HW4_solution

Issac Li

2/20/2017

Part I

```
require(data.table)
communities=fread("/Users/lizhuo/Documents/STAT665/HW4/communities.csv",heade
r=T)
# Unique States
length(unique(communities$state))
```

```
## [1] 46
```

```
# Unique County
# Apparently counties with same ID in different states are distinct counties.
state_county=unique(communities[,list(state,county)])
lb=dim(state_county)[1]
# We do not have information on whether NAs counties are distinct or not, so
the maximum possible
# no of counties should treat all NAs as different, but excludes the NAs alre
ady counted above.
ub=dim(state_county)[1]+sum(is.na(communities$county))-sum(is.na(state_county
$county))
paste("The number of unique counties is between",lb,"and",ub)
```

```
## [1] "The number of unique counties is between 280 and 1421"
```

```
# Frequency Table 1
find_na<-function(x){
 sum(is.na(x))
}
ftable <-data.frame(table(communities[, apply(.SD,2, find_na)]))
colnames(ftable)<-c("# Missing","Freq")
ftable
```

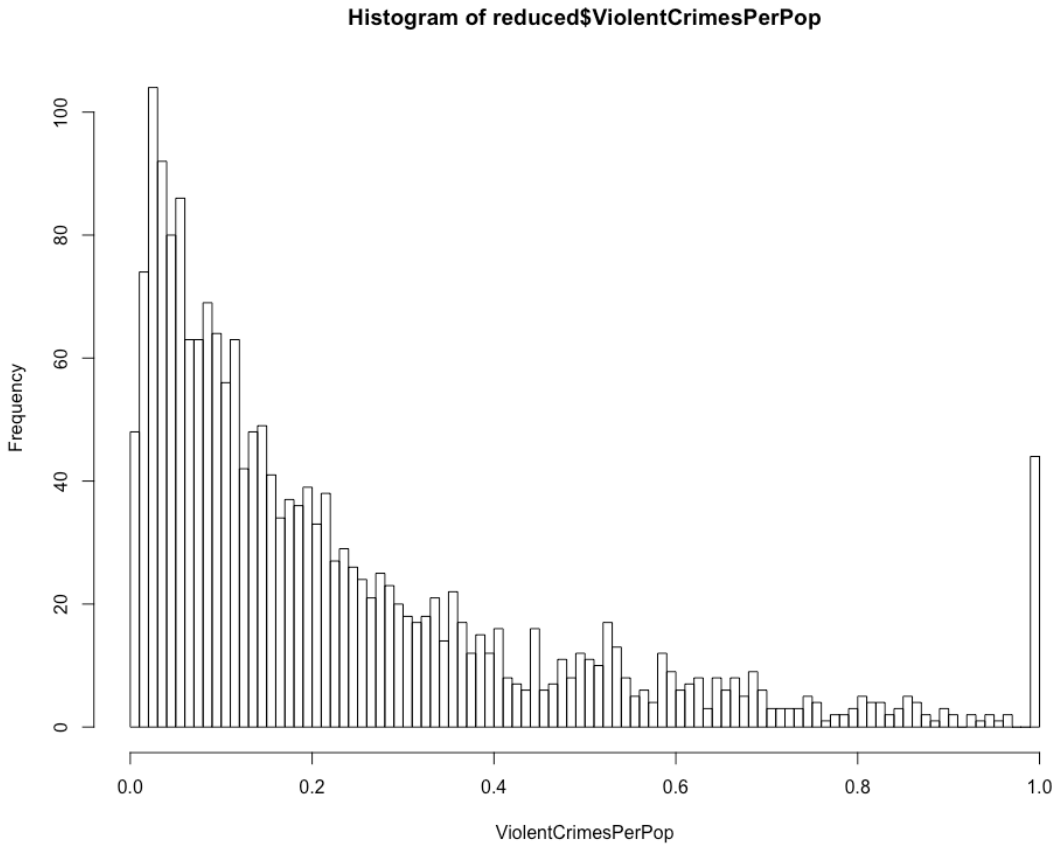
```
##   # Missing Freq
## 1         0  103
## 2         1    1
## 3      1174    1
## 4      1177    1
## 5      1675   22
```

```
# Reduce data
reduced<-communities[,colSums(is.na(communities))<(nrow(communities)*0.5),wit
h=F]
ftable <-data.frame(table(reduced[, apply(.SD,2, find_na)]))
```

```
colnames(ftable)<-c("# Missing","Freq")
ftable
```

```
##   # Missing Freq
## 1         0  103
## 2         1    1
```

```
# Plot hist of ```ViolentCrimesPerPop```
hist(reduced$ViolentCrimesPerPop,nclass=100,xlab="ViolentCrimesPerPop")
```



We can see that the distribution of data is left-skewed and the data is normalized between 0 and 1.

```
# Remove any column with missing values
reduced<-reduced[,colSums(is.na(reduced))==0,with=F]

# Drop the name, fold and goal column
reduced_mat<-reduced[,-c("communityname","fold","ViolentCrimesPerPop"),with=F
]
# Now we have 100 columns
dim(reduced_mat)
```

```
## [1] 1994  100
```

```r
# Remove those highly correlated variables
reduced_mat=matrix(as.numeric(unlist(reduced_mat)),nrow = 1994,ncol = 100,dim
names = list(1:1994,colnames(reduced_mat)))
cor_mat=cor(reduced_mat)

cor_mat[lower.tri(cor_mat)]=0
res=which(cor_mat>=0.9 & cor_mat!=1,arr.ind = T)

# These predictors are probably redundant
tail(res[order(res[,1]),])
```

```
##            row col
## RentLowQ    83  84
## RentLowQ    83  85
## RentLowQ    83  86
## RentMedian  84  85
## RentMedian  84  86
## RentHighQ   85  86
```

```r
# Remove all the predictors in the second column
to_remove=unique(res[,2])
upper=colnames(reduced_mat)[-to_remove]

# We have successfully reduced the maximal possible # of predictors by 30%.
length(upper)
```

```
## [1] 69
```

```r
# State can be considered a nominal predictive factor here (factorized), but
when
# doing k-fold cross validation, levels maybe missing due to truncated datase
t, so
# I have to remove this factor, even though it explains more variance in-samp
le

reduced<-reduced[,-c("state"),with=F]
upper=upper[-1]

fit.full=lm(formula = as.formula(paste("ViolentCrimesPerPop~",paste(upper,col
lapse = "+"))),data = reduced)
fit.null=lm(formula = ViolentCrimesPerPop~1,data = reduced)
fit.for=step(object = fit.null,scope = list(upper=fit.full),trace = F,directi
on = "forward")

fit.back=step(object = fit.full,trace = F,direction = "backward")
fit.bi=step(object = fit.null,scope = list(upper=fit.full),trace = F,directio
n = "both")

table1<-data.frame(matrix(NA,3,2,dimnames = list(c("fit.for","fit.back","fit.
bi"),c("R^2","No Terms"))))
```

```
f1 <- formula(fit.for)
f2 <- formula(fit.back)
f3 <- formula(fit.bi)

table1$No.Terms[1]<-length(strsplit(as.character(f1[3])[[1]],split = "+",fixe
d=T)[[1]])
table1$No.Terms[2]<-length(strsplit(as.character(f2[3])[[1]],split = "+",fixe
d=T)[[1]])
table1$No.Terms[3]<-length(strsplit(as.character(f3[3])[[1]],split = "+",fixe
d=T)[[1]])

table1$R.2[1]<-summary(fit.for)$`r.squared`
table1$R.2[2]<-summary(fit.back)$`r.squared`
table1$R.2[3]<-summary(fit.bi)$`r.squared`

round(table1,3)
```

```
##            R.2 No.Terms
## fit.for  0.671       24
## fit.back 0.680       42
## fit.bi   0.671       23
```

Part III

We can see that the model selected by bi-directional stepwise procedure is the best in terms of model complexity with an R^2 value very close to the other two.

```
summary(fit.bi)
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ PctIlleg + MalePctDivorce +
##     HousVacant + PctWorkMom + pctWWage + pctUrban + PctPersDenseHous +
##     racepctblack + NumStreet + population + MedOwnCostPctIncNoMtg +
##     PctFam2Par + PctPopUnderPov + pctWRetire + PctVacantBoarded +
##     MedRentPctHousInc + pctWInvInc + LemasPctOfficDrugUn + HispPerCap +
##     NumInShelters + pctWFarmSelf + AsianPerCap + indianPerCap,
##     data = reduced)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53327 -0.07318 -0.01362  0.04784  0.80155
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.415707   0.071921   5.780 8.66e-09 ***
## PctIlleg              0.187925   0.038976   4.822 1.53e-06 ***
## MalePctDivorce        0.101928   0.031946   3.191 0.001442 **
## HousVacant            0.269213   0.049514   5.437 6.09e-08 ***
```

```
## PctWorkMom             -0.090170   0.022288  -4.046 5.42e-05 ***
## pctWWage               -0.149076   0.032500  -4.587 4.78e-06 ***
## pctUrban                0.037375   0.008665   4.314 1.69e-05 ***
## PctPersDenseHous        0.199603   0.023341   8.552  < 2e-16 ***
## racepctblack            0.219800   0.023298   9.434  < 2e-16 ***
## NumStreet               0.159693   0.044431   3.594 0.000333 ***
## population             -0.267133   0.068469  -3.901 9.88e-05 ***
## MedOwnCostPctIncNoMtg -0.081517   0.018146  -4.492 7.45e-06 ***
## PctFam2Par             -0.229561   0.057719  -3.977 7.23e-05 ***
## PctPopUnderPov         -0.184587   0.033342  -5.536 3.50e-08 ***
## pctWRetire             -0.085930   0.027916  -3.078 0.002111 **
## PctVacantBoarded        0.050846   0.018256   2.785 0.005401 **
## MedRentPctHousInc       0.080686   0.022015   3.665 0.000254 ***
## pctWInvInc             -0.133940   0.035836  -3.738 0.000191 ***
## LemasPctOfficDrugUn     0.032479   0.015069   2.155 0.031257 *
## HispPerCap              0.040108   0.021879   1.833 0.066930 .
## NumInShelters           0.106084   0.059370   1.787 0.074119 .
## pctWFarmSelf            0.031579   0.018536   1.704 0.088595 .
## AsianPerCap             0.030118   0.017835   1.689 0.091443 .
## indianPerCap           -0.031956   0.019184  -1.666 0.095915 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1343 on 1970 degrees of freedom
## Multiple R-squared:  0.6714, Adjusted R-squared:  0.6675
## F-statistic:   175 on 23 and 1970 DF,  p-value: < 2.2e-16
```

From the summary above, we can see that the PctPersDenseHous pctWWage and a bunch o f other predictors are significant. Take these two for example, 1- unit increase in percent of persons in dense housing correlates with 13% percent increase (since normalized to 1) in pe rcent violent crimes and 1-unit increase in percent of percentage people working on wage co rrelates with 14.9 percent decrease in violent crimes. These results make sence.

```
MSE=0
k=10
n=nrow(reduced)
for (i in 1:k){
valid_ind=which(reduced$fold==i,arr.ind = F)
fit.temp=lm(formula = f3,data = reduced[-valid_ind,])
pred_lm=predict.lm(fit.temp,newdata = reduced[valid_ind,])
# Calculate Average MSE
MSE=MSE+1/n*sum((reduced[valid_ind,c(ViolentCrimesPerPop)]-pred_lm)^2)
}

MSE
```

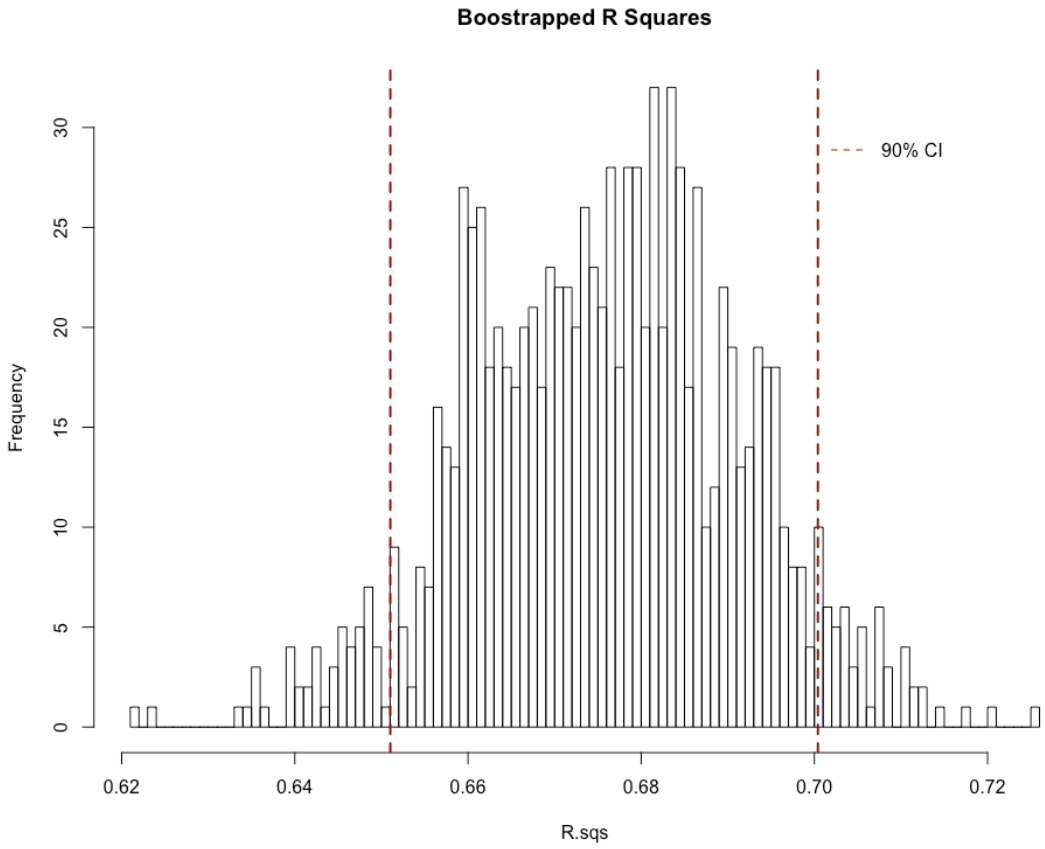
```
## [1] 0.01836488
```

The raw MSE is 0.0184. However, this value is not very interpretative in absolute sense since all the data are normalized. This value can nonetheless help us determining difference between models relatively.

```
# Do a bootstrap 1000 times and produce practical 90% CI for R^2
R.sqs=rep(NA,1000)

for( i in 1:1000){
set.seed(i*8+13)
samples=sample(1:nrow(reduced),nrow(reduced),replace = T)
R.sqs[i]=summary(lm(formula = f3,data = reduced[samples,]))$`r.squared`
}

lb=quantile(R.sqs,0.05)
ub=quantile(R.sqs,0.95)
hist(R.sqs,nclass = 100, main = "Boostrapped R Squares")
abline(v=c(lb,ub),col="dark red",lty=2,lwd=2)
legend(0.7,y=30,bty = "n" ,legend = "90% CI",lty=2,col="dark red")
```



The 90% CI for R^2 is [0.651,0.700].