MLHW5_PCA.R

rohan

Tue Apr 17 01:00:32 2018

```
mnist<-read.csv("/Users/rohan/Desktop/DS630_MachineLearning/HW5/mnist.csv",sep = ",
", header = F)
# Removing the first variable
mnist <-mnist[,-1]
# Cheacking for NA's and cleaning
mnist_clean<-na.omit(mnist)
```

```
#Employ either the princomp or prcomp function to perform PCA on the 784 columns of
the data set which contain the grey scale pixel information.
#Do not include the first column as this column merely indicates the digit 0-9 that
a given row encodes.
pca_mnist_model<-prcomp(mnist_clean)
#View(head(pca_mnist_model))
#View(head(pca_mnist_model$x))
#View(head(pca_mnist_model$rotation))
#View(head(pca_mnist_model$center,50))
#View(head(pca_mnist_model$scale))
#summary(pca_mnist_model)
#names(pca_mnist_model)
```

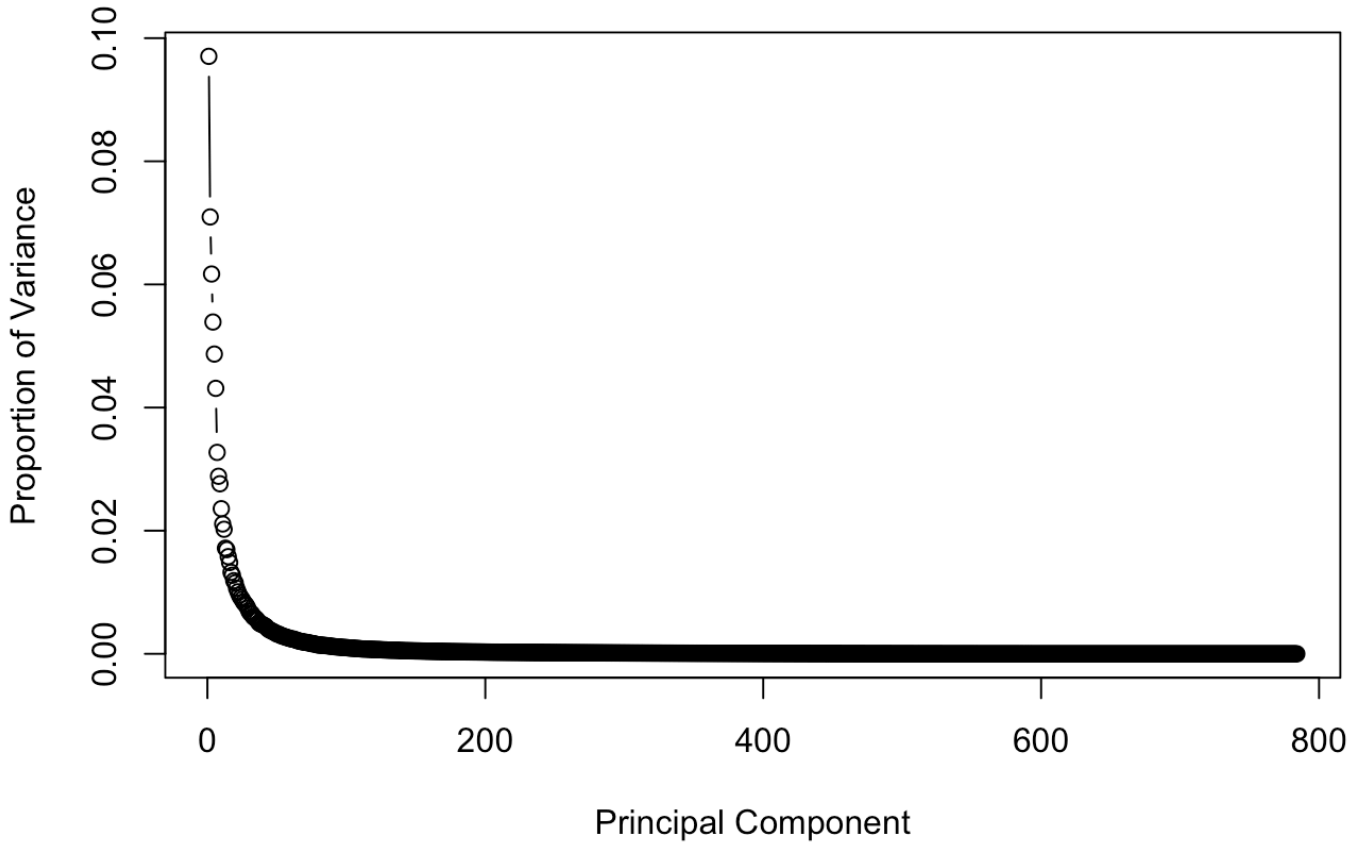
```
#compute standard deviation of each principal component
std_dev <- pca_mnist_model$sdev
#View(head(std_dev))
#compute variance of each principal component
variance <- std_dev^2
#View(head(variance))
#checking variance of first 10 components for drawing some inferences
variance[1:10]
```

```
##  [1] 332724.67 243283.94 211507.37 184776.39 166926.83 147844.96 112178.20
##  [8]  98874.43  94696.25  80809.82
```

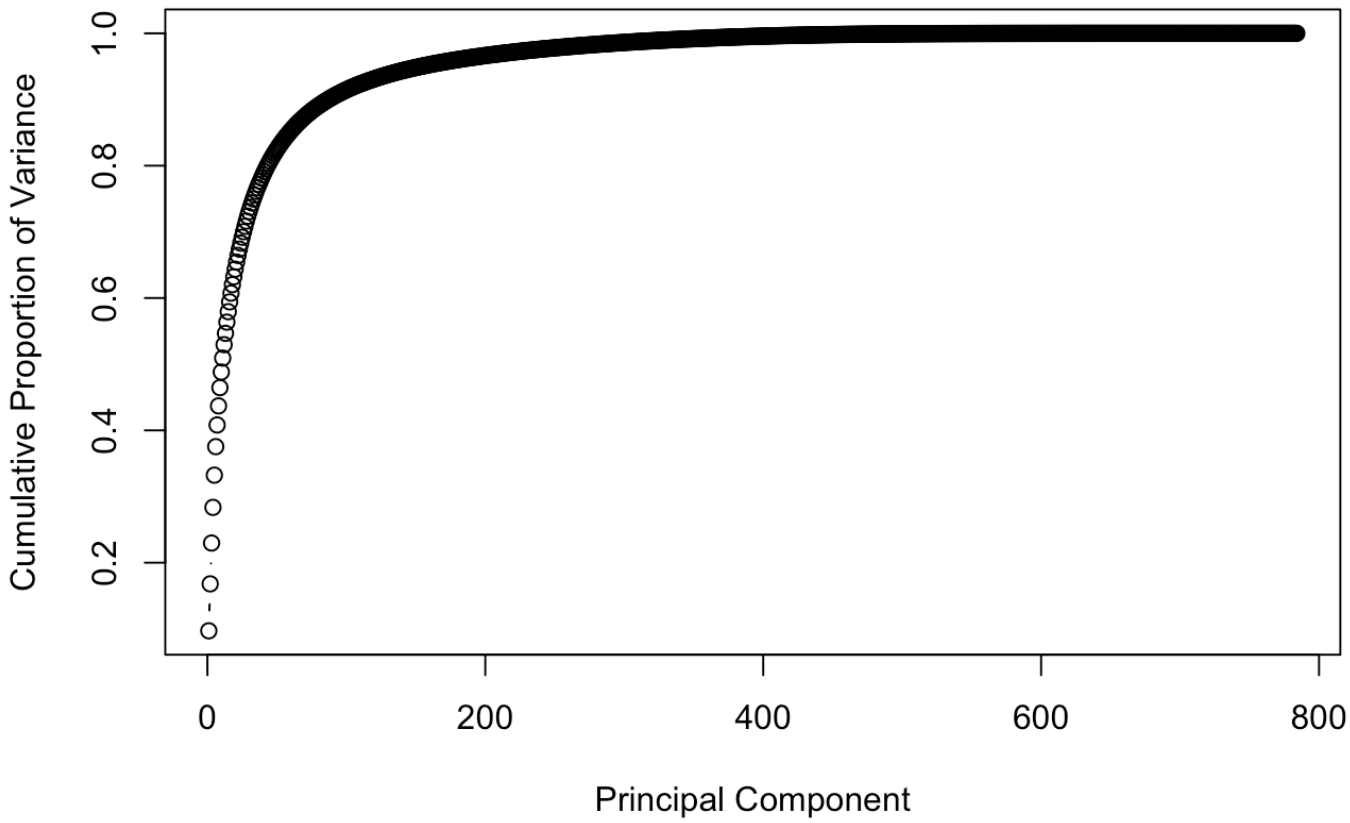
```
#proportion of variance
proportion_variance <- variance/sum(variance)
proportion_variance[1:20]
```

```
##  [1] 0.09704664 0.07095924 0.06169089 0.05389419 0.04868797 0.04312231
##  [7] 0.03271930 0.02883895 0.02762029 0.02357001 0.02109190 0.02022991
## [13] 0.01715818 0.01692111 0.01578641 0.01482953 0.01324561 0.01276897
## [19] 0.01187263 0.01152684
```

```
#scree plot
plot(proportion_variance, xlab = "Principal Component", ylab = "Proportion of Varia
nce", type = "b")
```




```
#cumulative scree plot
plot(cumsum(proportion_variance), xlab = "Principal Component", ylab = "Cumulative
Proportion of Variance", type = "b")
```




```
# Report the number of principal components needed to account for 98% of the varian
ce of the original data set.
# Taking inferences from the graph and then selecting the Principal components
sum(proportion_variance[1:261])
```

```
## [1] 0.9801387
```

```
# 261 Principal components account for 0.9801387 of the variance of the original da
ta set
```

```
# For each of the principal components reported in 2. supply the percentage of the
variance that each component contributes to the overall variance.
cumsum(proportion_variance[1:261])
```

```
##   [1] 0.09704664 0.16800588 0.22969677 0.28359097 0.33227894 0.37540125
##   [7] 0.40812055 0.43695950 0.46457980 0.48814980 0.50924170 0.52947161
##  [13] 0.54662979 0.56355091 0.57933732 0.59416685 0.60741246 0.62018143
##  [19] 0.63205406 0.64358090 0.65424256 0.66430969 0.67384542 0.68297086
##  [25] 0.69180491 0.70019810 0.70832389 0.71618755 0.72363488 0.73054347
##  [31] 0.73712441 0.74360589 0.74963204 0.75549786 0.76119807 0.76663435
##  [37] 0.77169222 0.77657080 0.78138510 0.78610776 0.79067523 0.79512359
##  [43] 0.79930860 0.80329076 0.80714050 0.81089154 0.81451162 0.81802754
##  [49] 0.82142812 0.82464686 0.82783703 0.83096508 0.83392491 0.83681446
##  [55] 0.83965576 0.84237012 0.84506533 0.84765006 0.85018776 0.85263557
##  [61] 0.85504063 0.85743326 0.85973735 0.86195267 0.86408988 0.86616213
##  [67] 0.86819256 0.87016038 0.87208891 0.87397523 0.87584500 0.87765583
##  [73] 0.87943146 0.88118044 0.88283802 0.88447696 0.88609158 0.88764274
##  [79] 0.88911887 0.89055063 0.89197158 0.89338311 0.89478485 0.89614222
##  [85] 0.89748069 0.89880465 0.90010622 0.90136495 0.90259324 0.90380908
##  [91] 0.90497942 0.90612816 0.90726060 0.90836946 0.90945948 0.91052871
##  [97] 0.91157067 0.91261074 0.91362330 0.91462857 0.91561259 0.91656228
## [103] 0.91750362 0.91841978 0.91932763 0.92022451 0.92108990 0.92194507
## [109] 0.92279069 0.92361319 0.92440477 0.92519071 0.92597531 0.92674415
## [115] 0.92750816 0.92826125 0.92899803 0.92972517 0.93044482 0.93115163
## [121] 0.93184706 0.93253922 0.93322252 0.93389658 0.93456346 0.93520872
## [127] 0.93584432 0.93647597 0.93709890 0.93770419 0.93830778 0.93890226
## [133] 0.93949057 0.94007709 0.94065843 0.94123526 0.94180063 0.94235539
## [139] 0.94289056 0.94341649 0.94394158 0.94445184 0.94495481 0.94545589
## [145] 0.94595460 0.94644567 0.94693121 0.94741405 0.94788807 0.94835642
## [151] 0.94882302 0.94928635 0.94974564 0.95019602 0.95064483 0.95108754
## [157] 0.95152421 0.95195166 0.95237664 0.95279865 0.95321416 0.95362380
## [163] 0.95402463 0.95442251 0.95481659 0.95520708 0.95559293 0.95597271
## [169] 0.95635161 0.95672555 0.95709496 0.95746011 0.95782393 0.95818380
## [175] 0.95853845 0.95889134 0.95923863 0.95958459 0.95992604 0.96026368
## [181] 0.96060078 0.96093100 0.96125936 0.96158526 0.96190821 0.96222948
## [187] 0.96254678 0.96286295 0.96317390 0.96348343 0.96378940 0.96409267
## [193] 0.96439508 0.96469665 0.96499336 0.96528911 0.96558258 0.96587525
## [199] 0.96616300 0.96644735 0.96672931 0.96700838 0.96728412 0.96755659
## [205] 0.96782603 0.96809374 0.96835886 0.96862318 0.96888507 0.96914452
## [211] 0.96940307 0.96966000 0.96991473 0.97016755 0.97041984 0.97066940
## [217] 0.97091792 0.97116399 0.97140794 0.97164981 0.97189078 0.97213122
## [223] 0.97236949 0.97260634 0.97283892 0.97307023 0.97330036 0.97352845
## [229] 0.97375603 0.97397880 0.97420067 0.97442111 0.97463898 0.97485563
## [235] 0.97507080 0.97528499 0.97549785 0.97570822 0.97591797 0.97612494
## [241] 0.97632966 0.97653393 0.97673525 0.97693623 0.97713523 0.97733286
## [247] 0.97752868 0.97772241 0.97791527 0.97810704 0.97829859 0.97848766
## [253] 0.97867568 0.97886207 0.97904792 0.97923253 0.97941623 0.97959938
## [259] 0.97978126 0.97996112 0.98013868
```

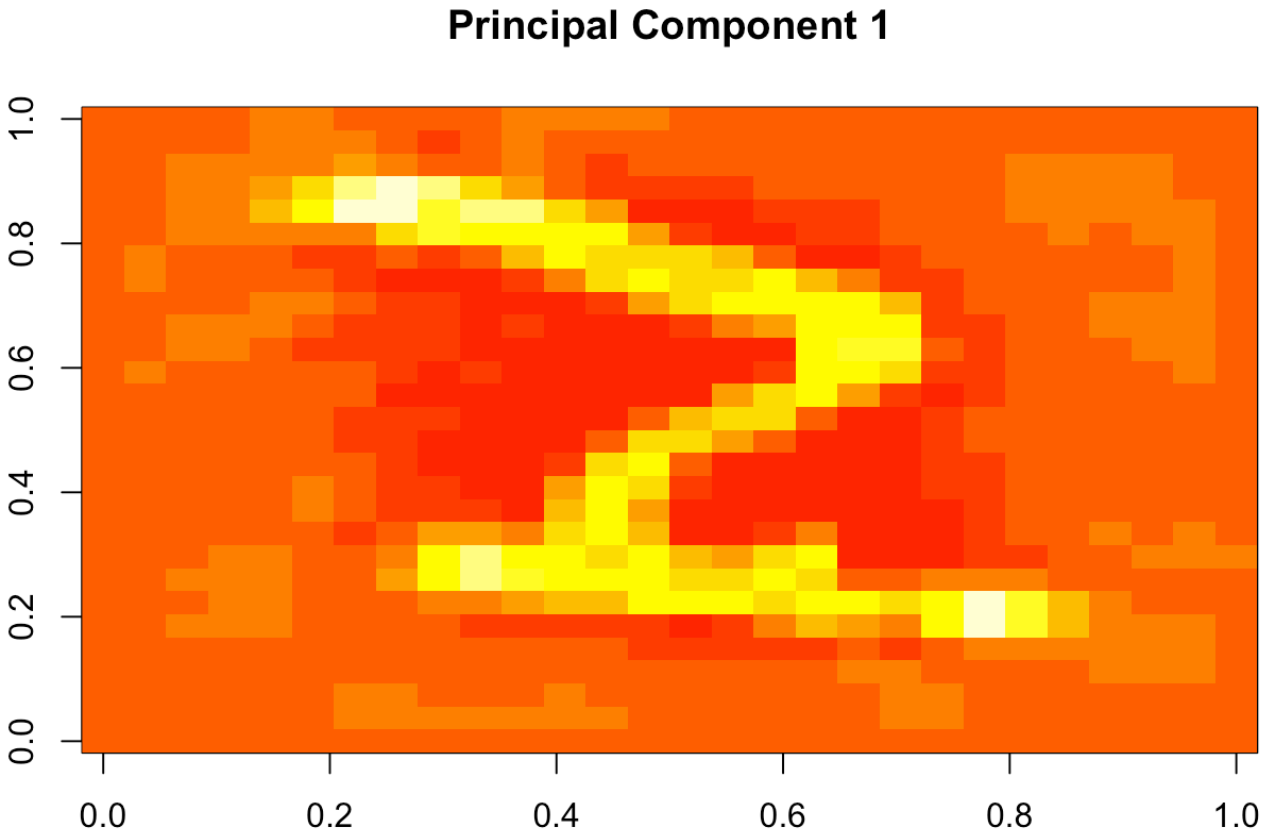
```
#Are there pixels in the original data set that consistently appear with significan
t loadings in the principal components reported in 2.?

significant_loadings1<-pca_mnist_model$rotation[,(1:261)]
#significant_loadings1<-abs(significant_loadings1)
#View(head(significant_loadings1))
significant_loadings2<-rowSums(significant_loadings1!=0)
#View(significant_loadings2)
significant_loadings<-(significant_loadings2[rev(order(significant_loadings2))])
significant_loadings
```

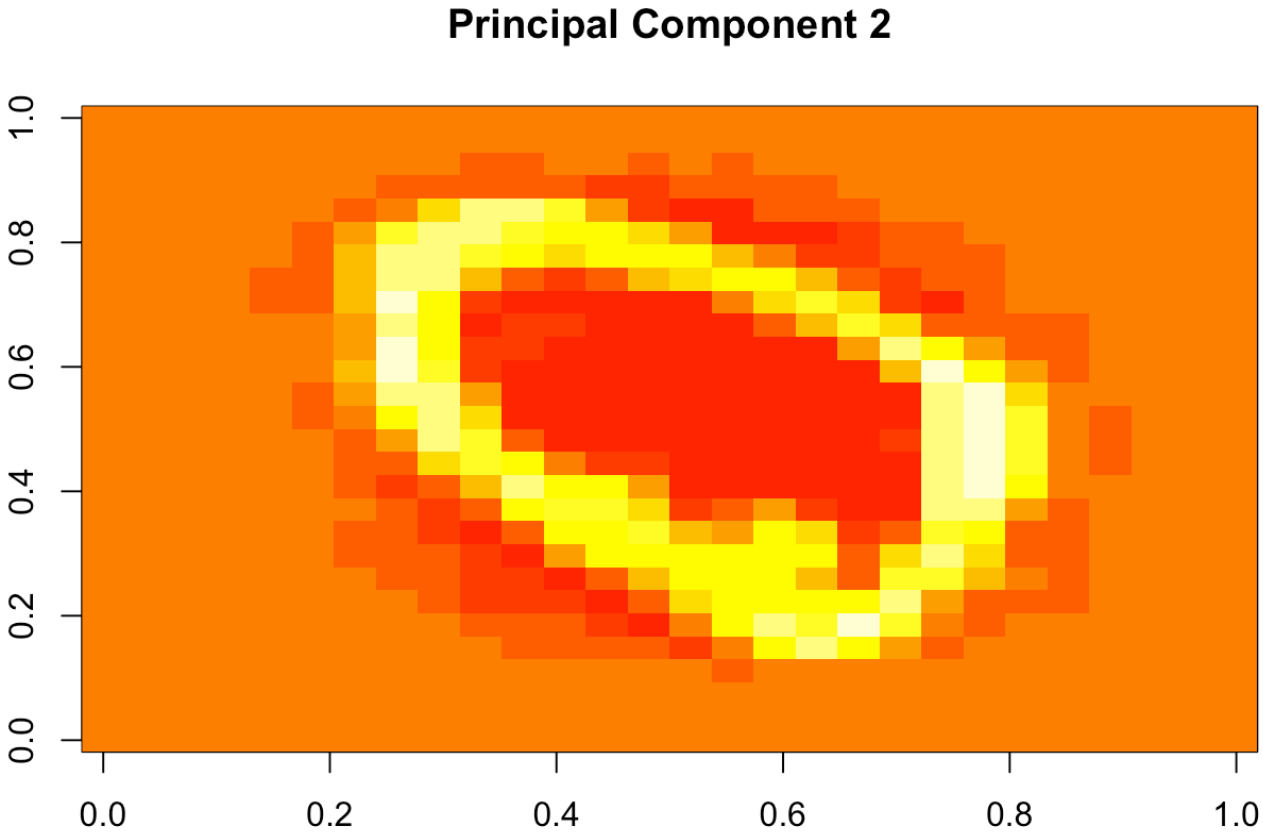
```
## V781 V780 V779 V778 V777 V776 V775 V774 V773 V772 V771 V770 V769 V768 V767
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V766 V765 V764 V763 V762 V755 V754 V753 V752 V751 V750 V749 V748 V747 V746
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V745 V744 V743 V742 V741 V740 V739 V738 V737 V736 V735 V734 V733 V728 V727
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V726 V725 V724 V723 V722 V721 V720 V719 V718 V717 V716 V715 V714 V713 V712
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V711 V710 V709 V708 V707 V706 V705 V704 V700 V699 V698 V697 V696 V695 V694
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V693 V692 V691 V690 V689 V688 V687 V686 V685 V684 V683 V682 V681 V680 V679
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V678 V677 V676 V672 V671 V670 V669 V668 V667 V666 V665 V664 V663 V662 V661
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V660 V659 V658 V657 V656 V655 V654 V653 V652 V651 V650 V649 V648 V645 V644
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V643 V642 V641 V640 V639 V638 V637 V636 V635 V634 V633 V632 V631 V630 V629
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V628 V627 V626 V625 V624 V623 V622 V621 V620 V619 V618 V617 V616 V615 V614
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V613 V612 V611 V610 V609 V608 V607 V606 V605 V604 V603 V602 V601 V600 V599
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V598 V597 V596 V595 V594 V593 V592 V591 V590 V589 V588 V587 V586 V585 V584
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V583 V582 V581 V580 V579 V578 V577 V576 V575 V574 V573 V572 V571 V570 V569
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V568 V567 V566 V565 V564 V563 V561 V560 V559 V558 V557 V556 V555 V554 V553
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V552 V551 V550 V549 V548 V547 V546 V545 V544 V543 V542 V541 V540 V539 V538
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V537 V536 V535 V534 V533 V532 V531 V530 V529 V528 V527 V526 V525 V524 V523
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V522 V521 V520 V519 V518 V517 V516 V515 V514 V513 V512 V511 V510 V509 V508
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V507 V506 V505 V504 V503 V502 V501 V500 V499 V498 V497 V496 V495 V494 V493
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V492 V491 V490 V489 V488 V487 V486 V485 V484 V483 V482 V481 V480 V479 V477
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V476 V475 V474 V473 V472 V471 V470 V469 V468 V467 V466 V465 V464 V463 V462
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V461 V460 V459 V458 V457 V456 V455 V454 V453 V452 V451 V450 V449 V448 V447
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
```

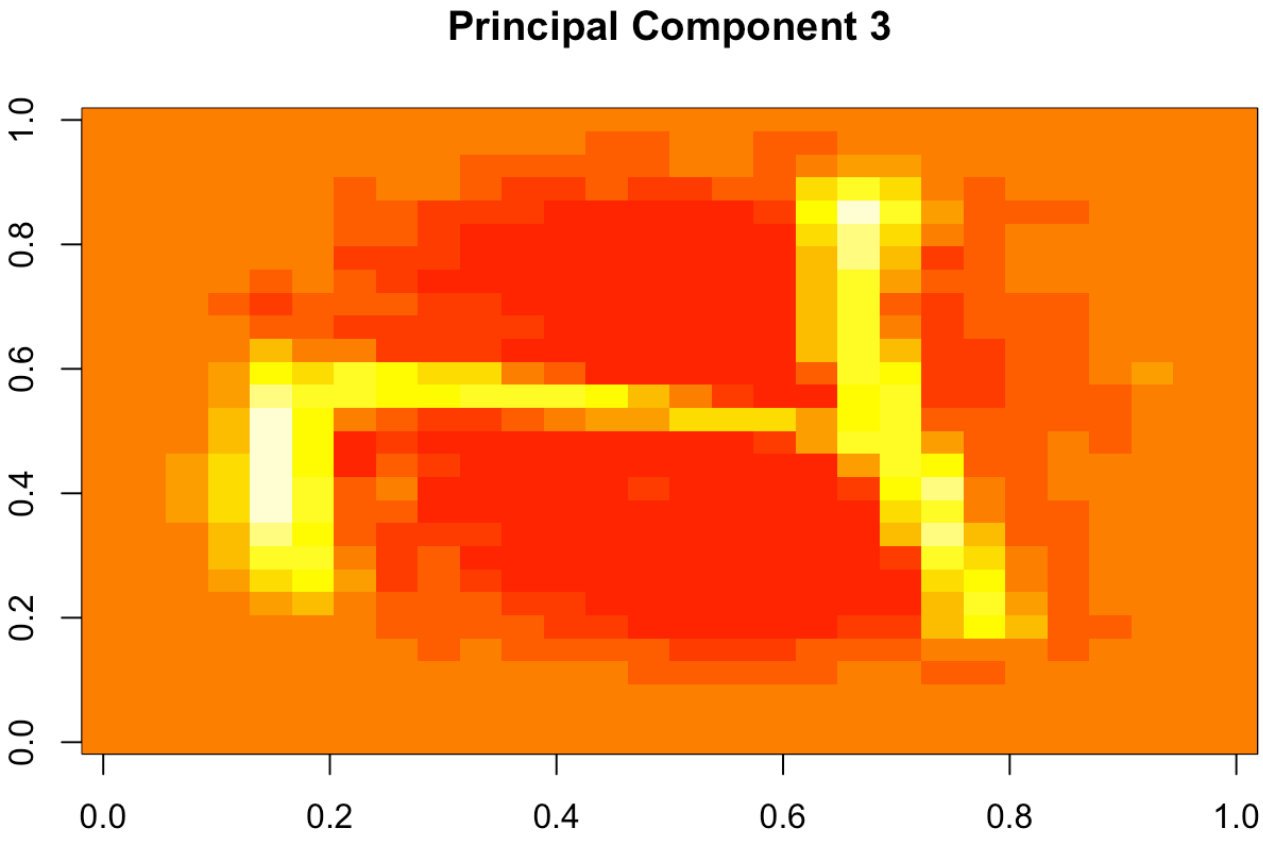
```
## V446 V445 V444 V443 V442 V441 V440 V439 V438 V437 V436 V435 V434 V433 V432
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V431 V430 V429 V428 V427 V426 V425 V424 V423 V422 V421 V420 V419 V418 V417
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V416 V415 V414 V413 V412 V411 V410 V409 V408 V407 V406 V405 V404 V403 V402
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V401 V400 V399 V398 V397 V396 V395 V394 V393 V392 V391 V390 V389 V388 V387
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V386 V385 V384 V383 V382 V381 V380 V379 V378 V377 V376 V375 V374 V373 V372
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V371 V370 V369 V368 V367 V366 V365 V364 V363 V362 V361 V360 V359 V358 V357
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V356 V355 V354 V353 V352 V351 V350 V349 V348 V347 V346 V345 V344 V343 V342
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V341 V340 V339 V338 V337 V336 V335 V334 V333 V332 V331 V330 V329 V328 V327
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V326 V325 V324 V323 V322 V321 V320 V319 V318 V317 V316 V315 V314 V313 V312
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V311 V310 V309 V308 V307 V306 V305 V304 V303 V302 V301 V300 V299 V298 V297
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V296 V295 V294 V293 V292 V291 V290 V289 V288 V287 V286 V285 V284 V283 V282
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V281 V280 V279 V278 V277 V276 V275 V274 V273 V272 V271 V270 V269 V268 V267
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V266 V265 V264 V263 V262 V261 V260 V259 V258 V257 V256 V255 V254 V253 V252
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V251 V250 V249 V248 V247 V246 V245 V244 V243 V242 V241 V240 V239 V238 V237
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V236 V235 V234 V233 V232 V231 V230 V229 V228 V227 V226 V225 V224 V223 V222
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V221 V220 V219 V218 V217 V216 V215 V214 V213 V212 V211 V210 V209 V208 V207
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V206 V205 V204 V203 V202 V201 V200 V199 V198 V197 V196 V195 V194 V193 V192
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V191 V190 V189 V188 V187 V186 V185 V184 V183 V182 V181 V180 V179 V178 V177
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V176 V175 V174 V173 V172 V171 V169 V168 V167 V166 V165 V164 V163 V162 V161
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V160 V159 V158 V157 V156 V155 V154 V153 V152 V151 V150 V149 V148 V147 V146
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V145 V144 V141 V140 V139 V138 V137 V136 V135 V134 V133 V132 V131 V130 V129
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V128 V127 V126 V125 V124 V123 V122 V121 V120 V119 V118 V117 V116 V115 V112
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
## V111 V110 V109 V108 V107 V106 V105 V104 V103 V102 V101 V100  V99  V98  V97
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
##  V96  V95  V94  V93  V92  V91  V90  V89  V88  V83  V82  V81  V80  V79  V78
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
##  V77  V76  V75  V74  V73  V72  V71  V70  V69  V68  V67  V66  V65  V64  V63
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
##  V62  V61  V60  V53  V52  V51  V50  V49  V48  V47  V46  V45  V44  V43  V42
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  261
##  V41  V40  V39  V38  V37  V36  V35  V34  V17  V16  V15  V14   V7   V2   V6
##  261  261  261  261  261  261  261  261  261  261  261  261  261  261  260
```

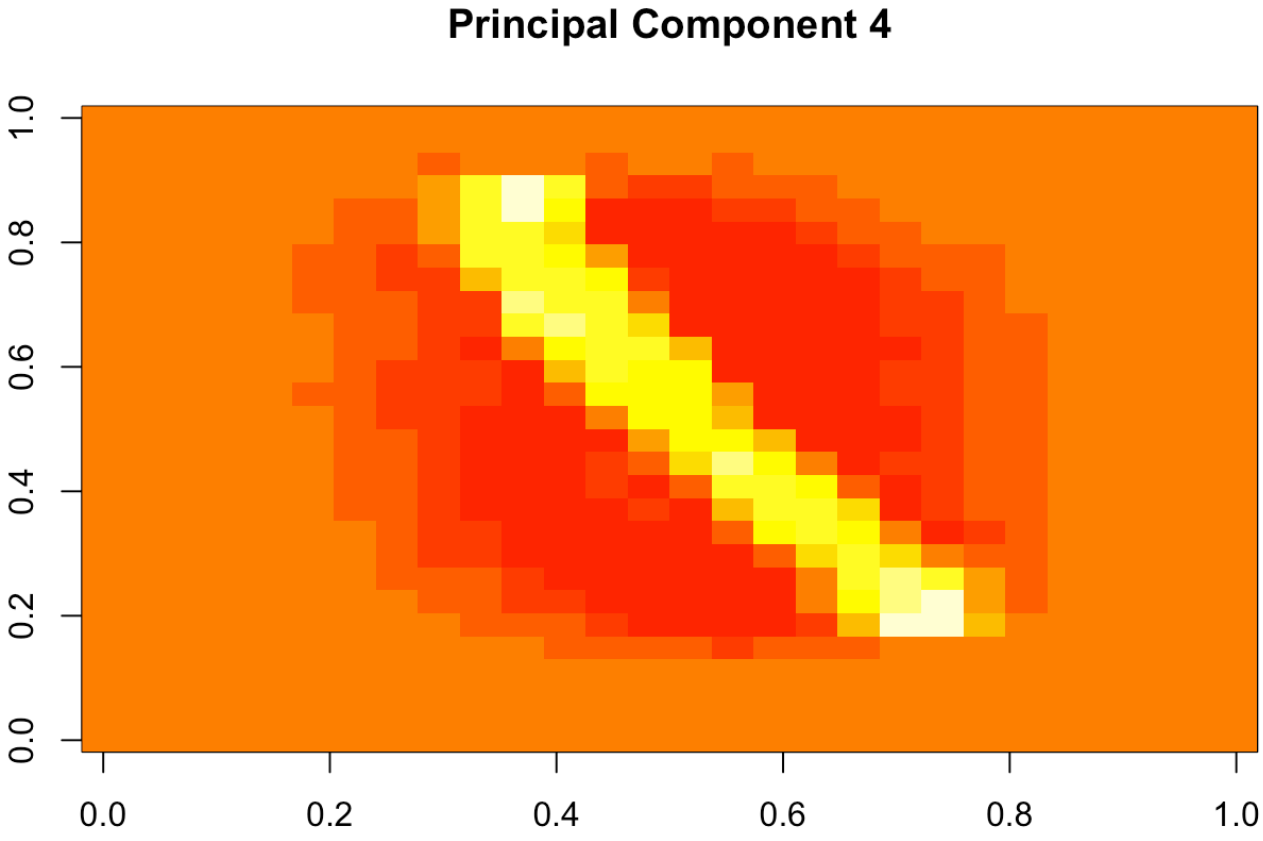
```
##   V4  V32  V33  V29  V28  V27  V26  V24  V13  V11   V8   V5  V31  V30  V23
##  260  259  258  258  258  258  258  258  258  258  258  258  257  257  257
##  V22  V21   V9  V25  V12  V10  V20  V19   V3  V18  V57  V54  V59  V56  V55
##  257  257  257  256  256  256  255  254  254  253  249  248  247  247  247
##  V87  V85  V58  V86  V84 V113 V143 V170 V114 V142 V731 V703 V702 V701 V675
##  245  244  243  241  240  224  222  220  219  218   59   59   59   59   59
## V674 V673 V647 V646 V562 V478 V732 V729 V730 V785 V784 V783 V782 V761 V760
##   59   59   59   59   59   59   53   52   50    0    0    0    0    0    0
## V759 V758 V757 V756
##    0    0    0    0
```

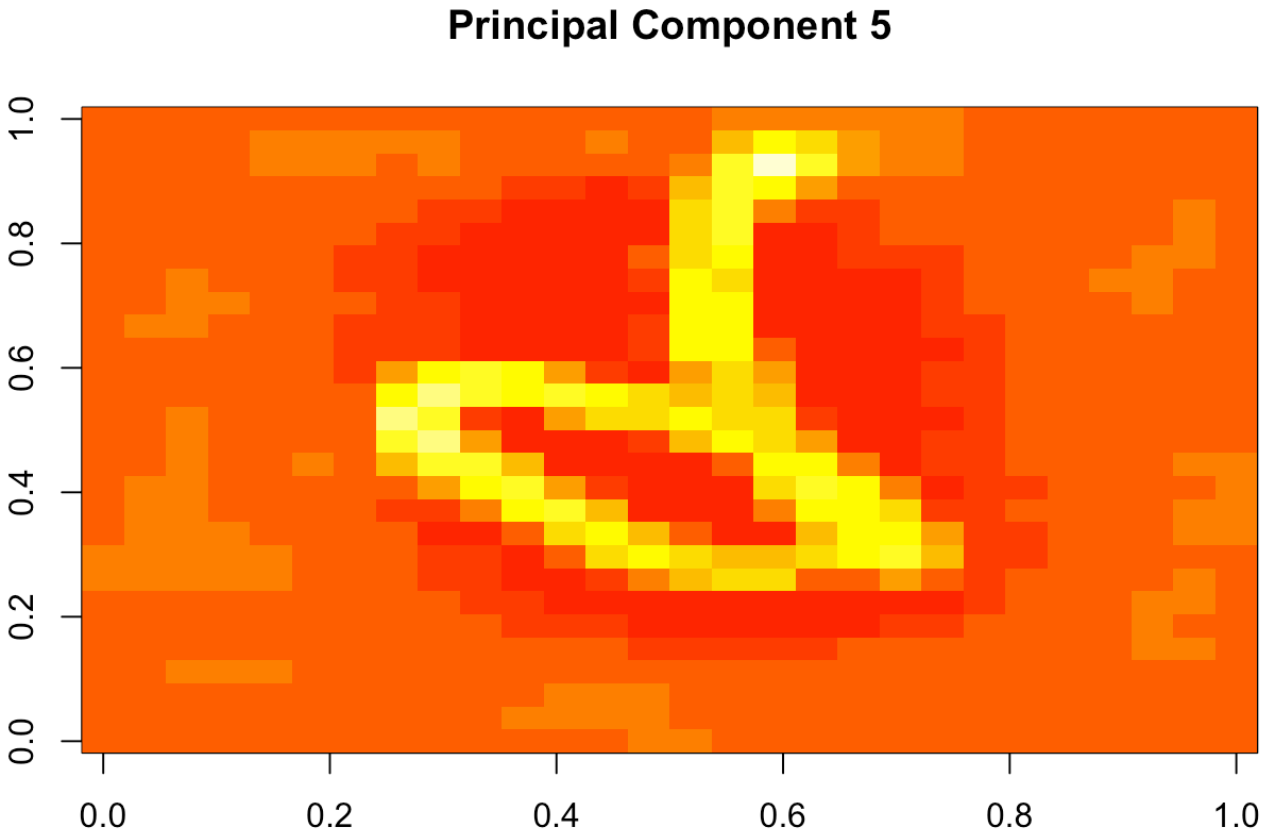
```
#Graph the first 10 principal components in a 28 × 28 pixel grid.
Bonus<-pca_mnist_model$x[,1:261]%*%t(pca_mnist_model$rotation[,1:261])
#View(head(Bonus))
for (i in 1:10)
{
  principal_components <- as.matrix(Bonus[i,])
  image(matrix(unlist(principal_components), ncol = 28, nrow = 28))
  title(paste("Principal Component", i))
}
```

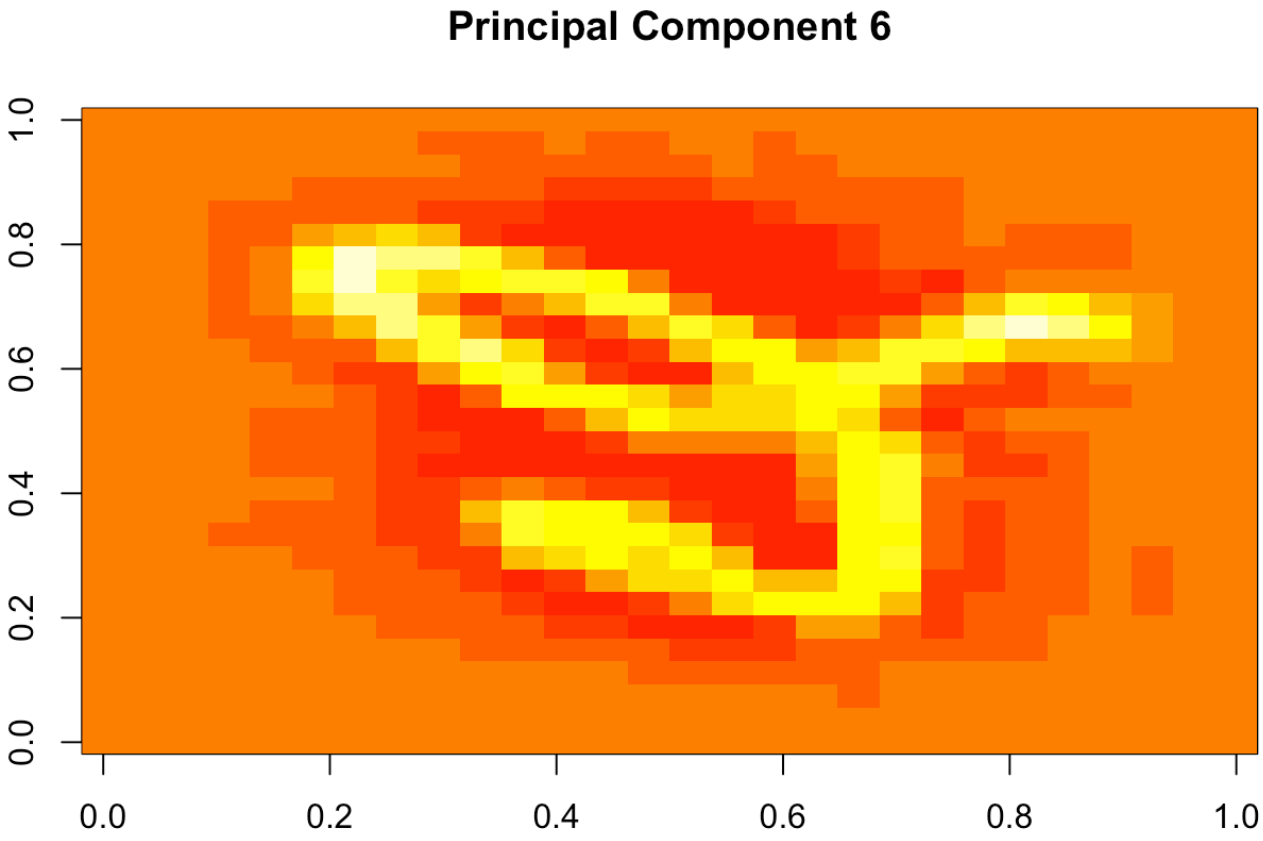

Principal Component 1
0.0
0.2
0.4
0.6
0.8
1.0

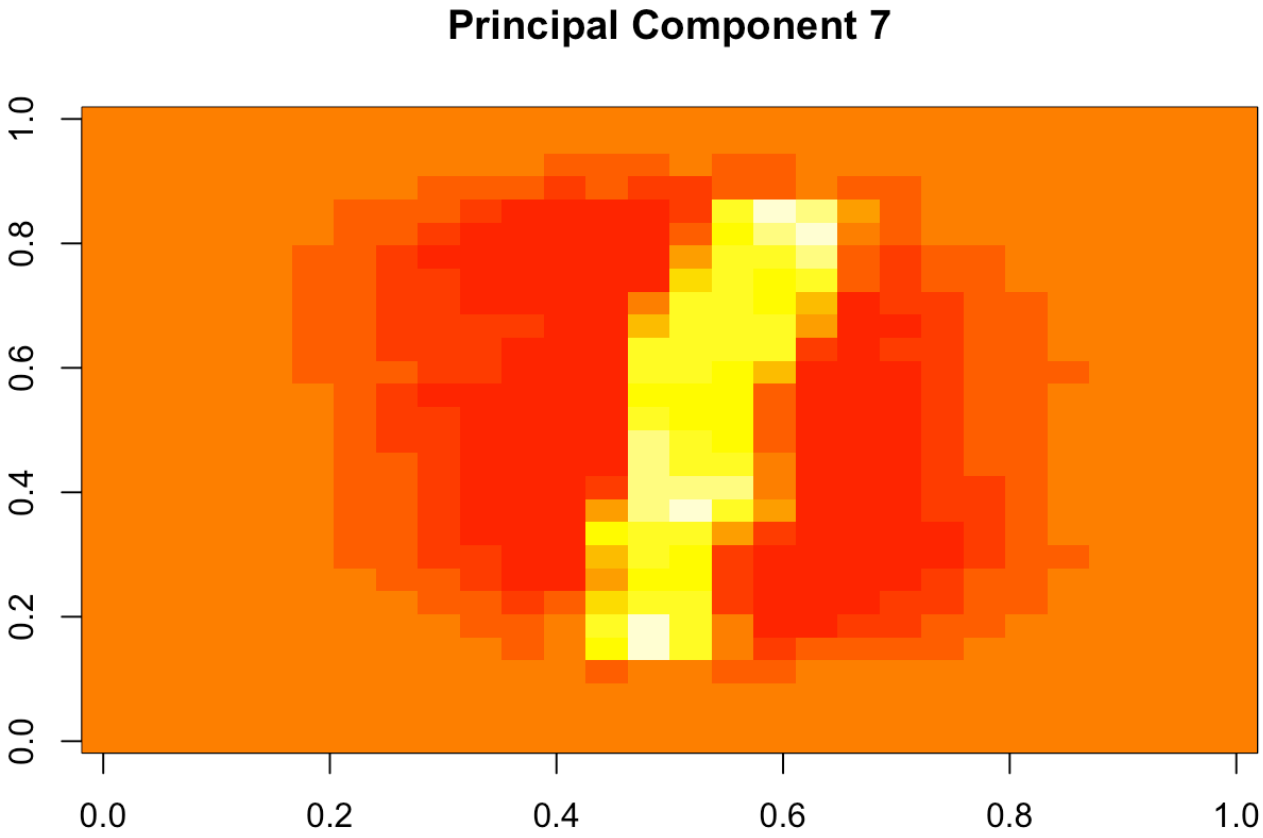

Principal Component 2
0.0
0.2
0.4
0.6
0.8
1.0

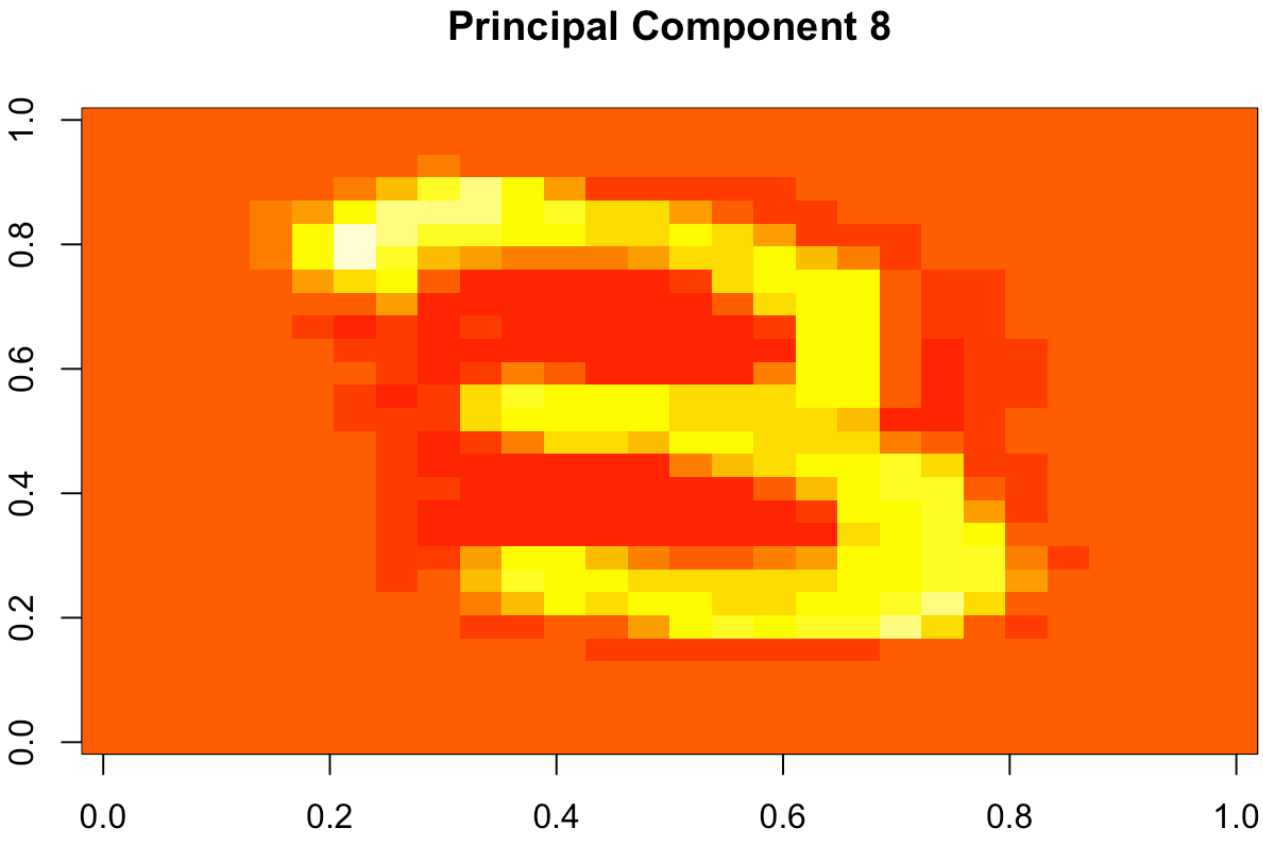

Principal Component 3
0.0
0.2
0.4
0.6
0.8
1.0

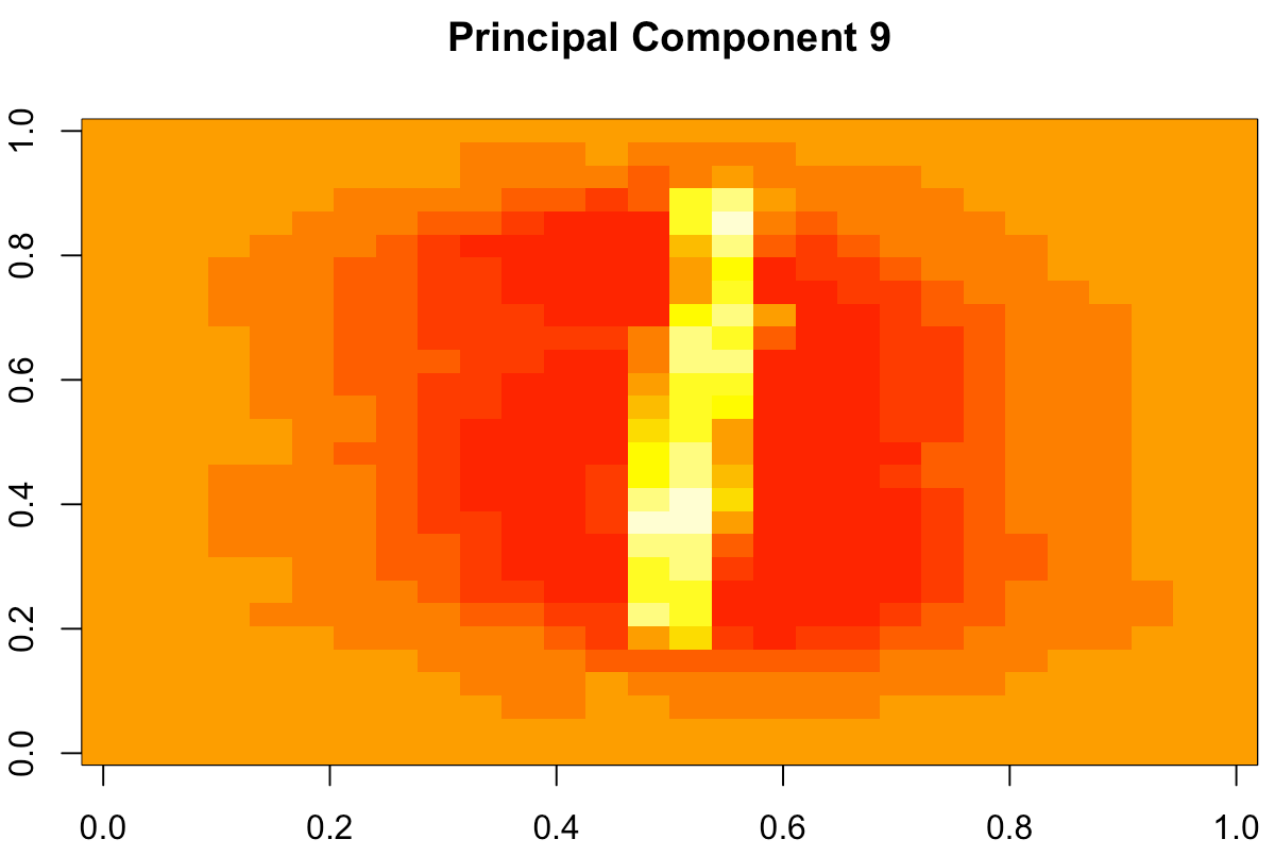

Principal Component 4
0.0
0.2
0.4
0.6
0.8
1.0

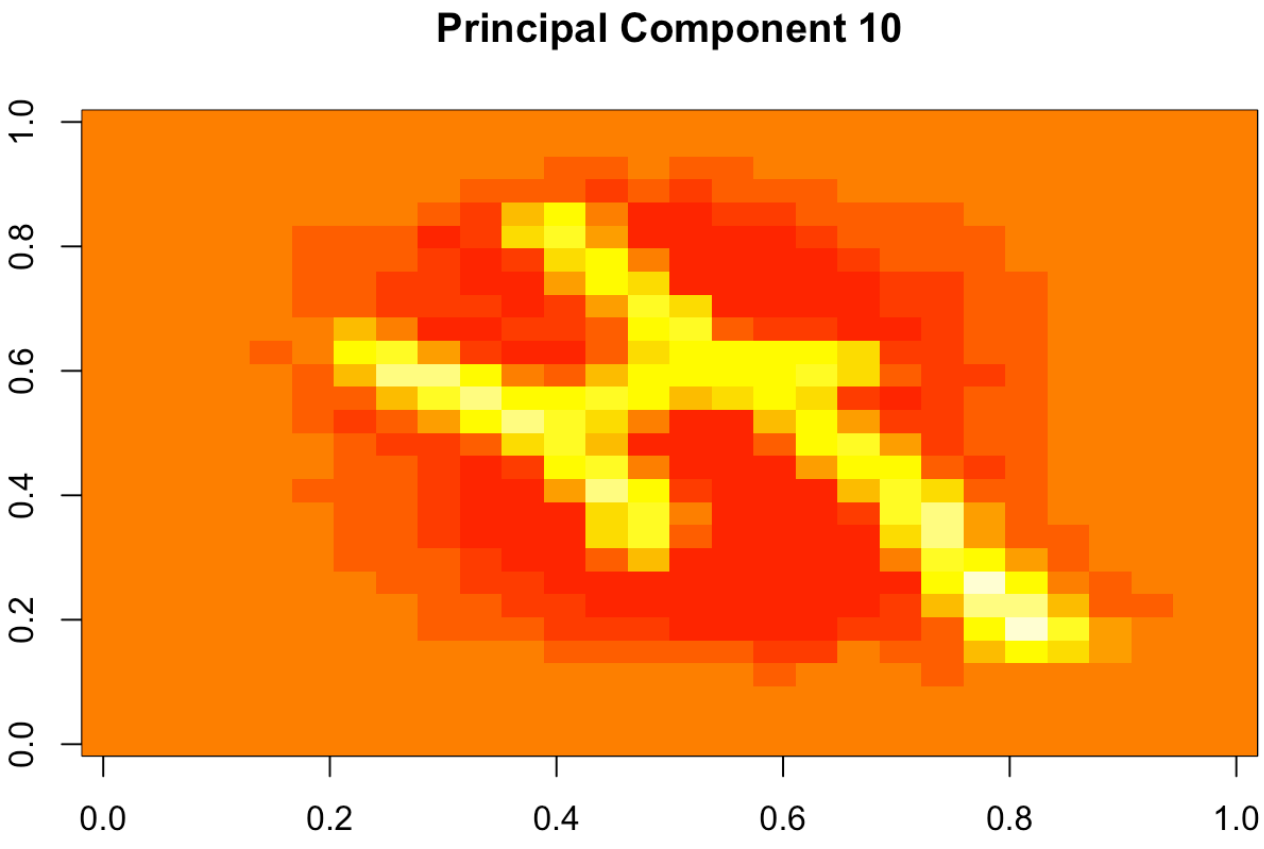

Principal Component 5
0.0
0.2
0.4
0.6
0.8
1.0


Principal Component 6
0.0
0.2
0.4
0.6
0.8
1.0


Principal Component 7
0.0
0.2
0.4
0.6
0.8
1.0


Principal Component 8
0.0
0.2
0.4
0.6
0.8
1.0


Principal Component 9
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Principal Component 10
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0