MSMB-Chapter5-Clustering

Aleeza Gerstein

2019-11-06

```
turtles <-read.table("../../data/PaintedTurtles.txt", header = TRUE)
turtles[1:4, ]
```

```
##   sex length width height
## 1   f     98    81     38
## 2   f    103    84     38
## 3   f    103    86     42
## 4   f    105    86     40
```

```
load("../../data/athletes.RData")
athletes[1:3, ]
```

```
##    m100 long weight highj  m400  m110  disc pole javel  m1500
## 1 11.25 7.43  15.48  2.27 48.90 15.13 49.28  4.7 61.32 268.95
## 2 10.87 7.45  14.97  1.97 47.71 14.46 44.36  5.1 61.76 273.02
## 3 11.18 7.44  14.20  1.97 48.29 14.81 43.66  5.2 64.16 263.20
```

```
load("../../data/Msig3transp.RData")
round(Msig3transp,2)[1:5, 1:6]
```

```
##              X3968 X14831 X13492 X5108 X16348  X585
## HEA26_EFFE_1 -2.61  -1.19  -0.06 -0.15   0.52 -0.02
## HEA26_MEM_1  -2.26  -0.47   0.28  0.54  -0.37  0.11
## HEA26_NAI_1  -0.27   0.82   0.81  0.72  -0.90  0.75
## MEL36_EFFE_1 -2.24  -1.08  -0.24 -0.18   0.64  0.01
## MEL36_MEM_1  -2.68  -0.15   0.25  0.95  -0.20  0.17
```

```
data("GlobalPatterns", package = "phyloseq")
GPOTUs = as.matrix(t(phyloseq::otu_table(GlobalPatterns)))
GPOTUs[1:4, 6:13]
```

```
## OTU Table:          [8 taxa and 4 samples]
##                      taxa are columns
##         246140 143239 244960 255340 144887 141782 215972 31759
## CL3          0      7      0    153      3      9      0     0
## CC1          0      1      0    194      5     35      3     1
## SV1          0      0      0      0      0      0      0     0
## M31Fcsw      0      0      0      0      0      0      0     0
```

```
data("airway", package = "airway")
assay(airway)[1:3, 1:4]
```

```
##                 SRR1039508 SRR1039509 SRR1039512 SRR1039513
## ENSG00000000003        679        448        873        408
## ENSG00000000005          0          0          0          0
## ENSG00000000419        467        515        621        365
```

```
metab = t(as.matrix(read.csv("../../data/metabolites.csv", row.names = 1)))
metab[1:4, 1:4]
```

```
##          146.0985388 148.7053275 310.1505057 132.4512963
```

```
## KOGCHUM1    29932.36    17055.70     1132.82    785.5129
## KOGCHUM2    94067.61    74631.69    28240.85   5232.0499
## KOGCHUM3   146411.33   147788.71    64950.49  10283.0037
## WTGCHUM1   229912.57   384932.56   220730.39  26115.2007
```

Task: Tabulate the frequency of zeros in these data matrices

```
table(assay(airway))[1]
```

```
##      0
## 314674
```

```
table(GPOTUs)[1]
```

```
##      0
## 395038
```

\textcolor{red}{Question 7.1) a) Columns are usually taxa b) Rows are usually genes c) a cell represents the number of reads d) athletes[5, 3]

\textcolor{red}{Question 7.2)

```
head(turtles)
```

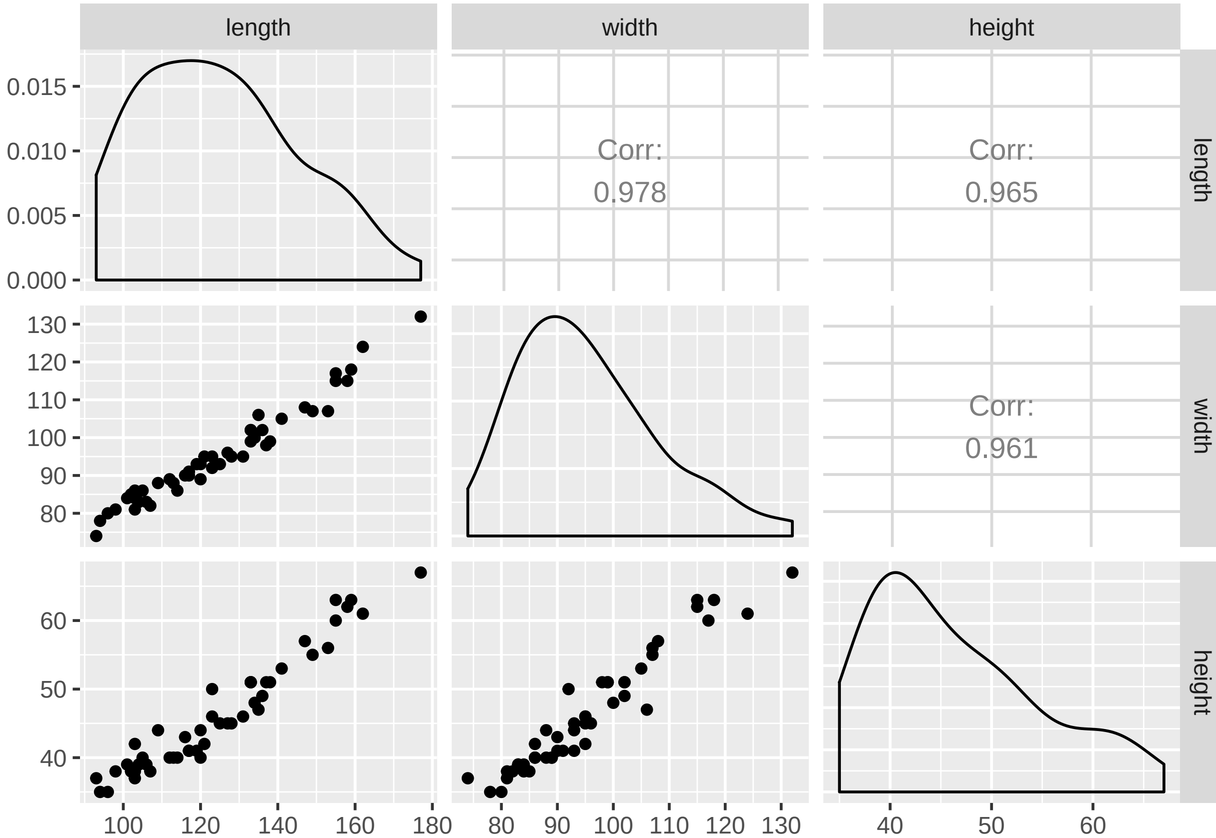
```
##   sex length width height
## 1   f     98    81     38
## 2   f    103    84     38
## 3   f    103    86     42
## 4   f    105    86     40
## 5   f    109    88     44
## 6   f    123    92     50
```

```
cor(turtles[,2:4])
```

```
##           length     width    height
## length 1.0000000 0.9783116 0.9646946
## width  0.9783116 1.0000000 0.9605705
## height 0.9646946 0.9605705 1.0000000
```

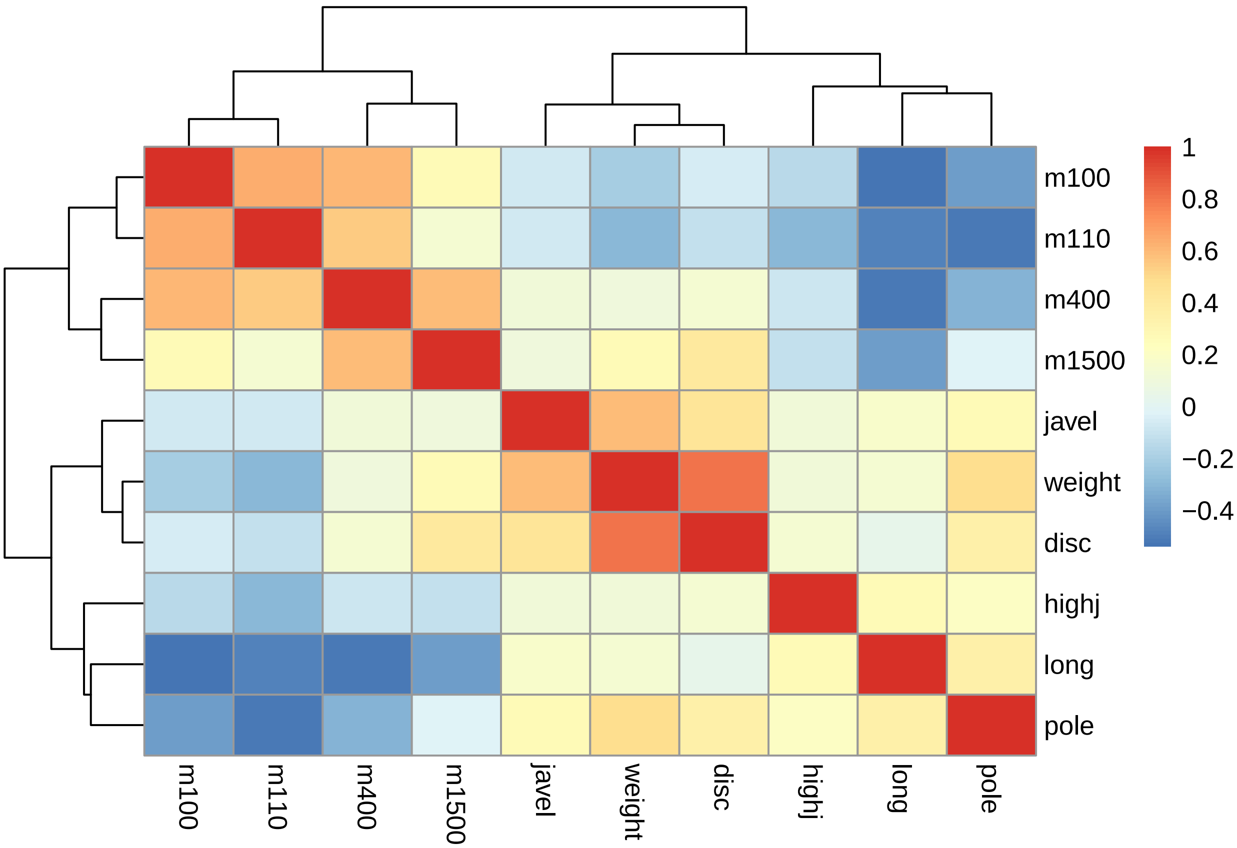
\textcolor{red}{Question 7.3)

```
ggpairs(turtles[,-1])
```



\textcolor{red}{Question 7.4}

```
pheatmap(cor(athletes), cell.width=10, cell.height =10)
```



Question 7.5Compute the means and standard deviations of the tutles data, then use the scale function to center and standarize the continuous variables. Make a scatterplot of the scaled and centered width and height variables and color the points by their sex

```
apply(turtles[,-1], 2, sd)
```

```
##    length     width    height
## 20.481602 12.675838  8.392837
```

```
apply(turtles[,-1], 2, mean)
```

```
##    length     width    height
## 124.68750  95.43750  46.33333
```

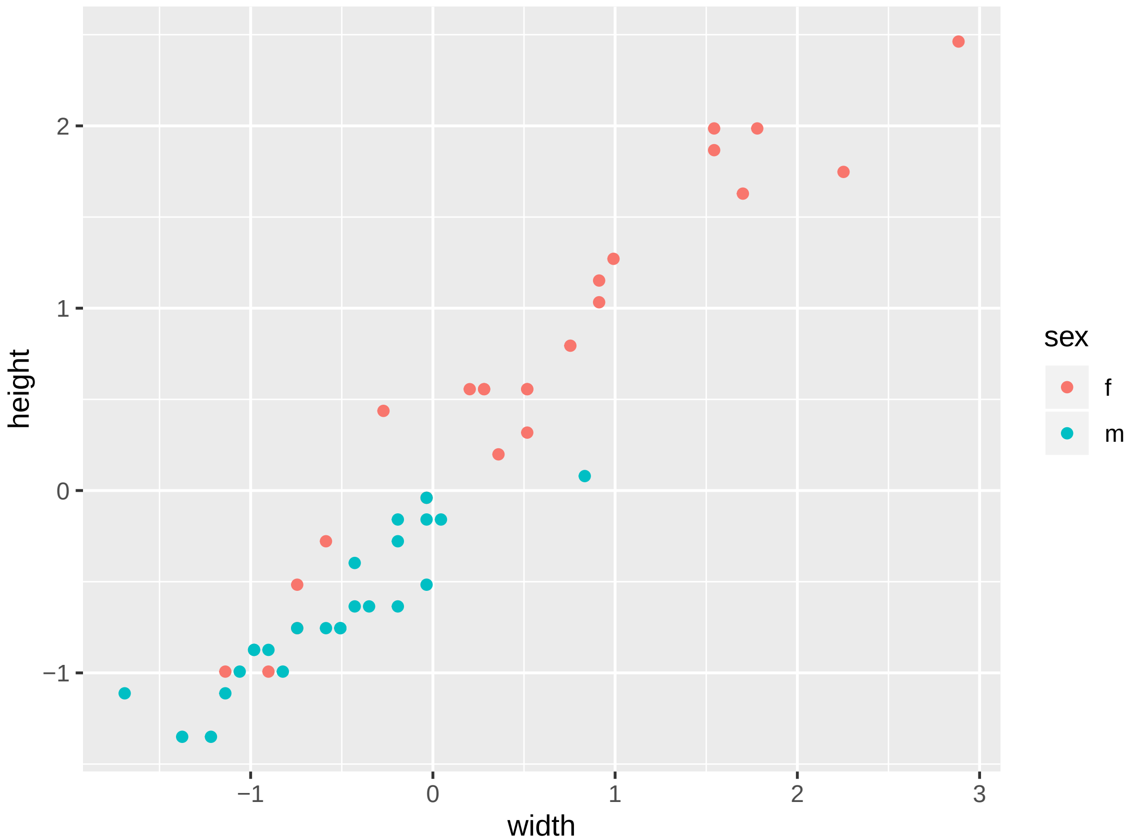
```
scaledTurtles <- scale(turtles[,-1])
apply(scaledTurtles, 2, mean)
```

```
##        length         width        height
## -1.432050e-18  1.940383e-17 -2.870967e-16
```

```
apply(scaledTurtles, 2, sd)
```

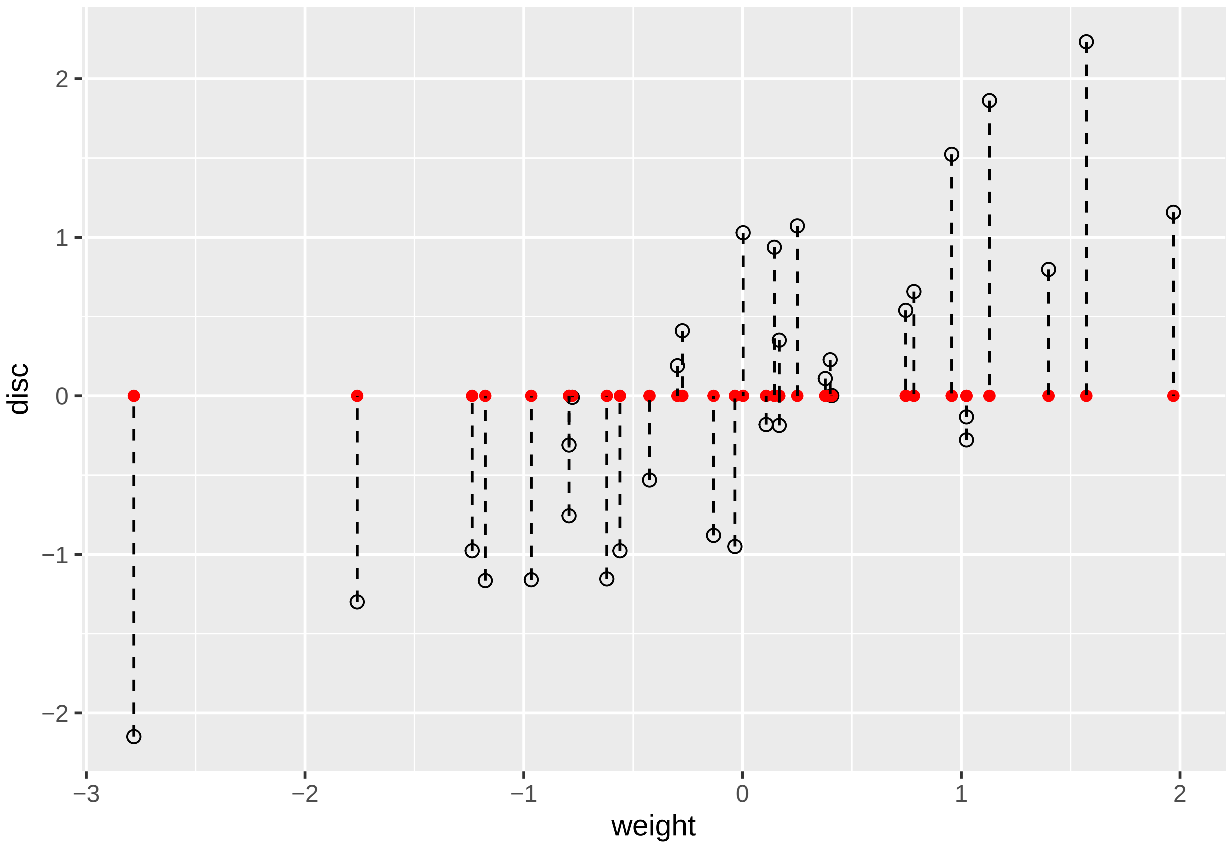
```
## length  width height
##      1      1      1
```

```
data.frame(scaledTurtles, sex = turtles[,1]) %>%
  ggplot(aes(x = width, y = height)) +
  geom_point(aes(color= sex)) +
  coord_fixed()
```



group =
sex in textbook unnecessary

```
athletes = data.frame(scale(athletes))
ath_gg = ggplot(athletes, aes(x = weight, y = disc)) +
  geom_point(size = 2, shape = 21)
ath_gg + geom_point(aes(y = 0), colour = "red") +
  geom_segment(aes(xend = weight, yend = 0), linetype = "dashed")
```



—-Reg1, fig.keep = 'high', fig.cap = "The blue line minimizes the sum

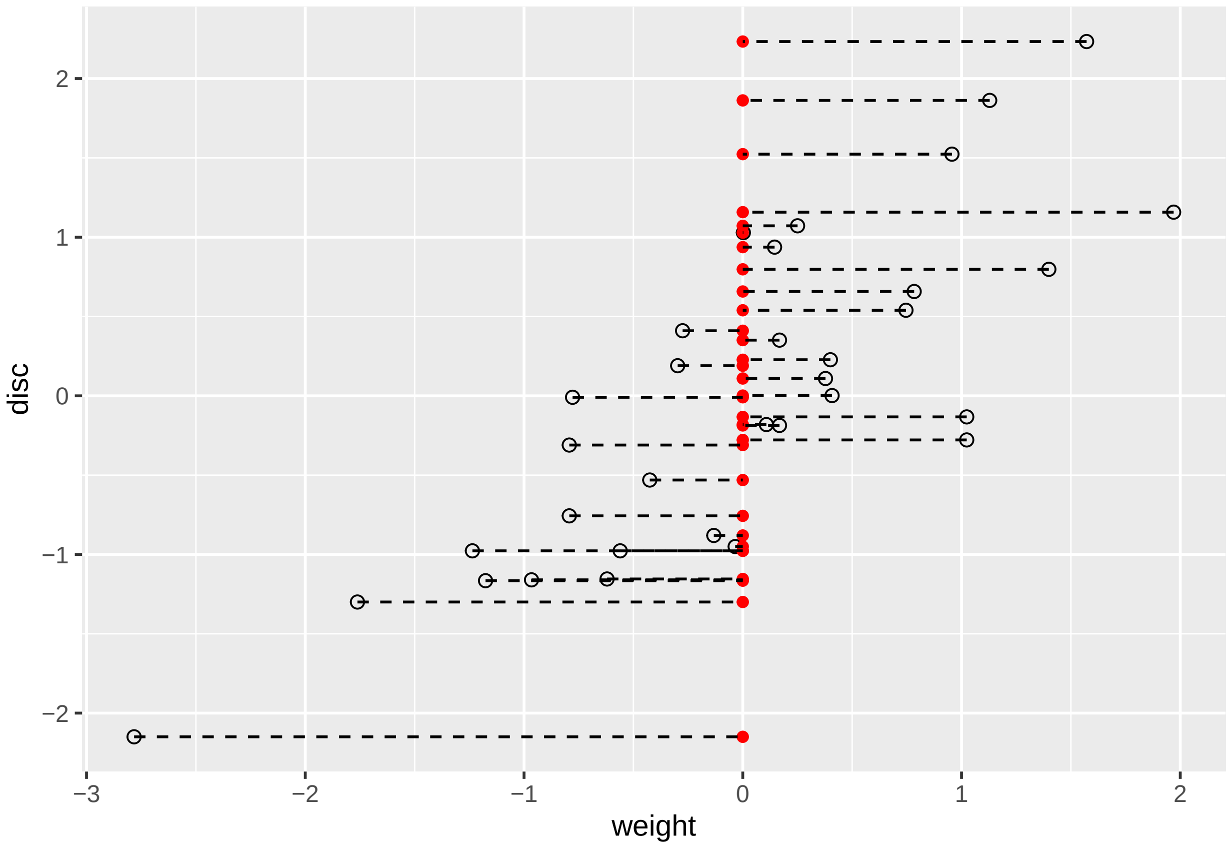
Task: Calculate the variance of the red points in Figure 7.6

```
var(athletes$weight)
```

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

Make a plot showing projection lines on the y-axis and projected points

```
ath_gg = ggplot(athletes, aes(x = weight, y = disc)) +
  geom_point(size = 2, shape = 21)
ath_gg + geom_point(aes(x = 0), colour = "red") +
  geom_segment(aes(yend = disc, xend = 0), linetype = "dashed")
```



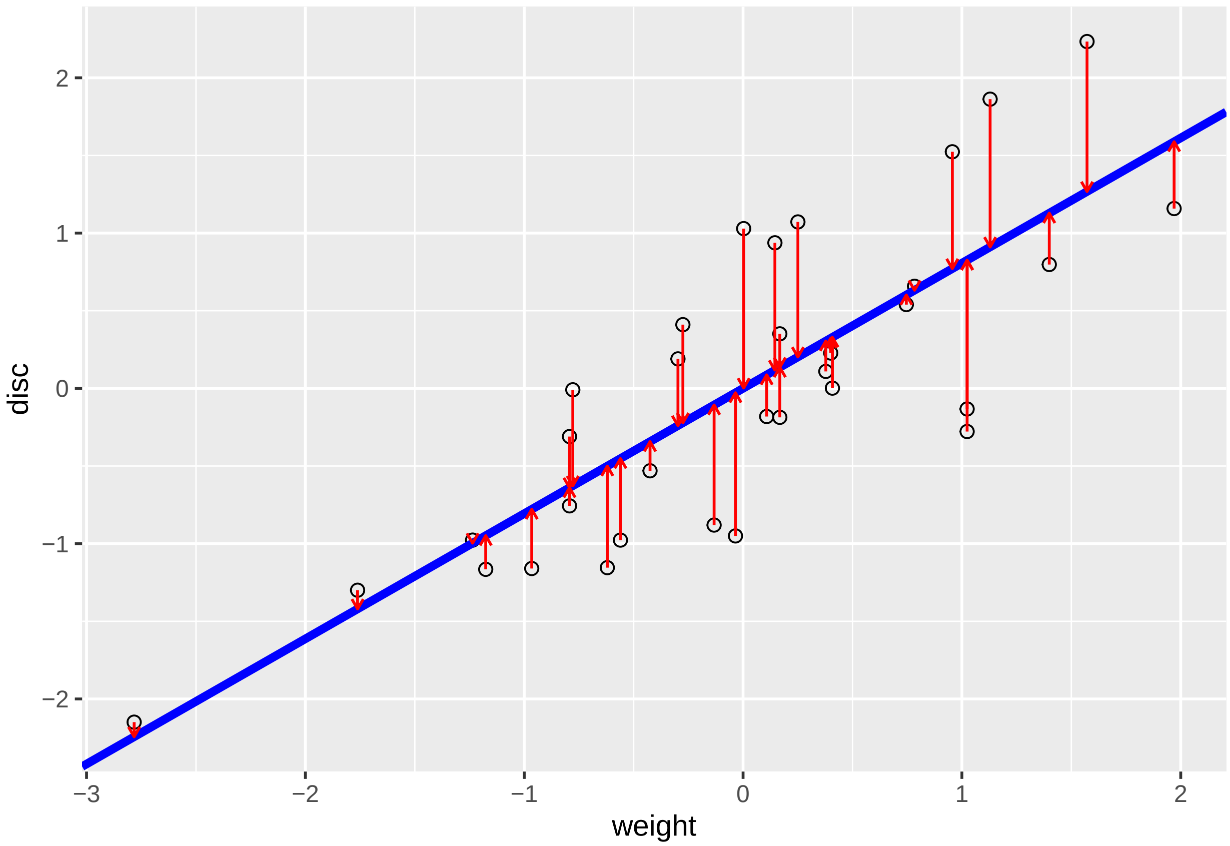
Compute the variance of the points projected onto the vertical y axis

```
var(athletes$disc)
```

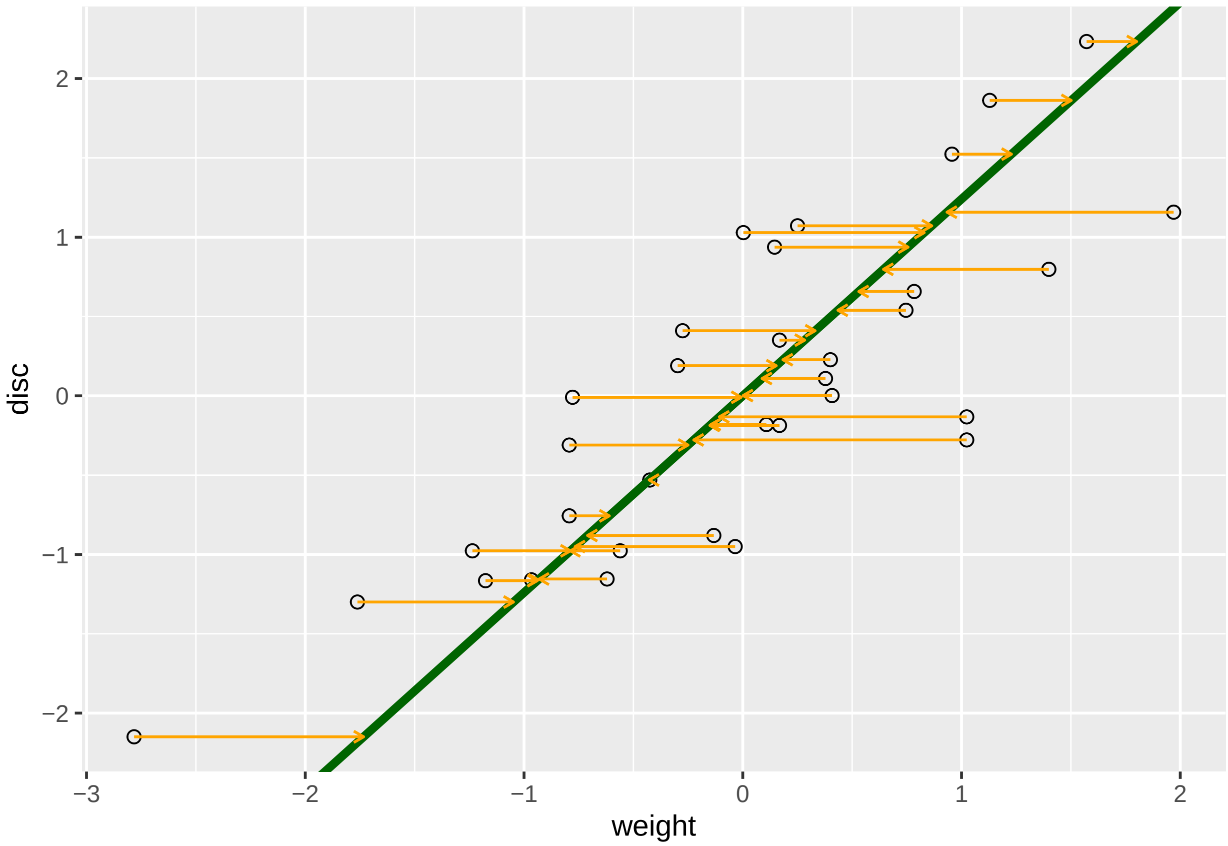
```
## [1] 1
```

7.3.2 How do we summarize two dimensional data by a line?

```
reg1 = lm(disc ~ weight, data = athletes)
a1 = reg1$coefficients[1] # intercept
b1 = reg1$coefficients[2] # slope
pline1 = ath_gg + geom_abline(intercept = a1, slope = b1,
                              col = "blue", lwd = 1.5)
pline1 + geom_segment(aes(xend = weight, yend = reg1$fitted),
                      colour = "red", arrow = arrow(length = unit(0.15, "cm")))
```




```
reg2 = lm(weight ~ disc, data = athletes)
a2 = reg2$coefficients[1] # intercept
b2 = reg2$coefficients[2] # slope
pline2 = ath_gg + geom_abline(intercept = -a2/b2, slope = 1/b2,
                              col = "darkgreen", lwd = 1.5)
pline2 + geom_segment(aes(xend=reg2$fitted, yend=disc),
                      colour = "orange", arrow = arrow(length = unit(0.15, "cm")))
```

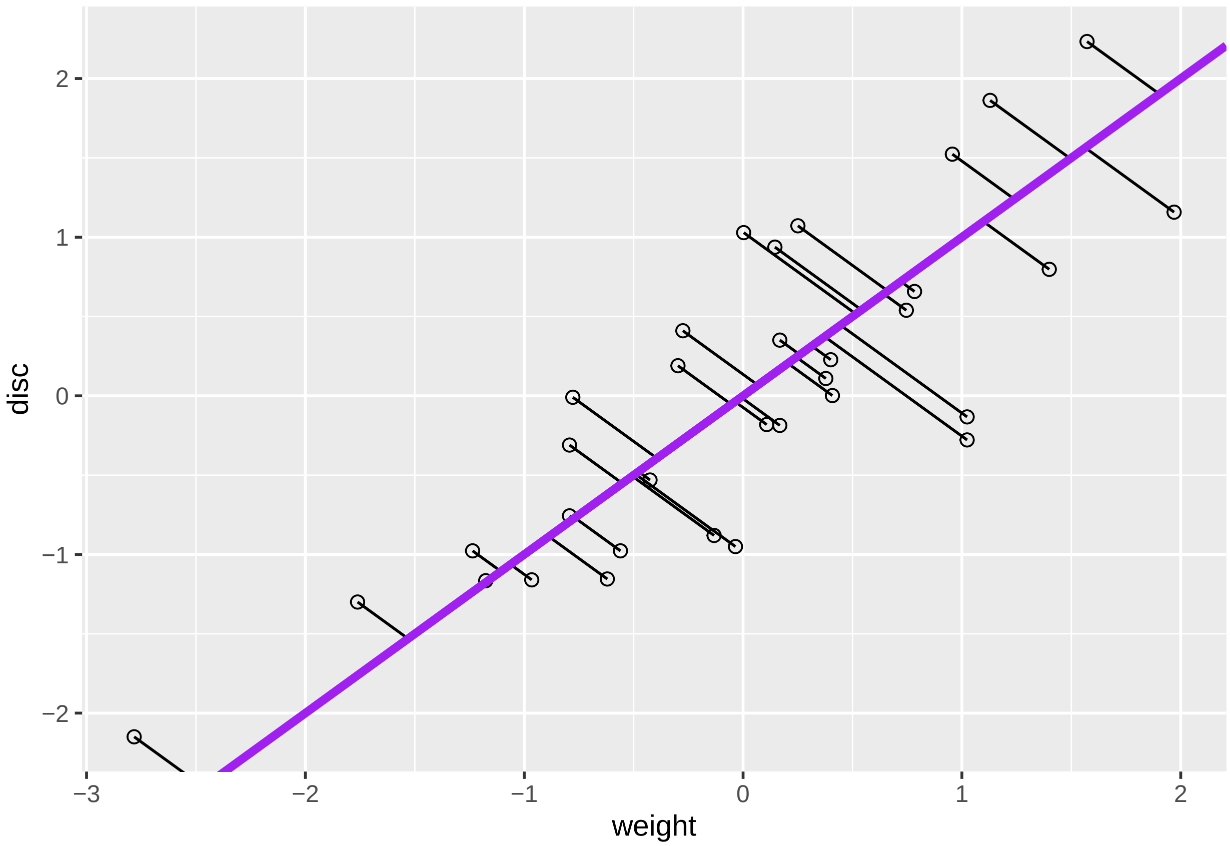


7.6 How large is the variance of the projected points that lie on the blue line of Figure 7.7? Pythagorus? But this is not squared?

```
var(athletes$weight) + var(reg1$fitted)
```

```
## [1] 1.650204
```

```
xy = cbind(athletes$disc, athletes$weight)
svda = svd(xy)
pc = xy %*% svda$v[, 1] %*% t(svda$v[, 1])
bp = svda$v[2, 1] / svda$v[1, 1]
ap = mean(pc[, 2]) - bp * mean(pc[, 1])
ath_gg + geom_segment(xend = pc[, 1], yend = pc[, 2]) +
  geom_abline(intercept = ap, slope = bp, col = "purple", lwd = 1.5)
```



7.6 The inner workings of PCA

```
.savedopt = options(digits = 3)
X = matrix(c(780,  75, 540,
             936,  90, 648,
             1300, 125, 900,
             728,  70, 504), nrow = 3)
u = c(0.8196, 0.0788, 0.5674)
v = c(0.4053, 0.4863, 0.6754, 0.3782)
s1 = 2348.2
sum(u^2)
```

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

```
sum(v^2)
```

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

```
s1 * u %*% t(v)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  780  936 1300  728
## [2,]   75   90  125   70
## [3,]  540  648  900  504
```

```
X - s1 * u %*% t(v)
```

```
##          [,1]   [,2]   [,3]   [,4]
## [1,] -0.03419 0.0745 0.1355 0.1221
## [2,]  0.00403 0.0159 0.0252 0.0186
## [3,] -0.00903 0.0691 0.1182 0.0982
```

```r
options(.savedopt)
```

```r
svd(X)
```

```
## $d
## [1] 2.348244e+03 2.141733e-13 6.912584e-15
##
## $u
##             [,1]         [,2]        [,3]
## [1,] 0.81963482  0.569413084  0.06298807
## [2,] 0.07881104 -0.003168944 -0.99688454
## [3,] 0.56743949 -0.822045435  0.04747341
##
## $v
##           [,1]        [,2]       [,3]
## [1,] 0.4052574  0.88432390 -0.1978361
## [2,] 0.4863089 -0.13032307  0.7123009
## [3,] 0.6754290 -0.44787634 -0.5829493
## [4,] 0.3782403  0.01984752  0.3371327
```

```r
svd(X)$u[, 1]
```

```
## [1] 0.81963482 0.07881104 0.56743949
```

```r
svd(X)$v[, 1]
```

```
## [1] 0.4052574 0.4863089 0.6754290 0.3782403
```

```r
sum(svd(X)$u[, 1]^2)
```

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

```r
sum(svd(X)$v[, 1]^2)
```

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

```r
svd(X)$d
```

```
## [1] 2.348244e+03 2.141733e-13 6.912584e-15
```

```r
Xtwo = matrix(c(12.5, 35.0, 25.0, 25, 9, 14, 26, 18, 16, 21, 49, 32,
                18, 28, 52, 36, 18, 10.5, 64.5, 36), ncol = 4, byrow = TRUE)
USV = svd(Xtwo)
```

```r
names(USV)
```

```
## [1] "d" "u" "v"
```

```r
USV$d
```

```
## [1] 1.350624e+02 2.805191e+01 3.103005e-15 1.849559e-15
```

```r
Xtwo - USV$d[1] * USV$u[, 1] %*% t(USV$v[, 1])
```

```
##             [,1]       [,2]        [,3]        [,4]
## [1,]  0.87481760  19.045230 -10.1044650  1.74963521
## [2,]  0.08079747   1.759002  -0.9332405  0.16159494
## [3,] -0.04700978  -1.023427   0.5429803 -0.09401956
## [4,]  0.16159494   3.518005  -1.8664809  0.32318987
## [5,] -0.69632883 -15.159437   8.0428540 -1.39265765
```

```r
Xtwo - USV$d[1] * USV$u[, 1] %*% t(USV$v[, 1]) -
  USV$d[2] * USV$u[, 2] %*% t(USV$v[, 2])
```

```
##              [,1]          [,2]         [,3]         [,4]
## [1,] 7.216450e-15 -1.065814e-14 8.881784e-15 4.884981e-15
## [2,] 2.040035e-15 -5.995204e-15 1.054712e-14 3.219647e-15
## [3,] 2.865763e-15 -9.547918e-15 1.554312e-15 6.231127e-15
## [4,] 4.385381e-15 -5.773160e-15 1.776357e-14 7.049916e-15
## [5,] 5.107026e-15 -1.776357e-15 1.776357e-14 1.776357e-14
```

```r
stopifnot(max(abs(
  Xtwo - USV$d[1] * USV$u[, 1] %*% t(USV$v[, 1]) -
    USV$d[2] * USV$u[, 2] %*% t(USV$v[, 2]))) < 1e-12,
  max(abs(USV$d[3:4])) < 1e-13)
```

```r
t(USV$u) %*% USV$u
```

```
##               [,1]          [,2]          [,3]          [,4]
## [1,]  1.000000e+00 -1.665335e-16  0.000000e+00 -8.326673e-17
## [2,] -1.665335e-16  1.000000e+00  1.665335e-16 -5.551115e-17
## [3,]  0.000000e+00  1.665335e-16  1.000000e+00 -5.551115e-17
## [4,] -8.326673e-17 -5.551115e-17 -5.551115e-17  1.000000e+00
```

```r
t(USV$v) %*% USV$v
```

```
##               [,1]          [,2]          [,3]          [,4]
## [1,]  1.000000e+00  8.326673e-17  1.387779e-17 -5.551115e-17
## [2,]  8.326673e-17  1.000000e+00 -3.642919e-17 -6.938894e-17
## [3,]  1.387779e-17 -3.642919e-17  1.000000e+00  2.775558e-17
## [4,] -5.551115e-17 -6.938894e-17  2.775558e-17  1.000000e+00
```

```r
turtles.svd = svd(scaledTurtles)
turtles.svd$d
```

```
## [1] 11.746475  1.419035  1.003329
```

```r
turtles.svd$v
```

```
##           [,1]       [,2]        [,3]
## [1,] 0.5787981 -0.3250273 -0.74789704
## [2,] 0.5779840 -0.4834699  0.65741263
## [3,] 0.5752628  0.8127817  0.09197088
```

```r
dim(turtles.svd$u)
```

```
## [1] 48  3
```

```r
sum(turtles.svd$v[,1]^2)
```

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

```r
sum(turtles.svd$d^2) / 47
```

```
## [1] 3
```

7.18 Compute the first principal component for the turtles data

```r
turtles.svd$d[1] %*% turtles.svd$u[,1]
```

```
##           [,1]      [,2]      [,3]      [,4]       [,5]       [,6]        [,7]
## [1,] -1.983668 -1.705579 -1.340216 -1.420782 -0.9423811 0.04689258 -0.09048395
```

```
##         [,8]      [,9]     [,10]     [,11]     [,12]     [,13]    [,14]
## [1,] 0.71721 0.8540019 0.8540019 0.5854404 0.8016959 0.7846503 0.858507
##         [,15]    [,16]    [,17]    [,18]    [,19]    [,20]    [,21]    [,22]
## [1,] 1.353953 1.93447 1.808307 1.989887 2.890979 2.776548 2.907215 3.140809
##         [,23]    [,24]     [,25]     [,26]     [,27]     [,28]     [,29]
## [1,] 3.362087 4.562009 -2.512689 -2.439124 -2.291411 -1.693556 -1.688242
##          [,30]     [,31]     [,32]     [,33]     [,34]     [,35]     [,36]
## [1,] -1.910913 -1.654375 -1.597856 -1.683736 -1.086174 -1.103512 -1.166447
##           [,37]      [,38]      [,39]      [,40]      [,41]     [,42]
## [1,] -0.7219127 -0.8307375 -0.7851402 -0.6374268 -0.8600987 -0.403541
##           [,43]      [,44]         [,45]       [,46]     [,47]     [,48]
## [1,] -0.4211712 -0.1937018 -0.0003910952 -0.01772898 0.1355914 0.8187415
```

```
scaledTurtles %*% turtles.svd$v[,1]
```

```
##               [,1]
##  [1,] -1.9836684602
##  [2,] -1.7055794726
##  [3,] -1.3402164350
##  [4,] -1.4207818191
##  [5,] -0.9423811150
##  [6,]  0.0468925757
##  [7,] -0.0904839545
##  [8,]  0.7172099610
##  [9,]  0.8540018654
## [10,]  0.8540018654
## [11,]  0.5854403531
## [12,]  0.8016958980
## [13,]  0.7846503260
## [14,]  0.8585070441
## [15,]  1.3539533202
## [16,]  1.9344701590
## [17,]  1.8083074735
## [18,]  1.9898872486
## [19,]  2.8909792543
## [20,]  2.7765475313
## [21,]  2.9072153955
## [22,]  3.1408088253
## [23,]  3.3620866667
## [24,]  4.5620089799
## [25,] -2.5126887623
## [26,] -2.4391243571
## [27,] -2.2914109209
## [28,] -1.6935561972
## [29,] -1.6882415877
## [30,] -1.9109134857
## [31,] -1.6543752488
## [32,] -1.5978564155
## [33,] -1.6837364090
## [34,] -1.0861739982
## [35,] -1.1035118831
## [36,] -1.1664470694
## [37,] -0.7219127043
## [38,] -0.8307375049
## [39,] -0.7851402035
```

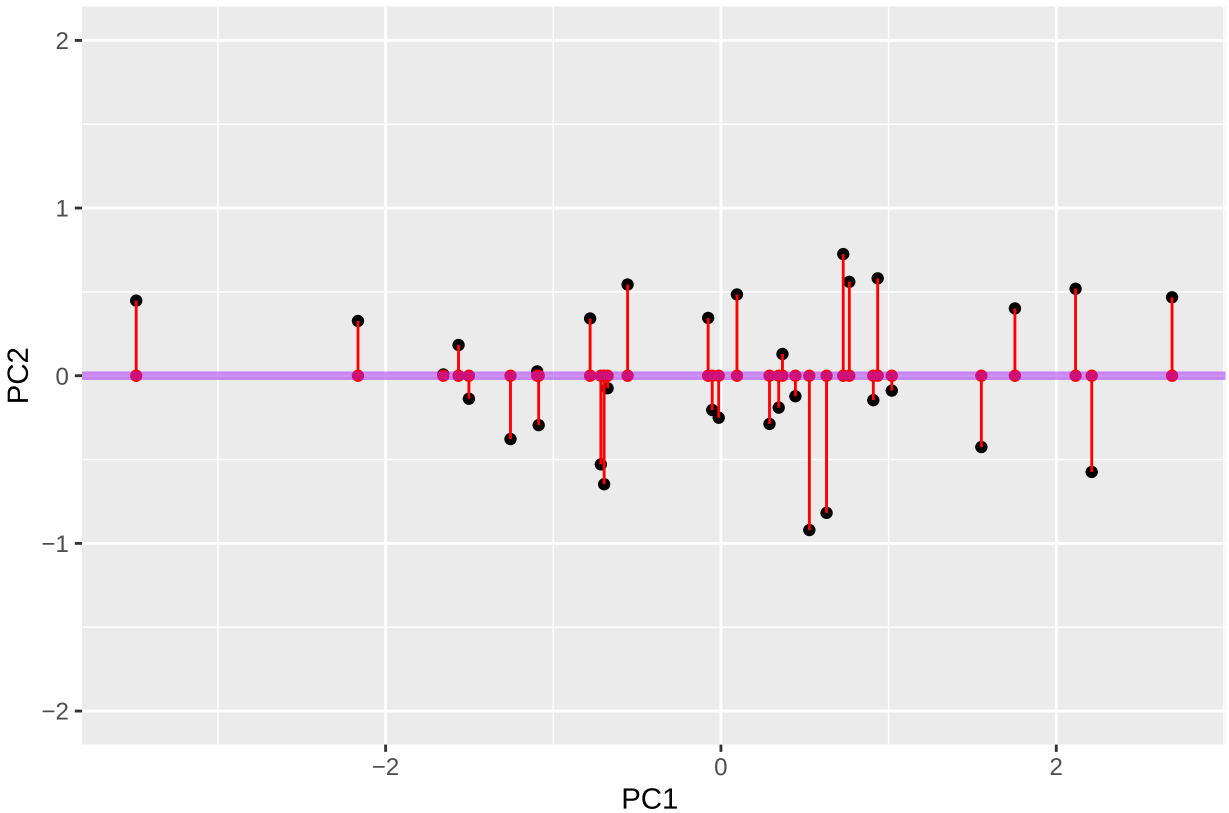
```
## [40,] -0.6374267672
## [41,] -0.8600986652
## [42,] -0.4035410247
## [43,] -0.4211712224
## [44,] -0.1937018329
## [45,] -0.0003910952
## [46,] -0.0177289800
## [47,]  0.1355913785
## [48,]  0.8187414700
```

7.19 What part of the output of the svd functions leads us to the first PC coefficients, also known as PC loadings?

```r
svda$v[,1]
```

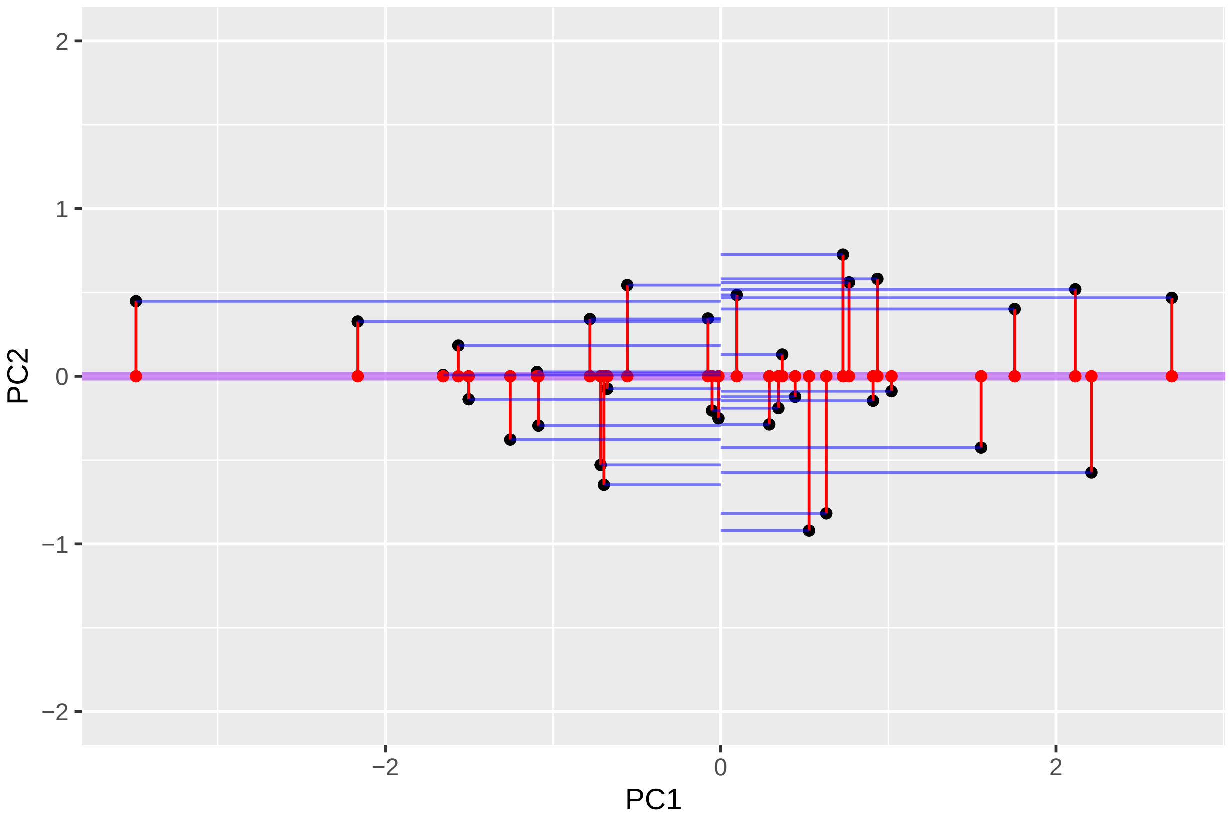
```
## [1] -0.7071068 -0.7071068
```

```r
ppdf = tibble(PC1n = -svda$u[, 1] * svda$d[1],
              PC2n = svda$u[, 2] * svda$d[2])
ggplot(ppdf, aes(x = PC1n, y = PC2n)) + geom_point() + xlab("PC1 ")+
  ylab("PC2") + geom_point(aes(x=PC1n,y=0),color="red") +
  geom_segment(aes(xend = PC1n, yend = 0), color = "red") +
  geom_hline(yintercept = 0, color = "purple", lwd=1.5, alpha=0.5) +
  xlim(-3.5, 2.7) + ylim(-2,2) + coord_fixed()
```




```r
segm = tibble(xmin = pmin(ppdf$PC1n, 0), xmax = pmax(ppdf$PC1n, 0), yp = seq(-1, -2, length = nrow(ppdf)
ggplot(ppdf, aes(x = PC1n, y = PC2n)) + geom_point() + ylab("PC2") + xlab("PC1") +
  geom_hline(yintercept=0,color="purple",lwd=1.5,alpha=0.5) +
  geom_point(aes(x=PC1n,y=0),color="red")+
  xlim(-3.5, 2.7)+ylim(-2,2)+coord_fixed() +
```

```r
  geom_segment(aes(xend=PC1n,yend=0), color="red")+
  geom_segment(data=segm,aes(x=xmin,xend=xmax,y=yo,yend=yo), color="blue",alpha=0.5)
```



7.20:

```r
svda$d[2]^2
```

```
## [1] 6.196729
```