Dimension Deduction:

Prepare Dataset

```
# get dataset file
data <- read.csv('communities.data', header = FALSE)
# get dataset variable names
name_data <- read.delim('names', header = FALSE, sep = ' ')
# set dataset column names
names(data) <- name_data[,2]
# drop ‘communityname’ column
data['communityname'] <- NULL

# inpulate missing value with column mean
for(i in 1:ncol(data)){
  # transfer every column type to numeric
  data[,i] <- as.numeric(data[,i])
  data[data[,i] == '?',i] <- mean(data[,i], na.rm = TRUE)
}

# set random seed is 100
set.seed(100)
# split data to training and validation set, training set is 60%
train_ind <- sample(seq_len(nrow(data)), size = (0.6 * nrow(data)))
TS <- data[train_ind, ]
VS <- data[-train_ind, ]
```

I Baseline Regression Model:

In this part using lm function to do the linear regression model.

```
> ptm <- proc.time()
> fit_model <- lm(data = TS, ViolentCrimesPerPop ~ .)
> proc.time() - ptm
   user  system elapsed
  0.044   0.002   0.049

# validate the training model
pred <- predict.lm(fit_model, VS[,1:126])

SSE <- sum((VS$ViolentCrimesPerPop - pred)^2)
RMSE <- sqrt(mean((VS$ViolentCrimesPerPop - pred)^2))
RSE <- sum((VS$ViolentCrimesPerPop -
pred)^2)/sum((mean(VS$ViolentCrimesPerPop)-VS$ViolentCrimesPerPop)^2)
```

```
SST <- sum((mean(VS$ViolentCrimesPerPop)-VS$ViolentCrimesPerPop)^2)
R_square <- 1-SSE/SST
```

II Feature Selection: Sequential Subset Selection

feature selection is the process of selecting a subset of relevant features for use in model construction. In this part, I used the step function to do the stepwise feature selection.

```
model_step <- step(fit_model,direction = "both")
> proc.time() - ptm
   user  system elapsed
110.339   2.492 112.627
```

At the beginning, the algorithm chooses one variable that makes the error minimum. On every iteration, the algorithm adds a variable into the model, if the error less than before, then go to next iteration. The program will stop until there is no variable can be added in the model to make the error less than before.

```
> dim(model_step$model)
[1] 1196   51
```

There are 51 variables remained.

```
prediction <- predict(model_step, newdata = VS[,1:126])
```

III Feature Selection: Ranking Attributes

The caret R package provides tools automatically report on the relevance and importance of attributes in the data and can select the most important features out.

```
# load the library
library(mlbench)
library(caret)

# get the top 50 important variables
varimp <- varImp(fit_model)
varimp[,'names'] <- rownames(varimp)
imp <- varimp[order(varimp$Overall,decreasing = TRUE),]
top50 <- head(imp,50)$names

var <- top50[1]
for (i in 2:length(top50)){
  var <- paste(var,top50[i],sep = "+")
}
# print the top 50 variable names
```

var
ptm <- proc.time()
rank_model <- lm(data = TS,ViolentCrimesPerPop ~
PctPopUnderPov+NumStreet+PctIlleg+pctWRetire+RentLowQ+PctKids2Par+NumImmig+PctNotSpeakEnglWell+PctHousNoPhone+PersPerRentOccHous+MalePctNevMarr+PolicBudgPerPop+PersPerOccupHous+whitePerCap+MedOwnCostPctInc+PolicReqPerOffic+PctVacMore6Mos+PctLess9thGrade+NumUnderPov+PctPolicAsian+county+PctSpeakEnglOnly+pctWFarmSelf+LemasPctOfficDrugUn+PctVacantBoarded+LemasTotalReq+PctPolicBlack+PolicOperBudg+PctHousOccup+racepctblack+PctOccupMgmtProf+MedOwnCostPctIncNoMtg+PctPersDenseHous+PolicCars+MedRentPctHousInc+HousVacant+PctEmplManu+PersPerOwnOccHous+MedRent+medIncome+MedNumBR+PctUsePubTrans+PctHousLess3BR+PctSameState85+PctPersOwnOccup+agePct12t29+PctPolicHisp+pctWSocSec+MedYrHousBuilt+perCapInc)
proc.time() – ptm
user system elapsed
  0.015   0.000   0.015

IV Feature Extraction: Principal Components Analysis

Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, where as feature selection methods include and exclude attributes present in the data without changing them.

The function prcomp() comes with the default "stats" package.
pca <- prcomp(TS[,1:126], retx=TRUE, center=TRUE, scale=TRUE)
summary(pca)

There are 126 components were constructed. The minimum number of components needed to capture at least 90% of the data variance is 28, because the cumulative proportion of PC28 is 0.90476.

Prepare the pca data:
newdata <- pca$x[,1:28]
newdata <- data.frame(newdata)
newdata[,29] <- TS[,127]

Fit a linear model with pca data:
fitmodel <- lm(data = newdata, V29 ~ .)
user system elapsed
  0.103   0.008   0.114

Transform the validation data using pca model:
pred.vs <- predict(pca, VS[,1:126])
pred.vs <- data.frame(pred.vs)

```
pred.vs <- pred.vs[,1:28]
```

Validation the result:

```
prediction <- predict(fitmodel, newdata = pred.vs)
```

V Feature Extraction: Factor Analysis (FA)

The factor.pa( ) function in the psych package offers a number of factor analysis related functions, including principal axis factoring.

```
library(psych)
ptm <- proc.time()
fa_fit <- factor.pa(TS[,1:126], nfactors=30)
fa_data <- predict(object = fa_fit, data = TS[,1:126])
# set 'ViolentCrimesPerPop' column
fa_data <- data.frame(fa_data)
fa_data[,31] <- TS[,127]
colnames(fa_data)[31] <- "ViolentCrimesPerPop"
fit_fa_model <- lm(data = fa_data, ViolentCrimesPerPop ~ .)
proc.time() - ptm

user  system elapsed
  0.593   0.020   0.613

# transfer Validation data using FA model
fa_VS_data <- predict(object = fa_fit, data = VS[,1:126])
fa_VS_data <- data.frame(fa_VS_data)
fa_VS_data[,31] <- VS[,127]
colnames(fa_VS_data)[31] <- "ViolentCrimesPerPop"
prediction <- predict(fit_fa_model, newdata = fa_VS_data[,1:30])
```

VI Comparison of Results

	Baseline	Sequential Subset Selection	Relief	PCA	FA
Number of attributes used to construct the linear regression model	127	127	127	28	30
Number of attributes appearing in the	127	51	50	28	30

linear regression model					
Time taken constructing the linear regression model	0.049	112.627	0.015	0.114	0.613
Sum of Square Errors(SSE)	15.23605	15.54977	14.2841	14.70018	14.83396
Root Mean Square Error(RMSE)	0.1381767	0.139592	0.1337904	0.135725	0.1363412
Relative Square Error(RSE)	0.3811096	0.3889568	0.3572977	0.3677056	0.3710519
Coeffient of Determination(R^2)	0.6188904	0.6110432	0.6427023	0.6322944	0.6289481

From the table we can see that, using top 50 important variables model has the largest R_square. However, PCA only uses 28 principle components and ranks the second best R_Square, which is 0.632. It is surprise that using stepwise model get the least R_square value and model takes the longest time. The stepwise model uses 51 variables but the result of stepwise is worse than top 50 important variable model. The reason I think is the two models uses different criteria to assess the importance of the variables, and different libraries use different methods to solve the problem. Moreover, the top 50 importance variables, PCA, and FA are better than baseline model and stepwise model, and the time taken of top 50 importance variables model is the fastest model among these methods.