Homework 3 Solution

Byteflow Dynamics

10/8/2017

Perform PCA on the Boston data from class

1. Remove the column *medv* before calculating the principal components (we call this 'data' throughout this homework)
2. Perform PCA
3. Plot PVE to find the number of PCs to include in further analysis

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.1
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
data("Boston")

data <- Boston[,-14]

pr.out=prcomp(data, scale=TRUE)
summary(pr.out)
```

```
## Importance of components%s:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.4752 1.1972 1.11473 0.92605 0.91368 0.81081
## Proportion of Variance 0.4713 0.1103 0.09559 0.06597 0.06422 0.05057
## Cumulative Proportion  0.4713 0.5816 0.67713 0.74310 0.80732 0.85789
##                            PC7     PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.73168 0.62936 0.5263 0.46930 0.43129 0.41146
## Proportion of Variance 0.04118 0.03047 0.0213 0.01694 0.01431 0.01302
## Cumulative Proportion  0.89907 0.92954 0.9508 0.96778 0.98209 0.99511
##                           PC13
## Standard deviation     0.25201
## Proportion of Variance 0.00489
## Cumulative Proportion  1.00000
```

```
pr.out$sdev
```

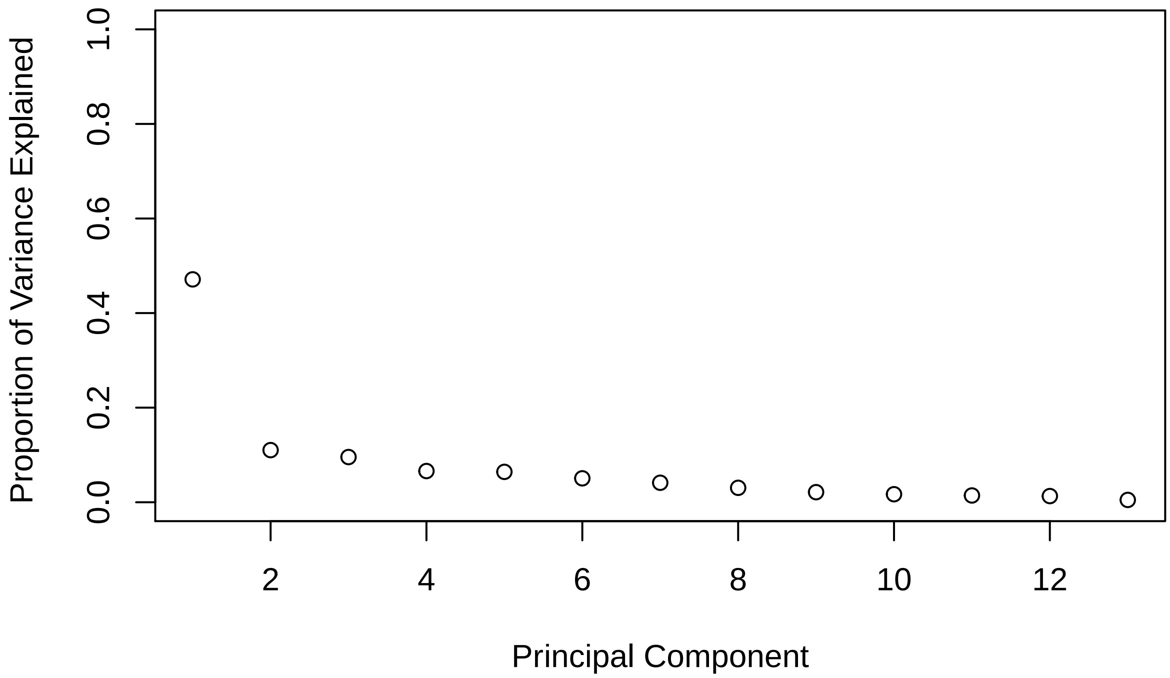
```
##  [1] 2.4752472 1.1971947 1.1147272 0.9260535 0.9136826 0.8108065 0.7316803
##  [8] 0.6293626 0.5262541 0.4692950 0.4312938 0.4114644 0.2520104
```

```
pr.var <- pr.out$sdev^2
pve <- pr.var / sum(pr.var)

pve
```

```
##  [1] 0.471296064 0.110251932 0.095585898 0.065967316 0.064216611
##  [6] 0.050569783 0.041181237 0.030469024 0.021303333 0.016941371
## [11] 0.014308797 0.013023306 0.004885328
```

```r
plot(pve, xlab="Principal Component", ylab="Proportion of Variance Explained ", ylim=c(0,1))
```




```r
# we'll keep 2 pcs
```

Perform multiple linear regression to model *medv*

1. First use all the variables for multiple linear regression

```r
lm.fit=lm(medv ~ ., data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
```

```
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

2. Use the most sigificant variables (pick 2-3 variables with lowest P values) for regression

```r
lm.fit=lm(medv ~ lstat + rm, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.076  -3.516  -1.010   1.909  28.131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.35827    3.17283  -0.428    0.669
## lstat       -0.64236    0.04373 -14.689   <2e-16 ***
## rm           5.09479    0.44447  11.463   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.54 on 503 degrees of freedom
## Multiple R-squared:  0.6386, Adjusted R-squared:  0.6371
## F-statistic: 444.3 on 2 and 503 DF,  p-value: < 2.2e-16
```

3. Use the PCs from past part as variables for regression
4. Check how different the results from steps 2, 3, and 4 are. Which one gives the best result?

```r
pc <- as.data.frame(pr.out$x[,1:2])

lm.fit=lm(Boston$medv ~ pc$PC1 + pc$PC2)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Boston$medv ~ pc$PC1 + pc$PC2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.589  -4.288  -1.759   2.446  33.917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

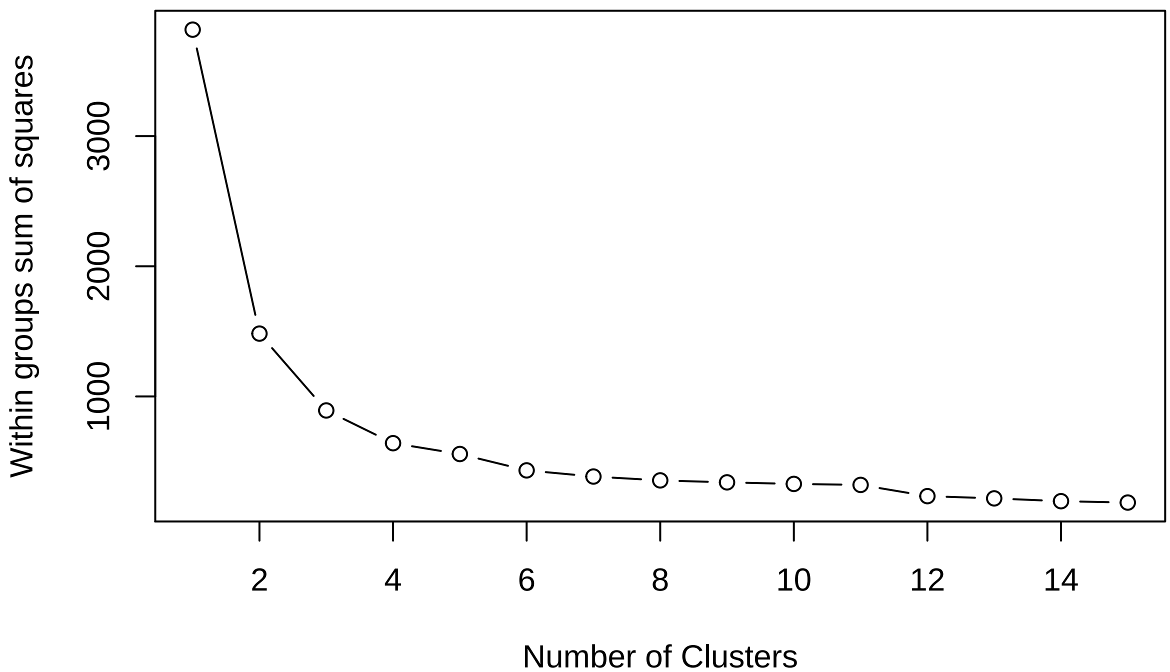
```
## (Intercept)  22.5328     0.3022  74.563   <2e-16 ***
## pc$PC1       -2.2730     0.1222 -18.599   <2e-16 ***
## pc$PC2        2.1949     0.2527   8.687   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.798 on 503 degrees of freedom
## Multiple R-squared:  0.4559, Adjusted R-squared:  0.4537
## F-statistic: 210.7 on 2 and 503 DF,  p-value: < 2.2e-16
```

Cluster data using PCs

1. Find the optimal number of clusters
2. Cluster data into clusters based on PCs

```
wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
     set.seed(seed)
     wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")}
```

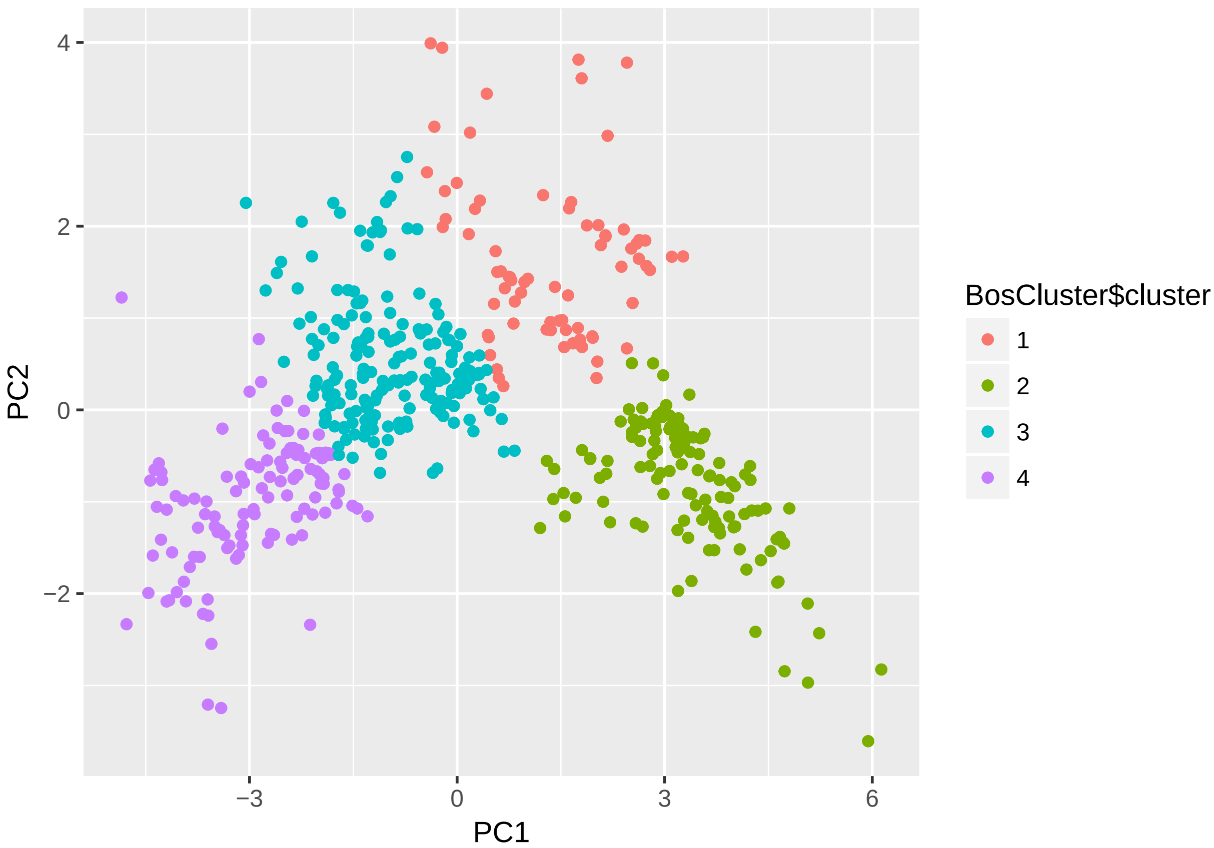
```
wssplot(pc)
```




```
# Looks like K = 4 wil be good

set.seed(20)
BosCluster <- kmeans(pc, 4, nstart = 20)

BosCluster$cluster <- as.factor(BosCluster$cluster)
ggplot(pc) +
  geom_point(mapping = aes(PC1, PC2, color = BosCluster$cluster))
```



4
2
0
−2
PC2
−3
0
3
6
PC1
BosCluster$cluster
1
2
3
4