Decision Making with Business Analytics
LAB 1: Variable Selection and Dimension Reduction.

- Basic guidance on how to solve this lab in R is given. But, it is not required to solve the lab in R, MATLAB or other packages are also a good option.
- Any question outside lab time should be submitted through Canvas. We want to make sure the answers are accessible for everyone.
- Students are required to work in pairs for this lab. In each pair (as long as it is possible) there should be one students that took the Data Science course and one than not. This requirement is only for this lab.

Monthly US Unemployment Rates

In this lab we will analyze the (seasonally adjusted) monthly unemployment rates covering the period January 1976 through August 2010 for the 50 US states. You are given a data matrix X with 50 rows (representing the 50 states) and 416 columns (representing the monthly observations).

```
##Libraries needed in this lab
library(cluster)
library(mixOmics)
library(lars)
```

To read the data use:

```
## read the data; change directory name as needed
## series are stored column-wise with labels in first row
raw <- read.csv("C:/DMwBA/Data/unempstates.csv")
```

Now we create a handle to compute the transpose of `raw`

```
## transpose the data
rawt=matrix(nrow=50,ncol=416) rawt=t(raw)
```

Clustering the States according to their unemployment rates

We are interested now in finding similarities/differences between the different states in terms of unemployment rates. The objective is to cluster states into groups that are alike. Here each state is characterized by a feature vector of very large dimension ($p = 415$), with its components representing the 415 monthly observations.

k-means on raw data

k-means is a clustering algorithm. Use k-means to create 2 clusters from the data:

```
## k-means clustering in 415 dimensions
set.seed(1)
grpunemp <- kmeans(rawt, centers=2, nstart=10)
sort(grpunemp$cluster)
```

To visualize the clusters, we use a 2-dimensional plot. The data file unemp.csv includes the average and the standard deviation for the unemployment rate of each state.

```
## load data set unemp.csv
unemp <- read.csv("C:/DMwBA/Data/unemp.csv")
## list of cluster assignments
o=order(grpunemp$cluster)
## plot clusters
data.frame(unemp$state[o],grpunemp$cluster[o])
plot(unemp$mean,unemp$stddev,type="n",xlab="mean", ylab="stddev")
text(x=unemp$mean,y=unemp$stddev,labels=unemp$state, col=grpunemp$cluster+1)
```

1. Repeat the previous exercise for $k = 3$, 4, 5 clusters.

2. How many clusters seem to be more appropriate, 2, 3, 4 or 5? Call that number **numClusters**.

For the rest of the exercise pick $N = 4$ states, each one in a different cluster (when $k = 4$ clusters are chosen). We will predict future unemployment rate changes for this states. Call this states the *chosen states*.

Estimating unemployment rate changes

Now we consider the problem of estimating unemployment rate changes from past data for the chosen states. We investigate the performance of the following methods:

i **Naive estimator**: For each chosen state, predict monthly unemployment rate as the average unemployment rate of the last 4 periods. Then predict the rate change as predicted rate - last rate.

ii **Univariate AR(4) model fit to individual series**: Current unemployment rate change is regressed on its previous four lags. This is done separately for each chosen state.

iii **Multivariate VAR(4)**: Current unemployment rate change of each chosen state is regressed on its previous four lags and the four lags from all other states. This amounts to a regression on 200 features (variables).

Let $300 < n < 416$ the number of months used in the training set (more on this below). A simplified way of looking at the Multivariate VAR(4) is to interpret it (for each state) as a linear regression with n samples and $m = 200$ features. A problem with fitting a regression on so many similar predictor variables is that some predictor variables (perhaps most) are not needed. Thus we are going to use dimensionality reduction methods next.

iv **VAR(4) restricted to chosen states**: Assuming that there is lots of redundancy in the information given by two estates in the same cluster, we can use as predictors the chosen states only. That is, current unemployment rate change of each chosen state is regressed on its previous four lags and the four lags from the other chosen states. This amounts to a regression on 16 features (variables).

v **VAR(4) with LASSO constraints**: The least absolute shrinkage and selection operator (LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. You will need to consider several values for the LASSO parameter, usually called lambda (λ). Run LASSO to predict current unemployment rate change of each chosen state from its previous four lags and the four lags from all other states. Try $\lambda = 0.01$, 0.1, 1.0 10.0.

vi **VAR(4) with Random Projection**: In the lecture we will see that random projections preserve lots of information. Reduce the number of variables from $m = 200$ to $m' = 20$ by performing a random projection from $\mathbb{R}^m$ to $\mathbb{R}^{m'}$. That is generate a random matrix $M \in \mathbb{R}^{m \times m'}$ and instead of using the predictor matrix $X \in \mathbb{R}^{n \times m}$ use $X' = X \times M \in \mathbb{R}^{n \times m'}$ as predictor.

To evaluate the methods we separate the last 50 observations as test set. On the remaining (362) observations we use the following possibilities:

1. Compare the methods using R-square as measurement. What is the best model according to R-square?

2. Compare the methods using a validation set: Divide the remaining of the sample set in a validation set (40 randomly selected time periods) and a training set (the remaining 322 observations). Compare the methods using the root mean square error (RMSE) calculated from the $200 = 50 \times 4$ forecast errors. Repeat the random selection of the validation and training samples 100 times, and output box-plots of the RMSE. What is the best model according to RMSE in this case?

3. Compare the methods using the last 40 time periods as validation set. What is the best model according to RMSE in this case?

4. Which models are overfitting the data?

5. Pick the best model in each of the previously given options (1, 2 and 3) and use your test set to evaluate it and conclude. Notice that in reality you will be using only one of these three model selection methods.

6. Would your results change if you use 50 randomly selected time periods instead of the last 50 as test set? In which situations could this choice matter? In which ones could make no difference? What are the implications about the chosen model about using one or the other test set?