Lab 9: Letter Recognition

Paul Nguyen

11/20/2019

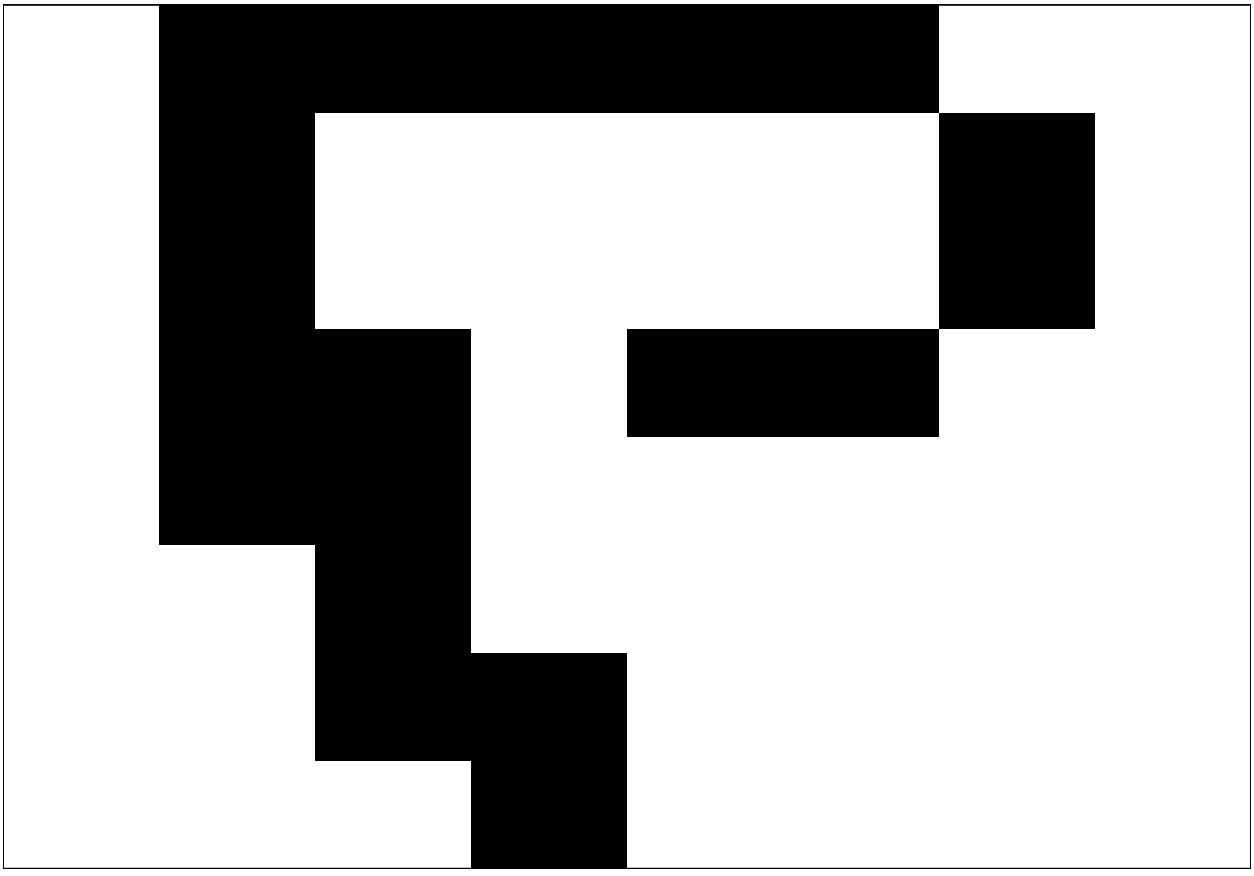
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

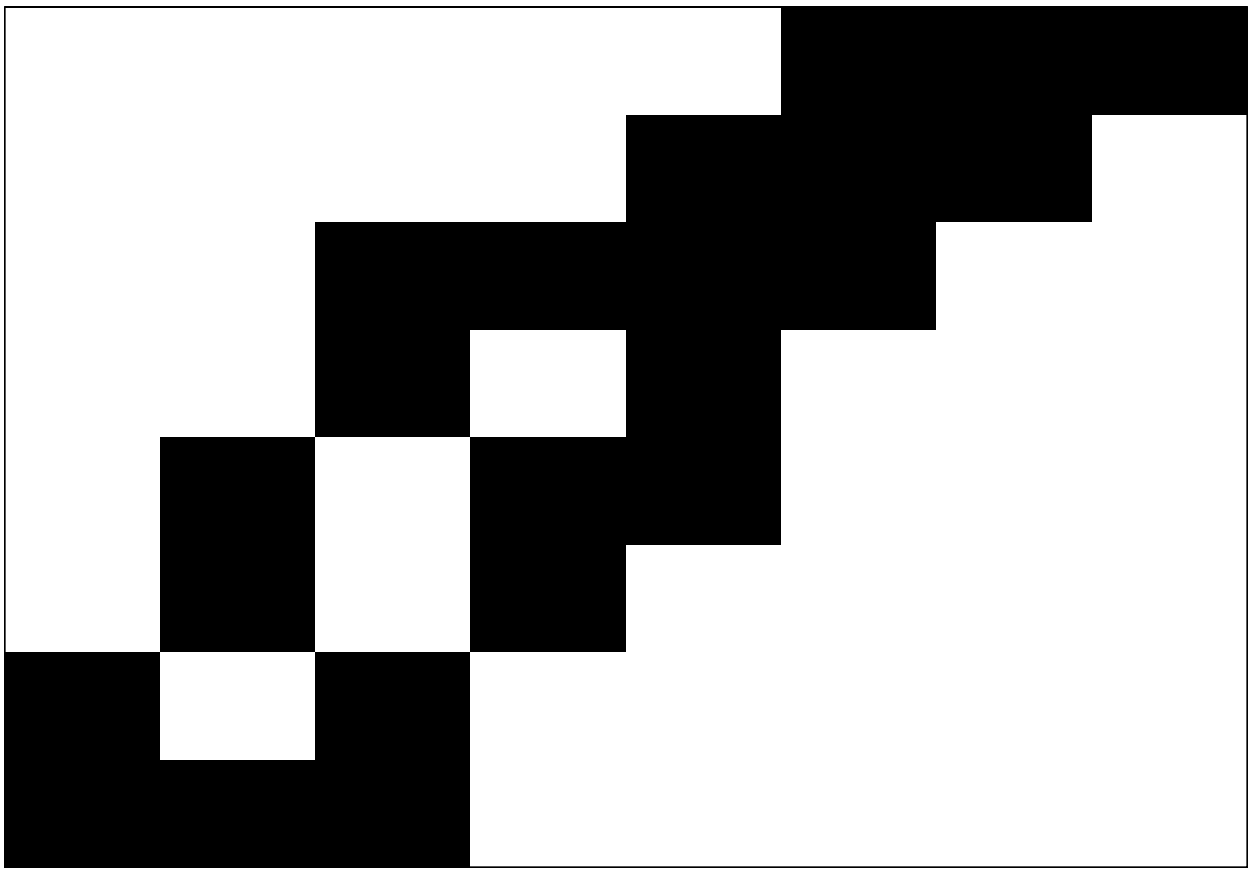
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

each column, besides the first, represents a pixel or box, which is represented with a 1 if the box is filled in in the actual picture. Each row represents a different version of the letter.

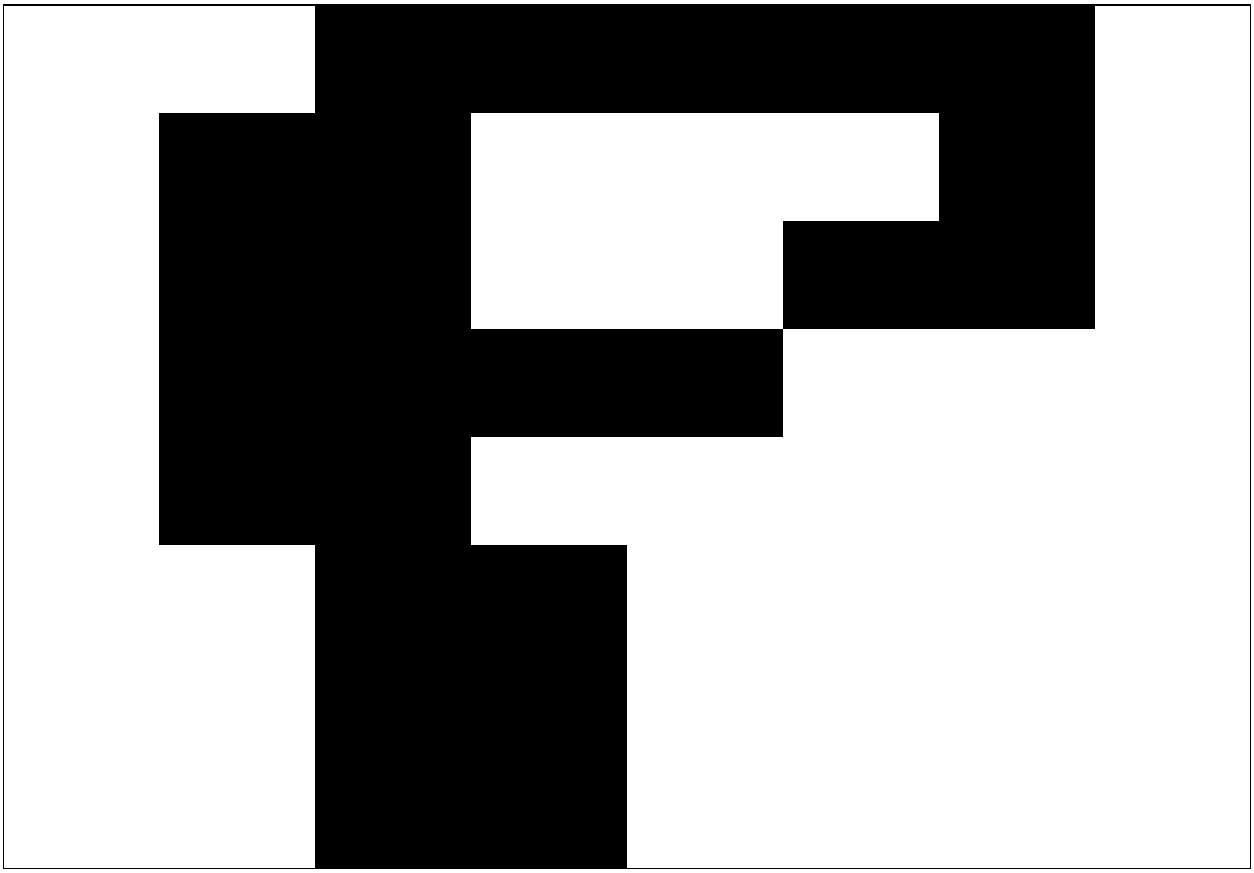
```
letterp <- d %>%
  filter(letter == "p")

plot_letter(letterp[2,])
```



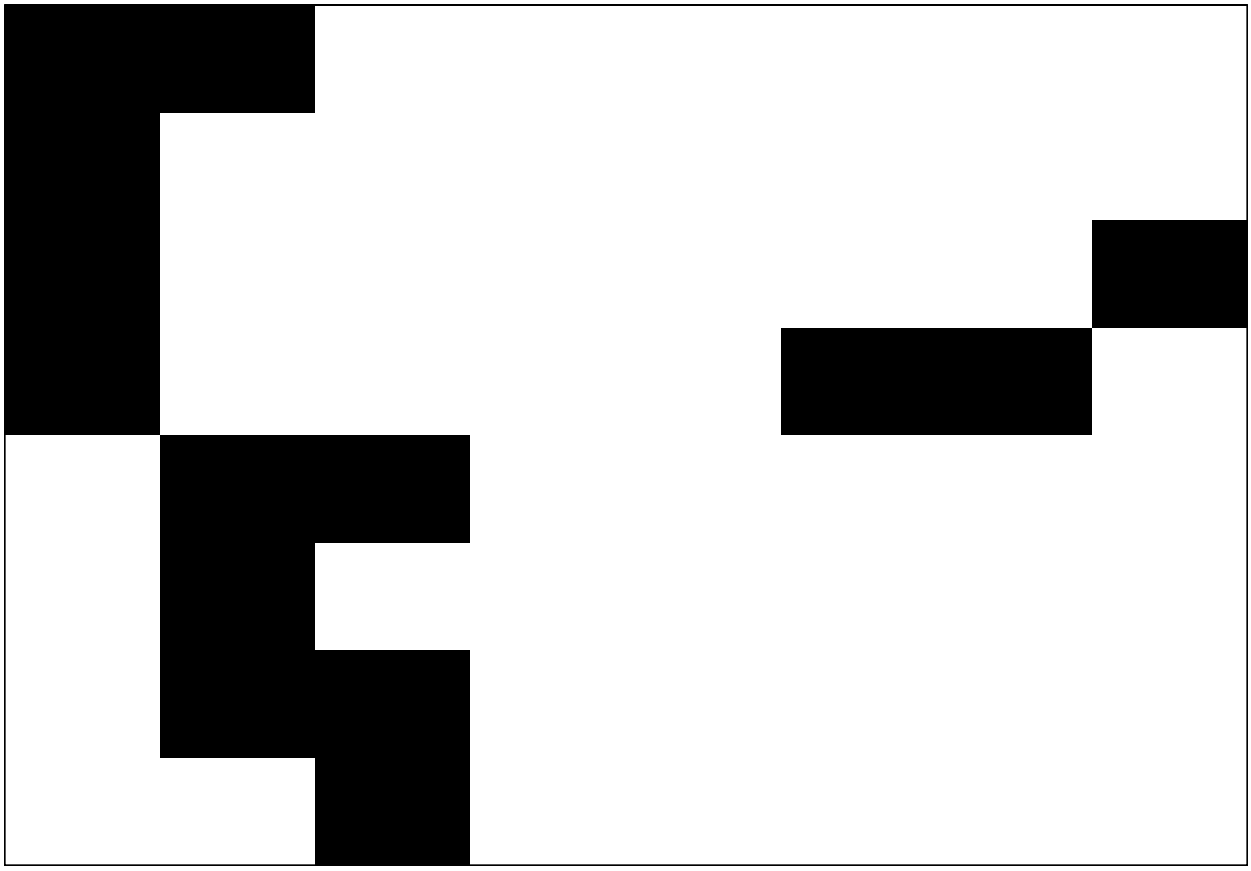
```
plot_letter(letterp[7,])
```



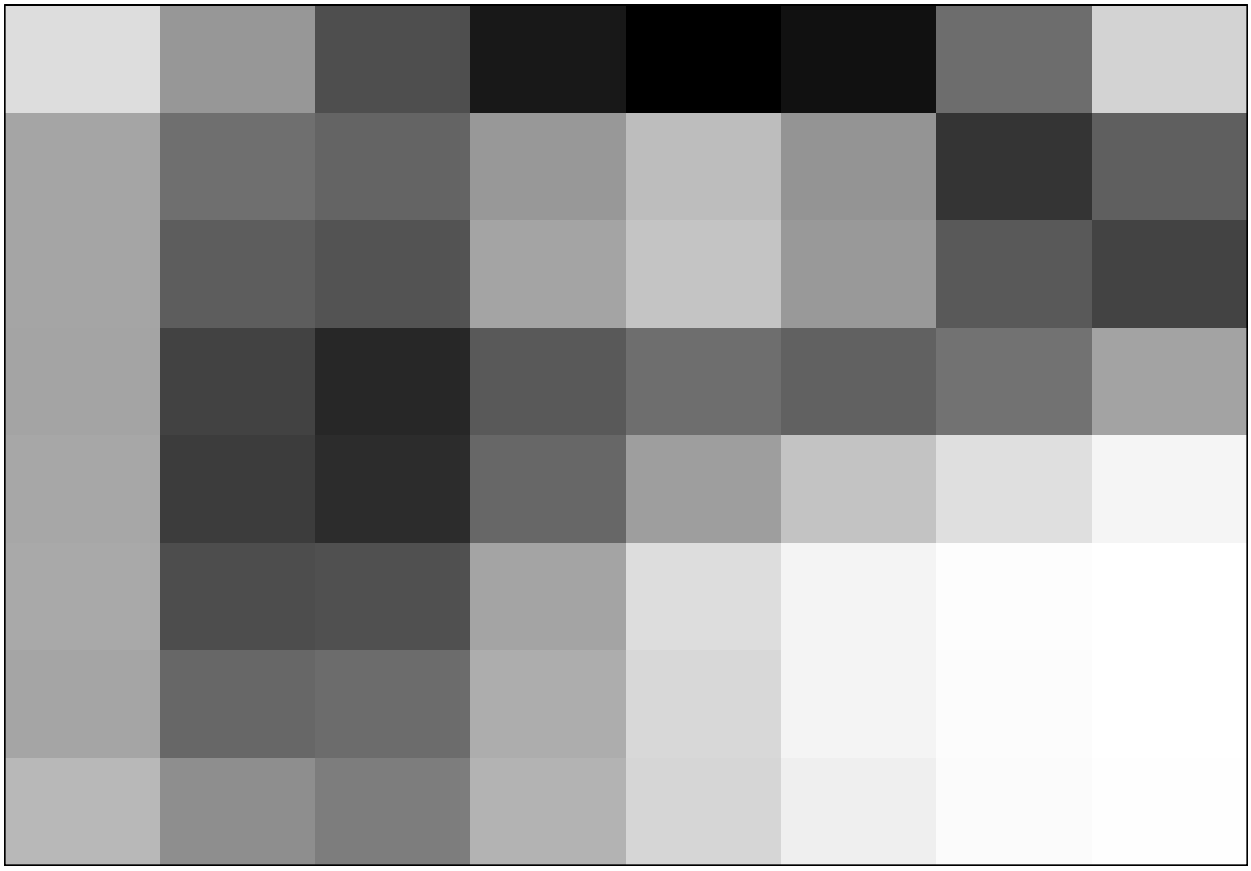
```
plot_letter(letterp[11,])
```



```
plot_letter(letterp[18,])
```



```
meanvalues <- colMeans(letterp[,-1])
prow <- letterp[1,]
prow[,2:65] <- meanvalues
plot_letter(prow)
```



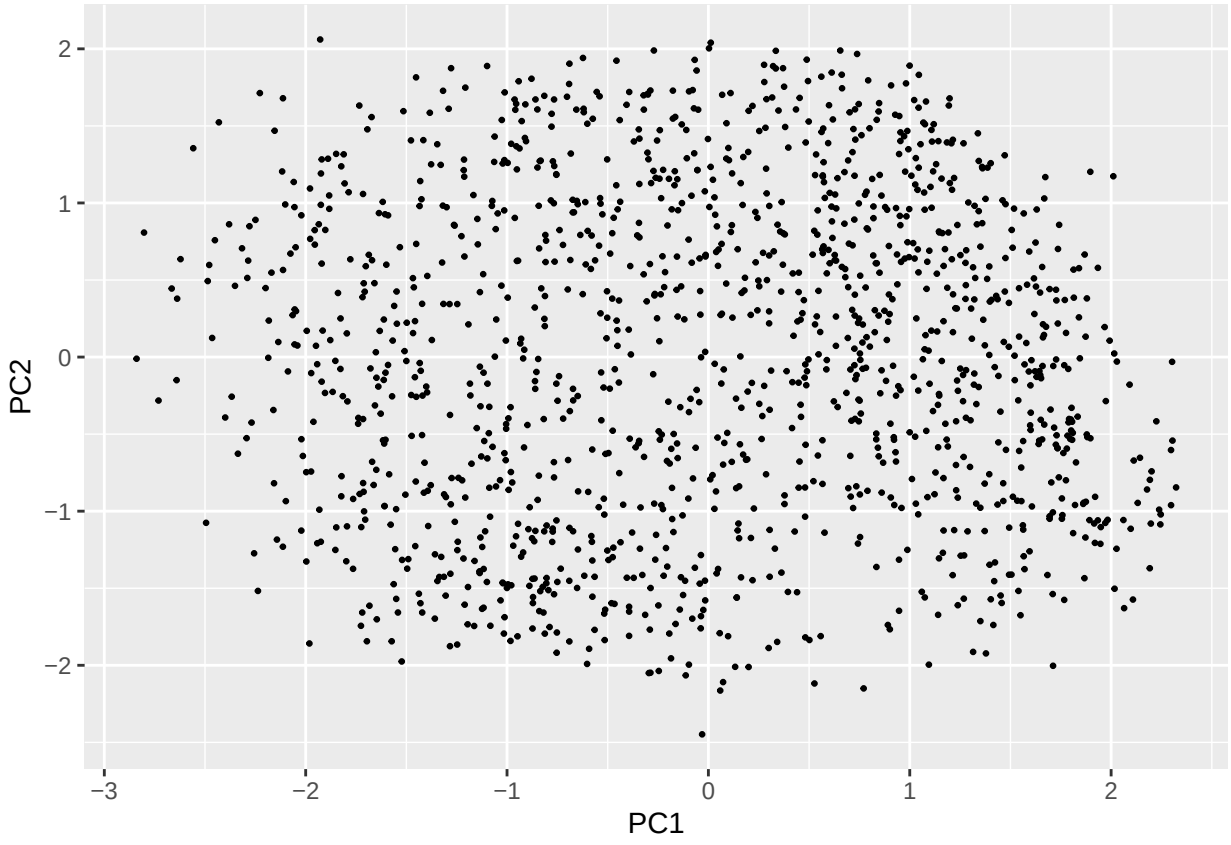
```
pca <- prcomp(letterp[,-1])
head(pca$rotation)
```

```
##              PC1        PC2         PC3        PC4         PC5         PC6
## V7  -0.033977834 0.04930187 -0.09606437 0.01469077  0.01119641 -0.03433832
## V8  -0.001786524 0.19496031 -0.22146946 0.02174224  0.09081987 -0.02971092
## V9   0.068189113 0.26566500 -0.26918949 0.11876385  0.06224744 -0.01864194
## V10  0.110115130 0.28936905 -0.18637524 0.18229947 -0.01223438  0.05727747
## V11  0.124897163 0.27244701 -0.04383096 0.22532259 -0.12429551  0.08042067
## V12  0.087148786 0.25157945  0.09274038 0.24905552 -0.21316644  0.10152295
##             PC7          PC8         PC9         PC10        PC11
## V7  0.09002771 -0.034146810  0.11637189 -0.033691191  0.09412599
## V8  0.12776562 -0.072073145  0.21781048  0.079998244  0.24033978
## V9  0.05225022  0.022577055  0.25937030  0.080294023  0.12822221
## V10 0.01614215  0.062343353  0.16998153 -0.006522148 -0.04632990
## V11 0.04129874  0.002174467  0.04283675 -0.028038205 -0.11620384
## V12 0.08462797 -0.078067809 -0.05560820 -0.023992409 -0.07490708
##              PC12         PC13        PC14         PC15        PC16
## V7  -0.030319234  0.051699263 -0.06545097  0.039017546 -0.04905646
## V8   0.011727023  0.114344743 -0.14719271  0.003270776 -0.14670575
## V9   0.036764118 -0.081776734 -0.08531442  0.023732040 -0.14306323
## V10  0.007618152 -0.139277291 -0.06982791 -0.062999160 -0.12900928
## V11  0.003478763 -0.008155401 -0.08022722 -0.163706225 -0.07195197
## V12  0.005528528  0.052849973 -0.01474955 -0.166681319  0.02275479
##             PC17        PC18        PC19        PC20        PC21
## V7   0.04053568 -0.02112143 -0.19675012  0.07331147  0.02074582
```

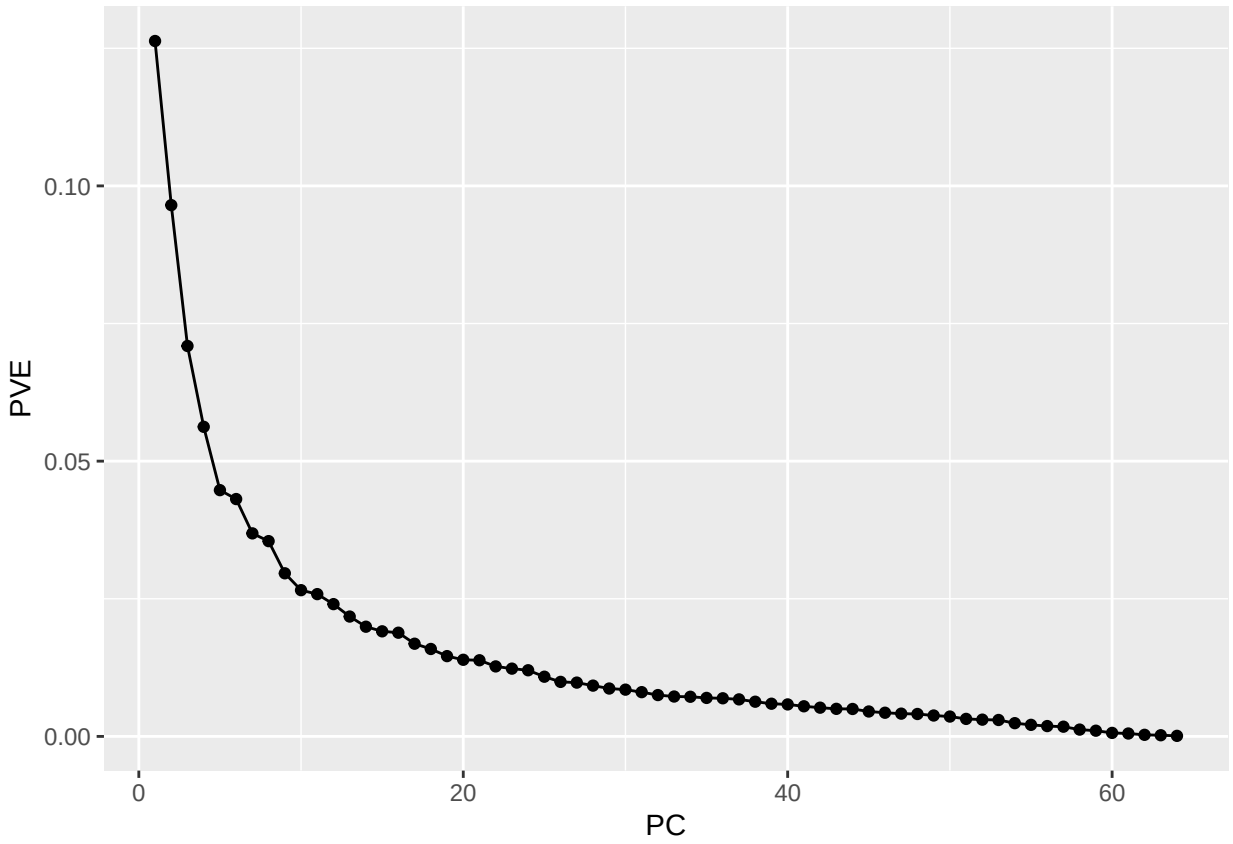
```
## V8  -0.01465850 -0.13570251 -0.24349090 -0.02773626  0.06499894
## V9  -0.04339977 -0.04281522 -0.05105723 -0.17845052 -0.09505272
## V10  0.01767950  0.16570408  0.03980842 -0.10307470 -0.03799167
## V11  0.04088879  0.23581692  0.08511083 -0.02260897  0.03357646
## V12  0.06576127  0.16197361  0.11990473  0.07675960 -0.00395762
##            PC22        PC23        PC24        PC25         PC26
## V7  -0.04854822 -0.11518418  0.07988507  0.07591731 -0.096881625
## V8   0.03218963 -0.09977858  0.15636804  0.11979520 -0.209254560
## V9  -0.03307921 -0.01299656 -0.13354383  0.02073013  0.125949690
## V10 -0.08195568 -0.01275220 -0.11581109 -0.07653563 -0.008830962
## V11 -0.01181906  0.07041730  0.01135948 -0.02171951 -0.127577649
## V12  0.16606872  0.12611188  0.08048473 -0.11900266  0.010589340
##            PC27         PC28         PC29         PC30         PC31
## V7  -0.15074549 -0.006907738  0.100973841  0.010711477  0.113273071
## V8  -0.09170487 -0.281489935  0.144937216 -0.233547366  0.307306478
## V9  -0.01142318  0.014047670  0.287842875 -0.134832252 -0.277876919
## V10 -0.00572772  0.036817573 -0.157626312  0.042465823 -0.003585598
## V11  0.03381974 -0.039439784 -0.141336713  0.038258670 -0.017014997
## V12 -0.02913093 -0.090961304  0.005977454 -0.003239693  0.001893299
##              PC32        PC33        PC34         PC35        PC36
## V7   0.1028586770 -0.15881888 -0.18222705 -0.116766598 -0.14249749
## V8  -0.0004276855 -0.01932608 -0.27423578 -0.059548382 -0.02762135
## V9  -0.2051450045  0.07715460  0.42364876  0.075944006  0.09414279
## V10  0.0935027752 -0.01784398  0.12452729 -0.006538945 -0.09117971
## V11  0.0640665610 -0.01238318 -0.02416920  0.103861010 -0.12732525
## V12 -0.0425516876  0.09022095 -0.07969108  0.231520153 -0.08391969
##            PC37        PC38         PC39        PC40         PC41
## V7  -0.02322056  0.26093124  0.091058739  0.08208765  0.103477535
## V8  -0.06153160 -0.07458048  0.009046771  0.05254269 -0.120050452
## V9   0.10650174 -0.16345148 -0.103389707 -0.14364890  0.102741680
## V10  0.07113839  0.12958533  0.038523266  0.03628018 -0.006023218
## V11 -0.01345769  0.10413316  0.002837140  0.00760365  0.030172111
## V12 -0.17330165  0.27077466  0.158149248  0.01527297 -0.162089886
##            PC42       PC43        PC44        PC45        PC46
## V7   0.04138360 -0.4037474 -0.07856428 -0.54114013  0.24076795
## V8   0.01005867  0.1884553  0.01520767  0.26936162 -0.14997599
## V9   0.10927068  0.1297542 -0.15517997 -0.18833062  0.10835848
## V10 -0.28626086 -0.3922387  0.38030978  0.26388612  0.01745921
## V11  0.06513525 -0.0192483 -0.01626659  0.04601287 -0.09833978
## V12  0.19450110  0.2815253 -0.24253488 -0.04982153  0.12579378
##             PC47        PC48        PC49         PC50        PC51
## V7   0.063703239  0.18309821  0.12748256 -0.037328700 -0.09563341
## V8  -0.073738762 -0.11856615 -0.09312921 -0.005823741  0.06345060
## V9  -0.063495822  0.08848462  0.03928982 -0.013133612 -0.06000558
## V10  0.002026246 -0.03011470  0.02210759  0.043538818  0.35950360
## V11  0.129024335 -0.03941962 -0.06300975  0.106562828 -0.70166591
## V12 -0.075697899  0.16545086  0.07369283 -0.101954662  0.36619921
##            PC52        PC53        PC54         PC55          PC56
## V7  -0.02535135 -0.14916177  0.05386178  0.010726332 -0.0106740789
## V8   0.04981552  0.07406897 -0.03871161  0.003395997  0.0004819222
## V9  -0.00727973 -0.03346015  0.03577925 -0.012989945  0.0140908430
## V10 -0.06023130 -0.06898352  0.08446538 -0.001583391  0.0144779881
## V11  0.06126486  0.25407584 -0.08336532  0.036150608  0.0179252799
## V12  0.01777305 -0.11044052  0.01485329  0.019495179  0.0031569927
```

```
##             PC57         PC58         PC59         PC60         PC61
## V7  -0.029485876  0.008144679  0.001555576  0.023422323 -0.020731553
## V8   0.004250860 -0.018830483 -0.013730100 -0.001005980  0.002098832
## V9  -0.025181427  0.011881989  0.009549539  0.003863121  0.003520426
## V10 -0.008669207 -0.003568455  0.010820246  0.007958533 -0.013641713
## V11  0.046974328 -0.020313319  0.003315976  0.003797095 -0.008129668
## V12 -0.042521644 -0.004897042 -0.010499487  0.009212336 -0.012593808
##             PC62          PC63          PC64
## V7   0.007308831  1.613479e-03  0.0005452024
## V8  -0.002695422  7.975665e-06 -0.0022394233
## V9   0.003455575  2.404153e-03 -0.0012988742
## V10  0.011912251 -2.808267e-03 -0.0018314651
## V11 -0.026442216  2.239720e-03  0.0006712100
## V12  0.013047163  7.747387e-03 -0.0021949239
```

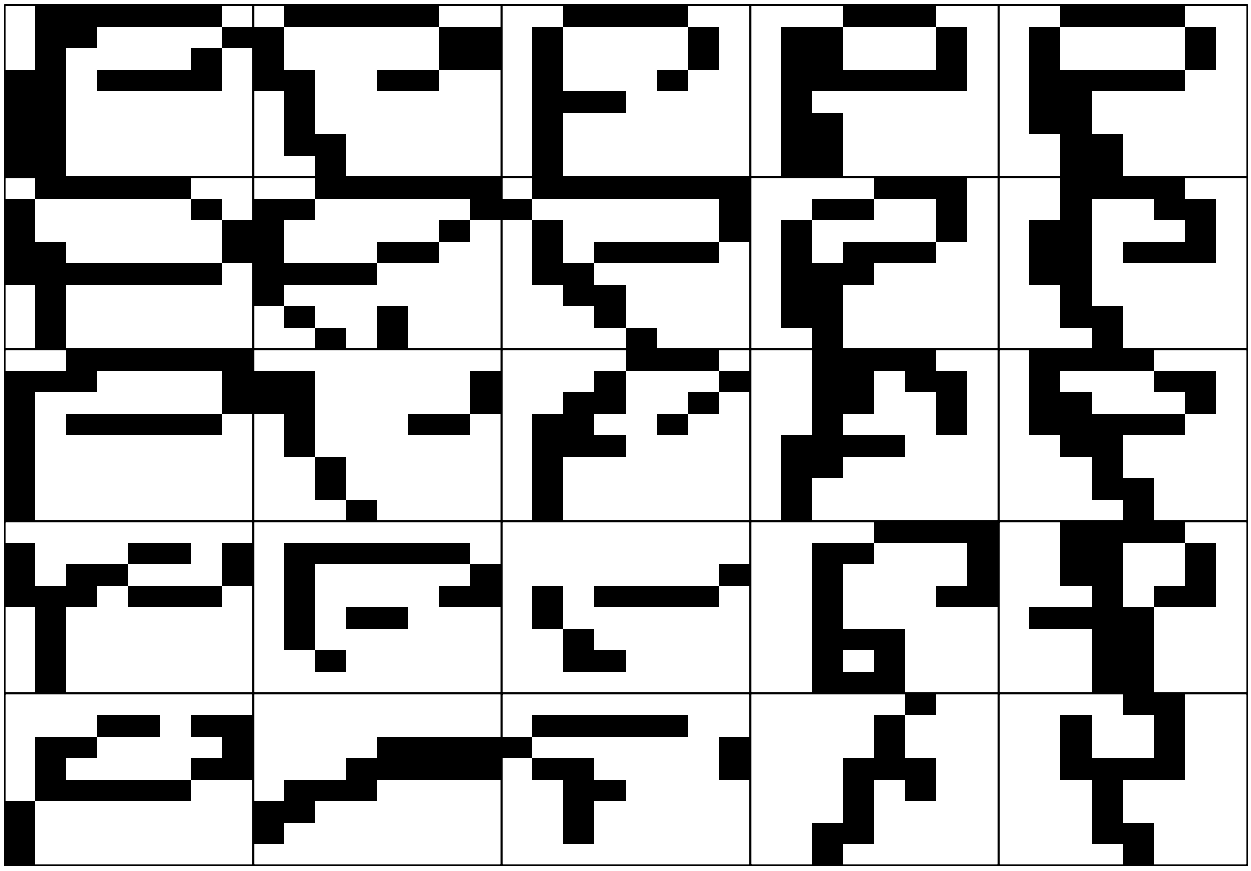
```
d <- as.data.frame(pca$x)
p1 <- ggplot(d, aes(x = PC1, y = PC2)) +
  geom_point(size = .5)
p1
```



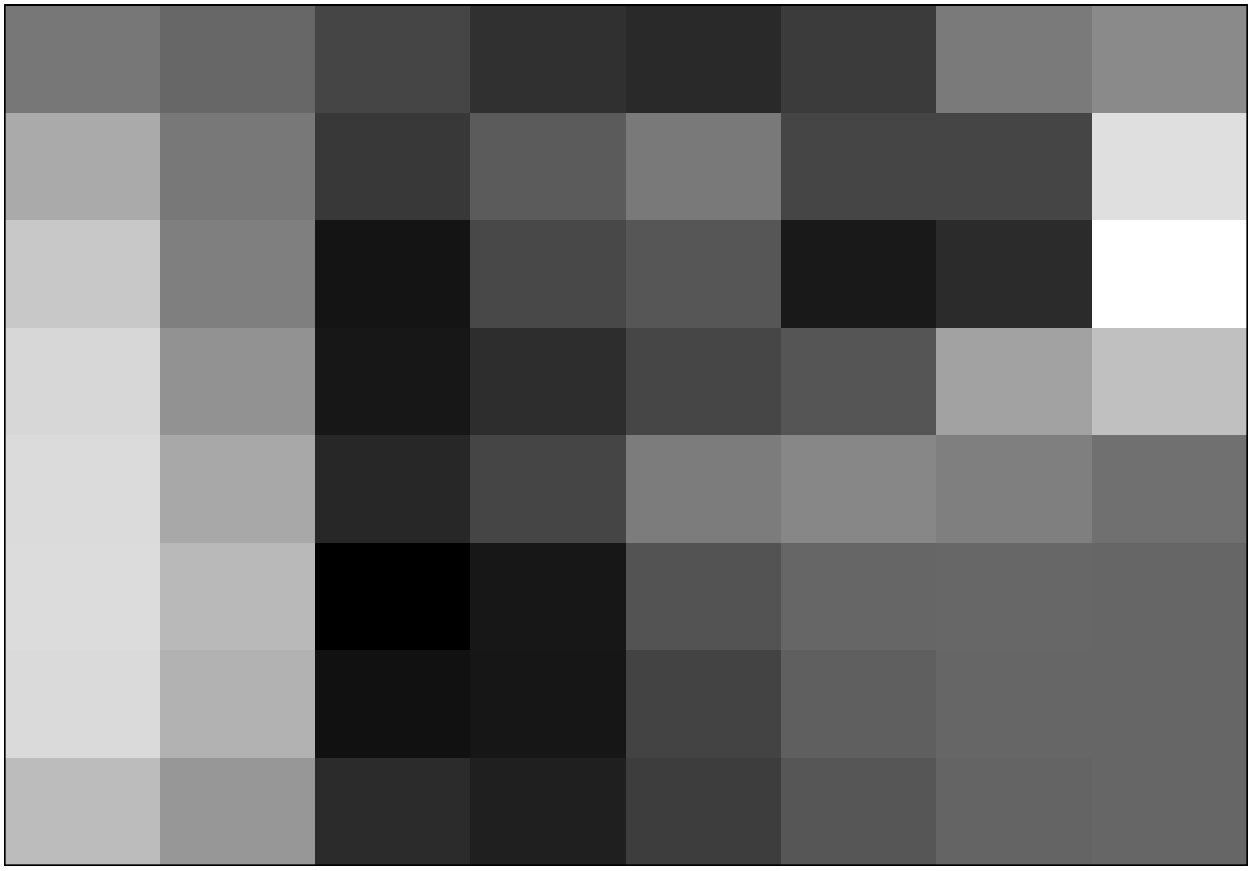

```
d1 <- data.frame(PC = 1:64,
                 PVE = pca$sdev^2 /
                   sum(pca$sdev^2))
ggplot(d1, aes(x = PC, y = PVE)) +
  geom_line() +
  geom_point()
```



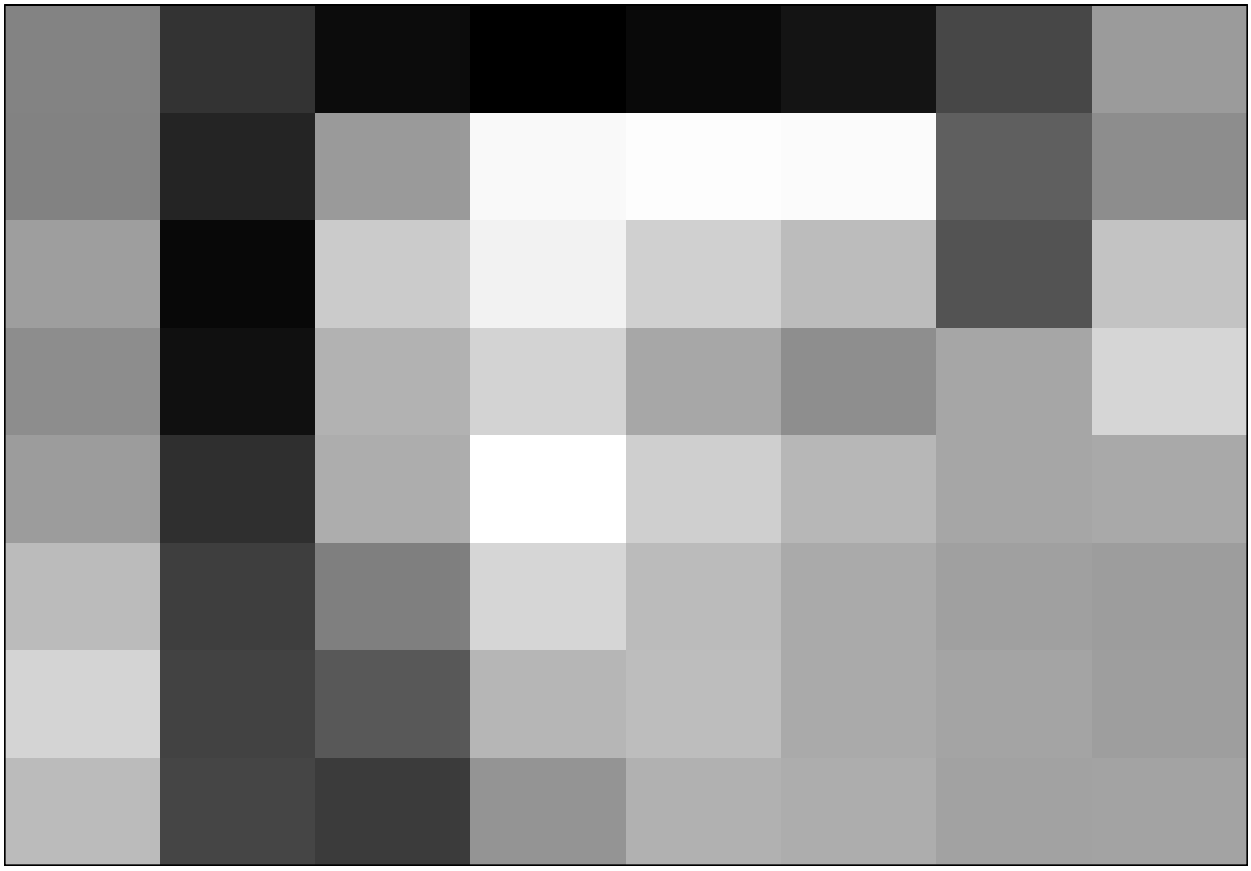

```
#work from 11/27
pc_grid <- function(pca, data) {
  d <- data
  grid_points <- as.matrix(expand.grid(seq(-1.5, 1.5, length.out = 5),
                                       seq(-1.5, 1.5, length.out = 5)))
  pc_points <- pca$x[, 1:2]
  nearest_ind <- rep(NA, nrow(grid_points))
  for(i in 1:nrow(grid_points)) {
    gp <- matrix(rep(grid_points[i, ], nrow(pc_points)),
                 ncol = 2, byrow = TRUE)
    nearest_ind[i] <- which.min(rowSums((pc_points - gp)^2))
  }
  nearest_grid <- data.frame(d[nearest_ind, ])
  par(mfrow = c(5, 5))
  regrid <- c(21:25, 16:20, 11:15, 6:10, 1:5)
  for(i in regrid) {
    plot_letter(nearest_grid[i, ])
  }
}

pc_grid(pca, letterp)
```



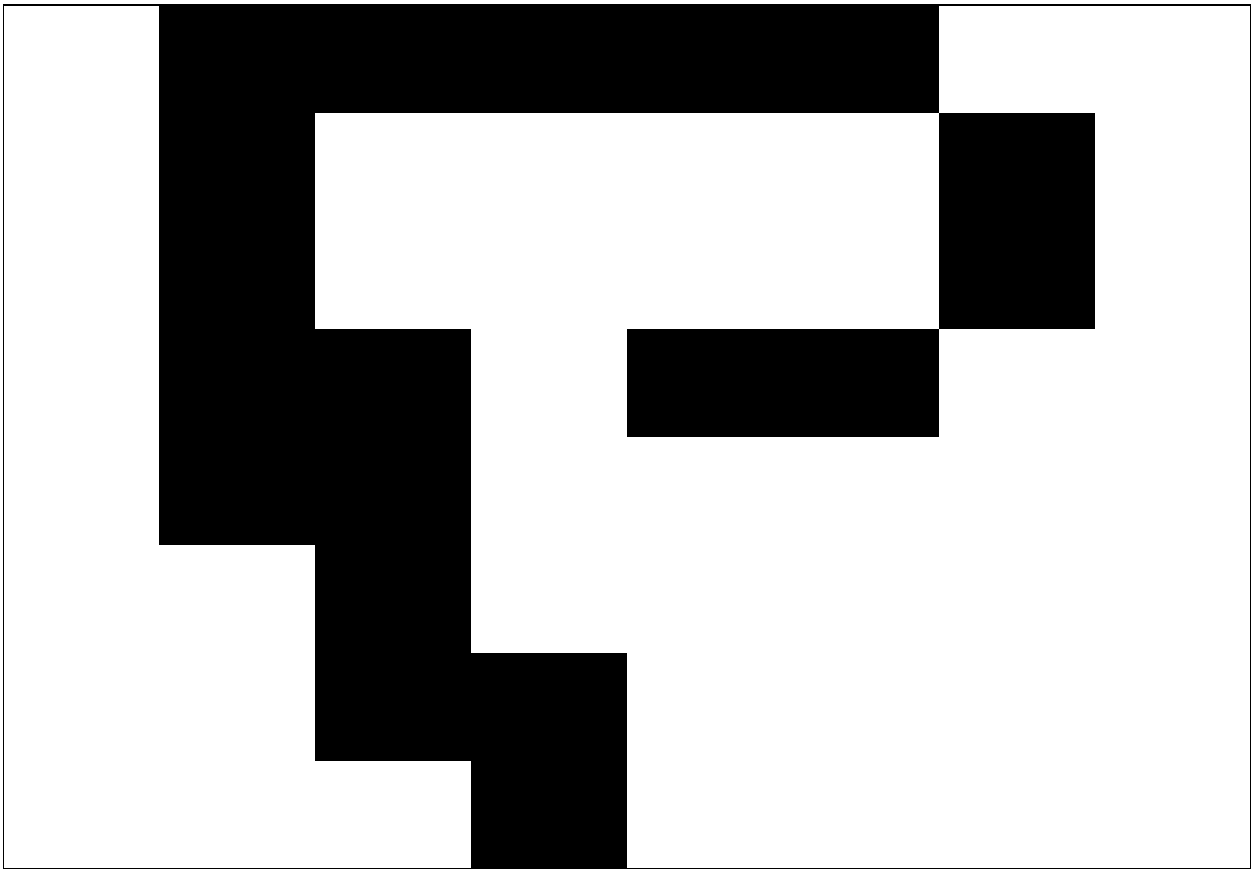
```r
loading <- pca$rotation

par(mfrow = c(1,1))
plot_letter(loading[,1], hasletter = FALSE)
```



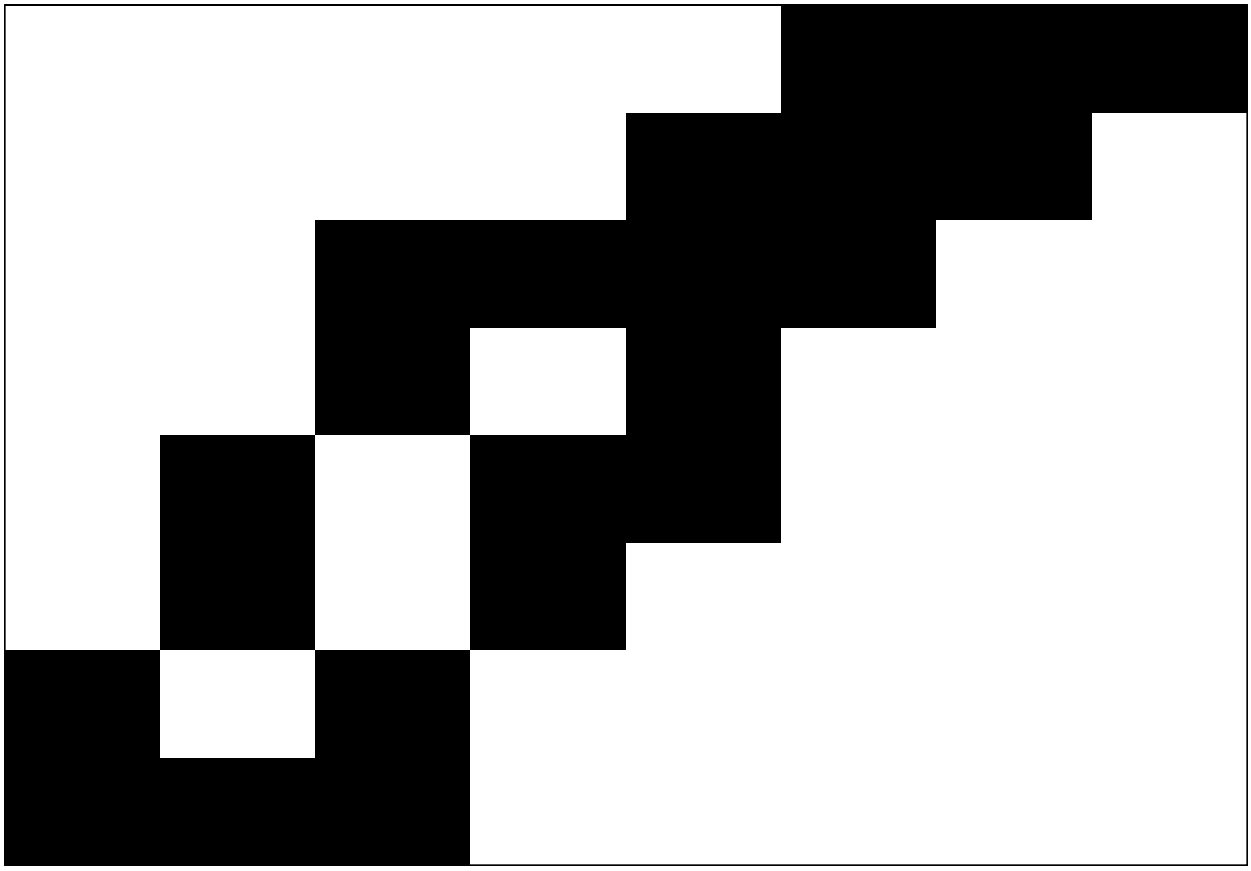
```
plot_letter(loading[,2], hasletter = FALSE)
```



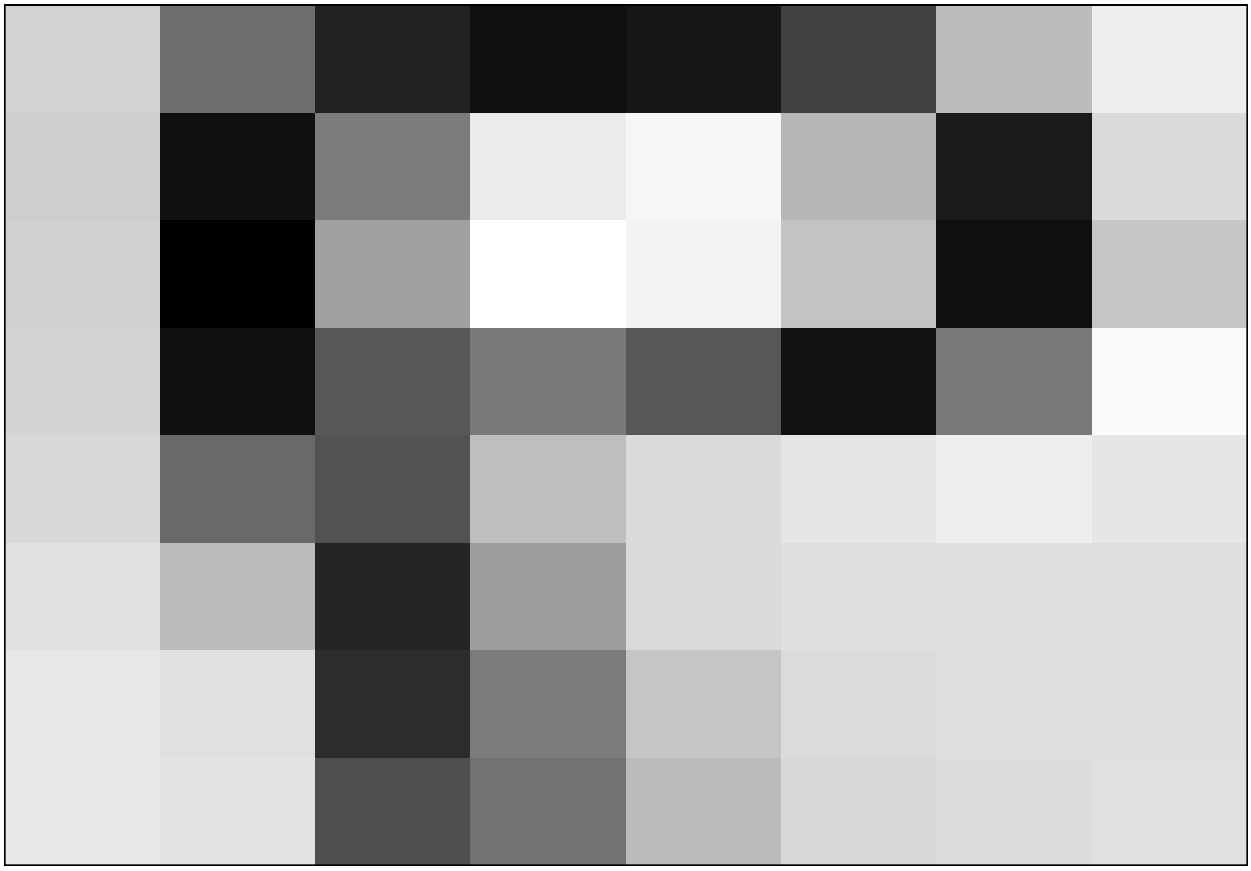
```
#reconstructing some letters
plot_letter(letterp[2,])
```



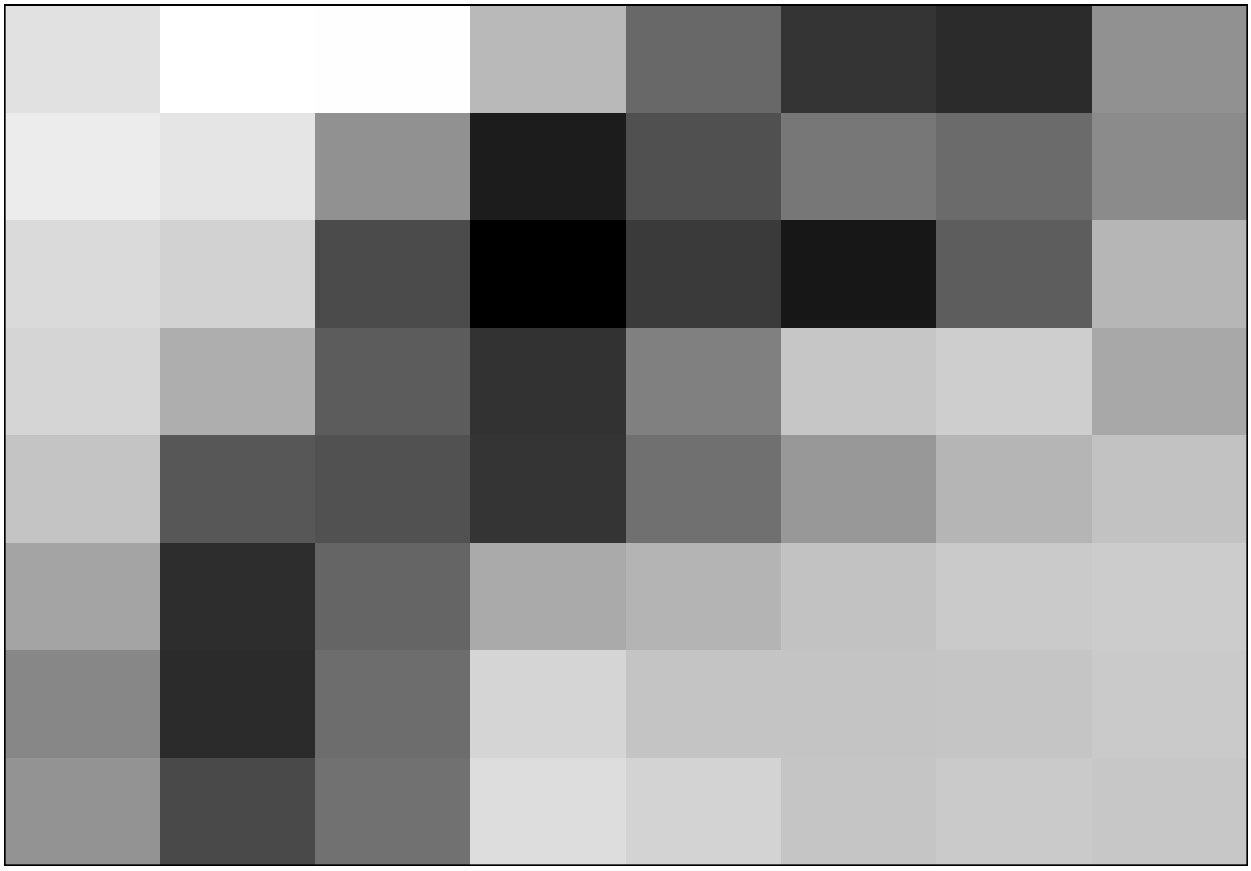
```
plot_letter(letterp[7,])
```



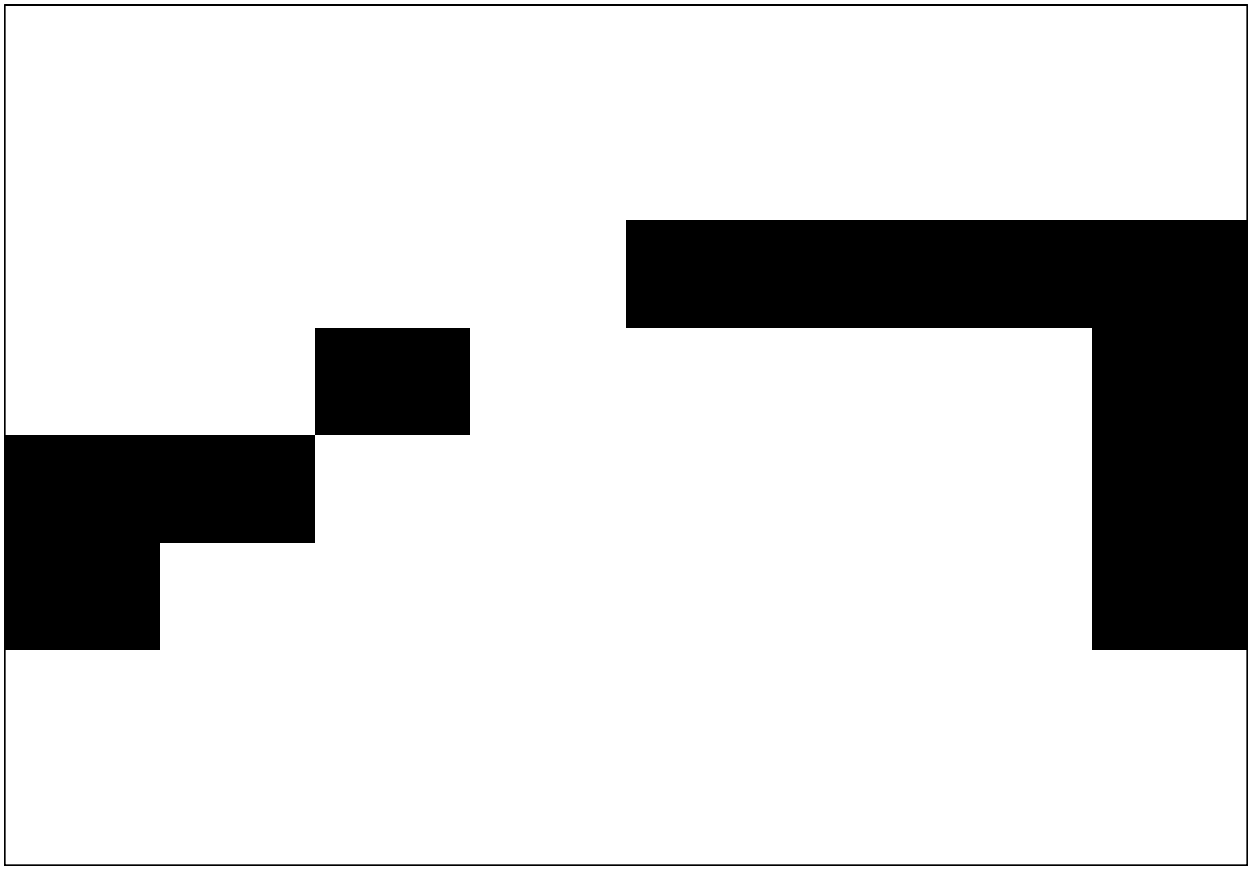
```
m <- 10
loadings1tom <- loading[,1:m]

zs <-  pca$x
zs2 <- zs[2,1:m]
zs7 <- zs[7,1:m]

reconstructed2 <- meanvalues + zs2%*%t(loadings1tom)
plot_letter(reconstructed2, hasletter = FALSE)
```



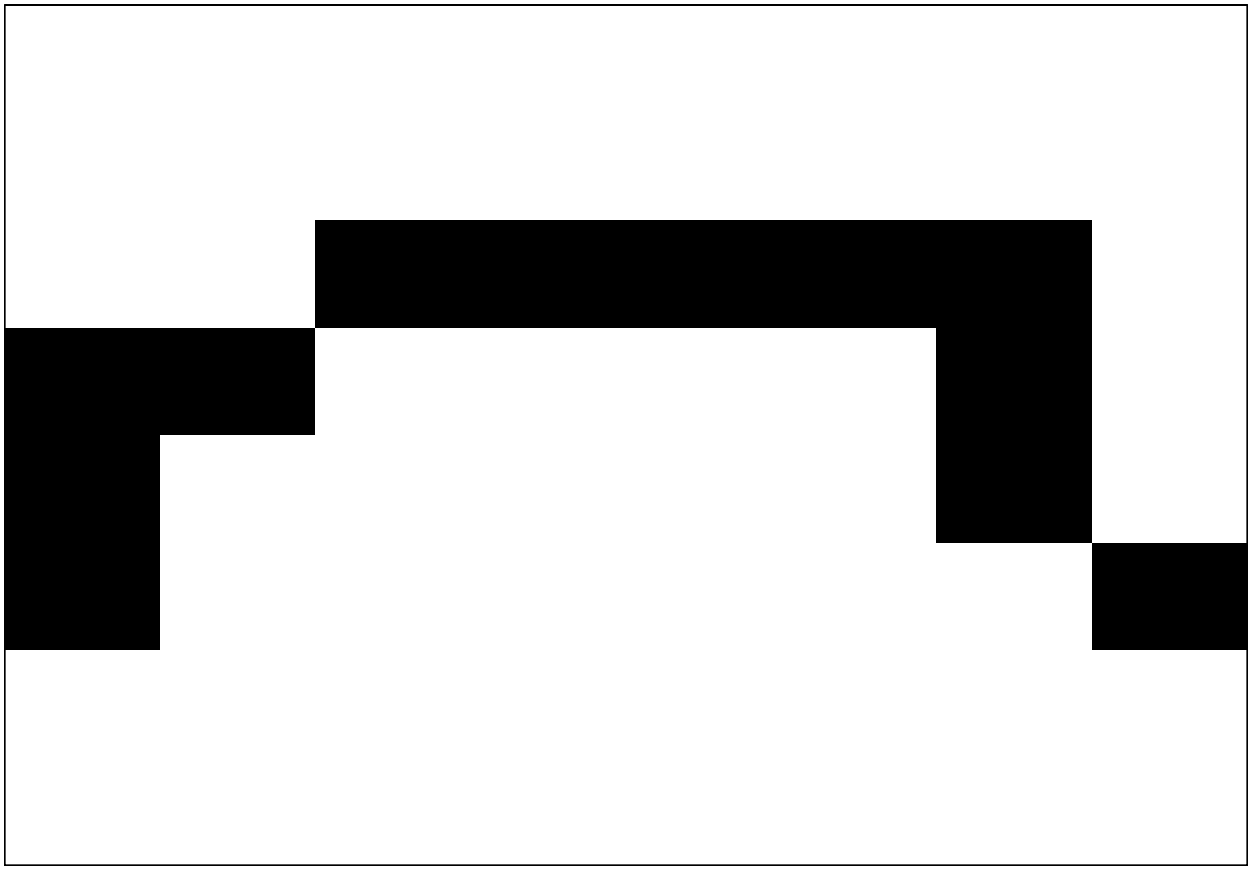
```
reconstructed7 <- meanvalues + zs7 %*% t(loadings1tom)
plot_letter(reconstructed7, hasletter = FALSE)
```



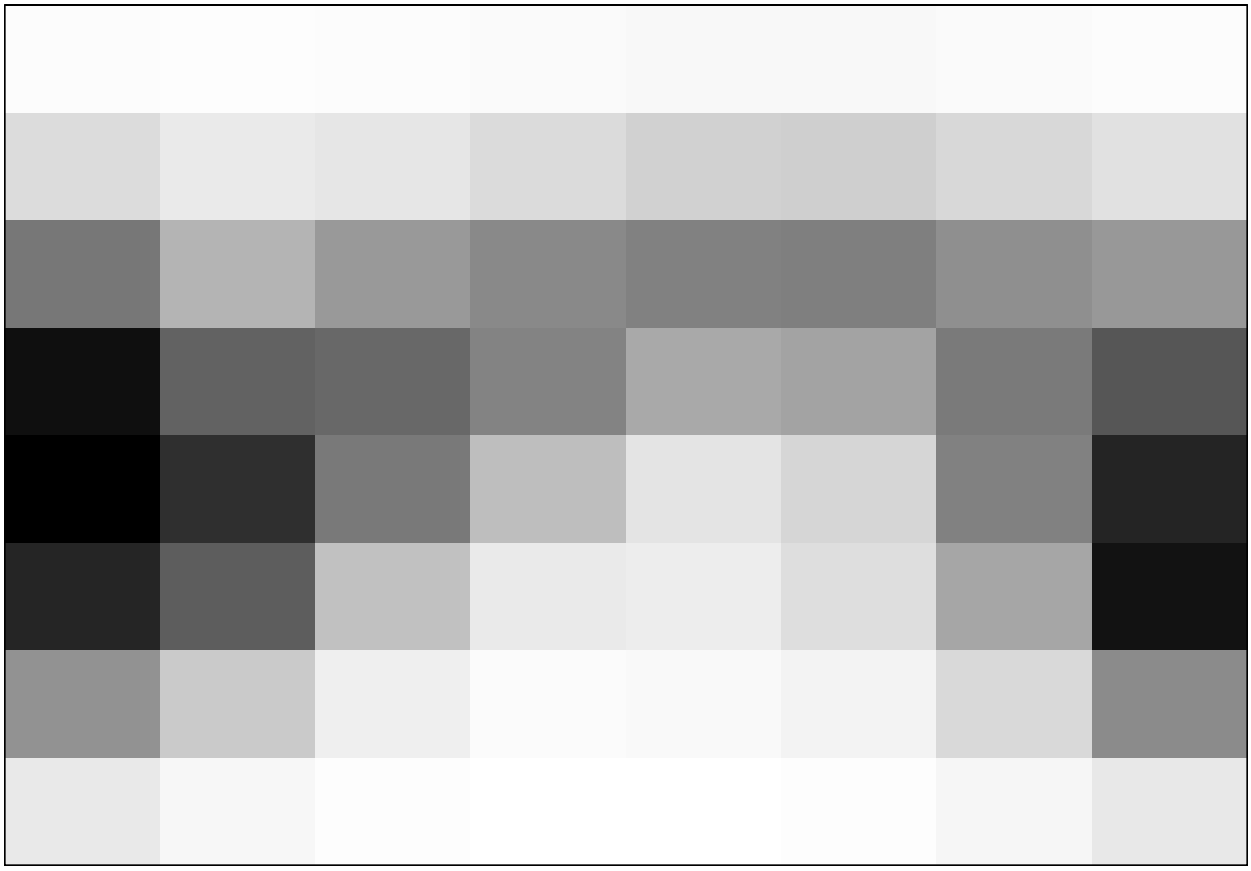
```r
d <- read.csv("https://raw.githubusercontent.com/stat-learning/course-materials/master/data/handwritten
newletter <- d %>%
  filter(letter == "n")

plot_letter(newletter[2,])
```



```
plot_letter(newletter[7,])
```



```
mean = colMeans(newletter[,-1])
plot_letter(mean, hasletter = FALSE)
```



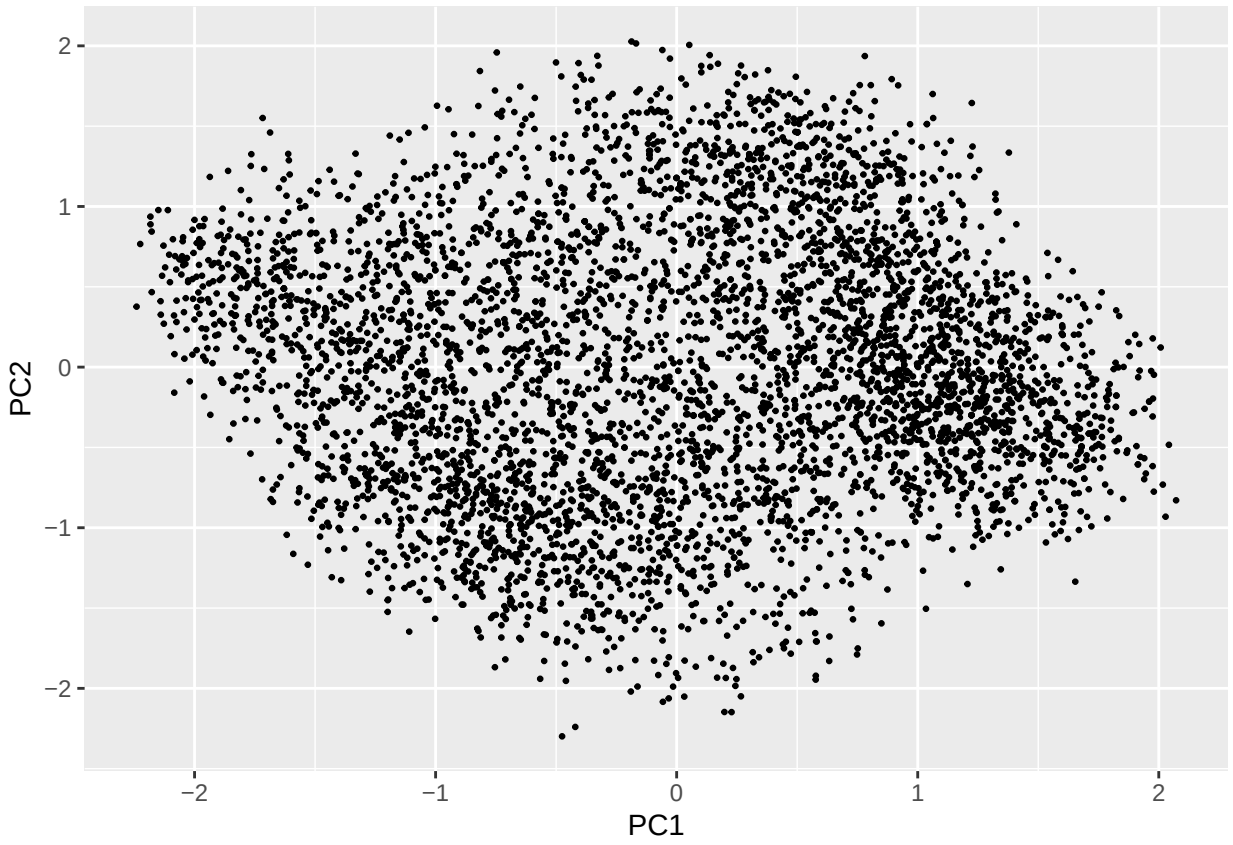
```
npca <- prcomp(newletter[,-1])
head(npca$rotation)
```

```
##              PC1         PC2         PC3        PC4           PC5
## V7  -0.009791180 0.008197438 -0.01023425 0.02028013 -0.0035797277
## V8  -0.003211310 0.008119746 -0.01392390 0.01535955  0.0003258952
## V9  -0.002991065 0.011512284 -0.01795612 0.02002915  0.0040471711
## V10 -0.010737692 0.017099649 -0.02645659 0.02405351 -0.0028819699
## V11 -0.018511781 0.021292938 -0.03457626 0.02971677  0.0071982240
## V12 -0.027965562 0.007509140 -0.03137951 0.02674644  0.0105738790
##              PC6          PC7          PC8          PC9         PC10
## V7  -0.009072337  0.006430225 -0.007389711  0.005474417 -0.006247795
## V8  -0.001398950 -0.007016111 -0.006840398 -0.007250385 -0.001840887
## V9  -0.006508137 -0.013583101 -0.021246077 -0.010820286 -0.012139969
## V10 -0.013521867 -0.014776536 -0.028467719 -0.013026220 -0.019463466
## V11 -0.026140285 -0.007116292 -0.033001829 -0.017717020 -0.023869946
## V12 -0.032635113  0.008208192 -0.005246964 -0.035877607 -0.017092563
##             PC11          PC12         PC13        PC14          PC15
## V7  0.011926266 -0.0078144166 0.0006827346 -0.01089953  0.0119801981
## V8  0.010545432 -0.0013319232 0.0040770638 -0.02541163  0.0016568117
## V9  0.008876926 -0.0004390072 0.0102561114 -0.04519652 -0.0046175566
## V10 0.006264789 -0.0156611640 0.0116376655 -0.05884688 -0.0029438041
## V11 0.005550231 -0.0169636458 0.0261113206 -0.05805567 -0.0049844173
## V12 0.007086038 -0.0083557628 0.0194377072 -0.02472039 -0.0007383929
##             PC16         PC17          PC18         PC19         PC20
## V7  -0.002538023 -0.006802789  1.426870e-02  0.005358966 -0.001060920
```

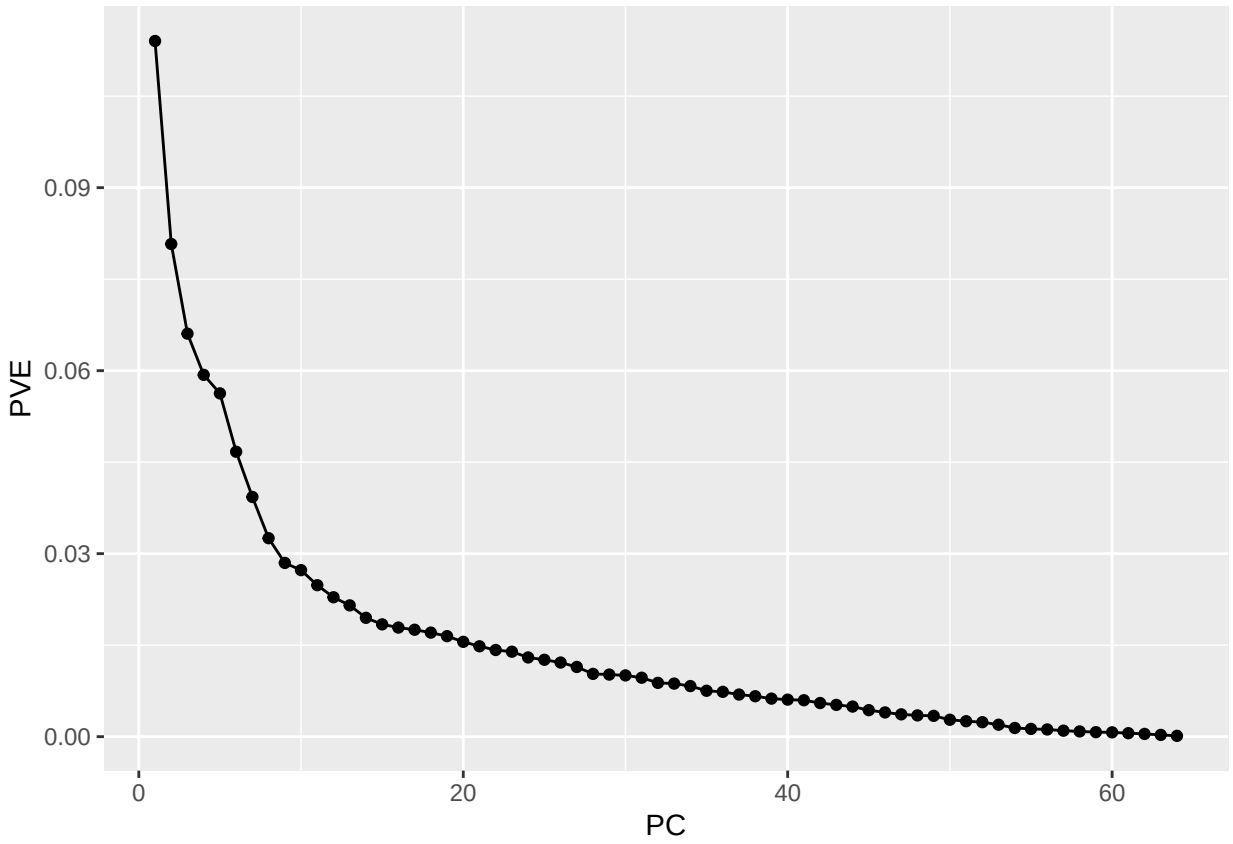
```
## V8  -0.003449656 -0.012168094  1.909046e-03  0.018176786 -0.003814684
## V9  -0.003773427 -0.013043304  3.148733e-05  0.013211546  0.007175892
## V10 -0.002287549 -0.011198235 -6.523354e-03 -0.006023348  0.003109592
## V11  0.005293017 -0.015198384 -7.846832e-03 -0.014574985 -0.009896224
## V12  0.004631195 -0.008752140 -8.043381e-03 -0.021333321 -0.023001270
##             PC21         PC22       PC23         PC24       PC25
## V7  -0.007058359 -0.019768762 0.01730217  0.005570236 0.01683663
## V8  -0.009223100 -0.012757605 0.01708104  0.002734845 0.01117619
## V9  -0.021403267 -0.007656245 0.03309439  0.002392552 0.02541215
## V10 -0.017685249 -0.018504008 0.05521169 -0.001274648 0.06285578
## V11 -0.010307721 -0.031267371 0.08983078  0.007072545 0.07937666
## V12  0.008087771 -0.026486460 0.10240792  0.006150253 0.08824370
##            PC26        PC27       PC28        PC29         PC30
## V7  -0.03809355 -0.07407436 0.03584557 0.004131254  0.014700302
## V8  -0.03640173 -0.02903074 0.02385571 0.002890842  0.022916691
## V9  -0.04975314 -0.03572643 0.02789270 0.007757911  0.038490299
## V10 -0.07861426 -0.08761950 0.03398414 0.035826747  0.009184612
## V11 -0.10001994 -0.12559469 0.04257888 0.054178121 -0.016578646
## V12 -0.09887025 -0.11687017 0.06588982 0.088165741 -0.022050068
##              PC31         PC32         PC33          PC34        PC35
## V7  -0.0248095628 -0.011798552 -0.031704398  0.0503300245 0.011658663
## V8  -0.0019587126 -0.001412058 -0.023332589  0.0004263536 0.009917345
## V9   0.0284656601  0.003224927 -0.014610377 -0.0641879425 0.066840667
## V10 -0.0004176219  0.004782037 -0.009725701 -0.1147149699 0.097200716
## V11 -0.0318183631  0.006266050  0.031128627 -0.1616803009 0.090754664
## V12 -0.0689347529  0.020025665  0.069247829 -0.1595449561 0.024775526
##              PC36          PC37          PC38         PC39         PC40
## V7   0.0169775294 -0.0180617661 -0.0148990208 -0.002105638 -0.02854036
## V8   0.0020974056  0.0275864413 -0.0226958719  0.033475088  0.02791323
## V9  -0.0009352279  0.0299872329 -0.0329905458  0.066277168  0.10883623
## V10  0.0003777189  0.0006689178  0.0001827163  0.051485326  0.12130737
## V11 -0.0124893905 -0.0012689353  0.0190536829  0.015343626  0.07139639
## V12 -0.0357022561  0.0258162052 -0.0312458683 -0.051113114 -0.04592737
##             PC41       PC42          PC43       PC44          PC45
## V7  -0.021363439 0.04542232  0.0006422963 0.01942151 -0.0001785813
## V8   0.009176709 0.02658576 -0.0655907802 0.03616028 -0.0075295290
## V9   0.023433647 0.05955075 -0.0195519496 0.05316868 -0.0691323948
## V10 -0.008469131 0.10615724  0.0752771161 0.07572404 -0.1055658710
## V11 -0.028935518 0.12933178  0.0796697104 0.10803848 -0.1032843116
## V12 -0.051838708 0.10804802  0.0276882138 0.08498487 -0.1189543242
##            PC46       PC47        PC48        PC49        PC50        PC51
## V7  -0.04503586 0.01302316 0.002356865  0.03171273  0.06199940 -0.09889524
## V8  -0.02934896 0.03785259 0.050066073 -0.01202212  0.05760214 -0.15062891
## V9   0.06290611 0.09932147 0.072731574 -0.02540977  0.02447609 -0.30293443
## V10  0.09731715 0.16396251 0.124504493 -0.07826842 -0.10749449 -0.33874114
## V11  0.20541283 0.20871765 0.186532846 -0.16864315 -0.20409267 -0.06237481
## V12  0.30952814 0.14251024 0.163038877 -0.13308272 -0.20226567  0.29945436
##            PC52        PC53        PC54        PC55         PC56
## V7  -0.00329730 -0.01983110  0.72125097  0.24570034 -0.420550081
## V8  -0.02150236  0.02711185  0.48183766 -0.25063499 -0.005902322
## V9  -0.21973162  0.05972118  0.17143586 -0.36653973  0.233391283
## V10 -0.35757304  0.04181826 -0.09255482 -0.09634215  0.100182228
## V11 -0.27355529 -0.04357063 -0.16586201  0.22763465 -0.139155175
## V12  0.13083304 -0.08214066  0.09423400  0.10822654 -0.106427906
```

```
##             PC57         PC58       PC59        PC60        PC61
## V7  -0.16241607  0.200782043  0.2568795 -0.26702236 -0.01813483
## V8   0.37707222 -0.204494922 -0.1451171  0.66029791 -0.08325772
## V9   0.17726396  0.007058394 -0.3112737 -0.48082792  0.07021692
## V10 -0.05805154  0.265240924  0.1128446 -0.05878718 -0.10071870
## V11 -0.15933739  0.022787805  0.2180792  0.36152387 -0.07642341
## V12  0.16974531 -0.361894936 -0.1383407 -0.17313139  0.49959325
##             PC62         PC63         PC64
## V7  -0.01939602 -0.016220512  0.014009440
## V8   0.08736788  0.029083563  0.013766433
## V9  -0.43912661 -0.086834196 -0.014162698
## V10  0.66942684  0.059390303  0.030495450
## V11 -0.52164318 -0.044396473 -0.021999010
## V12  0.22739553 -0.004740243  0.009295561
```

```
d <- as.data.frame(npca$x)
p1 <- ggplot(d, aes(x = PC1, y = PC2)) +
  geom_point(size = .5)
p1
```

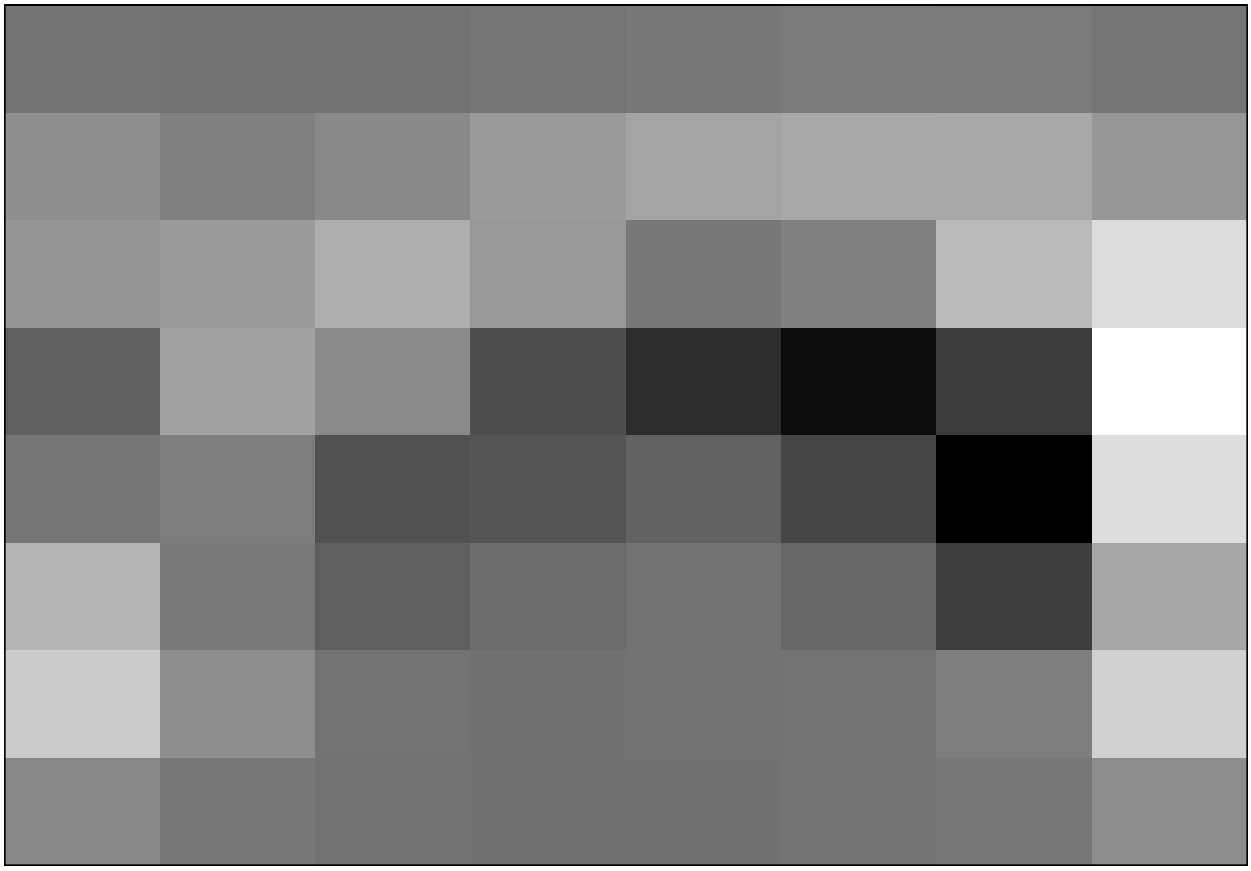



```
d2 <- data.frame(PC = 1:64,
                 PVE = npca$sdev^2 /
                   sum(npca$sdev^2))
ggplot(d2, aes(x = PC, y = PVE)) +
  geom_line() +
  geom_point()
```

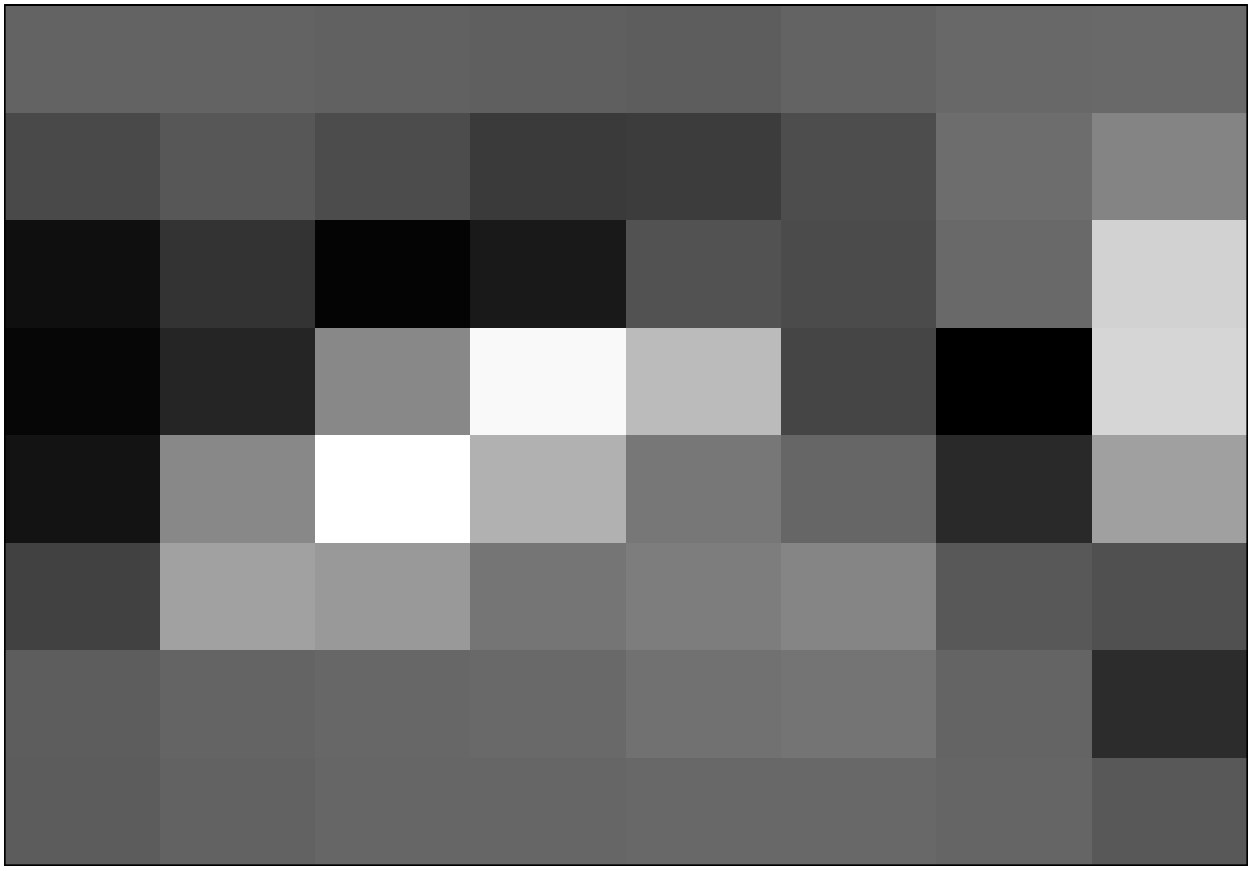



```
pc_grid(npca, newletter)
```

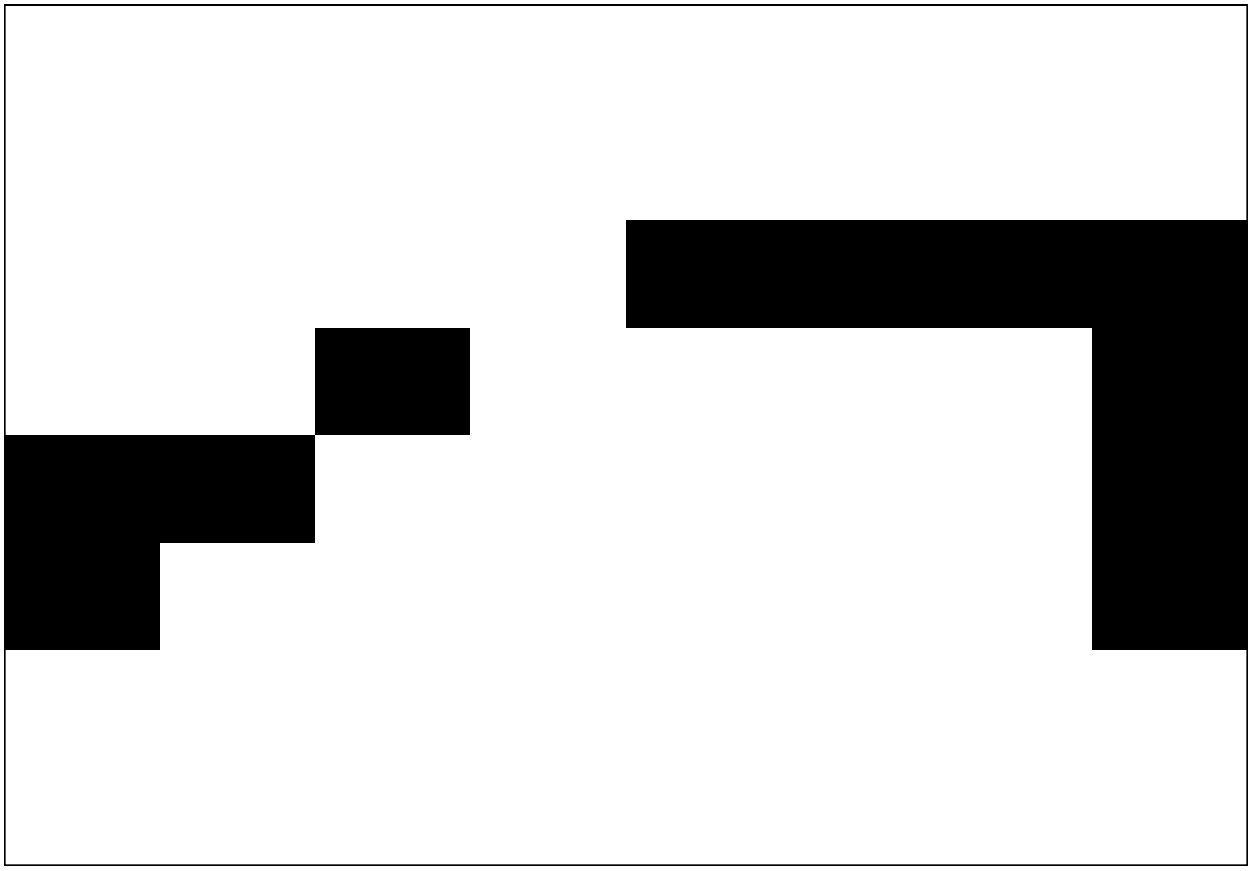
```r
loadingsforn <- npca$rotation

par(mfrow = c(1,1))
plot_letter(loadingsforn[,1], hasletter = FALSE)
```



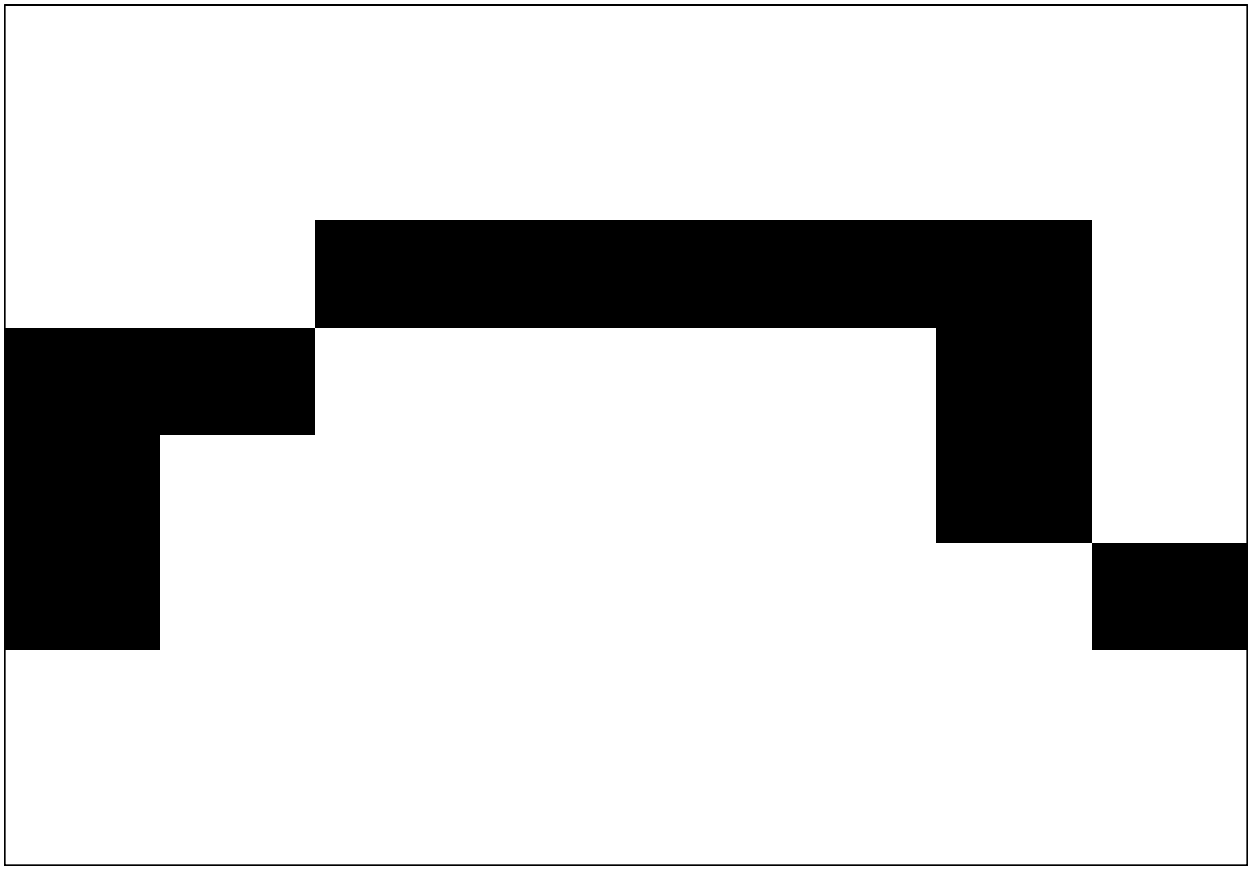
```
plot_letter(loadingsforn[,2], hasletter = FALSE)
```



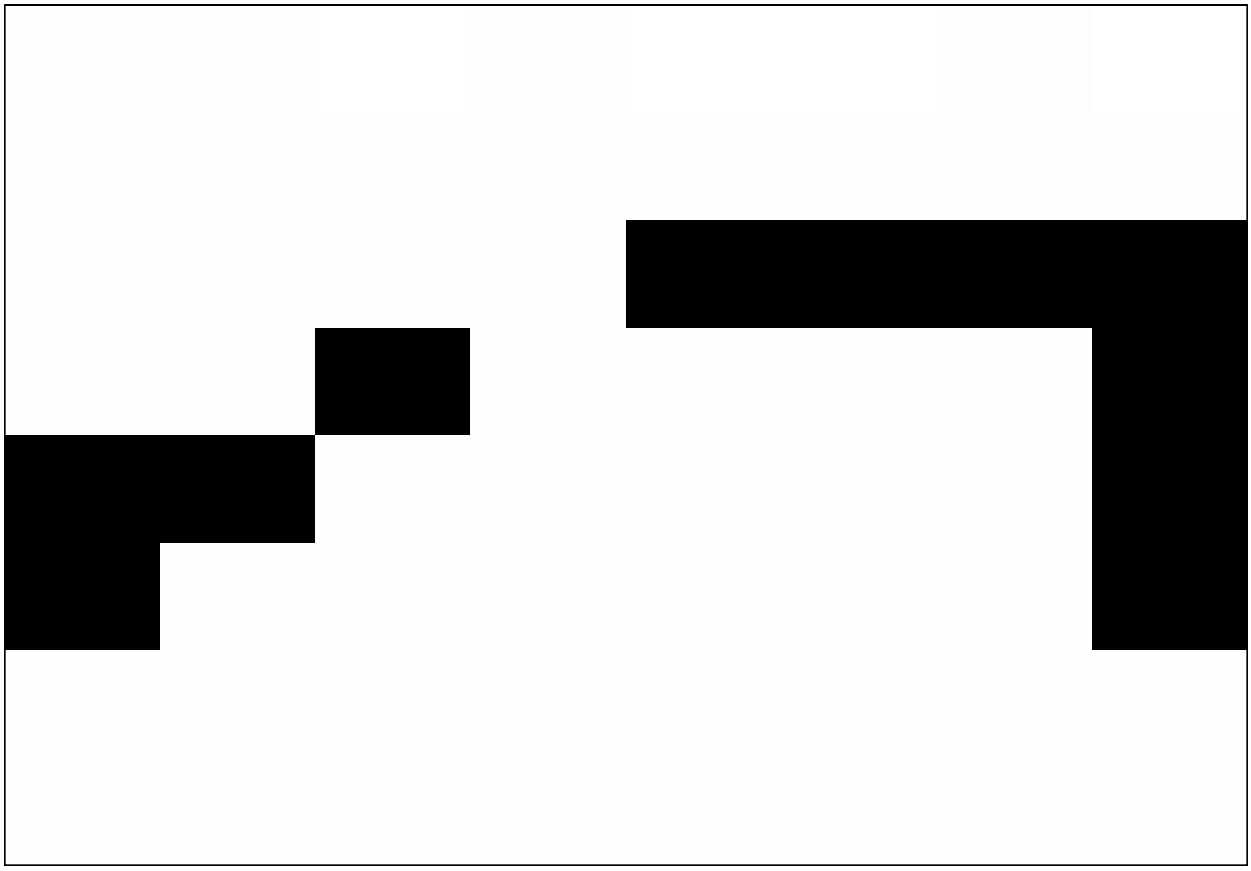
```
#reconstructing n
plot_letter(newletter[2,])
```



```
plot_letter(newletter[7,])
```



```
m <- 60
loadings1tom <- loadingsforn[,1:m]

zs <-  npca$x
zs2 <- zs[2,1:m]
zs7 <- zs[7,1:m]

reconstructed2n <- mean + zs2%*%t(loadings1tom)
plot_letter(reconstructed2n, hasletter = FALSE)
```



```
reconstructed7n <- mean + zs7 %*% t(loadings1tom)
plot_letter(reconstructed7n, hasletter = FALSE)
```