BIOST544 Homework 3

Ivy Zhang

11/15/2021

Screening Approach

First, we need to load in the data and organzie them to have a table with both expression data and the percentage of necrotic tissue in a tumor found by pathology data into one table. We will use the percentage as a biomaker to represent the existence and extent of necrotic tissue to help us figure out the association.

```
library(readr)
library(dplyr)
clinical <- read_csv("~/Desktop/R hw/clinical_data.csv")[,-1]
expression <- read_csv("~/Desktop/R hw/expression_data_probeID.csv")[,-1]
if(typeof(clinical$patid) != typeof(expression$patid)){
  expression$centerid <- as.numeric(expression$centerid)
  clinical$patid <- as.numeric(clinical$patid)
}
clinical.keep <- clinical[,c("centerid", "patid","necrotic_cells.pct")]
NOAH <- inner_join(expression, clinical.keep, by=c("centerid","patid"))
```

Then we need to do the screening-based approach, we will use the correlation as a measure of association between continous variables which is a common approach. The following code will use crossing-validation to help us to figure out how many gene we should keep in our prediction model by using correlation as our measure of association. We will use kendall correlation since it is more robust. We will think at most we can put 30 gene into our model. We will put the genes that is highly correlated with our come first in to our model.

```
set.seed(56)
genes = NOAH %>% select(-c(centerid, patid, necrotic_cells.pct))
all.cors = apply(genes, 2, cor, NOAH$necrotic_cells.pct, method = "kendall")
stats = data.frame(cors = all.cors, name = colnames(genes))
train_id = sample(1:nrow(NOAH), floor(nrow(NOAH)*0.75))
nmost = 30
train.cors = apply(genes[train_id,],2,cor, NOAH[train_id,]$necrotic_cells.pct,
                   method = "kendall")
stats_train = data.frame(cors = train.cors, name = colnames(genes))
sort_order =order(train.cors, decreasing = TRUE)
top_n = sort_order[1:nmost]

#-----------Evaluating-------------------
eval.my.model <- function(dat, mod){
  preds <- predict(mod, dat)
  MSE <- mean((dat$cells.pct - preds)^2)
  return(MSE)
}
#--------------Choosing number of genes-------------------
```

```
MSEs = rep(NA,nmost)
for(i in 1:nmost){
  n_gene = top_n[1:i]
  genes_topn =genes[,n_gene]
  data = cbind(NOAH$necrotic_cells.pct,genes_topn)
  colnames(data)[1] = "cells.pct"
  mod = lm(cells.pct~., data = data[train_id,] )
  MSEs[i] = eval.my.model(data[-train_id,], mod)
}
best_n = which.min(MSEs)
best_n
```

```
## [1] 6
```

After we know how many genes we should put into our model, we can fit a full model as following:

```
ngene = top_n[1:best_n]
genes_topn = genes[,ngene]
full_dat = cbind(NOAH$necrotic_cells.pct,genes_topn)
colnames(full_dat)[1] = "cells.pct"
full_fit = lm(cells.pct~., data = full_dat)
```

Prediction Based

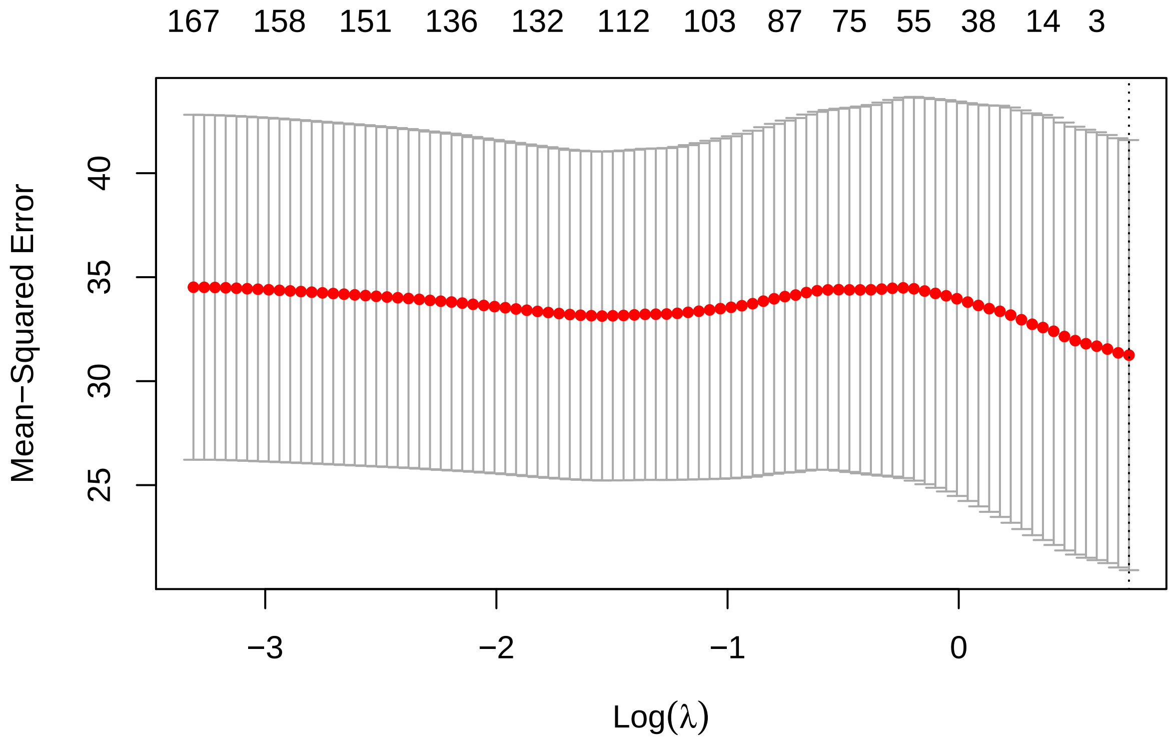
For the prdection approach, we will use LASSO to do the gene selection. We will use the cv.glmnet to help us to do cross-validation LASSO. At first, we try to do the choose a good lambda value by applying cross validation approach to all 54675 genes.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
set.seed(1)
gene_mat = as.matrix(genes)
cells.pct = as.matrix(NOAH$necrotic_cells.pct)
fit.cv = cv.glmnet(x = gene_mat, y = cells.pct, alpha=1, nlambda=100)
plot(fit.cv)
```



```
lanbda = fit.cv$lambda.min
lanbda
```

```
## [1] 2.088453
```

Then we will use lambda value to fit the LASSO model using full data to do the variable selection.

```
full.fit = glmnet(x = gene_mat, y = cells.pct, alpha=1, lambda=lanbda)
sum(coef(full.fit)!=0)
```

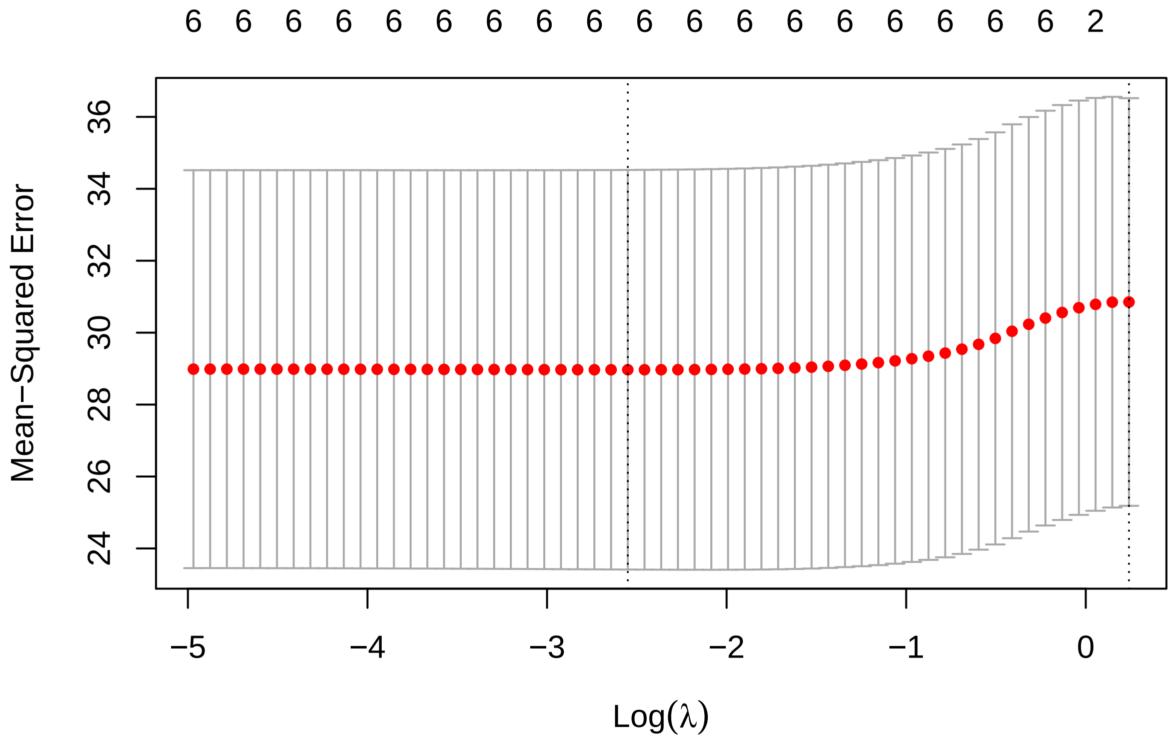
```
## [1] 1
```

As we can see, based on the lasso, it seems we only have none gene but only intercept that can help us to research the existance and extent of the tumor.

Extra Trying Approach

I think that may due to we only have 152 observations but need to research 54675 genes. Therefore, I will try to use the top 30 gene that is corrlated with the percentage to figure out the suitble lambda value to do the lasso and then fit the full model with that lambda value.

```
set.seed(183)
gene_top_mat = as.matrix(genes_topn)
fit.cv = cv.glmnet(x = gene_top_mat, y = cells.pct, alpha=1, nlambda=100)
plot(fit.cv)
```



```
lanbda = fit.cv$lambda.min
full.fit = glmnet(x = gene_mat, y = cells.pct, alpha=1, lambda=lanbda)
sum(coef(full.fit)!=0)
```

```
## [1] 154
```

Then based on the lasso model, we may need to have 153 genes (except for the intercept) to predict the existance and extence of the disase. I personally will prefer to use only 1 variable compare to 153 genes since the first approach in the prediction based are always fitting in the same data.