HW4

Qiuying Li UNI ql2280

10/3/2017

1. Investigate whether there is any multicollinearity, and suggest remedial measures if appropriate.

```
library("MASS", lib.loc="/Library/Frameworks/R.framework/Versions/3.3/Resources/library")
data = Boston
multi_lm = lm(medv~crim+zn+indus+nox+rm+age+tax,data)
summary(multi_lm)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + nox + rm + age + tax,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.625  -3.161  -0.833   2.089  41.042
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.615259   3.221482  -6.089 2.27e-09 ***
## crim         -0.132538   0.038482  -3.444 0.000621 ***
## zn            0.022103   0.014823   1.491 0.136547
## indus        -0.014980   0.072282  -0.207 0.835909
## nox           0.010643   4.230468   0.003 0.997994
## rm            7.606508   0.418424  18.179  < 2e-16 ***
## age          -0.023198   0.014893  -1.558 0.119964
## tax          -0.009006   0.002662  -3.384 0.000772 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.989 on 498 degrees of freedom
## Multiple R-squared:  0.5818, Adjusted R-squared:  0.576
## F-statistic: 98.99 on 7 and 498 DF,  p-value: < 2.2e-16
```

```r
anov = anova(multi_lm)
ss = anov$`Sum Sq`
VIF = 1/(1-ss[-length(ss)]/sum(ss))
anov$VIF = c(VIF," ")
anov
```

```
## Analysis of Variance Table
##
## Response: medv
##            Df  Sum Sq Mean Sq  F value  Pr(>F)    VIF
## crim        1  6440.8  6440.8 179.5726 0.00000 1.1776
## zn          1  3554.3  3554.3  99.0969 0.00000 1.0908
## indus       1  2551.2  2551.2  71.1299 0.00000 1.0635
## nox         1    28.7    28.7   0.7991 0.37180 1.0007
## rm          1 11794.6 11794.6 328.8410 0.00000 1.3814
## age         1    74.1    74.1   2.0656 0.15128 1.0017
## tax         1   410.6   410.6  11.4491 0.00077 1.0097
## Residuals 498 17861.9    35.9
```

```r
VIF_bar = mean(VIF);VIF_bar
```

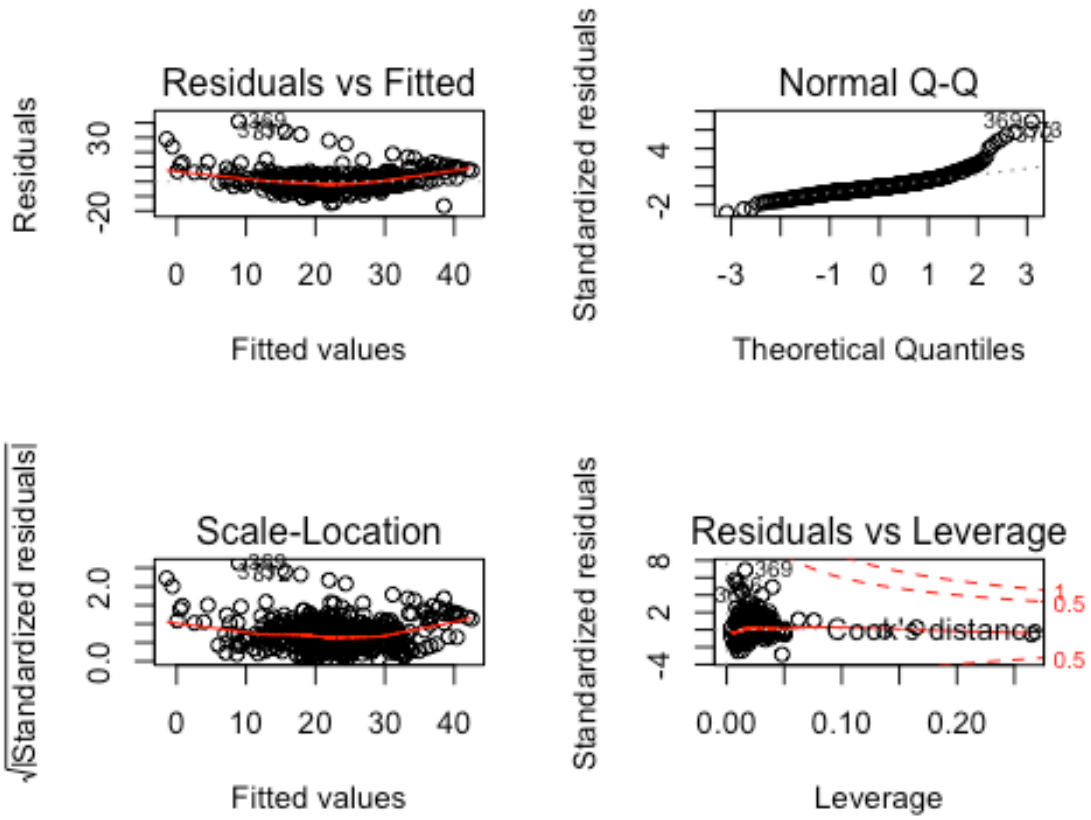
```
## [1] 1.103626
```

```r
names<-c("crim","zn","indus","nox","rm","age","tax")
explanatory<-as.matrix(Boston[names])
dependent<-as.matrix(Boston["medv"])
corr_mat<-cor(explanatory)
eigen_values<-eigen(corr_mat)$values
con_number<-max(eigen_values)/min(eigen_values);con_number
```

```
## [1] 19.45283
```

2.Compare results based on the usual linear regression and Principal Component Regression to predict ‘medv' the available variables.

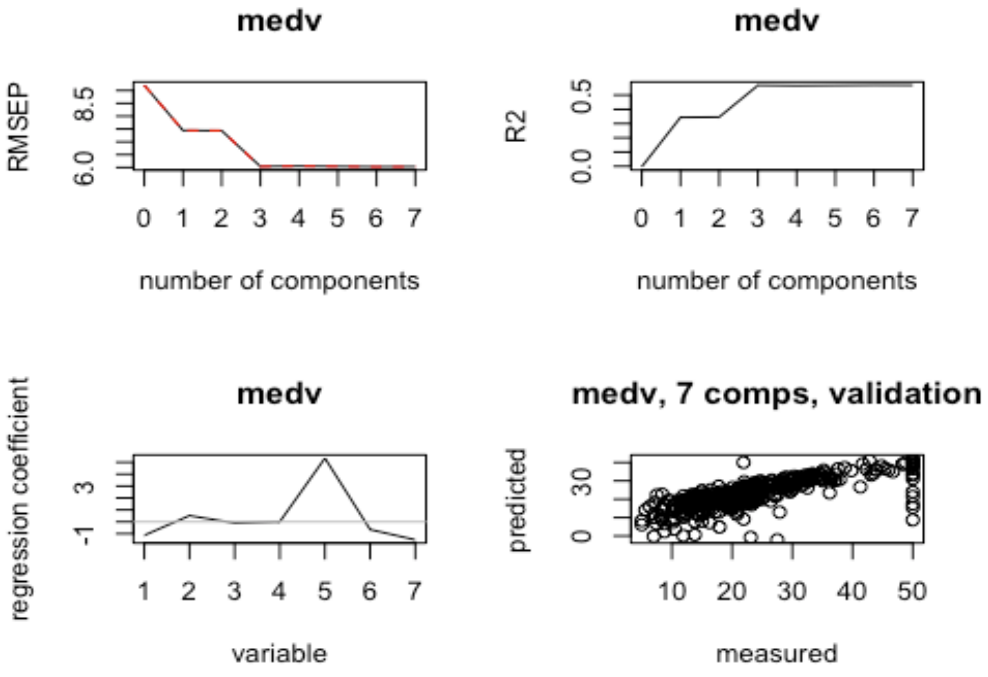
```
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings

data1 = as.data.frame(Boston)
lm = lm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)
par(mfrow = c(2,2))
plot(lm)
```



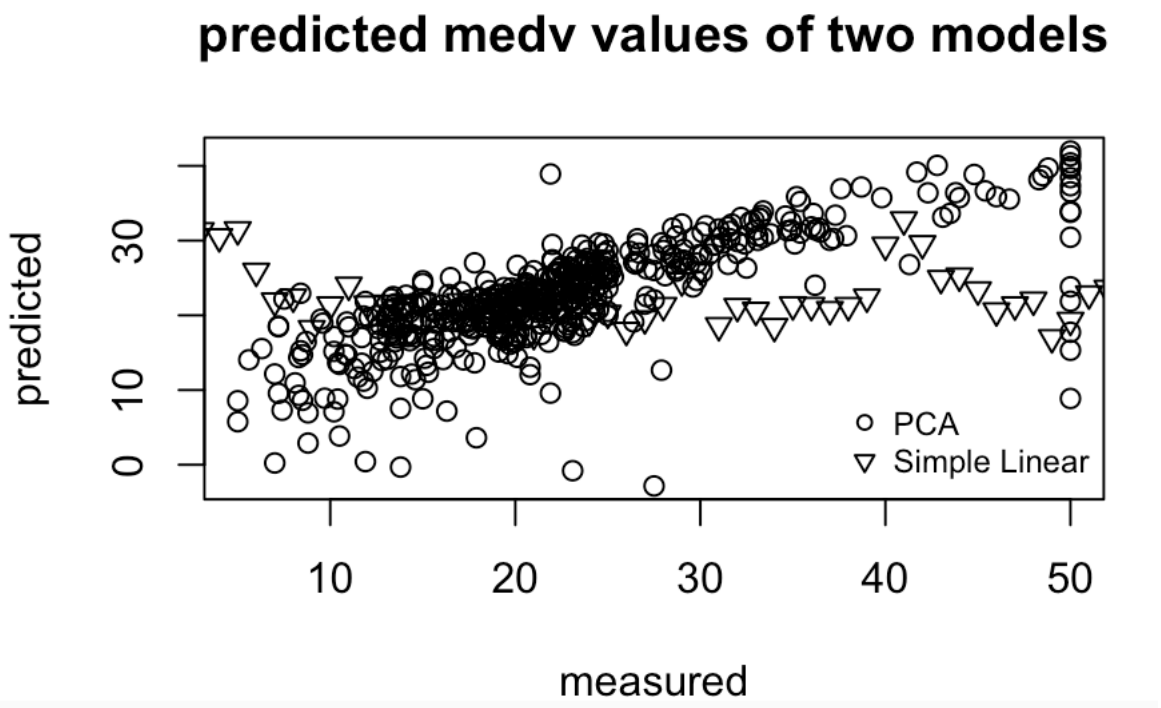

```
pcr_model <- pcr(medv~crim+zn+indus+nox+rm+age+tax,data=Boston, scale = TRUE,
 validation = "CV")
summary(pcr_model)
```

```
## Data:    X dimension: 506 7
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           9.206    7.445    7.433    6.038    6.052    6.041    6.029
## adjCV        9.206    7.444    7.438    6.034    6.048    6.037    6.025
##        7 comps
## CV       6.032
## adjCV    6.027
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       55.53    69.11    81.10    88.61    94.00    97.15   100.00
## medv    34.72    34.87    57.42    57.45    57.75    58.03    58.18
```

```r
pcr_pred <- predict(pcr_model, Boston, ncomp = 3);pcr_pred

validationplot(pcr_model)
validationplot(pcr_model, val.type = "R2")
coefplot(pcr_model)
predplot(pcr_model)
```

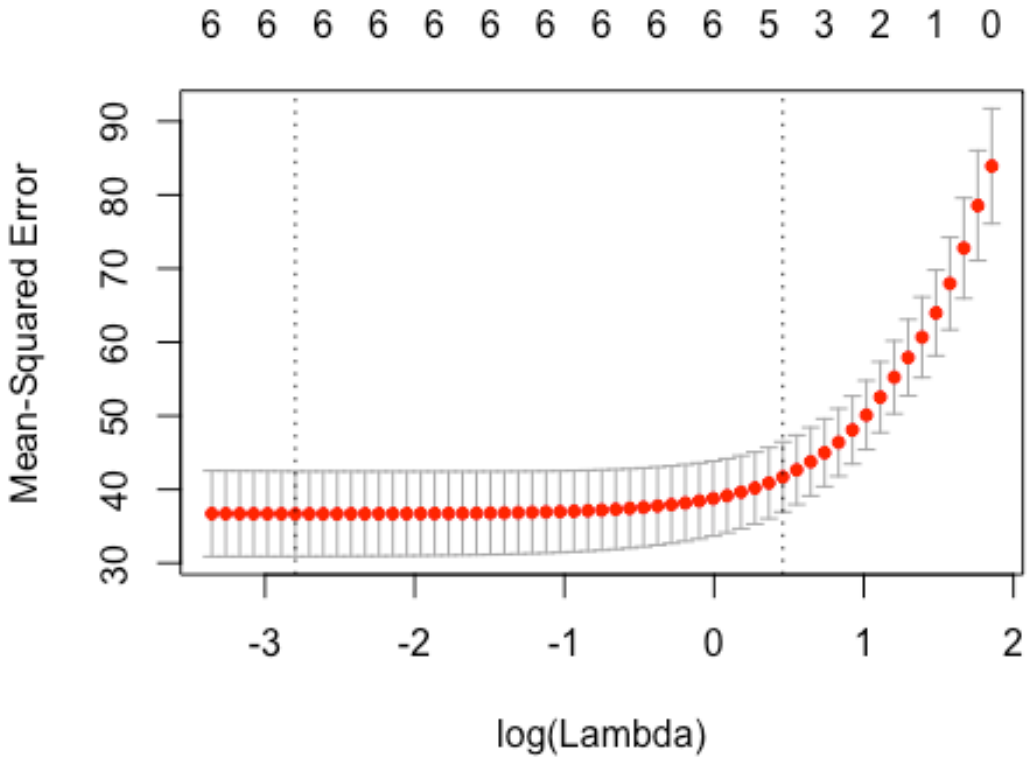


```r
par(mfrow = c(1,1))

predplot(pcr_model,main = "prediced medv values", pch = 1)
points(predict(lm), pch = 25)
legend('bottomright',c("PCA", "Simple Linear"), pch = c(1,25),bty='n', cex=.7
5)
```



3.Compare models selected using lasso vs a stepwise procedure to predict ‘ medv' using all available variables.

```
#Lasso regression
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-5

names<-c("crim","zn","indus","nox","rm","age","tax")
explanatory<-as.matrix(Boston[names])
dependent<-as.matrix(Boston["medv"])
lasso_reg<-glmnet(explanatory,dependent)
summary(lasso_reg)

##           Length Class     Mode
## a0         58    -none-    numeric
## beta      406    dgCMatrix S4
## df         58    -none-    numeric
## dim         2    -none-    numeric
## lambda     58    -none-    numeric
## dev.ratio  58    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        3    -none-    call
## nobs        1    -none-    numeric

cv_reg<-cv.glmnet(explanatory,dependent)
plot(cv_reg)
```




```
#choose model coefficient
lambda<-cv_reg$lambda.min
coeff<-coef(cv_reg,s="lambda.min")
predict_lasso<-predict(cv_reg,newx = explanatory,s="lambda.min")
MSE_lasso<-mean((dependent-predict_lasso)^2);MSE_lasso

## [1] 35.3082

#stepwise regression
glm_model1<-glm(medv~1,data=Boston)
glm_model2<-glm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)

backward<-stepAIC(glm_model2,direction = "backward",scope=list(
  upper = glm_model2,lower = glm_model1),trace = F)
summary(backward)

##
## Call:
## glm(formula = medv ~ crim + zn + rm + age + tax, data = Boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -16.669   -3.167   -0.808    2.075   41.083
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.713176   2.862677  -6.886 1.73e-11 ***
## crim         -0.131852   0.038261  -3.446 0.000617 ***
## zn            0.022947   0.014231   1.612 0.107487
## rm            7.625253   0.408770  18.654  < 2e-16 ***
## age          -0.024121   0.012709  -1.898 0.058271 .
## tax          -0.009323   0.002139  -4.358 1.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 35.72726)
##
##     Null deviance: 42716  on 505  degrees of freedom
## Residual deviance: 17864  on 500  degrees of freedom
## AIC: 3253.3
##
## Number of Fisher Scoring iterations: 2
```

```
forward<-stepAIC(glm_model1,direction = "forward",scope=list(
  upper = glm_model2,lower = glm_model1),trace = F)
summary(forward)
```

```
##
## Call:
## glm(formula = medv ~ rm + tax + crim + age + zn, data = Boston)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -16.669   -3.167   -0.808    2.075   41.083
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -19.713176   2.862677  -6.886 1.73e-11 ***
## rm            7.625253   0.408770  18.654  < 2e-16 ***
## tax          -0.009323   0.002139  -4.358 1.59e-05 ***
## crim         -0.131852   0.038261  -3.446 0.000617 ***
## age          -0.024121   0.012709  -1.898 0.058271 .
## zn            0.022947   0.014231   1.612 0.107487
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 35.72726)
##
##     Null deviance: 42716  on 505  degrees of freedom
## Residual deviance: 17864  on 500  degrees of freedom
## AIC: 3253.3
##
## Number of Fisher Scoring iterations: 2
```

```r
#calculate MSE
predict_back<-predict(backward,newx = explanatory)
predict_for<-predict(forward,newx = explanatory)
MSE_back<-mean((dependent-predict_back)^2);MSE_back

## [1] 35.30361

MSE_for<-mean((dependent-predict_for)^2);MSE_for

## [1] 35.30361

#comparison plot
par(mfrow = c(1,3))
plot(predict_lasso)
plot(predict_back)
plot(predict_for)
```