Statistics W4240 (Section 01): Data Mining Homework 2

Yunyan Zhang (UNI: yz2861)

Due date: February 19, 2016

Problem 1

a

```
setwd("~/Learning/stat w4240 data mining/homework2/")
p1 <- read.csv("hw02_q1_p1.csv")
colMeans(p1)
```

```
##        x1        x2        x3        x4        x5
##  6.049104 -8.277221  4.665532  7.914270 62.138753
```

```
rowMeans(p1)
```

```
##   [1]  -0.1277116  20.8162864  -8.8984358  25.5999204  -9.7472153
##   [6]  64.0626702  22.0392371  23.3914888  31.7598224 -13.8680290
##  [11]  43.8318898   6.5478369  14.1665143  16.1945993  29.6357898
##  [16]  11.0316832  -2.5453007   8.6124471  33.8364419  24.9647839
##  [21]  34.8385372  34.1951748  25.8869897  -0.4545730   9.0418836
##  [26]  21.4051827   3.2291136  35.5748021  21.1031545   6.5535668
##  [31]   3.7478608  18.9230712  -9.2447158   6.3811655  16.8358750
##  [36]   7.9628124  16.6264489  16.7027735 -34.4147885   0.4138282
##  [41]  12.6572899  35.4589880  17.3456417  17.2383651   0.5124620
##  [46] -24.7073649  17.1498949  52.3665782   9.6993053   0.3079195
##  [51]  15.6758568 -13.3093667   8.2062088  34.8247664  12.1909900
##  [56]  -3.1939531  -5.4779341  10.7689107  36.2253846  19.5034554
##  [61]   8.9492321   4.4008921  14.3901288  14.7207124  27.9510161
##  [66] -14.3617846  39.3331820  24.0356530  -6.7256757  -4.2948679
##  [71]  27.1881673  47.2951022  19.1932996  23.5607379   7.6480638
##  [76]  18.1517706  16.9872267 -46.6660940   7.2223867  28.8378401
##  [81]   6.5043155  26.5206768  -2.4442159  15.3802055  16.1739005
##  [86]  26.1705488  20.1409435  63.2646829   9.1977728  29.2026018
##  [91]   1.2105932  21.2145724  -8.4896595  19.0639963  20.9767512
##  [96]   3.5962333  22.3461063   0.7145014   6.3080005  64.8829556
```

The column means of the data set tells us the average of each dimension, and row means is the average of each entry's position.
The column mean of x2 is extremely small and x5 is extremely large, while x1, x3 and x4's mean is in interval [4,8]. Howver, the row means vary a lot in interval [-47, 65].

b

```
colm<-as.matrix(colMeans(p1))
one<-matrix(c(rep(1,100)),nrow = 100,ncol = 1)
```

```
p1_cent<-as.matrix(p1)-one%*%t(colm)
cov(p1_cent)
```

```
##           x1         x2        x3         x4        x5
## x1  73.70119  -84.75614  53.77483  121.32950  574.1520
## x2 -84.75614  112.01112 -64.54111 -117.11419 -825.5948
## x3  53.77483  -64.54111  40.00284   84.58444  449.7486
## x4 121.32950 -117.11419  84.58444  234.47811  690.4634
## x5 574.15198 -825.59478 449.74860  690.46338 6352.3807
```

The diagonal values of the covariance matrix are variances of x1, x2, x3, x4, x5 respectively; the off diagonal elements are covariance between them. For example, variance of x1 is 73.70119, and covariance of x1 and x2 is -84.75614.

c

```
p1eigen<-eigen(cov(p1_cent),symmetric=T,only.values = F)
p1eigen$values
```

```
## [1] 6.623584e+03 1.887829e+02 2.058943e-01 9.874338e-04 9.468342e-05
```

```
p1eigen$vectors
```

```
##             [,1]       [,2]         [,3]        [,4]        [,5]
## [1,]  0.09009603 -0.3247102 -0.383470773  0.82286709  0.24957150
## [2,] -0.12797842  0.1364755  0.227047683 -0.11412319  0.94890526
## [3,]  0.07028767 -0.1941349  0.894987159  0.37278501 -0.13191135
## [4,]  0.11077853 -0.9008231 -0.019718518 -0.40719485  0.10024632
## [5,]  0.97892389  0.1636064  0.002946326 -0.07133967  0.09921159
```

Each column represents an eigenvector.

Because the covariance matrix is symmetric, so $\hat{\Sigma} = \hat{\Sigma}^T$

when transpose:

$\because \left(\hat{\Sigma} X_{right}\right)^T = \left(\lambda X_{right}\right)^T$

$\therefore X_{right}^T \hat{\Sigma}^T = X_{right}^T \hat{\Sigma} = \lambda X_{right}^T$

$\because X_{left}^T \hat{\Sigma}^T = \lambda X_{left}^T$

So, left eigenvectors and right eienvectors are the same.

d

loadings

```
t(p1eigen$vectors)
```

```
##             [,1]       [,2]        [,3]        [,4]         [,5]
## [1,]  0.09009603 -0.1279784  0.07028767  0.11077853  0.978923894
```

```
## [2,] -0.32471017  0.1364755 -0.19413491 -0.90082314  0.163606358
## [3,] -0.38347077  0.2270477  0.89498716 -0.01971852  0.002946326
## [4,]  0.82286709 -0.1141232  0.37278501 -0.40719485 -0.071339671
## [5,]  0.24957150  0.9489053 -0.13191135  0.10024632  0.099211592
```

scores

```
t(p1eigen$vectors)%*%t(p1_cent)
```

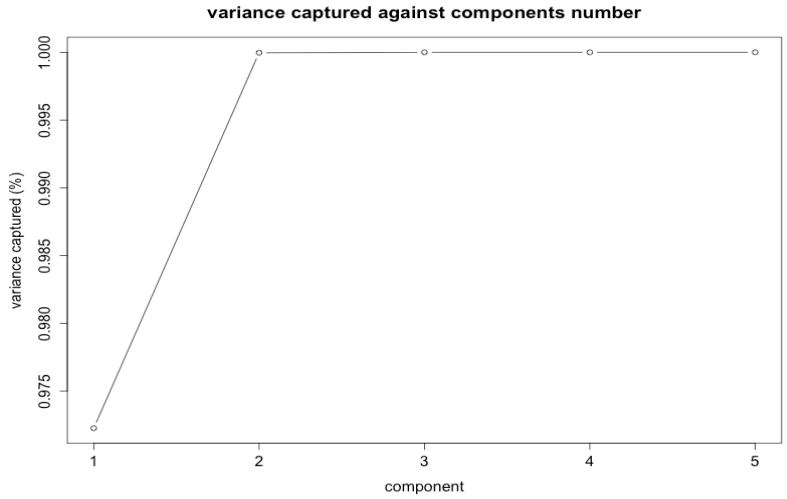
```
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,] -58.606720199  17.967889865 -1.035576e+02  38.865124468 -1.067853e+02
## [2,]   6.812884087 -10.025331361  7.721199e-01 -10.358921823  1.300995e+00
## [3,]   0.358690823   0.313590316  8.091266e-02   0.396822082  9.454513e-02
## [4,]  -0.008025112  -0.005100516  4.228485e-02   0.002165917 -2.588959e-03
## [5,]   0.006008437  -0.014540111  5.511172e-04   0.010168182 -9.527577e-03
##               [,6]         [,7]         [,8]        [,9]         [,10]
## [1,] 223.094855176 24.866274279 52.165663556 85.26822386 -134.47236742
## [2,]   2.170672761 -9.129222152 12.048094466  9.03987718   -8.36204776
## [3,]  -0.142606288 -0.579070066 -0.813698583  1.02615437   -0.46921998
## [4,]   0.023796497  0.006832407  0.023515723 -0.02781833    0.03527513
## [5,]   0.004184366 -0.003154517 -0.005105202  0.02019815    0.01364136
##              [,11]         [,12]         [,13]        [,14]        [,15]
## [1,] 112.720829981 -28.717690380 -11.231216491 16.340741459 82.515039310
## [2,] -18.191443046   6.456849462  -9.448060006  8.778505723 14.745682030
## [3,]  -0.272898296  -0.408980503   0.504892877 -0.056563844 -0.523378213
## [4,]   0.026671264  -0.022769215   0.009178745 -0.008887615 -0.012678513
## [5,]  -0.001498326   0.008701904  -0.002458606  0.016562482 -0.006137646
##             [,16]         [,17]        [,18]       [,19]       [,20]
## [1,] -8.15746164 -60.075356107 -33.62775512 85.47522162 57.82373736
## [2,]  7.50839792  15.994953894  -6.80734584 -0.23221099 10.99873188
## [3,]  0.35328135   0.099597926   0.97044908  0.61935050 -0.35618321
## [4,] -0.01762198   0.053128258  -0.02381278  0.06980573  0.01068183
## [5,] -0.01306550  -0.001670995  -0.00138528  0.02389607  0.01082133
##              [,21]         [,22]         [,23]         [,24]         [,25]
## [1,]  71.400818640 97.3463214295 45.2050456762 -64.893383707 -19.206100748
## [2,] -19.203688760  9.6712000334 -5.6010380130   1.704277727   5.150267677
## [3,]   0.129225070  0.1256020669 -0.0618845810  -0.038372645   0.035441792
## [4,]  -0.004711344 -0.0004988105 -0.0008785511  -0.011065882   0.028479667
## [5,]  -0.006144754 -0.0080843428 -0.0047918837  -0.002750028  -0.004969443
##              [,26]        [,27]         [,28]       [,29]         [,30]
## [1,]  3.086585e+00 -32.36984445  81.171977153 43.54909134 -30.483166312
## [2,] -2.820816e+01  18.10347971 -12.932667130 13.81889403   4.694738591
## [3,] -7.174322e-01   0.30828842  -0.249490262 -0.51591167  -0.344306355
## [4,]  1.345768e-02   0.01436504   0.017716245 -0.02530340  -0.038597101
## [5,]  1.950185e-04   0.01169650  -0.007872939  0.01393931   0.008489629
##              [,31]        [,32]         [,33]         [,34]        [,35]
## [1,] -68.338009765 11.958071514 -1.149034e+02 -3.383367e+01 39.579750678
## [2,] -20.141109636 -8.133667951 -9.026591e+00  1.879675e+00 29.315352358
## [3,]   0.505934409 -0.583771511  1.317556e-01 -6.786914e-01  0.153342720
## [4,]   0.008457724  0.001987747  3.009547e-03 -4.010050e-02 -0.018902226
## [5,]   0.009070093  0.016187696  1.346205e-02 -8.498259e-04 -0.001596966
```

```
##              [,36]         [,37]        [,38]         [,39]         [,40]
## [1,] -34.715693894  -7.881951968 24.273792684 -2.114039e+02 -49.417644035
## [2,]  -5.214127303 -17.020493304 14.308857250  6.106318e+00  13.085669175
## [3,]   0.587143103   0.633609597 -0.243328812 -7.063456e-01  -0.458158522
## [4,]   0.014542206  -0.047752583 -0.016326381  1.284566e-02   0.009250013
## [5,]   0.006431135   0.001386253 -0.007031015 -6.122073e-03   0.004956943
##              [,41]         [,42]        [,43]        [,44]         [,45]
## [1,] -10.458266268  67.352334231  8.489238788 18.766224840 -34.657955104
## [2,]  -2.241215521 -26.122787223 -4.474728823  6.784310384  28.245645991
## [3,]   0.005242291  -0.040371600 -0.351774095  0.315865710   0.802577657
## [4,]   0.035220981   0.038053958 -0.060864313 -0.001124995  -0.001665026
## [5,]  -0.002324704  -0.009945971 -0.005770854  0.009457018   0.005512764
##              [,46]        [,47]         [,48]         [,49]         [,50]
## [1,] -1.662082e+02  6.973371746 153.225902823 -39.280086332 -64.393623887
## [2,]  8.443082e+00 -4.915323350 -16.084773857 -18.029363266  -1.736086687
## [3,] -7.554525e-02 -0.109200644  -0.875597004  -0.085581544  -0.872933243
## [4,] -7.133580e-03  0.030742599   0.036600952  -0.053172634   0.001370069
## [5,] -9.593263e-03 -0.006858827   0.006417334  -0.007122636  -0.006878342
##              [,51]         [,52]        [,53]        [,54]       [,55]
## [1,] 36.774782659 -1.322400e+02 -10.37426231 108.53869633 13.54354790
## [2,] 31.406286101 -8.281412e+00  18.08309459  18.22143658 23.74828200
## [3,] -0.362465732  1.262685e-01   0.58007477   0.36741278 -0.13854099
## [4,]  0.063267443 -7.787320e-03  -0.01939543  -0.01018191 -0.02629835
## [5,]  0.007189871 -6.262407e-03   0.01598171  -0.01366259 -0.02266949
##              [,56]         [,57]         [,58]         [,59]         [,60]
## [1,] -74.734531819 -9.727834e+01 -13.421009001 116.515505620  11.912883275
## [2,]   3.731237653 -7.849776e+00   3.582758236  19.480626997 -10.487680169
## [3,]  -0.634494672  6.440079e-01   0.572948080  -0.368736800  -0.096072326
## [4,]   0.042916823  3.873931e-02   0.009272099  -0.043272000  -0.026501843
## [5,]   0.007525288 -2.810957e-05   0.004739217  -0.003138307   0.002182429
##              [,61]         [,62]        [,63]        [,64]        [,65]
## [1,] -38.15482746 -57.413979232 17.311617868 20.110532227 52.548347993
## [2,] -13.30893051 -12.229716200 17.806942367 19.454906590 -7.009717664
## [3,]   0.24187911   0.335149128 -0.036974680  0.436467976  0.624706845
## [4,]   0.01931550   0.024769966 -0.007545028  0.007805388 -0.009709644
## [5,]  -0.01357139  -0.008195619  0.001858829  0.006826586  0.005291829
##              [,66]        [,67]       [,68]         [,69]         [,70]
## [1,] -130.00686110 117.87750141 33.39342491 -1.101919e+02 -87.032439114
## [2,]   -1.36944075   7.46095121 -9.51237860 -1.583332e+01  -3.396640709
## [3,]    0.12271074   0.36517609 -0.55890175 -3.472238e-01  -0.164166665
## [4,]   -0.03739024   0.01725999 -0.02001877  6.854874e-02   0.029645545
## [5,]   -0.01334408  -0.01310343 -0.01393228  8.558465e-03  -0.005704212
##              [,71]        [,72]        [,73]        [,74]        [,75]
## [1,] 56.395779591 141.99942898   7.83983876 50.501842475 -16.82110677
## [2,]  0.343981229  -4.03603820 -13.45841489 10.169069501  13.60535085
## [3,]  0.772928302   0.12638110  -0.54360534  0.079054258  -0.16077282
## [4,] -0.017540455   0.03826470   0.03243906 -0.003458852  -0.01421665
## [5,]  0.004787099   0.01061099  -0.01274139 -0.019808674  -0.01425207
##              [,76]       [,77]         [,78]         [,79]       [,80]
## [1,] 19.558194266  5.10855801 -2.554885e+02 -25.436336775 65.87405685
```

```
## [2,]  3.096787614 -5.46602700  1.695277e+01   7.073935402  2.13289671
## [3,] -0.265003717  0.87393153 -2.552934e-01   0.137229884  0.20906089
## [4,] -0.025990456 -0.04667224  3.918307e-02  -0.004794063  0.01460751
## [5,] -0.003091874 -0.00710823  4.271328e-03  -0.009853021  0.00729218
##              [,81]        [,82]         [,83]         [,84]        [,85]
## [1,] -8.2861889105 50.449683954 -79.916539896  2.2032651947  -6.86042961
## [2,] 27.2816011304 -3.655877601  -4.184588063 -1.4092054021 -13.85032788
## [3,] -0.2236549610 -0.749463083   0.391364012  0.4844368587   0.77919710
## [4,]  0.0593172462 -0.075464838  -0.016847943  0.0174420200   0.06027191
## [5,] -0.0008629083 -0.002876452   0.004830459  0.0003017361  -0.02226949
##              [,86]       [,87]        [,88]        [,89]        [,90]
## [1,] 61.573828241 15.84738578 207.44702257 -34.17983284 58.908385082
## [2,]  9.326404291 -9.93716488  -9.86630413 -11.02418124 -6.815011556
## [3,] -0.448358927 -0.94612191   0.03329542  -0.67677574 -0.254493544
## [4,]  0.062115518 -0.05535051  -0.01664496  -0.03625568 -0.036271938
## [5,] -0.001800167  0.01782893  -0.00402701   0.01556885 -0.002694781
##              [,91]         [,92]        [,93]        [,94]        [,95]
## [1,] -97.025648430   1.282808144 -90.86775035  -5.11784569 34.525457124
## [2,] -37.733025112 -28.629050021  11.40376380 -25.30365175  5.845261948
## [3,]   0.280164304   0.112472322  -0.26573699   0.34864472  0.291766067
## [4,]  -0.021836503   0.010380016  -0.06535013  -0.05256495 -0.015630703
## [5,]   0.001045843   0.002244887   0.01846498   0.01796436 -0.003836123
##              [,96]        [,97]         [,98]         [,99]        [,100]
## [1,] -55.963040700 50.272385171 -47.475844160 -3.505450e+01 209.221179942
## [2,]  -7.335032370 15.492360634  14.188379928  1.160244e+00 -15.365537728
## [3,]   0.121237565  0.263135172   0.376152574 -4.210416e-01  -0.100374179
## [4,]  -0.011366780 -0.022343252  -0.031156534 -2.206544e-02   0.032325456
## [5,]  -0.004398222 -0.001862266  -0.009890383  5.471521e-04   0.004388429
```

e

```
plot(seq(1:5),cumsum(p1eigen$values)/sum(p1eigen$values),xlab="component",yla
b="variance captured (%)",main="variance captured against components number",
type="l")
```



We should include first 2 components as they contain almost 100% of original information.

f

```
p12 <- read.csv("hw02_q1_p2.csv")
p12 <- data.matrix(p12)
colm2<-as.matrix(colMeans(p12))
one2<-matrix(c(rep(1,5)),nrow = 5,ncol = 1)
p12_cent<-as.matrix(p12)-one2%*%t(colm)
p12scores<-t(as.matrix(p1eigen$vectors))%*% t(p12_cent)
p12scores
```

```
##             [,1]          [,2]        [,3]         [,4]          [,5]
## [1,] -70.371540299 27.4689251724  2.22357733 -67.08355664 -18.414179477
## [2,] -12.098356149  6.9698728181 -2.55656904   9.45171522   6.968267804
## [3,]   0.292497049  0.2336019632  0.15223063   0.43951496   0.726859012
## [4,]  -0.062071749 -0.0191897354 -0.06059532   0.05093323  -0.059568367
## [5,]   0.001295112 -0.0007608147  0.00376563  -0.01665144   0.007572305
```

g

```
p12proj<-t(p12scores[1:2,])%*%t(as.matrix(p1eigen$vectors)[,1:2])
          +one2%*%t(colm)
p12proj
```

```
##           x1          x2        x3         x4        x5
## [1,]  3.637367  -0.9223123  2.067993 11.0170934 -8.728997
## [2,]  6.260757 -10.8414338  5.243163  4.6786141 90.169056
## [3,]  7.079584  -8.9107004  5.318141 10.4636109 63.897195
## [4,] -3.063926   1.5979545 -1.884523 -8.0314715 -1.984582
## [5,]  2.127392  -4.9696055  2.018458 -0.4028028 45.252726
```

Above is the coordinates of the projections in the original space, x'.

```
sqrt(rowSums((p12-p12proj)^2))
```

```
## [1] 0.2990136 0.2343901 0.1638906 0.4427695 0.7293351
```

Above is the Euclidean distance from the original data points.

h

```
p12-p12proj
```

```
##             x1         x2        x3           x4           x5
## [1,] -0.1629176 0.07472354 0.2384708  0.019637518  0.005418460
## [2,] -0.1055600 0.05450684 0.2020175  0.003131408  0.001981775
## [3,] -0.1072981 0.04505217 0.1131587  0.022049832  0.005144966
## [4,] -0.1307856 0.07817755 0.4145439 -0.031075577 -0.003990621
## [5,] -0.3258562 0.17901519 0.6273244  0.010682446  0.007142412
```

The errors come from 3 directions that are not chosen. The 3 directions correspond to the 3 eigenvalues smaller than the first two.

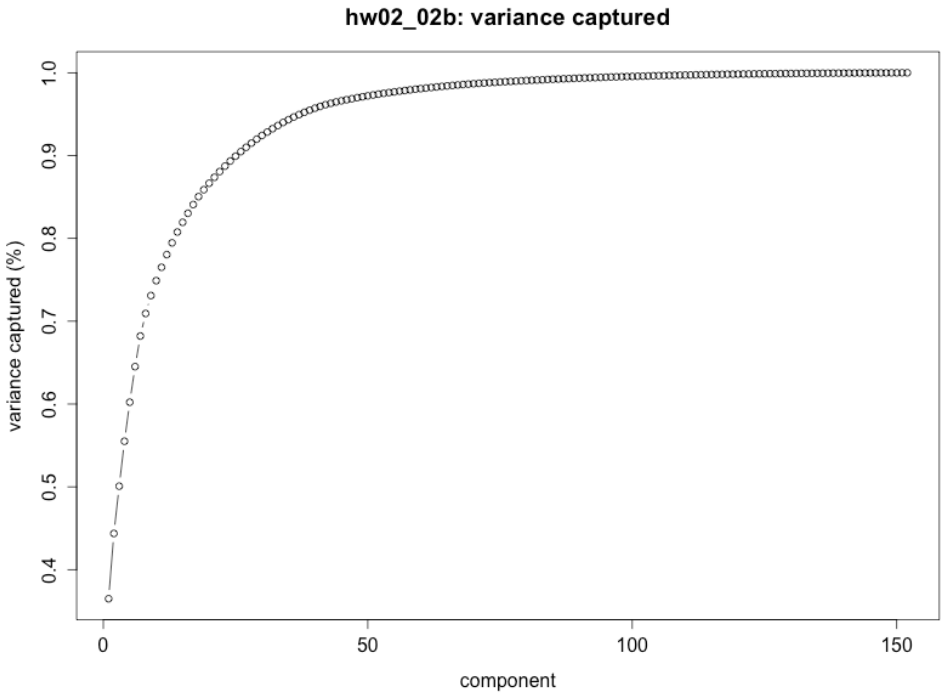
Problem 2

the list of pictures (note the absence of 14 means that 31 corresponds to yaleB32)

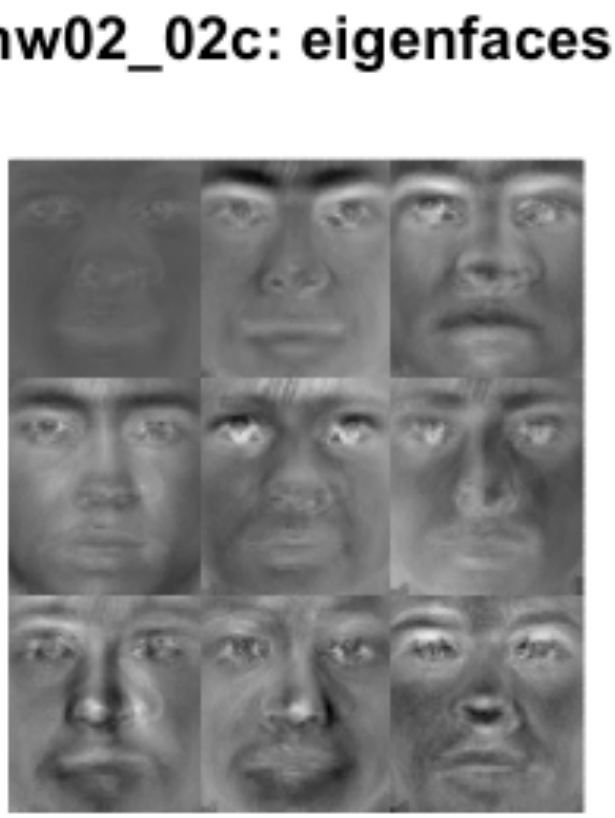
a

hw02_02a: mean face

b

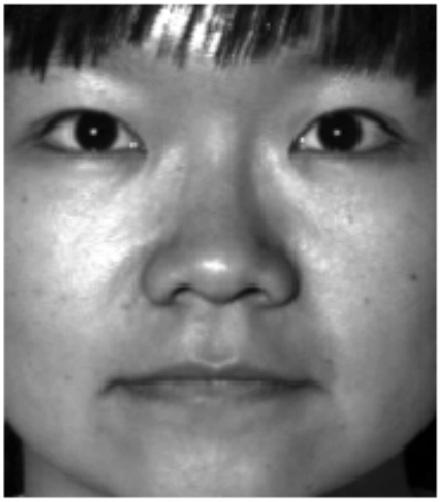


c



The 9 eigenfaces shown above describe the 9 most common features share by the subjects.

d



Above is the original picture of “yaleB05_P00A+010E+00.pgm. The index for “yaleB05_P00A+010E+00.pgm” is 20. Then I can choose the 20th column from both pc's x and rotation and sum up their product to reconstruct the picture.

From the pictures shown below, I think 18 pictures are enough to recognize the person.

hw02_02d: one face per time

hw02_02d: five faces per time

e

```
## The first four rows of original pixmapGrey matrix is the data for subject#
# 01.

## [1] 1 2 3 4

## After excluding 4 rows, the left matrix has a dimension of 148 x 32256.

## [1]   148 32256

## Following are the picture names related to subject 01.

## [1] "CroppedYale/yaleB01/yaleB01_P00A+000E+00.pgm"
## [2] "CroppedYale/yaleB01/yaleB01_P00A+005E+10.pgm"
## [3] "CroppedYale/yaleB01/yaleB01_P00A+005E-10.pgm"
## [4] "CroppedYale/yaleB01/yaleB01_P00A+010E+00.pgm"
```

hw02_02e: reconstruct face

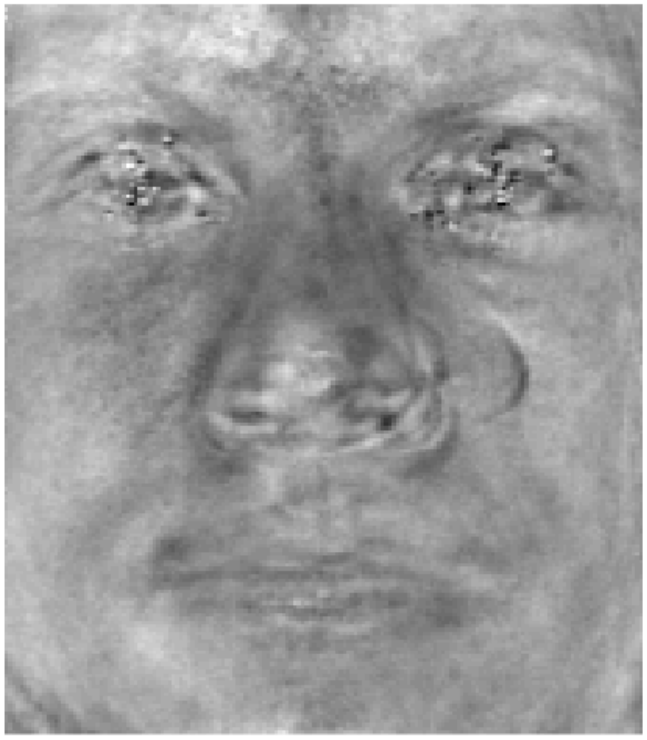
From the whole picture level, it looks like the original image, as a human face; however, some features are lost (such as his eyes and lips) because the data of original image is subtracted before PCA.

f

```
## Pixmap image
##   Type          : pixmapGrey
##   Size          : 480x640
##   Resolution    : 1x1
##   Bounding box  : 0 0 640 480
```

Because the original data has a size of 480 x 640, so I use its 192 x 168 subset (which is the picture below on the right) and get its channels before PCA.

hw02_02f: reconstruct picture



I use the principal components from previous PCA to reconstruct the picture. However, the picture does not look like the original image. Because the picture I want to reconstruct is not included in the PC, which only allows me to reduce dimensions instead of building something new. Also, the side face and front face have different features.