153912.97
461	0.6978814	268314.52	187251.71

No.	Prob Damage	Cond. Exp. Damage	Exp. Damage
462	0.9986390	3363890.98	3359312.84
463	0.9999514	9803874.62	9803397.90
464	0.4443453	44992503.76	19992207.27
465	0.9970308	5893846.29	5876346.48
466	0.9992852	4168760.87	4165781.11
467	0.9999099	69447400.36	69441144.38
468	0.2450326	240711.54	58982.17
469	0.9913691	1807333.36	1791734.41
470	0.9996915	1783852.73	1783302.46
471	0.3586859	869495.65	311875.84
472	0.9998696	4505547.05	4504959.59
473	0.9986655	4147801.24	4142265.86
474	0.9967640	1597311.16	1592142.30
475	0.2203282	7218893.50	1590526.17
476	0.2811242	41518091.89	11671739.12
477	0.8306988	1120427.55	930737.78
478	0.9956580	8902809.81	8864153.97
479	0.5439880	401242.66	218271.18
480	0.9929051	2891802.75	2871285.62
481	0.9994529	4583114.64	4580607.03
482	0.9985226	6900371.29	6890176.53
483	0.9999464	6506912.12	6506563.33
484	0.5193215	111139.43	57717.10
485	0.8896959	181257.99	161264.49
486	0.9043388	24106171.31	21800146.96
487	0.8985452	2275389.78	2044540.56
488	0.8496501	4385762.64	3726363.49
489	0.2759198	181533.63	50088.73
490	0.9052380	278109.95	251755.70

No.	Da
491	0.738
492	0.998
493	0.982
494	0.510
495	0.100
496	0.939
497	0.934
498	0.988
499	0.987
500	0.999
501	0.999
502	0.501
503	0.411
504	0.562
505	0.999
506	0.976
507	0.997
508	0.999
509	0.516
510	0.960
511	0.805
512	0.778
513	0.999
514	0.999
515	0.971

5. *A scientist has collected the following data on the effect of organic farming*

on butterfly populations. The data are in the file ***HW3Q5.txt****.*

Variable	*Meaning*
total.agriculture	*The proportion of the habitat that is used for agriculture.*
main.crop	*The most grown crop in the region.*
percent.organic	*The proportion of agricultural land that uses organic farming methods.*
ave.summer.temp	*The average temperature during the summer months (°C).*
ave.winter.temp	*The average temperature during the winter months (°C).*
rainfall	*The average total annual rainfall.*
year	*The year.*
butterflies	*The number of butterflies caught in the region.*

(a) Fit a decision tree to predict number of butterflies from the other variables. Choose an appropriate transformation for the response variable, and make any necessary adjustments to the data.

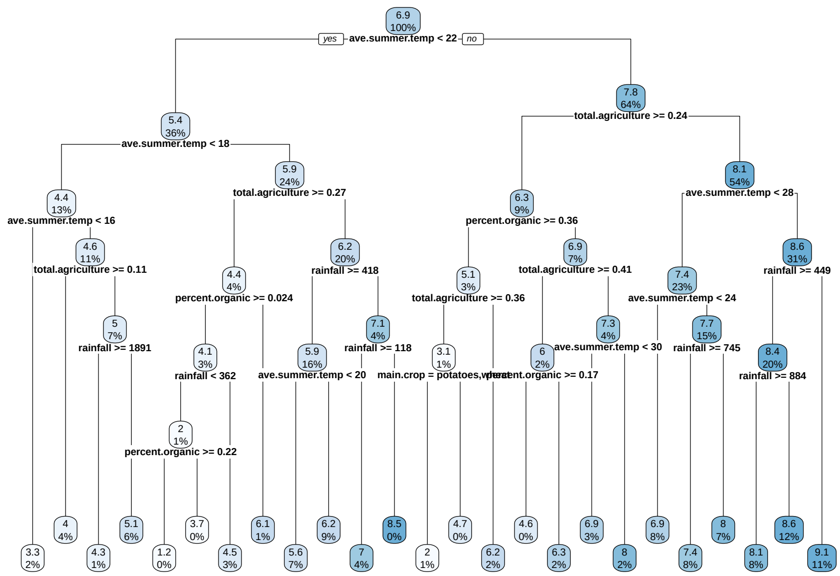
Given the skewed and heavy-tailed nature of the number of butterflies, a log-transformation is appropriate. This has the problem that it cannot handle cases with 0 butterflies. Since there are only two such cases, we remove these cases.

```
HW3Q5<-read.table("../HW3Q5.txt",stringsAsFactors=TRUE)
HW3Q5_test<-read.table("../HW3Q5_test.txt",stringsAsFactors=TRUE)

library(rpart.plot)
library(dplyr)

HW3Q5_dt<-rpart(log(butterflies)~.,data=HW3Q5%>%filter(butterflies>0),control=rpart.contr
HW3Q5_best_cp<-HW3Q5_dt$cptable[which(HW3Q5_dt$cptable[,4]==min(HW3Q5_dt$cptable[,4])),1]
HW3Q5_dt_best<-prune(HW3Q5_dt,cp=HW3Q5_best_cp)
rpart.plot(HW3Q5_dt_best)
```

This gives us the following decision tree.



(b) Fit a random forest model to predict number of butterflies from the other variables. Test this model on the dataset in the file `HW3Q4_test.txt`*.*

We use the caret package to train a random forest model. We use repeated cross-validation with 10 folds and 2 repeats to tune the `mtry` parameter in the range . This cross-validation selects `mtry=7`. The data set is not too large, so fitting 500 trees is not too time consuming.

The fitted model assigns the following variable importances:

Variable	Relative Importance
ave.summer.temp	100.0000
total.agriculture	25.4324
rainfall	8.1391
percent.organic	3.9402
ave.winter.temp	2.7445
main.crop	0.9327
year	0.0000

The cross-validated RMSE is 0.6872034, compared to a standard deviation of .

For the test data, it produces the following predictions, compared with the observed values:

```
library(caret)
library(dplyr)

RF.model<-train(HW3Q5%>%filter(butterflies>0)%>%select(-c("butterflies")),
                log((HW3Q5%>%filter(butterflies>0))$butterflies),
                method="rf",
                trControl=trainControl(method="repeatedcv",
                                       number=10,
                                       repeats=2),
                tuneGrid=expand.grid(mtry=seq_len(7)),
                ntree=500)


RF.predict<-exp(predict(RF.model,newdata=HW3Q5_test))
### remember that we log-transformed the response, so we need to exponentiate to get the
```

Obs.	butterflies	Obs.	butterflies	Obs.	butterflies	Obs.	butterflies
1	9912.5	16	1068.6	31	329.4	46	1567.5
2	1273.9	17	619.7	32	2507.2	47	34.5
3	2004.8	18	6284.3	33	21.8	48	692.7
4	491.2	19	61.0	34	273.2	49	739.8
5	735.4	20	2721.0	35	757.8	50	3592.3
6	2123.0	21	11271.5	36	762.0	51	1303.4
7	936.9	22	719.9	37	5013.6	52	6520.5
8	1396.4	23	134.7	38	4637.2	53	70.8
9	684.3	24	204.4	39	6694.3	54	519.1
10	410.3	25	5311.5	40	1147.4	55	2179.6
11	1118.5	26	3158.7	41	705.7	56	6169.1
12	3832.6	27	4382.9	42	2938.6	57	3246.0
13	753.0	28	4247.6	43	4005.6	58	30.9
14	4995.8	29	599.9	44	69.1		
15	2604.6	30	752.3	45	1334.8		