STAT 5525: Homework 4

Adam Wells

Friday, August 7

---

1. (This problem is optional) Read in the simple x,y dataset of 1000 points.

```
df <- read.csv("~/Desktop/STAT5525/hump1000.csv")
set.seed(1)
train_test_split<-initial_split(df,prop=0.8)
train_data<-train_test_split%>%training()
test_data<-train_test_split%>%testing()
```

a. Use R to fit a best tree model for this data using cross validation. Plot the data and fitted model. What is your estimated RMSE for this fitted model (be sure to compute this using a test/holdout sample)?

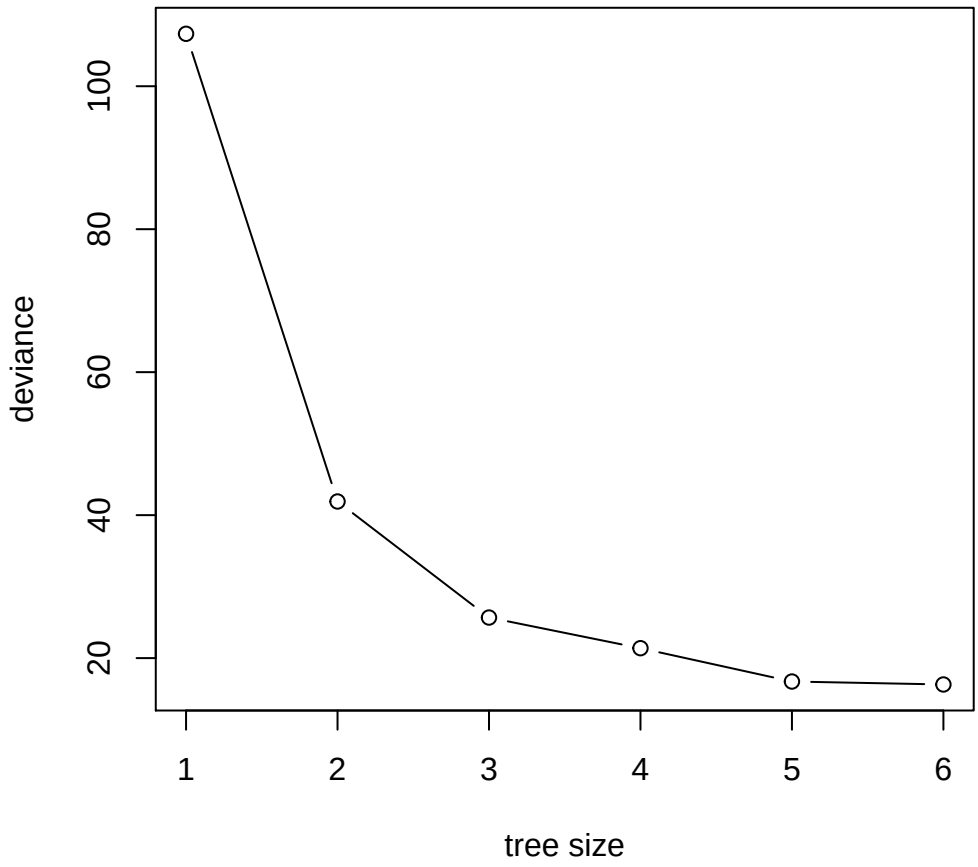
```
fit<-tree(y~x,train_data)
cv.fit<-cv.tree(fit,FUN=prune.tree)
cv.dev<-cbind(cv.fit$size,cv.fit$dev)
colnames(cv.dev)<-c("Size","Deviance")
rmse<-modelr::rmse(fit,test_data)
```

The following chart and graph show that the deviance is least when the number of terminal nodes is 6. When the 6-node model is applied to the test data, the rmse is 0.16.

```
pander(cv.dev)
```

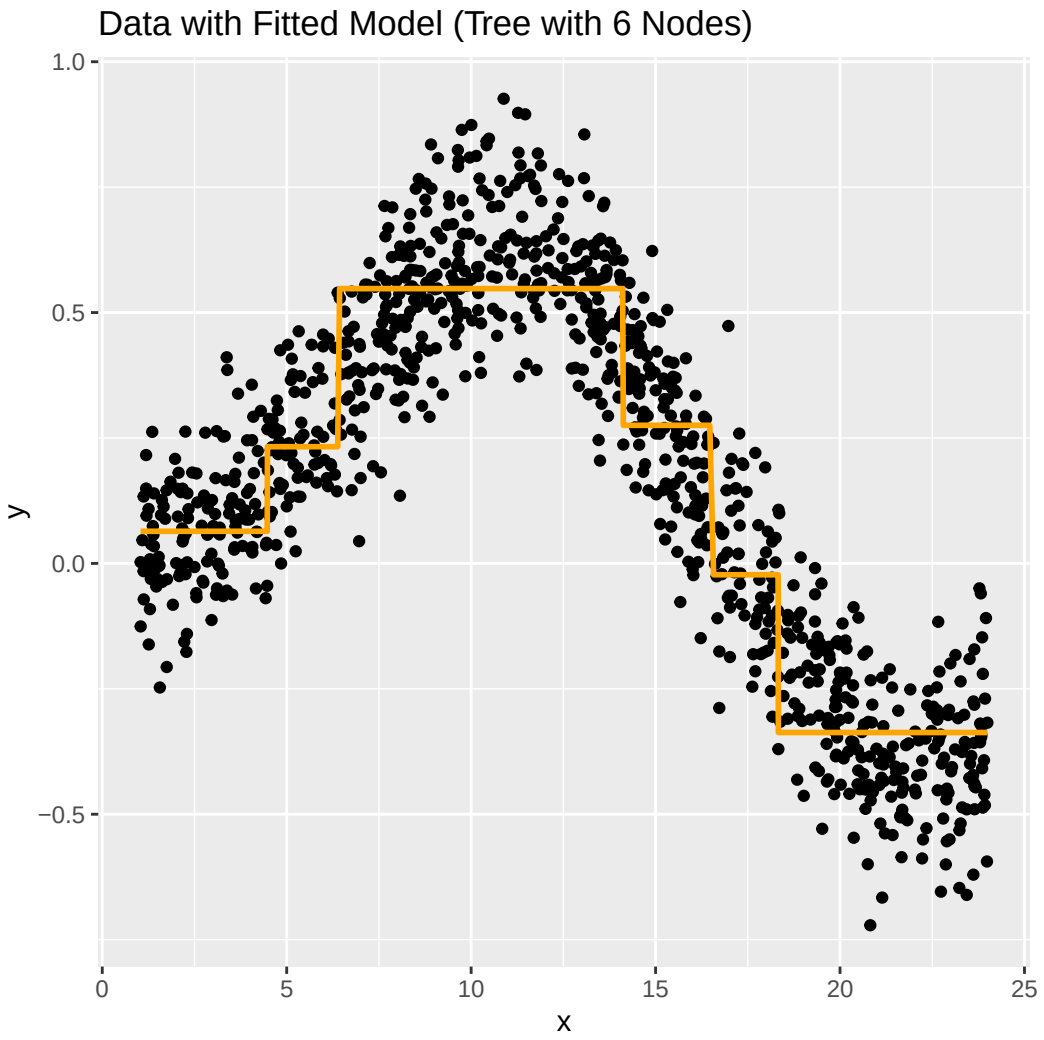
Size	Deviance
6	16.31
5	16.72
4	21.39
3	25.68
2	41.91
1	107.3

```
plot(cv.fit$size,cv.fit$dev,xlab='tree size',ylab='deviance',type="b")
```



The following graph shows the data and fitted model:

```
preds<-predict(fit,newdata = df)
df%>%
ggplot(mapping=aes(x=x,y=y))+
  geom_point()+
  geom_line(aes(y=preds,x=x),color="orange",size=1)+
  labs(title="Data with Fitted Model (Tree with 6 Nodes)")
```

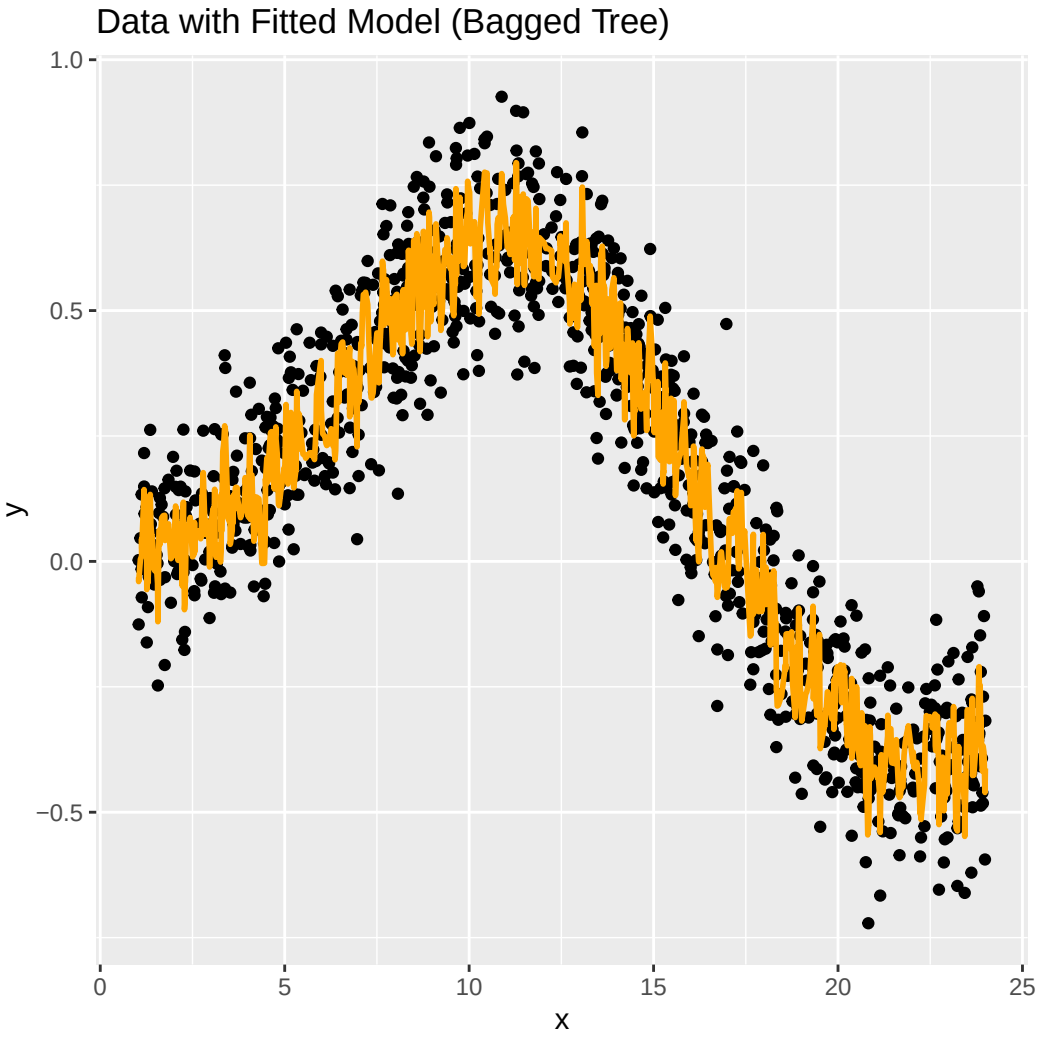


b. Use a bagged tree model to compute a fit to this data. Again, what is the estimated RMSE for this fitted model?

```
fit2<-randomForest(y ~ x,data=train_data,importance =TRUE,ntree=1000)
rmse<-modelr::rmse(fit2,test_data)
```

The rmse of the bagged tree model, when applied to the test data, is 0.14. The following graph shows the data and fitted model:

```
preds<-predict(fit2,newdata = df)
df%>%
ggplot(mapping=aes(x=x,y=y))+
  geom_point()+
  geom_line(aes(y=preds,x=x),color="orange",size=1)+
  labs(title="Data with Fitted Model (Bagged Tree)")
```

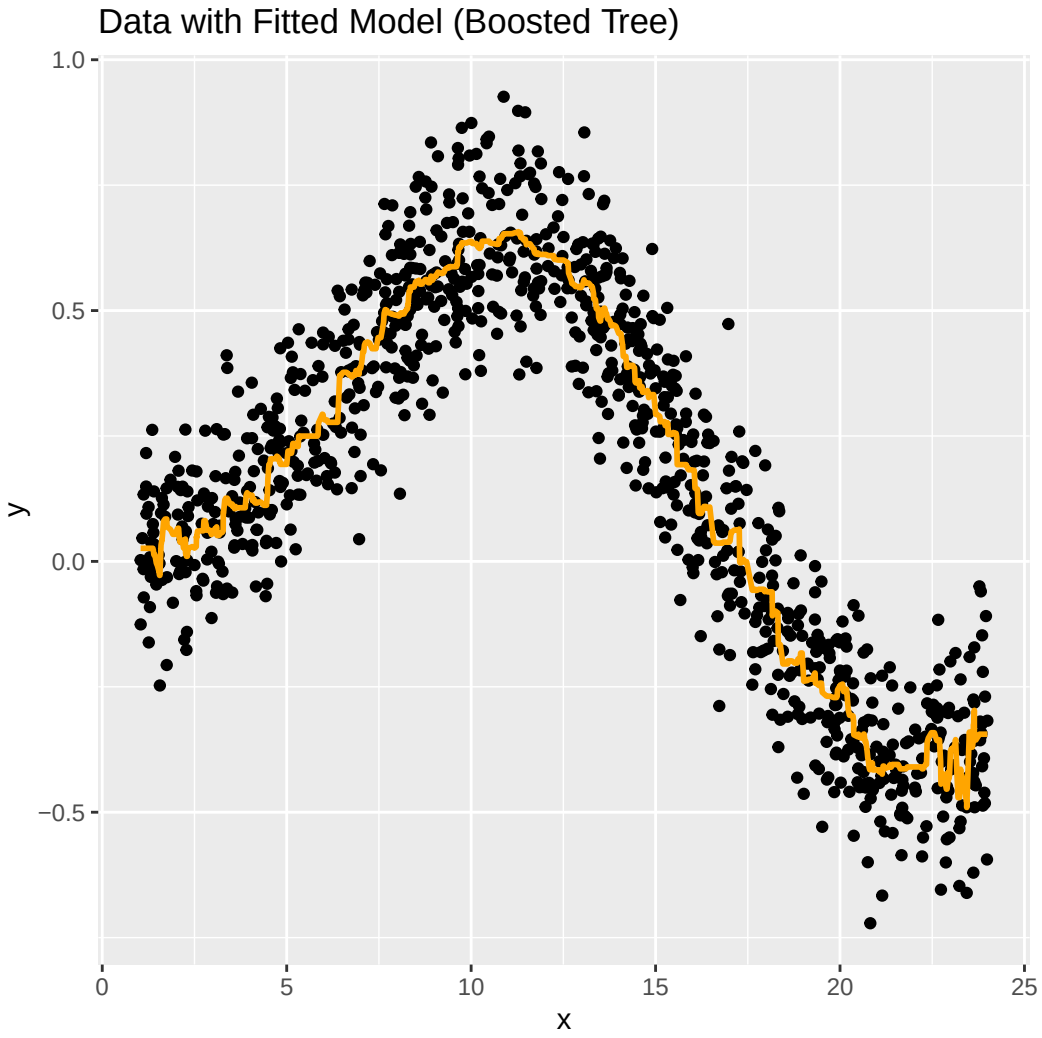


c. Use a boosted tree model to compute a fit to this data. Again, what is the estimated RMSE for this fitted model?

```
fit3=gbm(y~x,data=train_data,distribution="gaussian",n.trees=1000)
rmse<-modelr::rmse(fit3,test_data)
```

The boosted tree model has an RMSE of 0.13 when applied to the test data. The following graph shows the data and fitted model:

```
preds<-predict(fit3,newdata = df)
df%>%
ggplot(mapping=aes(x=x,y=y))+
  geom_point()+
  geom_line(aes(y=preds,x=x),color="orange",size=1)+
  labs(title="Data with Fitted Model (Boosted Tree)")
```



8.10: We now use boosting to predict Salary in the Hitters data set.

a. Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

```
hitters<-Hitters[-which(is.na(Hitters$Salary)),]
hitters<-hitters%>%
  mutate(logSalary=log(Salary))
```

b. Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

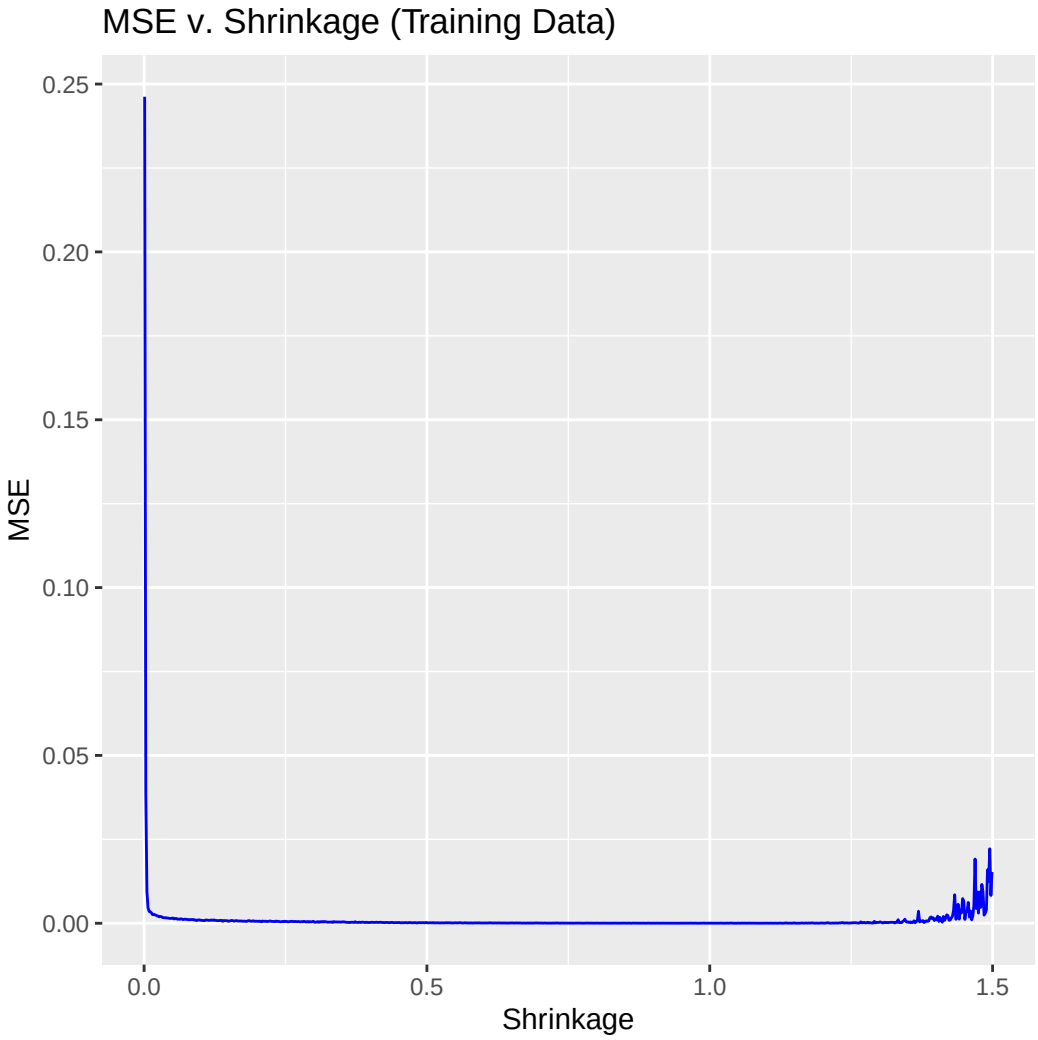
```
train_data<-hitters[1:200,]
test_data<-hitters[-(1:200),]
```

c. Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter lambda. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

```
#The following function performs the boosting function (gbm) on a range of lambda values.
#It return a dataframe of lambda values and MSEs.  The function takeds as arguments a data frame,
#a vector of response values, a vector of lambda values, a model formula, and the number of trees

shrink_gbm<-function(data,response,shrinkage,formula,ntrees){
  mse<-as.data.frame(NULL)
  for(i in 1:length(shrinkage)){
    fit<-gbm(formula=formula,
             data=data,
             n.trees=ntrees,
             distribution="gaussian",
             shrinkage = shrinkage[i])
    preds<-predict(fit,data)
    truth<-as.vector(response)
    m=mean((truth-preds)^2)
    ms<-c(shrinkage[i],m)
    mse<-rbind(mse,ms)
  }
  colnames(mse)<-c("lambda","MSE")
  return(mse)
}

formula=as.formula(logSalary~.)

shrinkage_vec<-seq(from=0.001,to=1.5,by=0.002)

shrink<-shrink_gbm(data=train_data,
                   formula=formula,
                   response=train_data$logSalary,
                   shrinkage=shrinkage_vec,
                   ntrees=1000)


bestlam=shrink$lambda[shrink$MSE==min(shrink$MSE)]
```

The value of lambda equivalent to the lowest mse is 1.085. The following plot shows that the MSE values decrease rapidly from a shrinkage value of 0.001 to a shrinkage value of 0.1. The MSE is lowest at 1.085, and MSE begins increasing appreciably at a shrinkage value of 1.3.

```
ggplot(shrink,mapping=aes(x=lambda,y=MSE))+
  geom_line(color="blue")+
  labs(title="MSE v. Shrinkage (Training Data)",x="Shrinkage")
```

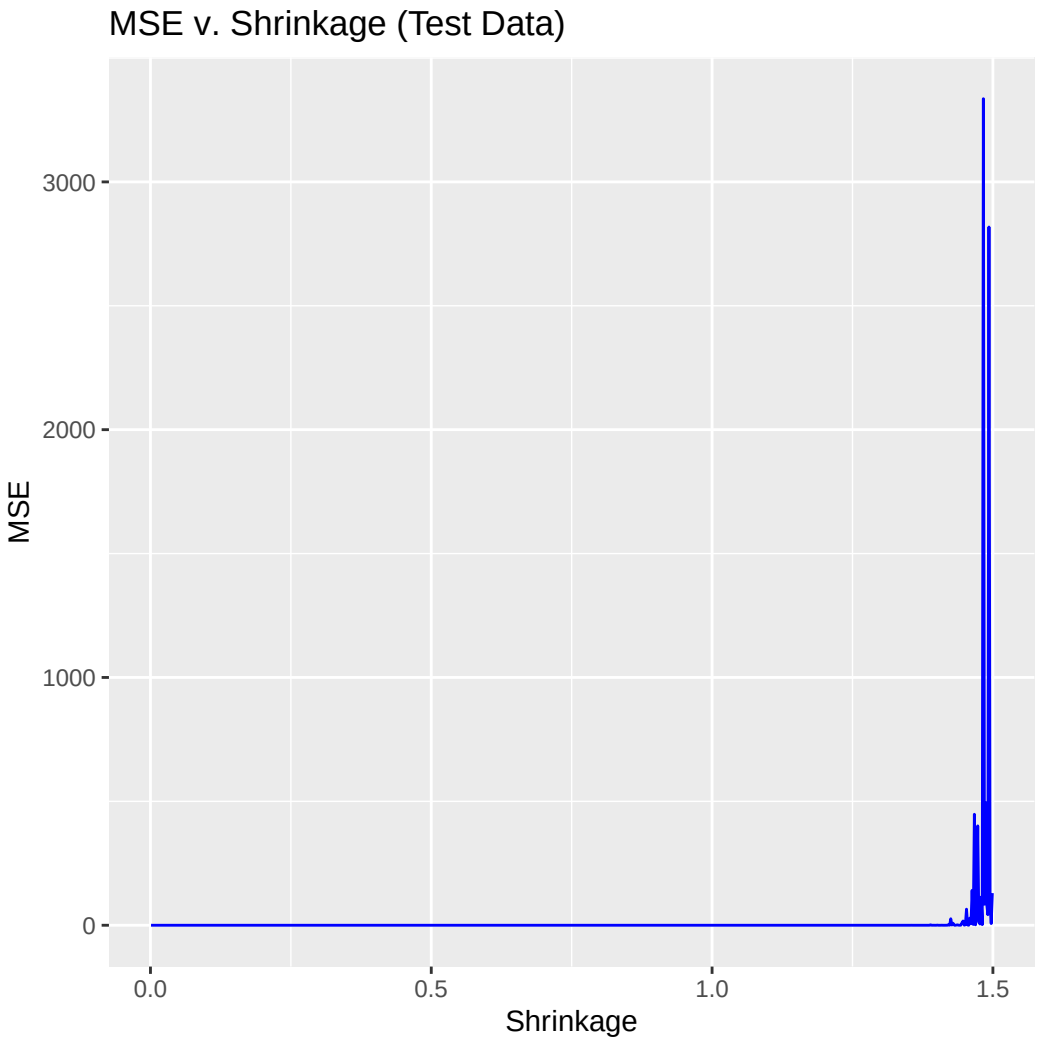


d. Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

```
shrink_test<-shrink_gbm(data=test_data,
                    formula=formula,
                    response=test_data$logSalary,
                    shrinkage=shrinkage_vec,
                    ntrees=1000)

bestlam_test=shrink_test$lambda[shrink_test$MSE==min(shrink_test$MSE)]
```

For the test data, the value of lambda equivalent to the lowest mse is 1.031. The following plot shows that the MSE values decrease slowly from a shrinkage value of 0.001 (MSE=0.1860794) to a shrinkage value of 1.031 (MSE=4.679062×10^{-10}*. MSE increases quickly after a shrinkage value of 1.4.*

```
ggplot(shrink_test,mapping=aes(x=lambda,y=MSE))+
  geom_line(color="blue")+
  labs(title="MSE v. Shrinkage (Test Data)",x="Shrinkage")
```


MSE v. Shrinkage (Test Data)
MSE
3000
2000
1000
0
0.0
0.5
1.0
1.5
Shrinkage