ridgereg

Niclas Lovsjö, Maxime Bonneau

11 oktober 2015

We have built the function ridgereg, which makes ridge regression out of a dataset by being fed a formula and some data. We can also define a lambda, which the default is set to 0. This is done like this,

```
library(lab4bis)
data(iris)
model1<-ridgereg(Sepal.Length~Sepal.Width+Petal.Length,iris)
model1$coefficients
```

```
##              Sepal.Length
## Intercept       5.8433333
## Sepal.Width     0.2595692
## Petal.Length    0.8330796
```

```
model2<-ridgereg(Sepal.Length~Sepal.Width+Petal.Length,iris, lambda=2)
model2$coefficients
```

```
##              Sepal.Length
## Intercept       5.7664474
## Sepal.Width     0.2497029
## Petal.Length    0.8178743
```

Predictions are done by,

```
newdata<-data.frame(c(1,2,3),c(4,3,2))
predict(model2,newdata)
```

```
##                  [,1]     [,2]     [,3]
## Sepal.Length 9.287648 8.719476 8.151305
```

Now we will comment on the caret-package part. We have decided to use our own API data from lab5. In there we have data of the election of year 2014 in Sweden. We will try to build a model explaining the size of a city, in terms of people allowed to vote, by the distribution of votes the parties in that particular city has. For example, say the distribution of the great little city Filipstad has 40%(S),30%(SD),20%(V),5%(M). Then can we find a good prediction of the size of this city? The features will then be the party percentages.

First we divide the data into a training and a test set:

```
#extract the data we need and partition:
library(lattice)
library(caret)
```

```
## Loading required package: ggplot2
```

```
divide_data<-function(theData){
  data <- data.frame(theData[,unlist(lapply(colnames(theData),function(y) substr(y,start=nchar(y)-2,sto
  data<-data.frame(data,theData$Rostb)
  colnames(data)<-c(colnames(theData)[unlist(lapply(colnames(theData),function(y) substr(y,start=nchar(
  set.seed(12345)
  in_train <- createDataPartition(y = data$Rostb,p=0.75,list = FALSE)
  train <- theData[in_train,]
  test<-theData[-in_train,]
  out<-list(train=train,test=test)
  #how to actually get the coefficients???
  return(out)
}
data<-divide_data(theData)
```

This is how the data will look like:

```
data$train[1:2,]
```

```
##   LAN KOM         LAAN     KOMMUN M.tal M.proc C.tal C.proc FP.tal FP.proc
## 1  10  82 Blekinge län  Karlshamn  3923  18.69  1015   4.84    616    2.93
## 2  10  80 Blekinge län Karlskrona  8848  20.71  2493   5.83   2225    5.21
##   KD.tal KD.proc S.tal S.proc V.tal V.proc MP.tal MP.proc SD.tal SD.proc
## 1    647    3.08  8114  38.65  1044   4.97   1192    5.68   3867   18.42
## 2   1674    3.92 15522  36.33  1954   4.57   2309    5.40   6603   15.45
##   FI.tal FI.proc OVR.tal OVR.proc BL.tal BL.proc OG.tal OG.proc
## 1    430    2.05     144     0.69    218    1.03      3    0.01
## 2    755    1.77     344     0.81    364    0.84      4    0.01
##   Rost.Giltiga Rostande Rostb   VDT
## 1        20992    21213 24702 85.88
## 2        42727    43095 49012 87.93
```

Then we fit a linear model using the caret package, and also a linear model using foreward-selection.