

Lab #7: Penalized Regression

Biostat 216: Machine Learning in R for the Biomedical Sciences

Subset Selection

1. Install the leaps package

```
## install.packages("leaps")
```

2. Load the leaps package

```
library(leaps)
```

3. Read in the breast cancer imaging data "ispy1doctored.csv" into a data frame called dat

```
dat <- read.csv("ispy1doctored.csv")
```

4. Perform best subset selection using the default parameters with MRI_LD_Tfinal as outcome (the longest diameter of the breast cancer tumor at the final visit pre-surgery) and all other variables as predictors. Look at a summary of the fitted models. What is the max number of predictors in any model imposed by the default settings?

```
regfit8 <- regsubsets(MRI_LD_Tfinal ~., data=dat)  
summary(regfit8)
```

```
## Subset selection object
```

```
## Call: regsubsets.formula(MRI_LD_Tfinal ~ ., data = dat)
```

```
## 18 Variables (and intercept)
```

```
##
```

	Forced in	Forced out
--	-----------	------------

## pixelVolT0	FALSE	FALSE
---------------	-------	-------

## pixelVolT1	FALSE	FALSE
---------------	-------	-------

## pixelVolT2	FALSE	FALSE
---------------	-------	-------

## pixelVolTfinal	FALSE	FALSE
-------------------	-------	-------

## pixelVolPctChgT0_T1	FALSE	FALSE
------------------------	-------	-------

## ftvPeT0	FALSE	FALSE
------------	-------	-------

## ftvPeT1	FALSE	FALSE
------------	-------	-------

## ftvPeT2	FALSE	FALSE
------------	-------	-------

## ftvPeTfinal	FALSE	FALSE
----------------	-------	-------

## ftvPePctChgT0_T1	FALSE	FALSE
---------------------	-------	-------

## age	FALSE	FALSE
--------	-------	-------

## race	FALSE	FALSE
---------	-------	-------

## HR_HER2statusHER2pos	FALSE	FALSE
-------------------------	-------	-------

## HR_HER2statusHRposHER2neg	FALSE	FALSE
------------------------------	-------	-------

## HR_HER2statusTripleNeg	FALSE	FALSE
---------------------------	-------	-------

## MRI_LD_T0	FALSE	FALSE
--------------	-------	-------

## MRI_LD_T1	FALSE	FALSE
--------------	-------	-------

## MRI_LD_T2	FALSE	FALSE
--------------	-------	-------

```
## 1 subsets of each size up to 8
```

```
## Selection Algorithm: exhaustive
```

```
##
```

	pixelVolT0	pixelVolT1	pixelVolT2	pixelVolTfinal
--	------------	------------	------------	----------------

## 1 (1)	" "	" "	" "	" "
------------	-----	-----	-----	-----

## 2 (1)	" "	" "	" "	" "
------------	-----	-----	-----	-----

## 3 (1)	" "	" "	" "	" "
------------	-----	-----	-----	-----

```

## 4 ( 1 ) " " " " " "
## 5 ( 1 ) " " " " " "
## 6 ( 1 ) " " " " " "
## 7 ( 1 ) " " " " " "
## 8 ( 1 ) " " " " " "
##
## pixelVolPctChgT0_T1 ftvPeT0 ftvPeT1 ftvPeT2 ftvPeTfinal
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) " " " " " " "*"
## 3 ( 1 ) " " "*" "*" " " " "
## 4 ( 1 ) " " " " " " "*"
## 5 ( 1 ) " " " " " " "*"
## 6 ( 1 ) " " "*" "*" "*" "*"
## 7 ( 1 ) " " "*" "*" "*" "*"
## 8 ( 1 ) " " "*" "*" "*" "*"
##
## ftvPePctChgT0_T1 age race HR_HER2statusHER2pos
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) " " " " " " " "
## 3 ( 1 ) " " " " " " " "
## 4 ( 1 ) " " " " " " " "
## 5 ( 1 ) " " " " " " " "
## 6 ( 1 ) " " " " " " " "
## 7 ( 1 ) " " " " " " " "
## 8 ( 1 ) " " " " " " " "
##
## HR_HER2statusHRposHER2neg HR_HER2statusTripleNeg MRI_LD_T0
## 1 ( 1 ) " " " " " "
## 2 ( 1 ) " " " " " "
## 3 ( 1 ) " " " " " "
## 4 ( 1 ) " " " " "*"
## 5 ( 1 ) "*" " " " "*"
## 6 ( 1 ) "*" " " " "
## 7 ( 1 ) " " " " "*"
## 8 ( 1 ) "*" " " " "*"
##
## MRI_LD_T1 MRI_LD_T2
## 1 ( 1 ) " " "*"
## 2 ( 1 ) " " "*"
## 3 ( 1 ) " " "*"
## 4 ( 1 ) "*" "*"
## 5 ( 1 ) "*" "*"
## 6 ( 1 ) " " "*"
## 7 ( 1 ) "*" "*"
## 8 ( 1 ) "*" "*"

```

Max number of predictors running regsubsets with defaults is 8.

5. How many predictors do you have available in the dataset?

```
ncol(dat) - 1
```

```
## [1] 16
```

6. Perform best subsets selection allowing models that could include all predictors

```
regfit16 <- regsubsets(MRI_LD_Tfinal ~., nvmax=16, data=dat)
```

7. Output a list of names of the objects in the summary of best subset selection fit, and examine the output of summary.

```

regSummary16 <- summary(regfit16)
names(regSummary16)

## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
regSummary16

## Subset selection object
## Call: regsubsets.formula(MRI_LD_Tfinal ~ ., nvmax = 16, data = dat)
## 18 Variables (and intercept)
##
## Forced in Forced out
## pixelVolT0 FALSE FALSE
## pixelVolT1 FALSE FALSE
## pixelVolT2 FALSE FALSE
## pixelVolTfinal FALSE FALSE
## pixelVolPctChgT0_T1 FALSE FALSE
## ftvPeT0 FALSE FALSE
## ftvPeT1 FALSE FALSE
## ftvPeT2 FALSE FALSE
## ftvPeTfinal FALSE FALSE
## ftvPePctChgT0_T1 FALSE FALSE
## age FALSE FALSE
## race FALSE FALSE
## HR_HER2statusHER2pos FALSE FALSE
## HR_HER2statusHRposHER2neg FALSE FALSE
## HR_HER2statusTripleNeg FALSE FALSE
## MRI_LD_T0 FALSE FALSE
## MRI_LD_T1 FALSE FALSE
## MRI_LD_T2 FALSE FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: exhaustive
## pixelVolT0 pixelVolT1 pixelVolT2 pixelVolTfinal
## 1 ( 1 ) " " " " " "
## 2 ( 1 ) " " " " " "
## 3 ( 1 ) " " " " " "
## 4 ( 1 ) " " " " " "
## 5 ( 1 ) " " " " " "
## 6 ( 1 ) " " " " " "
## 7 ( 1 ) " " " " " "
## 8 ( 1 ) " " " " " "
## 9 ( 1 ) " " " " "*"
## 10 ( 1 ) "*" " " " "*"
## 11 ( 1 ) "*" " " "*" "*"
## 12 ( 1 ) "*" " " "*" "*"
## 13 ( 1 ) "*" " " "*" "*"
## 14 ( 1 ) "*" " " "*" "*"
## 15 ( 1 ) "*" " " "*" "*"
## 16 ( 1 ) "*" " " "*" "*"
## pixelVolPctChgT0_T1 ftvPeT0 ftvPeT1 ftvPeT2 ftvPeTfinal
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) " " " " " " "*"
## 3 ( 1 ) " " "*" "*" " " " "
## 4 ( 1 ) " " " " " " "*"
## 5 ( 1 ) " " " " " " "*"

```

## 6	(1)	" "	"*"	"*"	"*"	"*"
## 7	(1)	" "	"*"	"*"	"*"	"*"
## 8	(1)	" "	"*"	"*"	"*"	"*"
## 9	(1)	" "	"*"	"*"	"*"	"*"
## 10	(1)	" "	"*"	"*"	"*"	"*"
## 11	(1)	" "	"*"	"*"	"*"	"*"
## 12	(1)	" "	"*"	"*"	"*"	"*"
## 13	(1)	"*"	"*"	"*"	"*"	"*"
## 14	(1)	"*"	"*"	"*"	"*"	"*"
## 15	(1)	"*"	"*"	"*"	"*"	"*"
## 16	(1)	"*"	"*"	"*"	"*"	"*"
##		ftvPePctChgT0_T1	age	race	HR_HER2status	HER2pos
## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "
## 3	(1)	" "	" "	" "	" "	" "
## 4	(1)	" "	" "	" "	" "	" "
## 5	(1)	" "	" "	" "	" "	" "
## 6	(1)	" "	" "	" "	" "	" "
## 7	(1)	" "	" "	" "	" "	" "
## 8	(1)	" "	" "	" "	" "	" "
## 9	(1)	" "	" "	" "	" "	" "
## 10	(1)	" "	" "	" "	" "	" "
## 11	(1)	" "	" "	" "	" "	" "
## 12	(1)	" "	" "	" "	"*"	" "
## 13	(1)	" "	" "	" "	"*"	" "
## 14	(1)	" "	"*"	" "	"*"	" "
## 15	(1)	"*"	"*"	" "	"*"	" "
## 16	(1)	"*"	"*"	"*"	"*"	" "
##		HR_HER2status	HRpos	HER2neg	HR_HER2status	TripleNeg
## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "
## 3	(1)	" "	" "	" "	" "	" "
## 4	(1)	" "	" "	" "	" "	"*"
## 5	(1)	"*"	" "	" "	" "	"*"
## 6	(1)	"*"	" "	" "	" "	" "
## 7	(1)	" "	" "	" "	" "	"*"
## 8	(1)	"*"	" "	" "	" "	"*"
## 9	(1)	"*"	" "	" "	" "	"*"
## 10	(1)	"*"	" "	" "	" "	"*"
## 11	(1)	"*"	" "	" "	" "	"*"
## 12	(1)	" "	" "	" "	"*"	"*"
## 13	(1)	" "	" "	" "	"*"	"*"
## 14	(1)	" "	" "	" "	"*"	"*"
## 15	(1)	" "	" "	" "	"*"	"*"
## 16	(1)	" "	" "	" "	"*"	"*"
##		MRI_LD_T1	MRI_LD_T2			
## 1	(1)	" "	"*"			
## 2	(1)	" "	"*"			
## 3	(1)	" "	"*"			
## 4	(1)	"*"	"*"			
## 5	(1)	"*"	"*"			
## 6	(1)	" "	"*"			
## 7	(1)	"*"	"*"			
## 8	(1)	"*"	"*"			

```
## 9 ( 1 ) "*"      "*"
## 10 ( 1 ) "*"     "*"
## 11 ( 1 ) "*"     "*"
## 12 ( 1 ) "*"     "*"
## 13 ( 1 ) "*"     "*"
## 14 ( 1 ) "*"     "*"
## 15 ( 1 ) "*"     "*"
## 16 ( 1 ) "*"     "*"

```

8. Generate plots for each of the metrics R^2 , Adjusted R^2 , C_p , and BIC , vs. number of variables in the model. Add a point to the plot indicating where optimal point is. Tip: writing a function may save you some typing/clutter in your code.

```
par(mfrow=c(2,2))
plot(regSummary16$rss, xlab="Number of Variables", ylab="RSS", type="l")

plotMetric <- function(metric,ylabel,optdir){
  plot(regSummary16[[metric]], xlab="Number of Variables", ylab=ylabel, type="l")
  if(optdir=="max") wopt <- which.max(regSummary16[[metric]])
  if(optdir=="min") wopt <- which.min(regSummary16[[metric]])
  print(paste("Number of variables that leads to best", metric,"is",wopt))
  points(wopt,regSummary16[[metric]][wopt], col="red", cex=2, pch=20)
}

plotMetric("adjr2","Adjusted RSq","max")

```

```
## [1] "Number of variables that leads to best adjr2 is 10"

```

```
plotMetric("cp","Cp","min")

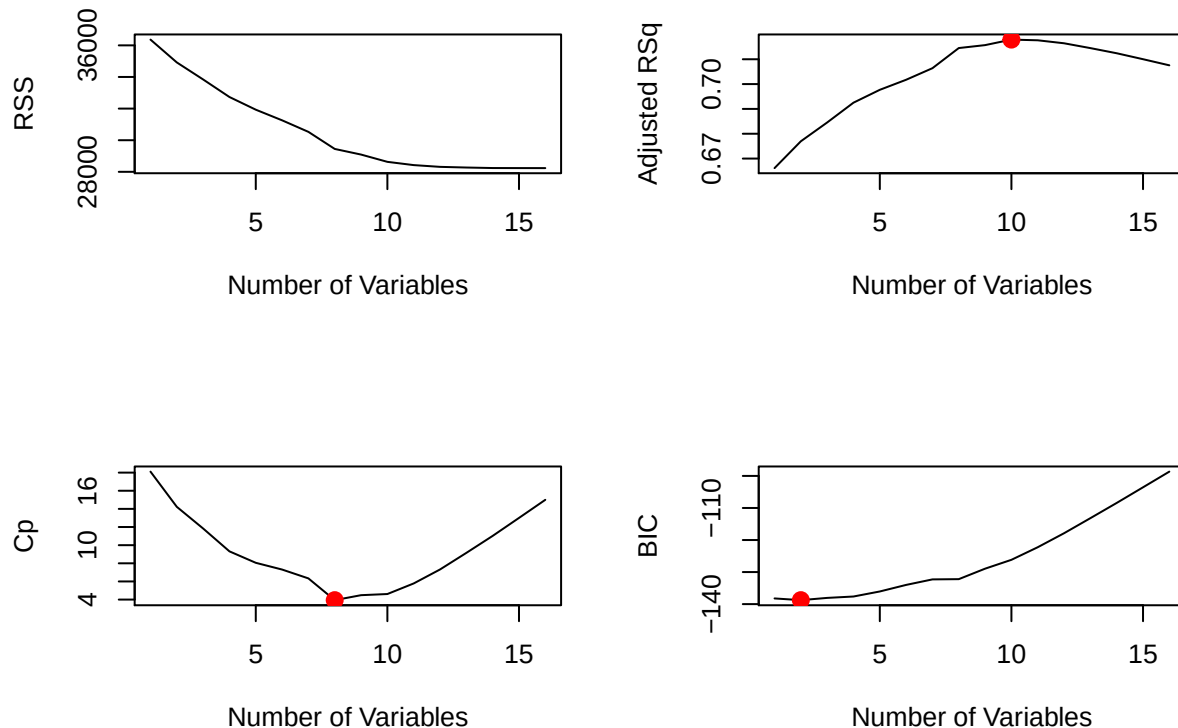
```

```
## [1] "Number of variables that leads to best cp is 8"

```

```
plotMetric("bic","BIC","min")

```



```
## [1] "Number of variables that leads to best bic is 2"
```

9. Now perform forward and backward stepwise selection allowing for potentially the full set of predictors in the models

```
regfit16fwd <- regsubsets(MRI_LD_Tfinal ~., nvmax=16, data=dat, method="forward")
summary(regfit16fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(MRI_LD_Tfinal ~ ., nvmax = 16, data = dat,
##      method = "forward")
## 18 Variables (and intercept)
##
```

	Forced in	Forced out
## pixelVolT0	FALSE	FALSE
## pixelVolT1	FALSE	FALSE
## pixelVolT2	FALSE	FALSE
## pixelVolTfinal	FALSE	FALSE
## pixelVolPctChgT0_T1	FALSE	FALSE
## ftvPeT0	FALSE	FALSE
## ftvPeT1	FALSE	FALSE
## ftvPeT2	FALSE	FALSE
## ftvPeTfinal	FALSE	FALSE
## ftvPePctChgT0_T1	FALSE	FALSE
## age	FALSE	FALSE
## race	FALSE	FALSE
## HR_HER2statusHER2pos	FALSE	FALSE
## HR_HER2statusHRposHER2neg	FALSE	FALSE
## HR_HER2statusTripleNeg	FALSE	FALSE
## MRI_LD_T0	FALSE	FALSE
## MRI_LD_T1	FALSE	FALSE
## MRI_LD_T2	FALSE	FALSE

```
## 1 subsets of each size up to 16
## Selection Algorithm: forward
##
```

	pixelVolT0	pixelVolT1	pixelVolT2	pixelVolTfinal
## 1 (1)	" "	" "	" "	" "
## 2 (1)	" "	" "	" "	" "
## 3 (1)	" "	" "	" "	" "
## 4 (1)	" "	" "	" "	" "
## 5 (1)	" "	" "	" "	" "
## 6 (1)	" "	" "	" "	"*"
## 7 (1)	" "	" "	" "	"*"
## 8 (1)	" "	" "	" "	"*"
## 9 (1)	" "	" "	"*"	"*"
## 10 (1)	" "	" "	"*"	"*"
## 11 (1)	" "	" "	"*"	"*"
## 12 (1)	"*"	" "	"*"	"*"
## 13 (1)	"*"	" "	"*"	"*"
## 14 (1)	"*"	" "	"*"	"*"
## 15 (1)	"*"	" "	"*"	"*"
## 16 (1)	"*"	" "	"*"	"*"

```
##
```

	pixelVolPctChgT0_T1	ftvPeT0	ftvPeT1	ftvPeT2	ftvPeTfinal
## 1 (1)	" "	" "	" "	" "	" "
## 2 (1)	" "	" "	" "	" "	"*"
## 3 (1)	" "	" "	" "	" "	"*"
## 4 (1)	" "	" "	" "	" "	"*"

## 5	(1)	" "	" "	" "	" "	"*"
## 6	(1)	" "	" "	" "	" "	"*"
## 7	(1)	" "	" "	" "	"*"	"*"
## 8	(1)	" "	" "	" "	"*"	"*"
## 9	(1)	" "	" "	" "	"*"	"*"
## 10	(1)	" "	" "	"*"	"*"	"*"
## 11	(1)	" "	"*"	"*"	"*"	"*"
## 12	(1)	" "	"*"	"*"	"*"	"*"
## 13	(1)	" "	"*"	"*"	"*"	"*"
## 14	(1)	" "	"*"	"*"	"*"	"*"
## 15	(1)	" "	"*"	"*"	"*"	"*"
## 16	(1)	"*"	"*"	"*"	"*"	"*"

##		ftvPePctChgT0_T1	age	race	HR_HER2status	HER2pos
----	--	------------------	-----	------	---------------	---------

## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "
## 3	(1)	" "	" "	" "	" "	" "
## 4	(1)	" "	" "	" "	" "	" "
## 5	(1)	" "	" "	" "	" "	" "
## 6	(1)	" "	" "	" "	" "	" "
## 7	(1)	" "	" "	" "	" "	" "
## 8	(1)	"*"	" "	" "	" "	" "
## 9	(1)	"*"	" "	" "	" "	" "
## 10	(1)	"*"	" "	" "	" "	" "
## 11	(1)	"*"	" "	" "	" "	" "
## 12	(1)	"*"	" "	" "	" "	" "
## 13	(1)	"*"	" "	" "	"*"	" "
## 14	(1)	"*"	"*"	" "	"*"	" "
## 15	(1)	"*"	"*"	" "	"*"	" "
## 16	(1)	"*"	"*"	" "	"*"	" "

##		HR_HER2status	HRpos	HER2neg	HR_HER2status	TripleNeg	MRI_LD_T0
----	--	---------------	-------	---------	---------------	-----------	-----------

## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "
## 3	(1)	"*"	" "	" "	" "	" "
## 4	(1)	"*"	" "	" "	" "	" "
## 5	(1)	"*"	" "	" "	" "	"*"
## 6	(1)	"*"	" "	" "	" "	"*"
## 7	(1)	"*"	" "	" "	" "	"*"
## 8	(1)	"*"	" "	" "	" "	"*"
## 9	(1)	"*"	" "	" "	" "	"*"
## 10	(1)	"*"	" "	" "	" "	"*"
## 11	(1)	"*"	" "	" "	" "	"*"
## 12	(1)	"*"	" "	" "	" "	"*"
## 13	(1)	"*"	" "	" "	" "	"*"
## 14	(1)	"*"	" "	" "	" "	"*"
## 15	(1)	"*"	"*"	" "	" "	"*"
## 16	(1)	"*"	"*"	" "	" "	"*"

##		MRI_LD_T1	MRI_LD_T2
----	--	-----------	-----------

## 1	(1)	" "	"*"
## 2	(1)	" "	"*"
## 3	(1)	" "	"*"
## 4	(1)	"*"	"*"
## 5	(1)	"*"	"*"
## 6	(1)	"*"	"*"
## 7	(1)	"*"	"*"

```
## 8 ( 1 ) "*"      "*"
## 9 ( 1 ) "*"      "*"
## 10 ( 1 ) "*"     "*"
## 11 ( 1 ) "*"     "*"
## 12 ( 1 ) "*"     "*"
## 13 ( 1 ) "*"     "*"
## 14 ( 1 ) "*"     "*"
## 15 ( 1 ) "*"     "*"
## 16 ( 1 ) "*"     "*"

```

```
regfit16bwd <- regsubsets(MRI_LD_Tfinal ~., nvmax=16, data=dat, method="backward")
summary(regfit16fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(MRI_LD_Tfinal ~ ., nvmax = 16, data = dat,
##      method = "forward")
## 18 Variables (and intercept)
##              Forced in Forced out
## pixelVolT0      FALSE      FALSE
## pixelVolT1      FALSE      FALSE
## pixelVolT2      FALSE      FALSE
## pixelVolTfinal  FALSE      FALSE
## pixelVolPctChgT0_T1  FALSE      FALSE
## ftvPeT0         FALSE      FALSE
## ftvPeT1         FALSE      FALSE
## ftvPeT2         FALSE      FALSE
## ftvPeTfinal     FALSE      FALSE
## ftvPePctChgT0_T1  FALSE      FALSE
## age             FALSE      FALSE
## race            FALSE      FALSE
## HR_HER2statusHER2pos  FALSE      FALSE
## HR_HER2statusHRposHER2neg  FALSE      FALSE
## HR_HER2statusTripleNeg  FALSE      FALSE
## MRI_LD_T0       FALSE      FALSE
## MRI_LD_T1       FALSE      FALSE
## MRI_LD_T2       FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: forward
##           pixelVolT0 pixelVolT1 pixelVolT2 pixelVolTfinal
## 1 ( 1 ) " "          " "          " "          " "
## 2 ( 1 ) " "          " "          " "          " "
## 3 ( 1 ) " "          " "          " "          " "
## 4 ( 1 ) " "          " "          " "          " "
## 5 ( 1 ) " "          " "          " "          " "
## 6 ( 1 ) " "          " "          " "          "*"
## 7 ( 1 ) " "          " "          " "          "*"
## 8 ( 1 ) " "          " "          " "          "*"
## 9 ( 1 ) " "          " "          "*"          "*"
## 10 ( 1 ) " "         " "          "*"          "*"
## 11 ( 1 ) " "         " "          "*"          "*"
## 12 ( 1 ) "*"         " "          "*"          "*"
## 13 ( 1 ) "*"         " "          "*"          "*"
## 14 ( 1 ) "*"         " "          "*"          "*"
## 15 ( 1 ) "*"         " "          "*"          "*"
## 16 ( 1 ) "*"         " "          "*"          "*"

```


##		pixelVolPctChgT0_T1	ftvPeT0	ftvPeT1	ftvPeT2	ftvPeTfinal
## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	"*"
## 3	(1)	" "	" "	" "	" "	"*"
## 4	(1)	" "	" "	" "	" "	"*"
## 5	(1)	" "	" "	" "	" "	"*"
## 6	(1)	" "	" "	" "	" "	"*"
## 7	(1)	" "	" "	" "	"*"	"*"
## 8	(1)	" "	" "	" "	"*"	"*"
## 9	(1)	" "	" "	" "	"*"	"*"
## 10	(1)	" "	" "	"*"	"*"	"*"
## 11	(1)	" "	"*"	"*"	"*"	"*"
## 12	(1)	" "	"*"	"*"	"*"	"*"
## 13	(1)	" "	"*"	"*"	"*"	"*"
## 14	(1)	" "	"*"	"*"	"*"	"*"
## 15	(1)	" "	"*"	"*"	"*"	"*"
## 16	(1)	"*"	"*"	"*"	"*"	"*"

##		ftvPePctChgT0_T1	age	race	HR_HER2status	HER2pos
## 1	(1)	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "
## 3	(1)	" "	" "	" "	" "	" "
## 4	(1)	" "	" "	" "	" "	" "
## 5	(1)	" "	" "	" "	" "	" "
## 6	(1)	" "	" "	" "	" "	" "
## 7	(1)	" "	" "	" "	" "	" "
## 8	(1)	"*"	" "	" "	" "	" "
## 9	(1)	"*"	" "	" "	" "	" "
## 10	(1)	"*"	" "	" "	" "	" "
## 11	(1)	"*"	" "	" "	" "	" "
## 12	(1)	"*"	" "	" "	" "	" "
## 13	(1)	"*"	" "	" "	"*"	" "
## 14	(1)	"*"	"*"	" "	"*"	" "
## 15	(1)	"*"	"*"	" "	"*"	" "
## 16	(1)	"*"	"*"	" "	"*"	" "

##		HR_HER2status	HRpos	HER2neg	HR_HER2status	TripleNeg	MRI_LD_T0
## 1	(1)	" "	" "	" "	" "	" "	" "
## 2	(1)	" "	" "	" "	" "	" "	" "
## 3	(1)	"*"	" "	" "	" "	" "	" "
## 4	(1)	"*"	" "	" "	" "	" "	" "
## 5	(1)	"*"	" "	" "	" "	" "	"*"
## 6	(1)	"*"	" "	" "	" "	" "	"*"
## 7	(1)	"*"	" "	" "	" "	" "	"*"
## 8	(1)	"*"	" "	" "	" "	" "	"*"
## 9	(1)	"*"	" "	" "	" "	" "	"*"
## 10	(1)	"*"	" "	" "	" "	" "	"*"
## 11	(1)	"*"	" "	" "	" "	" "	"*"
## 12	(1)	"*"	" "	" "	" "	" "	"*"
## 13	(1)	"*"	" "	" "	" "	" "	"*"
## 14	(1)	"*"	" "	" "	" "	" "	"*"
## 15	(1)	"*"	"*"	" "	" "	" "	"*"
## 16	(1)	"*"	"*"	" "	" "	" "	"*"

##		MRI_LD_T1	MRI_LD_T2
## 1	(1)	" "	"*"
## 2	(1)	" "	"*"

```
## 3 ( 1 ) " " "*"
## 4 ( 1 ) "*" "*"
## 5 ( 1 ) "*" "*"
## 6 ( 1 ) "*" "*"
## 7 ( 1 ) "*" "*"
## 8 ( 1 ) "*" "*"
## 9 ( 1 ) "*" "*"
## 10 ( 1 ) "*" "*"
## 11 ( 1 ) "*" "*"
## 12 ( 1 ) "*" "*"
## 13 ( 1 ) "*" "*"
## 14 ( 1 ) "*" "*"
## 15 ( 1 ) "*" "*"
## 16 ( 1 ) "*" "*"

```

10. Compare the coefficients of the different models based on restricting the number of variables from 1 through 8. Tip: writing a loop will save you a lot of typing. What is the largest size of model where all 3 methods agree? What is the smallest size model where all 3 methods disagree?

```
for(i in 1:8){
  print(paste("Number of variables =",i))
  print(coef(regfit16, id=i))
  print(coef(regfit16fwd, id=i))
  print(coef(regfit16bwd, id=i))
  cat("\n#####\n\n")
}
```

```
## [1] "Number of variables = 1"
## (Intercept) MRI_LD_T2
## -5.3892027 0.8005015
## (Intercept) MRI_LD_T2
## -5.3892027 0.8005015
## (Intercept) MRI_LD_T2
## -5.3892027 0.8005015
##
## #####
##
## [1] "Number of variables = 2"
## (Intercept) ftvPeTfinal MRI_LD_T2
## -4.4892490 0.3438281 0.7556036
## (Intercept) ftvPeTfinal MRI_LD_T2
## -4.4892490 0.3438281 0.7556036
## (Intercept) ftvPeT1 MRI_LD_T2
## -5.41206528 0.07498432 0.77198644
##
## #####
##
## [1] "Number of variables = 3"
## (Intercept) ftvPeT0 ftvPeT1 MRI_LD_T2
## -4.7149634 -0.2239072 0.4321604 0.7556418
## (Intercept) ftvPeTfinal
## -6.7432773 0.3297824
## HR_HER2statusHRposHER2neg MRI_LD_T2
## 5.6796121 0.7558100
## (Intercept) ftvPeT0 ftvPeT1 MRI_LD_T2

```

```

## -4.7149634 -0.2239072 0.4321604 0.7556418
##
## #####
##
## [1] "Number of variables = 4"
## (Intercept) ftvPeTfinal MRI_LD_T0 MRI_LD_T1 MRI_LD_T2
## -3.2787017 0.3504987 -0.2894782 0.3253792 0.7291301
## (Intercept) ftvPeTfinal
## -10.1945359 0.3480656
## HR_HER2statusHRposHER2neg MRI_LD_T1
## 6.1334420 0.1083988
## MRI_LD_T2
## 0.6845333
## (Intercept) ftvPeT0 ftvPeT1 MRI_LD_T1 MRI_LD_T2
## -7.54787835 -0.23609993 0.43611713 0.09383407 0.70112287
##
## #####
##
## [1] "Number of variables = 5"
## (Intercept) ftvPeTfinal
## -6.1216616 0.3412677
## HR_HER2statusHRposHER2neg MRI_LD_T0
## 5.0832681 -0.2569219
## MRI_LD_T1 MRI_LD_T2
## 0.3116322 0.7172658
## (Intercept) ftvPeTfinal
## -6.1216616 0.3412677
## HR_HER2statusHRposHER2neg MRI_LD_T0
## 5.0832681 -0.2569219
## MRI_LD_T1 MRI_LD_T2
## 0.3116322 0.7172658
## (Intercept) ftvPeT0 ftvPeT1 MRI_LD_T0 MRI_LD_T1 MRI_LD_T2
## -4.1491993 -0.1896251 0.3829421 -0.2359234 0.2749130 0.7290266
##
## #####
##
## [1] "Number of variables = 6"
## (Intercept) ftvPeT0
## -6.7016577 -0.2299142
## ftvPeT1 ftvPeT2
## 0.4658600 -0.3379737
## ftvPeTfinal HR_HER2statusHRposHER2neg
## 0.7854264 6.3092039
## MRI_LD_T2
## 0.7301859
## (Intercept) pixelVolTfinal
## -12.4937094 4333.9733811
## ftvPeTfinal HR_HER2statusHRposHER2neg
## 0.3642012 5.2962223
## MRI_LD_T0 MRI_LD_T1
## -0.2569753 0.3188632
## MRI_LD_T2
## 0.7191308
## (Intercept) ftvPeT0 ftvPeT1 ftvPeTfinal MRI_LD_T0 MRI_LD_T1

```

```

## -3.8807538 -0.1610298 0.3122117 0.2653564 -0.2366371 0.2870606
## MRI_LD_T2
## 0.6987543
##
## #####
##
## [1] "Number of variables = 7"
## (Intercept) ftvPeT0 ftvPeT1 ftvPeT2 ftvPeTfinal MRI_LD_T0
## -3.2785082 -0.2039563 0.4067118 -0.3511142 0.8492289 -0.2763390
## MRI_LD_T1 MRI_LD_T2
## 0.3174166 0.7069879
## (Intercept) pixelVolTfinal
## -11.8759559 4059.0447370
## ftvPeT2 ftvPeTfinal
## -0.2293799 0.7615046
## HR_HER2statusHRposHER2neg MRI_LD_T0
## 5.6942778 -0.2877339
## MRI_LD_T1 MRI_LD_T2
## 0.3437352 0.7300493
## (Intercept) ftvPeT0 ftvPeT1 ftvPeT2 ftvPeTfinal MRI_LD_T0
## -3.2785082 -0.2039563 0.4067118 -0.3511142 0.8492289 -0.2763390
## MRI_LD_T1 MRI_LD_T2
## 0.3174166 0.7069879
##
## #####
##
## [1] "Number of variables = 8"
## (Intercept) ftvPeT0
## -6.3750853 -0.2016613
## ftvPeT1 ftvPeT2
## 0.4242895 -0.3944662
## ftvPeTfinal HR_HER2statusHRposHER2neg
## 0.9053744 5.9939034
## MRI_LD_T0 MRI_LD_T1
## -0.2502244 0.3069329
## MRI_LD_T2
## 0.6917249
## (Intercept) pixelVolTfinal
## -10.16247136 3552.68975041
## ftvPeT2 ftvPeTfinal
## -0.27398603 0.79581974
## ftvPePctChgT0_T1 HR_HER2statusHRposHER2neg
## 0.05568065 5.89228116
## MRI_LD_T0 MRI_LD_T1
## -0.25484357 0.34194608
## MRI_LD_T2
## 0.70258167
## (Intercept) ftvPeT0 ftvPeT1
## -2.4909594 -0.2137085 0.4186091
## ftvPeT2 ftvPeTfinal HR_HER2statusHER2pos
## -0.3921801 0.9176423 -4.9425506
## MRI_LD_T0 MRI_LD_T1 MRI_LD_T2
## -0.2774223 0.3431684 0.6937087
##

```

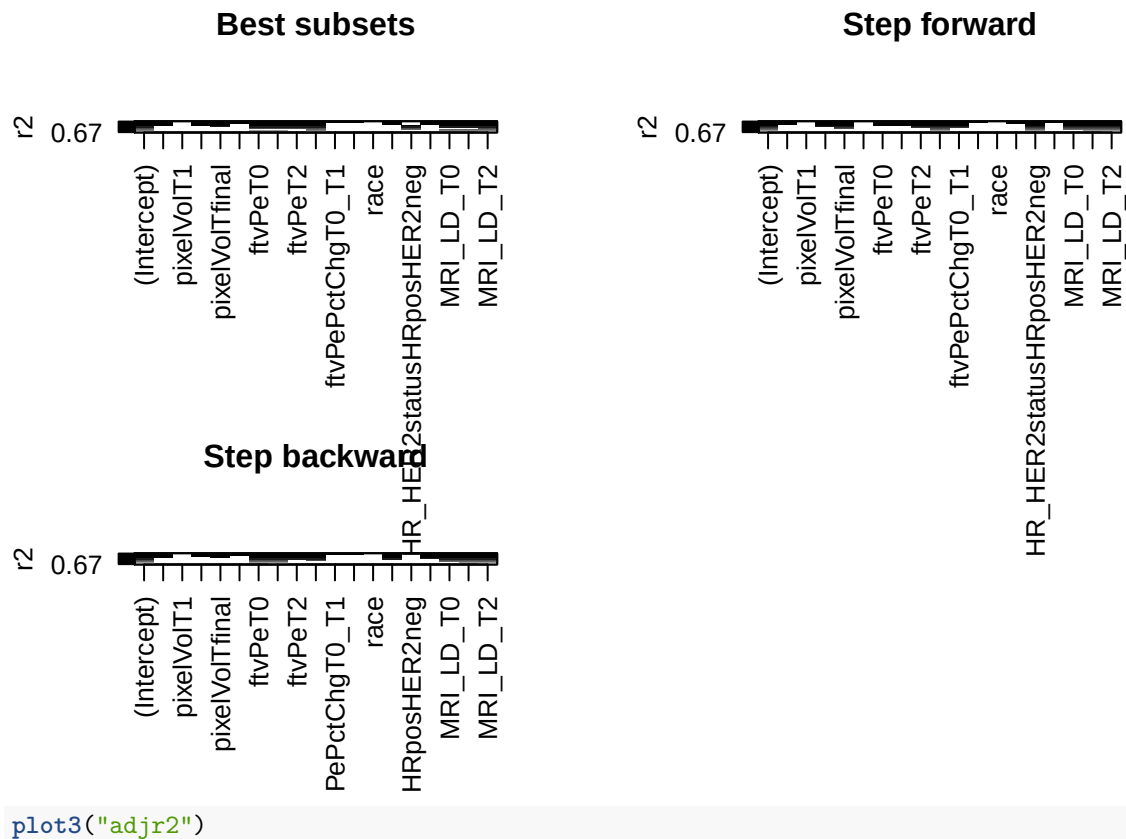
```
## #####
```

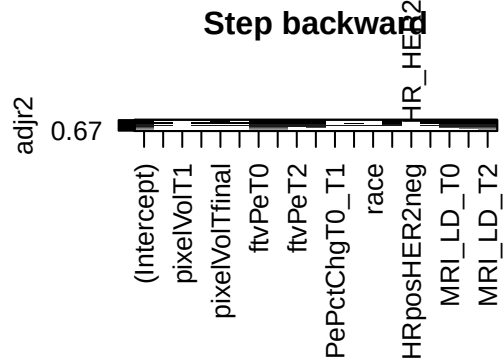
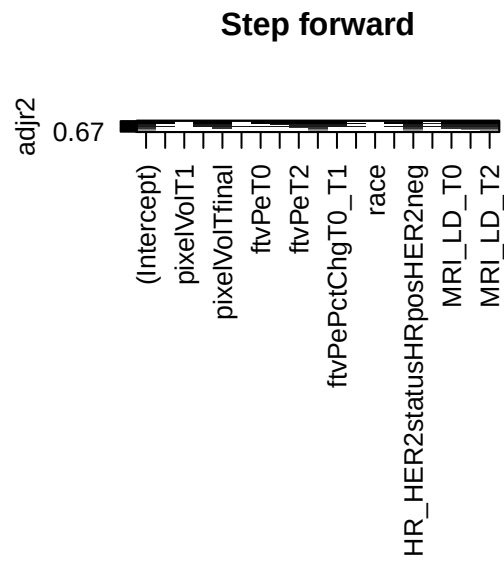
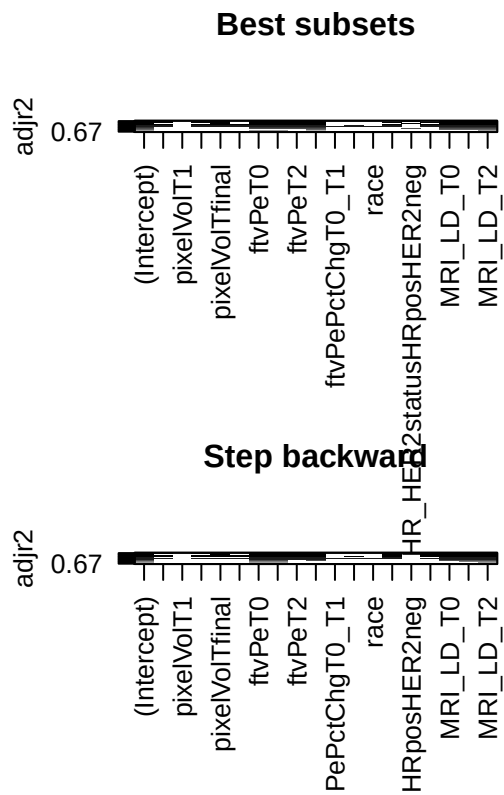
The largest size model where all 3 methods agree is with 1 variable. The smallest size model where all 3 methods disagree is with 4 predictor variables.

11. Try the plot function associated with `regsubsets` on each of the 3 selection procedures (best subset, forward, and backward). Use the `scale` argument to plot R^2 . Feel free to also look at the corresponding plots for the Adjusted R^2 , C_p , and BIC statistics too. Again you may want to play with writing a function to avoid clutter.

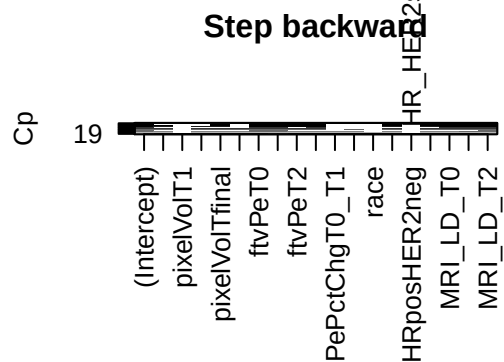
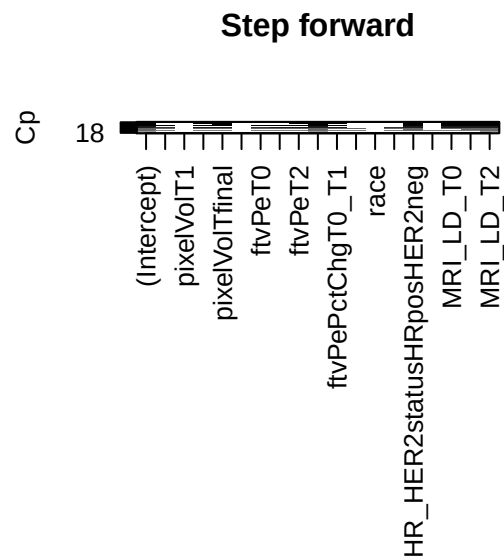
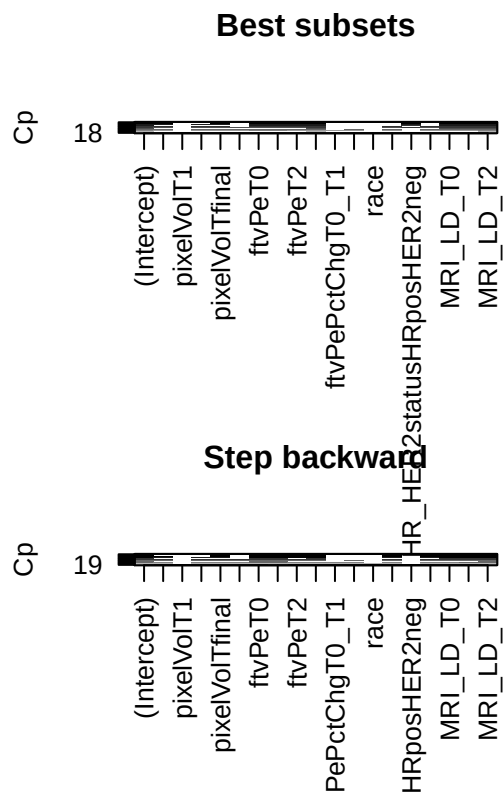
```
plot3 <- function(scaleVal){
  par(mfrow=c(2,2))
  plot(regfit16, scale=scaleVal, main="Best subsets")
  plot(regfit16fwd, scale=scaleVal, main="Step forward")
  plot(regfit16bwd, scale=scaleVal, main="Step backward")
  par(mfrow=c(1,1)) ## Resetting to default after generating plot with panels
}

plot3("r2")
```

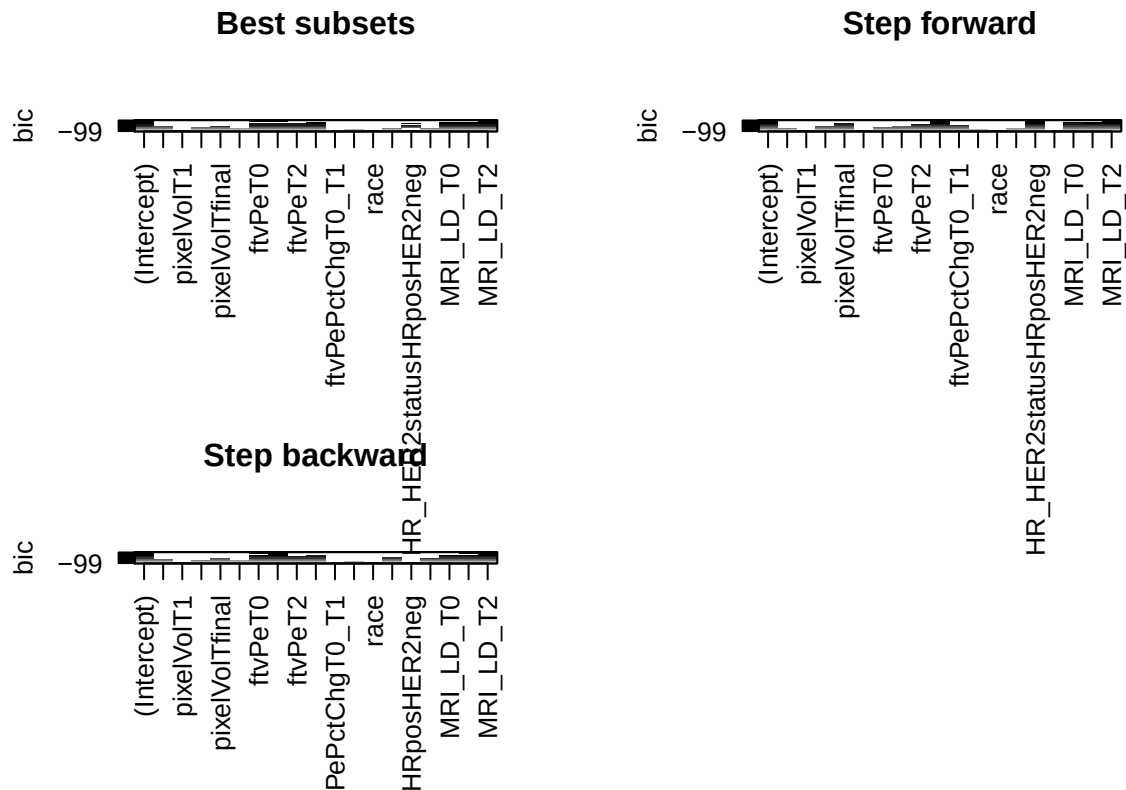




```
plot3("Cp")
```



```
plot3("bic")
```



Ridge regression

12. Install the package `glmnet`

```
## install.packages("glmnet")
```

13. Load the `glmnet` package

```
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

14. Convert the data into components `x` and `y` that are suitable for use in the `glmnet` package (we will continue to consider `MRI_LD_Tfinal` as the outcome)

```
x <- model.matrix(MRI_LD_Tfinal ~., data=dat)[, -1]
y <- dat$MRI_LD_Tfinal
head(x)
```

```
##      pixelVolT0 pixelVolT1 pixelVolT2 pixelVolTfinal pixelVolPctChgT0_T1
## 1 0.001220703 0.001220703 0.001220703 0.001220703 0
## 2 0.001220703 0.001220703 0.001220703 0.001220703 0
## 3 0.001220703 0.001220703 0.001220703 0.001220703 0
## 4 0.000741577 0.000741577 0.000988770 0.000741577 0
## 5 0.001220703 0.001220703 0.001220703 0.001220703 0
```

```
## 6 0.001220703 0.001220703 0.001220703 0.001220703 0
##      ftvPeT0      ftvPeT1      ftvPeT2 ftvPeTfinal ftvPePctChgT0_T1 age race
## 1 21.156006 0.01098633 0.0000000 0.0000000 -99.94807 38.73 1
## 2 5.310059 5.13061520 3.3630371 1.2145996 -3.37931 37.79 1
## 3 13.099365 9.20898440 5.2856445 1.9055176 -29.69900 49.83 1
## 4 42.757855 53.58117400 73.6119140 12.2286070 25.31305 48.28 1
## 5 3.208008 1.89941410 2.0129395 0.6811523 -40.79148 64.51 1
## 6 25.770264 11.01318400 0.8093262 0.1074219 -57.26399 40.66 4
##      HR_HER2statusHER2pos HR_HER2statusHRposHER2neg HR_HER2statusTripleNeg
## 1 0 1 0
## 2 0 1 0
## 3 0 1 0
## 4 0 0 1
## 5 0 1 0
## 6 0 0 1
##      MRI_LD_T0 MRI_LD_T1 MRI_LD_T2
## 1 88 78 30
## 2 29 26 66
## 3 50 64 54
## 4 91 90 99
## 5 45 49 47
## 6 75 66 57
```

```
head(y)
```

```
## [1] 14 16 46 43 32 7
```

15. Generate a set of values to try for labmba from $\lambda = 10^{10}$ to $\lambda = 10^{-2} = 0.01$

```
grid <- 10^seq(10,-2,length=100)
```

16. Fit ridge regression to the dataset with MRI_LD_Tfinal as outcome and all other variables as candidate predictors for all of the lambda values you have created

```
ridgeMod <- glmnet(x,y,alpha=0,lambda=grid)
```

17. Look at the value of lambda associated with ridgeMod[75]

```
ridgeMod$lambda[75]
```

```
## [1] 10.72267
```

18. Display the fitted coefficients associated with this value of lambda

```
coef(ridgeMod)[ ,75]
```

```
##      (Intercept)      pixelVolT0
## -2.741407e+00 -8.419582e+02
##      pixelVolT1      pixelVolT2
## -8.143580e+02 2.677042e+02
##      pixelVolTfinal pixelVolPctChgT0_T1
## 3.162981e+03 5.057216e-03
##      ftvPeT0      ftvPeT1
## -1.122046e-02 1.027397e-01
##      ftvPeT2      ftvPeTfinal
## 3.885998e-02 2.912748e-01
##      ftvPePctChgT0_T1      age
## 5.691822e-02 -2.389649e-02
##      race      HR_HER2statusHER2pos
```



```
##          1.659805e-01          -3.009956e+00
## HR_HER2statusHRposHER2neg    HR_HER2statusTripleNeg
##          2.671742e+00          -3.909710e+00
##          MRI_LD_T0           MRI_LD_T1
##          4.983714e-02          1.679492e-01
##          MRI_LD_T2
##          4.098273e-01
```

19. And then the l_2 norm associated with this value of `lambda`

```
sqrt(sum(coef(ridgeMod)[-1,75]^2))
```

```
## [1] 3383.522
```

20. Repeat steps 17 to 19 for `lambda` associated with index 55. Is the difference in the l_2 norm in the direction that you would expect? Why/why not?

```
ridgeMod$lambda[55]
```

```
## [1] 2848.036
```

```
coef(ridgeMod)[ ,55]
```

```
##          (Intercept)          pixelVolT0
##          2.891707e+01          -3.558791e+01
##          pixelVolT1          pixelVolT2
##          -6.817013e+01          -4.535014e+01
##          pixelVolTfinal    pixelVolPctChgT0_T1
##          -2.411387e+01          -9.335016e-04
##          ftvPeT0          ftvPeT1
##          1.994292e-03          5.204836e-03
##          ftvPeT2          ftvPeTfinal
##          5.304968e-03          1.104019e-02
##          ftvPePctChgT0_T1          age
##          1.570339e-03          -2.320637e-03
##          race          HR_HER2statusHER2pos
##          6.456858e-03          -3.060474e-02
## HR_HER2statusHRposHER2neg    HR_HER2statusTripleNeg
##          6.756150e-02          -8.117031e-02
##          MRI_LD_T0           MRI_LD_T1
##          5.239519e-03          5.953179e-03
##          MRI_LD_T2
##          7.823693e-03
```

```
sqrt(sum(coef(ridgeMod)[-1,55]^2))
```

```
## [1] 92.4759
```

Yes, it is in the expected direction. The l_2 norm is much smaller for index 55 because the value of λ is much larger. Therefore the shrinkage of coefficients toward zero is much stronger and therefore the sum of squares of the coefficients is much smaller.

21. Use the `predict` command to find the ridge regression coefficients for $\lambda = 3000$

```
predict(ridgeMod, s=3000, type="coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##          1
## (Intercept)          2.895176e+01
## pixelVolT0          -3.424155e+01
```

```
## pixelVolT1 -6.556136e+01
## pixelVolT2 -4.364712e+01
## pixelVolTfinal -2.325719e+01
## pixelVolPctChgT0_T1 -8.974192e-04
## ftvPeT0 1.917069e-03
## ftvPeT1 5.001561e-03
## ftvPeT2 5.097481e-03
## ftvPeTfinal 1.060735e-02
## ftvPePctChgT0_T1 1.508096e-03
## age -2.229673e-03
## race 6.201290e-03
## HR_HER2statusHER2pos -2.936737e-02
## HR_HER2statusHRposHER2neg 6.485855e-02
## HR_HER2statusTripleNeg -7.793758e-02
## MRI_LD_T0 5.034330e-03
## MRI_LD_T1 5.719253e-03
## MRI_LD_T2 7.514447e-03
```

Lasso

22. Fit lasso regression to the dataset with `MRI_LD_Tfinal` as outcome and all other variables as candidate predictors for the same set of lambda values you considered in fitting ridge regression

```
ridgeModLasso <- glmnet(x,y,alpha=1,lambda=grid)
```

23. Repeat steps 17 to 19 (again with index 75) but this time for the lasso fit

```
ridgeModLasso$lambda[75]
```

```
## [1] 10.72267
```

```
coef(ridgeMod)[ ,75]
```

```
## (Intercept) pixelVolT0
## -2.741407e+00 -8.419582e+02
## pixelVolT1 pixelVolT2
## -8.143580e+02 2.677042e+02
## pixelVolTfinal pixelVolPctChgT0_T1
## 3.162981e+03 5.057216e-03
## ftvPeT0 ftvPeT1
## -1.122046e-02 1.027397e-01
## ftvPeT2 ftvPeTfinal
## 3.885998e-02 2.912748e-01
## ftvPePctChgT0_T1 age
## 5.691822e-02 -2.389649e-02
## race HR_HER2statusHER2pos
## 1.659805e-01 -3.009956e+00
## HR_HER2statusHRposHER2neg HR_HER2statusTripleNeg
## 2.671742e+00 -3.909710e+00
## MRI_LD_T0 MRI_LD_T1
## 4.983714e-02 1.679492e-01
## MRI_LD_T2
## 4.098273e-01
```

```
sqrt(sum(coef(ridgeMod)[-1,75]^2))
```

[1] 3383.522

24. We have not performed any evaluation of prediction error here or comparison of methods. Outline the steps to compare the predictive accuracy in terms of mean squared error (MSE) between ridge regression and lasso using training/validation/test partitions of the data.

First split the data into a training, validation, and test partition. Then in the training set fit each of ridge and lasso regression for a large set of values of λ . Next apply all of the fitted models (i.e. for all values of λ and for both fitting procedures) to the validation set and determine the best values of λ for each of ridge and lasso with respect to minimizing mean square error. Finally, fit the best λ ridge and lasso models determined from the validation set to the test dataset and estimate mean squared error in each case. The one with the smallest mean squared error wins.

25. The dataset is not particularly large here. How might cross-validation be used to make better use of the full data?

The most straightforward place to use cross-validation here would be for estimation of λ . Combine the training and validation sets and use cross-validation in this combined dataset to estimate λ for each of ridge and lasso. Then perform ridge and lasso to the combined set with the best values of λ . Finally, evaluate the MSE for each of these two fitted models to the test set as was done with training-validation-test.

The next level of complexity would be to perform “nested-cross-validation”. In this setting cross-validation is used to estimate mean-square-error. However, within each training set (i.e. for each training partition of $k - 1$ folds, a further layer of cross-validation is performed to estimate the optimal λ for that fold. In the end you could take the average of the cross-validation λ values and refit to the full dataset to get the final model.