Prolem Set 2: PCA

Katia Williams

2/13/2018

Exploratory Phase

```
dec <- read.csv('decathlon.csv')
head(dec)
```

```
##     athlete X100m long_jump shot_put high_jump X400m X110m_hurdle discus
## 1    Sebrle 10.85      7.84    16.36      2.12 48.36        14.05  48.72
## 2      Clay 10.44      7.96    15.23      2.06 49.19        14.13  50.11
## 3    Karpov 10.50      7.81    15.93      2.09 46.81        13.97  51.65
## 4     Macey 10.89      7.47    15.73      2.15 48.97        14.56  48.34
## 5   Warners 10.62      7.74    14.48      1.97 47.97        14.01  43.73
## 6 Zsivoczky 10.91      7.14    15.31      2.12 49.40        14.95  45.62
##   pole_vault javeline X1500m rank points competition
## 1        5.0    70.52 280.01    1   8893     Olympic
## 2        4.9    69.71 282.00    2   8820     Olympic
## 3        4.6    55.54 278.11    3   8725     Olympic
## 4        4.4    58.46 265.42    4   8414     Olympic
## 5        4.9    55.39 278.05    5   8343     Olympic
## 6        4.7    63.45 269.54    6   8287     Olympic
```

```
summary(dec)
```

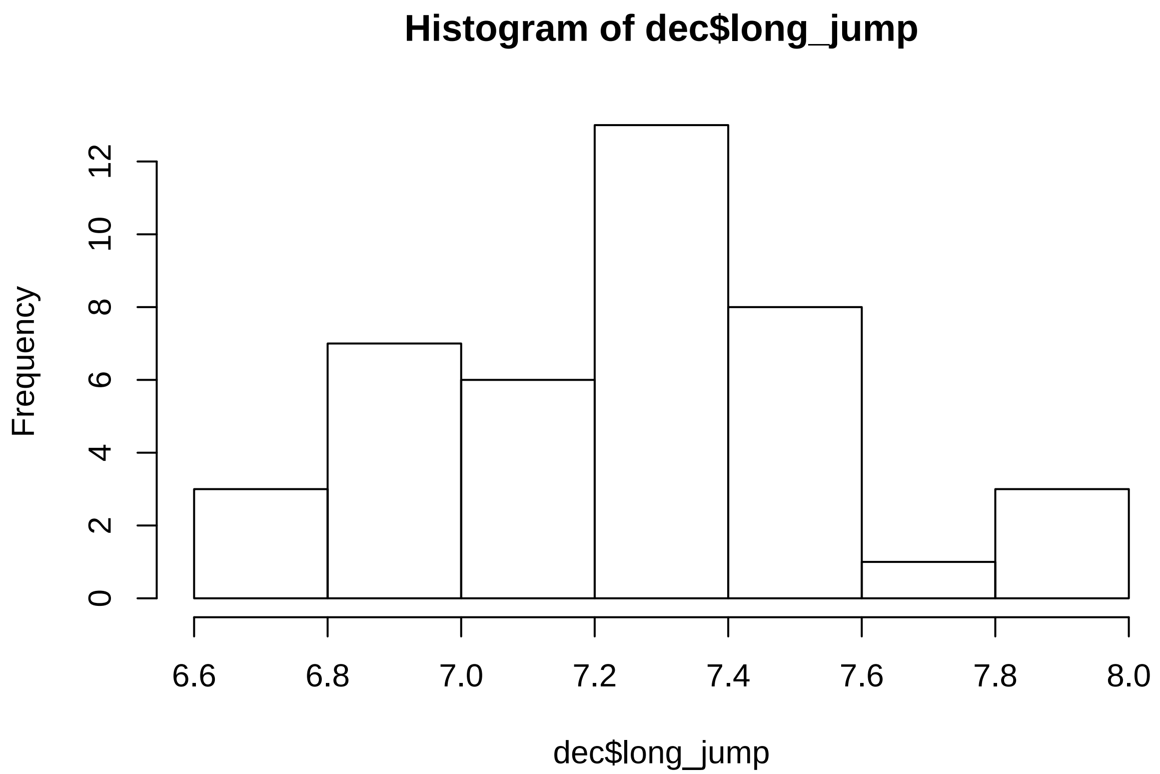
```
##         athlete       X100m         long_jump       shot_put
##  Averyanov  : 1   Min.   :10.44   Min.   :6.61   Min.   :12.68
##  Barras     : 1   1st Qu.:10.85   1st Qu.:7.03   1st Qu.:13.88
##  BARRAS     : 1   Median :10.98   Median :7.30   Median :14.57
##  Bernard    : 1   Mean   :11.00   Mean   :7.26   Mean   :14.48
##  BERNARD    : 1   3rd Qu.:11.14   3rd Qu.:7.48   3rd Qu.:14.97
##  BOURGUIGNON: 1   Max.   :11.64   Max.   :7.96   Max.   :16.36
##  (Other)    :35
##    high_jump        X400m        X110m_hurdle       discus
##  Min.   :1.850   Min.   :46.81   Min.   :13.97   Min.   :37.92
##  1st Qu.:1.920   1st Qu.:48.93   1st Qu.:14.21   1st Qu.:41.90
##  Median :1.950   Median :49.40   Median :14.48   Median :44.41
##  Mean   :1.977   Mean   :49.62   Mean   :14.61   Mean   :44.33
##  3rd Qu.:2.040   3rd Qu.:50.30   3rd Qu.:14.98   3rd Qu.:46.07
##  Max.   :2.150   Max.   :53.20   Max.   :15.67   Max.   :51.65
##
##    pole_vault       javeline         X1500m           rank
##  Min.   :4.200   Min.   :50.31   Min.   :262.1   Min.   : 1.00
##  1st Qu.:4.500   1st Qu.:55.27   1st Qu.:271.0   1st Qu.: 6.00
##  Median :4.800   Median :58.36   Median :278.1   Median :11.00
##  Mean   :4.762   Mean   :58.32   Mean   :279.0   Mean   :12.12
##  3rd Qu.:4.920   3rd Qu.:60.89   3rd Qu.:285.1   3rd Qu.:18.00
##  Max.   :5.400   Max.   :70.52   Max.   :317.0   Max.   :28.00
##
##      points       competition
```

```
##  Min.   :7313   Decastar:13
##  1st Qu.:7802   Olympic :28
##  Median :8021
##  Mean   :8005
##  3rd Qu.:8122
##  Max.   :8893
##
```

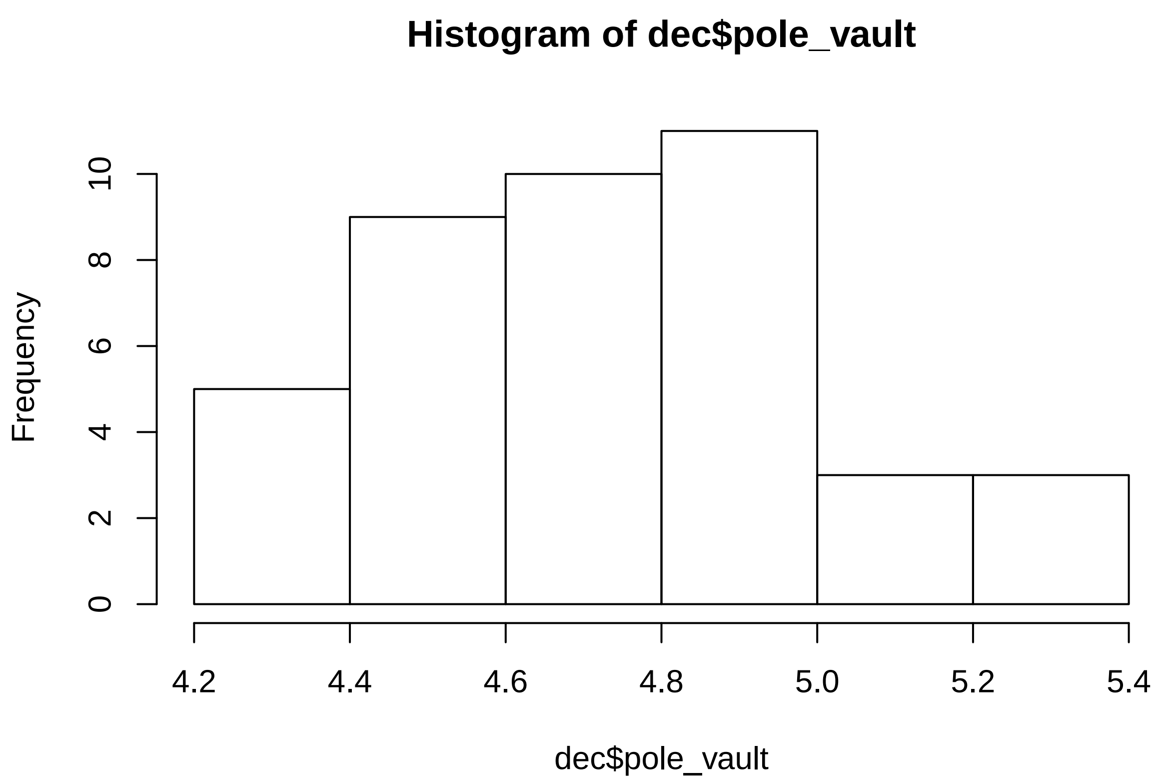
```r
pairs(dec)
```

Some interesting possible correlations: *The 400m and Long Jump* Points and the 100m seem negatively correlated (?) *Shot put and discuss look related (makes more sense, they're pretty similar)

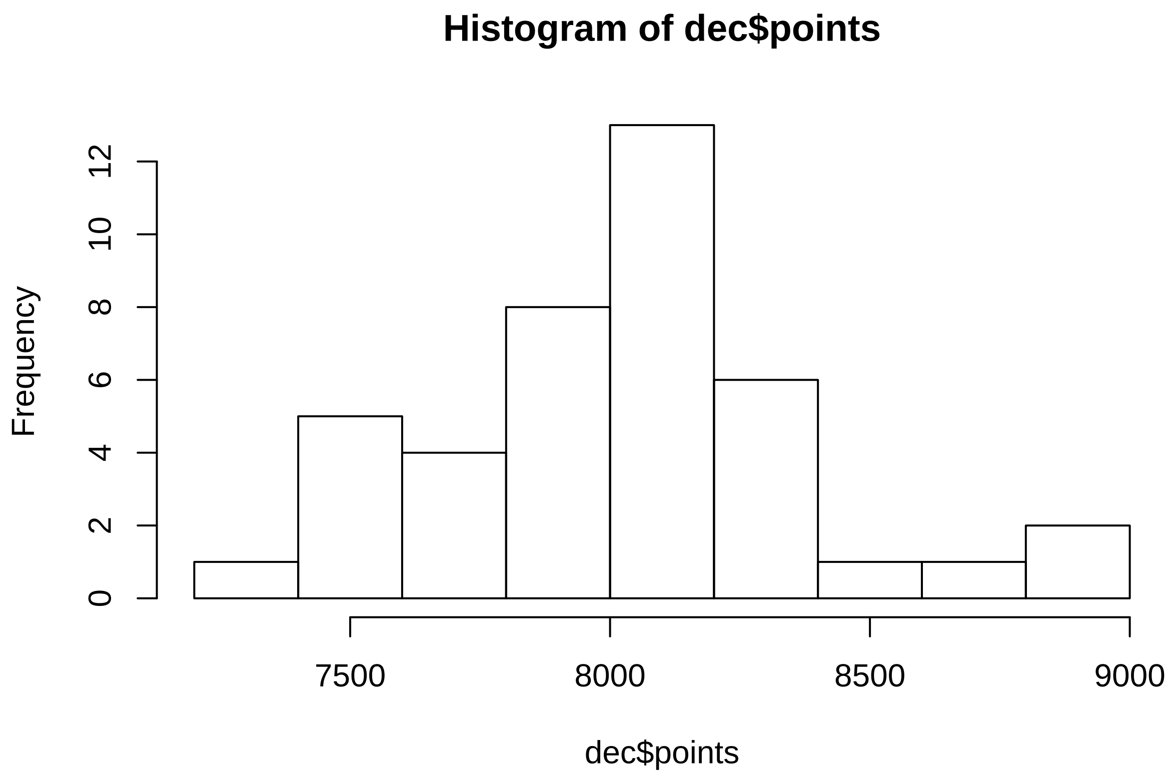
```r
hist(dec$long_jump)
```




```
hist(dec$pole_vault)
```

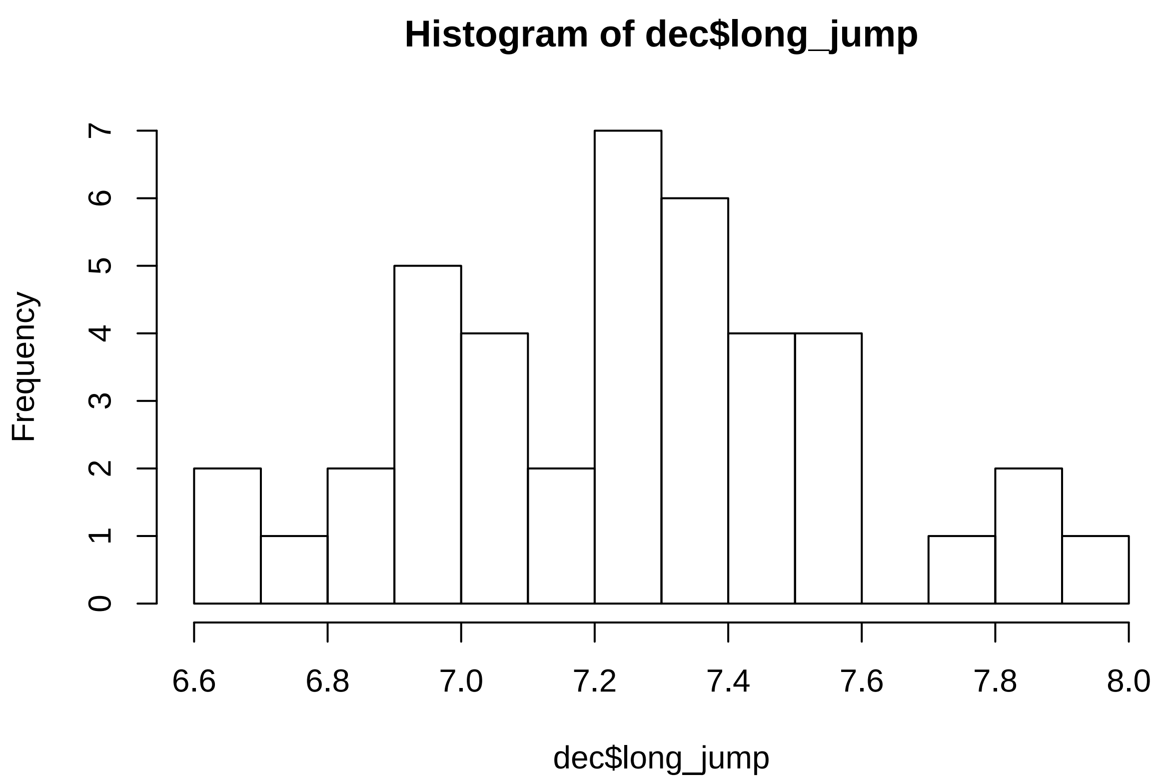



```
hist(dec$points)
```



Long Jump: Do less people get between 7.6 and 7.8 than between 7.8 and 8.0? Or is that just how its binned?

```
hist(dec$long_jump, 10)
```



I guess it's just because no one got 7.7 so it's dragging the histogram down.

1) Calculation of primary PCA outputs

a) Loadings

Initial Data Wrangling:

```
acdec_unscaled <- dec[dec$competition == "Olympic",]
supdec_unscaled <- dec[dec$competition != "Olympic",]
acathletes <- acdec_unscaled$athlete
supathletes <- supdec_unscaled$athlete
acdec <- scale(acdec_unscaled[,c(2:11)])[,c(1:10)]
supdec <- scale(supdec_unscaled[,c(2:4)])[,c(1:3)]
head(acdec)
```

```
##         X100m  long_jump   shot_put  high_jump      X400m X110m_hurdle
## 1 -0.28444399  1.6834526  2.0263454  1.5961767 -0.9854509  -1.13751179
## 2 -2.05912717  2.0352188  0.7065931  0.9291178 -0.3311115  -0.95680070
## 3 -1.79941744  1.5955110  1.5241388  1.2626473 -2.2074100  -1.31822289
## 4 -0.11130417  0.5988401  1.2905543  1.9297062 -0.5045509   0.01452143
## 5 -1.27999797  1.3903141 -0.1693488 -0.0714706 -1.2929116  -1.22786734
## 6 -0.02473426 -0.3685170  0.8000269  1.5961767 -0.1655558   0.89548801
##       discus pole_vault    javeline      X1500m
## 1  1.3166153  0.9256099  2.32544650  0.21721634
## 2  1.7378802  0.5800488  2.16266022  0.39298303
## 3  2.2046054 -0.4566342 -0.68509481  0.04939889
## 4  1.2014493 -1.1477562 -0.09826032 -1.07144499
## 5 -0.1956955  0.5800488 -0.71524042  0.04409940
## 6  0.3771036 -0.1110732  0.90458354 -0.70754611
```

```
n <- length(acdec[,1])
X <- acdec
```

```
get_S <- function(X,n) {
  return((1/(n-1)) * t(X) %*% X)
}

S <- get_S(X, n)

get_loadings <- function(S){
  return(eigen(S)$vectors)
}


loadings <- get_loadings(S)

get_lambdas <- function(S) {
  return(eigen(S)$values)
}
lambdas <- get_lambdas(S)
colnames(loadings) <- paste0( "V",(1:10))
loadings[,1:4]
```

```
##               V1         V2          V3           V4
##  [1,]  0.42270533  0.1806841  0.21199128  0.075009372
##  [2,] -0.42146649 -0.2315408 -0.13017356 -0.006144987
```

```
##  [3,] -0.33407359  0.4437320 -0.01889119  0.140442615
##  [4,] -0.33249211  0.3362530  0.01083254 -0.111008069
##  [5,]  0.38995573  0.3524322 -0.19266472  0.116944533
##  [6,]  0.37654258  0.1655859  0.03684219  0.115374735
##  [7,] -0.28793579  0.4754243 -0.01497490 -0.206205419
##  [8,] -0.09539301 -0.2324861 -0.52373161  0.643167759
##  [9,] -0.15213083  0.2415176  0.43702142  0.689806306
## [10,]  0.11193576  0.3372567 -0.65852601 -0.057300779
```

b)

```r
PCs <- X %*% loadings
colnames(PCs) <- paste0('PC', c(1:10))
rownames(PCs) <- acathletes
PCs[, c(1:4)]
```

```
##                     PC1        PC2        PC3         PC4
## Sebrle      -3.64687853  1.5046838  0.2162631  1.74472299
## Clay        -3.60330295  0.8537756 -0.3196647  1.16403050
## Karpov      -4.20070330  0.4155663 -0.3533370 -1.70482900
## Macey       -1.90491357  1.3402994  1.2384762 -1.09465304
## Warners     -1.89845545 -1.6971105 -0.8885198 -0.49575117
## Zsivoczky   -0.69529257  1.2474627  1.0235787  0.53486534
## Hernu       -0.69023057 -0.5068215  0.7320204 -0.17198585
## Nool        -0.17778111 -1.7420595 -0.9865815  1.97322479
## Bernard     -1.57350593  0.1338441  0.1139975 -1.63517179
## Schwarzl     0.09249421 -1.4510863 -0.7211613  0.49054597
## Pogorelov   -0.25580109  0.6051491 -1.7486821 -0.22767233
## Schoenbeck   0.12114605 -0.2872966 -0.5531648  1.00087299
## Barras       0.28609389  0.3817102  1.6529060  0.61055310
## Smith       -0.47451303  1.1087005  1.5460578 -1.09545064
## Averyanov   -0.21829441 -1.7113985 -0.5069687 -0.28850069
## Ojaniemi    -0.11595075 -0.7797977  0.1865277  0.10490637
## Smirnov      0.62752609 -1.0467549  1.2218190  0.31568082
## Qi           0.72606940 -0.1849499  1.0043829 -0.34313787
## Drews        0.41555968 -3.0780560 -0.8637160 -0.54571271
## Parkhomenko  1.31164623  1.8259536  0.7181978  1.66622181
## Terek        0.89200732  0.2586709 -2.3429373  0.24618999
## Gomez        0.64388302 -1.0754572  1.5068250 -0.16939887
## Turi         1.80329186  0.1925375 -0.8147291  0.36824998
## Lorenzo      2.57797811 -1.5344745  1.6135571 -0.08449894
## Karlivans    2.31672132 -0.1869460  0.1352429 -1.17059258
## Korkizoglou  1.45766664  1.7903823 -2.9486653 -0.88518819
## Uldal        2.88307554  0.1301175  0.5209506  0.04088944
## Casarsa      3.30046389  3.4933557 -0.3826751 -0.34841043
```

c) Eigenvalues

```r
lambdas
```

```
##  [1] 3.5446573 1.9699560 1.4217248 0.9034912 0.5636320 0.5282270 0.4328613
##  [8] 0.3658102 0.1634956 0.1061447
```

```r
sum(lambdas)
```

```
## [1] 10
```

2) Choosing the number of dimensions to retain/examine

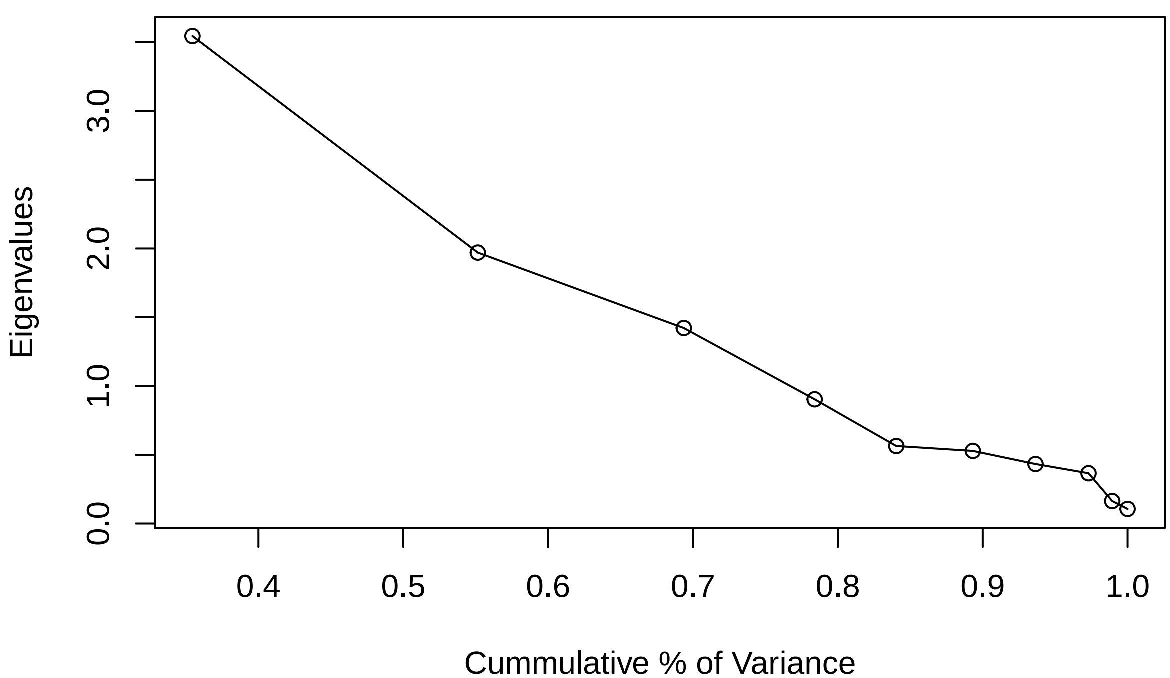
a)

```r
p <- 10
eigen_summary <- as.data.frame(cbind(lambdas, lambdas/p, cumsum(lambdas/p)))
colnames(eigen_summary) <- c('Eigenvalue', '% of Var', 'Cummulative % of Var')
eigen_summary
```

```
##    Eigenvalue   % of Var Cummulative % of Var
## 1   3.5446573 0.35446573            0.3544657
## 2   1.9699560 0.19699560            0.5514613
## 3   1.4217248 0.14217248            0.6936338
## 4   0.9034912 0.09034912            0.7839829
## 5   0.5636320 0.05636320            0.8403461
## 6   0.5282270 0.05282270            0.8931688
## 7   0.4328613 0.04328613            0.9364550
## 8   0.3658102 0.03658102            0.9730360
## 9   0.1634956 0.01634956            0.9893855
## 10  0.1061447 0.01061447            1.0000000
```

b)

```r
plot(eigen_summary$`Cummulative % of Var`, lambdas, type='o', xlab = "Cummulative % of Variance", ylab=
```



I see two “elbows” : at the second lambda, and the fifth lambda. This means that those are good dividing points between the lambdas that capture the “bulk” of the total variance, and the lambdas that don’t contribute as much.

c)

I would probably keep the first five PCs/dimensions. This is according to Cattel’s rule, and from my examination of the graph above

3) Studying the cloud of individuals

a)

```
project <- function(x,v) {
  return((x %*% v)/(t(v) %*% v)*v)
}
```

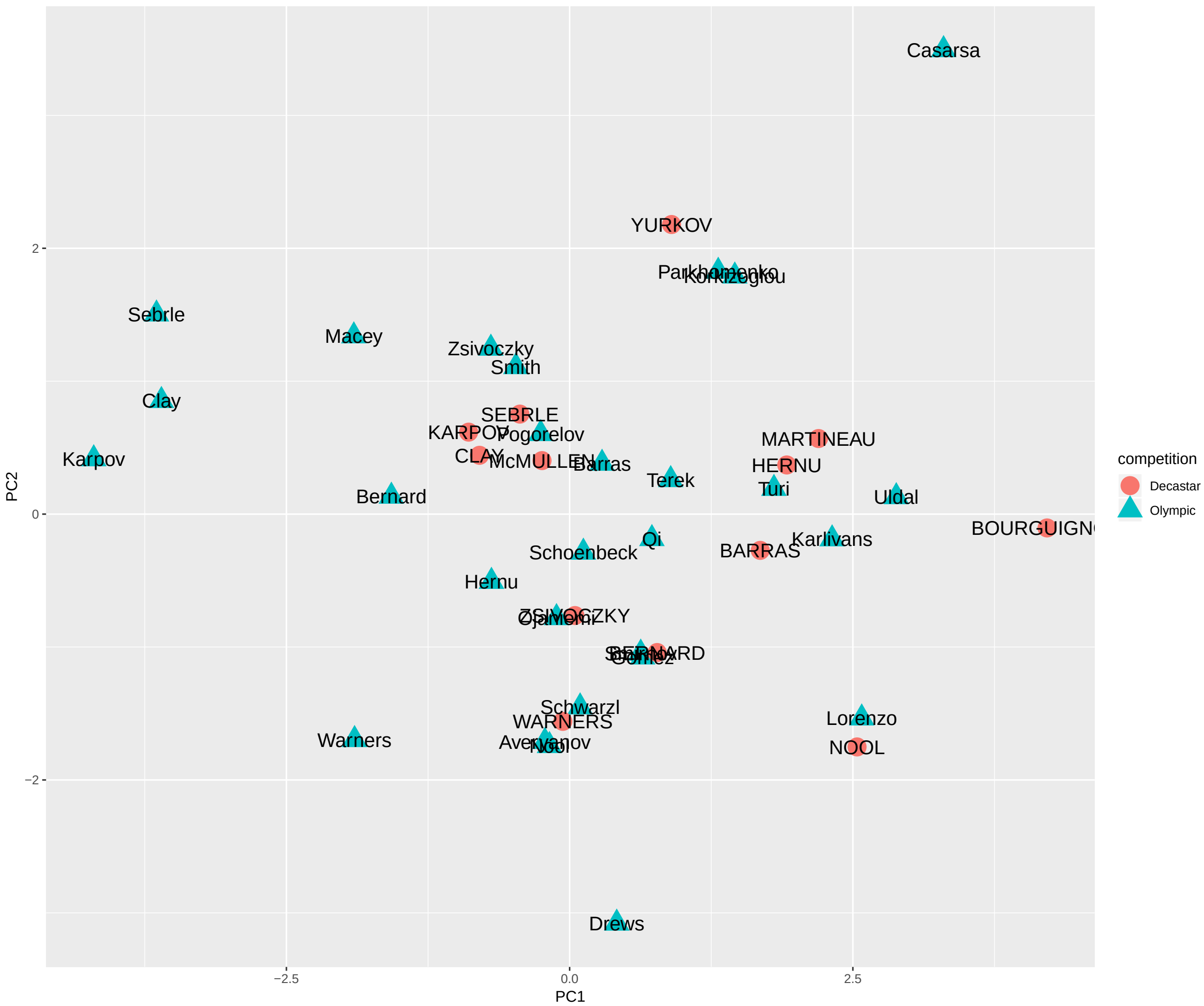
```
unscaled_sups <- dec[c(29:41),c(2:11)]
accentroids <- apply(acdec_unscaled[, c(2:11)], MARGIN=2, FUN=mean)
acsds <- apply(acdec_unscaled[, c(2:11)], MARGIN=2, FUN=sd)
cent_sups <- sweep(unscaled_sups, MARGIN=2, STAT=accentroids, FUN='-')
sups <- sweep(cent_sups, MARGIN=2, STAT=acsds, FUN='/')
sups_on_PCs <- as.matrix(sups) %*% loadings
colnames(sups_on_PCs) <- paste0('PC', c(1:10))
sups_on_PCs
```

```
##           PC1        PC2         PC3         PC4        PC5         PC6
## 29 -0.43930190  0.7524453 -0.98678469  1.20203985  0.4253897  0.21195717
## 30 -0.79561959  0.4425308 -1.86183358 -0.05618834 -1.6689269 -1.98517405
## 31 -0.89167000  0.6167016 -2.20004010 -1.43716735  0.3017575 -0.30634635
## 32  0.77252139 -1.0411739 -0.61027465  2.13612105 -0.2587693  0.72117386
## 33  0.89945975  2.1788060  0.87351341  0.83658736  0.5763792  1.36032380
## 34 -0.06010103 -1.5592556 -0.81515671 -0.54485493  0.9058410  0.13897421
## 35  0.04670533 -0.7630591  1.13152517 -1.57397173  0.6822395 -0.49483864
## 36 -0.24374426  0.4025704 -0.21882056 -1.46923585  1.0512592 -0.18757613
## 37  2.19682203  0.5686558  0.75008897 -0.20943934  0.8868093  1.05264267
## 38  1.91654578  0.3697215 -0.64426478  0.39480146 -0.7685903 -0.04159251
## 39  1.68233643 -0.2728426 -0.02294709 -0.34573839  0.7795294 -0.29152289
## 40  2.53692192 -1.7515236  1.28208989 -0.03443662  0.6745090  1.17668941
## 41  4.21392299 -0.1040495 -1.24669417  0.75829429 -0.3490381  0.97398392
##           PC7        PC8         PC9        PC10
## 29 -1.17427392  0.2115985  0.79205410 -0.50652936
## 30  0.09378109  0.8902251 -0.50900199 -0.60199028
## 31  0.50464529  1.4806595  0.50970439 -1.08676800
## 32 -0.00478260  0.3737301  0.27969723 -0.73074181
## 33 -0.64080372  0.8183894  0.35549356 -0.18053471
## 34 -0.10396183  0.8024939  1.21273323  0.07367955
## 35 -0.39906759  1.1354200 -0.07700471 -0.44420057
## 36 -1.24607333 -0.5587098 -0.39525698 -0.76630350
## 37  1.09738272  2.8858764  0.02668512  0.65250492
## 38 -0.87434020  1.3579655  1.00729953  1.06668037
```

```
## 39  0.38985082  0.8335699  0.82372139 -0.61629825
## 40 -1.51358867  0.4472553  0.55060105 -0.68741961
## 41 -0.25193768  0.5686850  0.26481285 -0.43749829
```

```r
all_ind_ac_var <- data.frame(rbind(PCs, sups_on_PCs))
all_ind_ac_var['athlete'] <- dec[['athlete']]
all_ind_ac_var['competition'] <- dec[['competition']]

all_ind_ac_var %>% ggplot(aes(x=PC1, y= PC2, label= athlete)) + geom_point(aes(color=competition, shape
```



The olympic competitors seem to have more spread than the decastar competitors. However, many of them are distributed fairly equally regarding the first two PCs.

b)

```r
dsqrds <- apply(acdec**2, MARGIN=1, FUN=sum)
cos2<- PCs**2/dsqrds
colnames(cos2) <- paste0('PC', c(1:10))
rownames(cos2) <- paste0('cos2 ', acathletes)
head(cos2)
```

```
##                      PC1         PC2         PC3        PC4          PC5
## cos2 Sebrle    0.66903592 0.113893074 0.002352728 0.15312983 0.0008771211
## cos2 Clay      0.68487230 0.038449930 0.005390107 0.07147214 0.0452161626
## cos2 Karpov    0.80752649 0.007903026 0.005713353 0.13300697 0.0219207077
## cos2 Macey     0.36529692 0.180841922 0.154408343 0.12062809 0.0349442502
## cos2 Warners   0.46973462 0.375380751 0.102893033 0.03203165 0.0027821723
## cos2 Zsivoczky 0.08591311 0.276553647 0.186194481 0.05084090 0.0478138026
##                       PC6          PC7          PC8          PC9
## cos2 Sebrle    0.004149693 0.0097357827 0.0138028806 0.0311268999
## cos2 Clay      0.035320991 0.0833875028 0.0044742602 0.0310538668
## cos2 Karpov    0.004561838 0.0072156010 0.0094648006 0.0002113434
## cos2 Macey     0.117816923 0.0047192673 0.0043473577 0.0005760518
## cos2 Warners   0.010060318 0.0001402244 0.0003008094 0.0056938553
## cos2 Zsivoczky 0.303669910 0.0057122125 0.0081004741 0.0172853142
##                       PC10
## cos2 Sebrle    0.0018960693
## cos2 Clay      0.0003627428
## cos2 Karpov    0.0024758626
## cos2 Macey     0.0164208733
## cos2 Warners   0.0009825697
## cos2 Zsivoczky 0.0179161472
```

Representation

```r
sort(cos2[,1] + cos2[,2])
```

```
##   cos2 Schoenbeck      cos2 Barras       cos2 Terek   cos2 Pogorelov
##        0.03544177       0.05153401       0.07725611       0.07773325
##     cos2 Ojaniemi       cos2 Smith          cos2 Qi       cos2 Gomez
##        0.13006276       0.14097634       0.16995775       0.25285569
##        cos2 Hernu cos2 Korkizoglou        cos2 Nool        cos2 Turi
##        0.25338222       0.31810583       0.34248386       0.34357165
##    cos2 Zsivoczky      cos2 Bernard     cos2 Smirnov cos2 Parkhomenko
##        0.36246676       0.37102582       0.39872234       0.46458606
##    cos2 Averyanov     cos2 Schwarzl       cos2 Macey   cos2 Karlivans
##        0.50042562       0.52321051       0.54613885       0.58080629
##      cos2 Lorenzo        cos2 Clay      cos2 Sebrle      cos2 Karpov
##        0.68239294       0.72332223       0.78292899       0.81542952
##        cos2 Drews      cos2 Warners       cos2 Uldal     cos2 Casarsa
##        0.82476002       0.84511537       0.85927471       0.95414664
```

The athletes that are best represented on the first two PCs are Casarsa, Uldal, Warners, and Drews

The athletes that are worst represented on the first two PCs are Schoenbeck, Barras, Terek, and Pogorelov. xs ##c)

```r
m <- 1/(n-1)
ctr <- sweep((m*PCs**2)*100, 2, lambdas, FUN="/")
rownames(ctr) <- paste0('Ctr of ', rownames(ctr))
ctr[,1:4]
```

```
##                         PC1        PC2        PC3        PC4
## Ctr of Sebrle     13.896472718 4.25667207 0.12183879 12.478583027
## Ctr of Clay       13.566366232 1.37046272 0.26620124  5.554449503
## Ctr of Karpov     18.437668318 0.32468361 0.32523627 11.914448816
```

```
## Ctr of Macey        3.791512875  3.37740675  3.99572873  4.912078298
## Ctr of Warners      3.765848145  5.41501879  2.05662407  1.007487819
## Ctr of Zsivoczky    0.505123020  2.92573389  2.72937503  1.172738593
## Ctr of Hernu        0.497794814  0.48293619  1.39594113  0.121254458
## Ctr of Nool         0.033024268  5.70565731  2.53563429 15.961196069
## Ctr of Bernard      2.587013828  0.03368047  0.03385408 10.960719825
## Ctr of Schwarzl     0.008939045  3.95882420  1.35483238  0.986442407
## Ctr of Pogorelov    0.068370188  0.68850078  7.96603923  0.212487210
## Ctr of Schoenbeck   0.015334884  0.15518173  0.79713121  4.106485050
## Ctr of Barras       0.085522257  0.27393487  7.11732807  1.528126098
## Ctr of Smith        0.235265509  2.31104393  6.22690381  4.919239127
## Ctr of Averyanov    0.049790581  5.50658088  0.66954990  0.341197634
## Ctr of Ojaniemi     0.014047826  1.14325620  0.09063735  0.045114489
## Ctr of Smirnov      0.411458048  2.06001181  3.88896838  0.408515646
## Ctr of Qi           0.550830837  0.06431140  2.62796365  0.482669233
## Ctr of Drews        0.180438327 17.81282299  1.94340179  1.220788555
## Ctr of Parkhomenko  1.797609753  6.26843595  1.34372003 11.380934728
## Ctr of Terek        0.831378555  0.12579836 14.30019672  0.248458059
## Ctr of Gomez        0.433187509  2.17453292  5.91488543  0.117634126
## Ctr of Turi         3.397770397  0.06969640  1.72920767  0.555901396
## Ctr of Lorenzo      6.944171406  4.42689362  6.78249358  0.029269465
## Ctr of Karlivans    5.608020249  0.06570709  0.04764855  5.617251120
## Ctr of Korkizoglou  2.220130040  6.02658464 22.65017943  3.212059105
## Ctr of Uldal        8.685084058  0.03183105  0.70699096  0.006853852
## Ctr of Casarsa     11.381826314 22.94379938  0.38148824  0.497616293
```

Influential athletes on the first two PCs: Sebrle, Clay, Karpov, Casara, and Drews

4) Studying the cloud of variables

a)

```r
correlations <- cor(as.matrix(acdec_unscaled[,c(2:13)]), PCs)
correlations[,1:4]
```

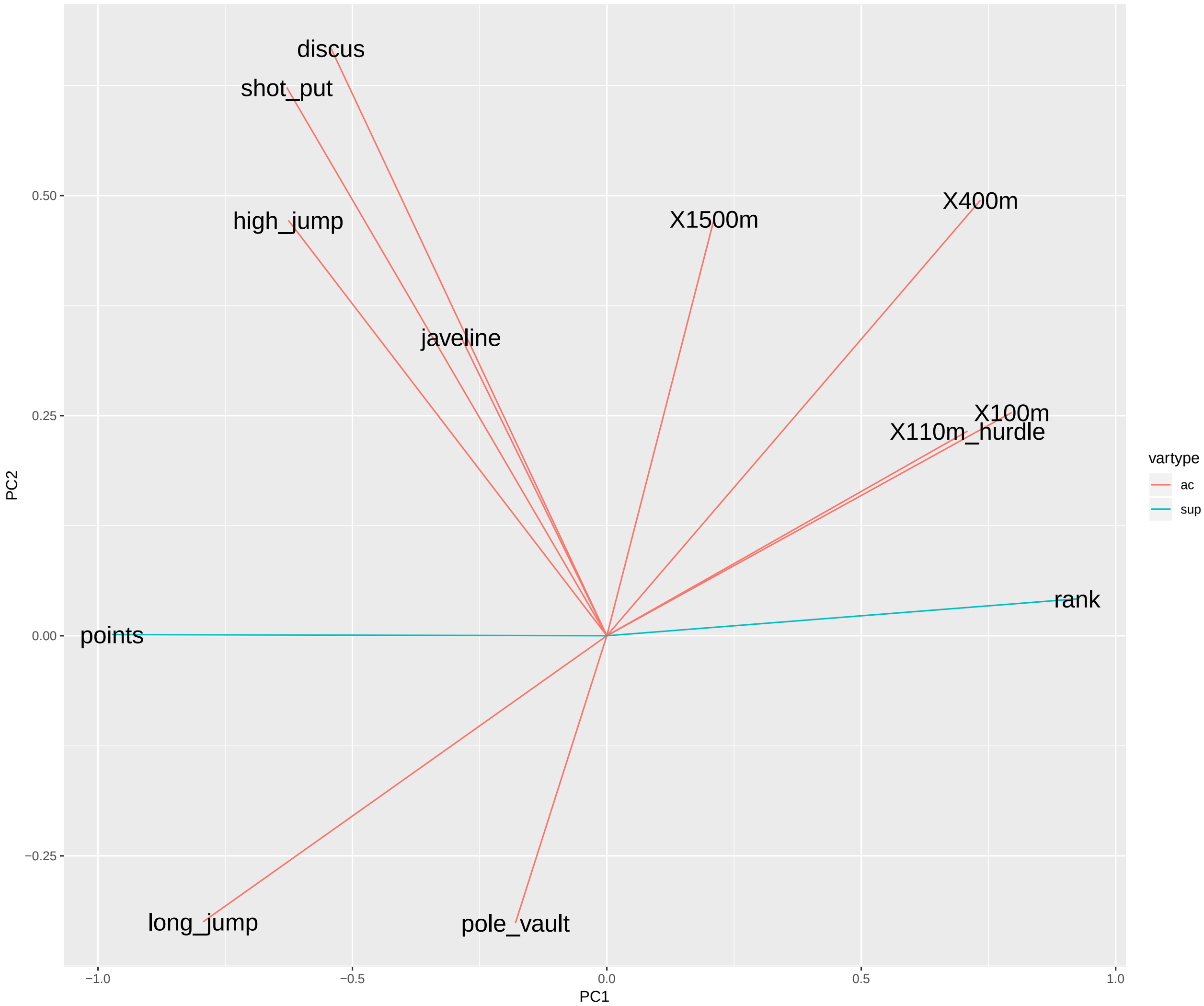
```
##                      PC1          PC2         PC3          PC4
## X100m          0.7958383  0.253599340  0.25277014  0.071298025
## long_jump     -0.7935059 -0.324979385 -0.15521388 -0.005840942
## shot_put      -0.6289690  0.622800538 -0.02252512  0.133493731
## high_jump     -0.6259915  0.471948288  0.01291630 -0.105515561
## X400m          0.7341798  0.494656647 -0.22972590  0.111158299
## X110m_hurdle   0.7089265  0.232408269  0.04392919  0.109666171
## discus        -0.5421042  0.667282351 -0.01785549 -0.196002694
## pole_vault    -0.1795989 -0.326306097 -0.62447715  0.611344812
## javeline      -0.2864207  0.338982261  0.52108731  0.655675756
## X1500m         0.2107444  0.473357014 -0.78520075 -0.054465625
## rank           0.9243932  0.041903953 -0.07680790 -0.148552939
## points        -0.9724931  0.001294792  0.06188580  0.196710089
```

```r
#I found this function online; ggplot doesn't have a good circle maker

circleFun <- function(center = c(0,0),diameter = 1, npoints = 100){
    r = diameter / 2
```

```
    tt <- seq(0,2*pi,length.out = npoints)
    xx <- center[1] + r * cos(tt)
    yy <- center[2] + r * sin(tt)
    return(data.frame(x = xx, y = yy))
}

plot_corr <- data.frame(correlations)
plot_corr['vartype'] <- c(rep('ac', 10), rep('sup', 2))
data.frame(plot_corr) %>% ggplot(aes(x= PC1, y=PC2, xend=0, yend=0, label = rownames(correlations), ylin
```



##c) Looking at the correlation matrix: Points, rank, X100m, and long jump are all very related to PC1 discuss and shot put seem more strongly related to PC2

Looking at the diagram: Points and rank are very related to PC1! Discuss and Shot put look fairly related to PC2.

5) Conclusions

High jump, shot put, and discuss (and to some extent javeline) are all very related variables!So are X110m hurdle and x100m

Olympiads Sebrle, Clay, Karpov, Casara, and Drews contribute the most to the first two PCs, and therefore contribute to a lot of the variance between competitors. Maybe they are particularly good (or bad).