HW05 Spatial Statistics

Drew Remmenga

2022-10-20

- You are encouraged to use web resources and class materials
- Please work alone and hand in your own work.
- Be spare in what you include and you will lose credit if you include too much extraneous output or information. All questions count for an equal number of points.
- Any subproblems marked GRAD are required for 500 level students but will serve as an extra credit question for the 400 level students.
- Please send me email if you have questions or any concerns. `nychka@mines.edu`
- Hand in your work in pdf format in Gradescope. You can keep the questions as part of what you hand in but you should begin your *answer* on a separate page. You can use `\newpage` to create a page break in your work.

Reformatting text and code that are not your answers

It is harder to grade homework when all the introduction and question are included and in Either comment out the answers without just deleting them or put your answers in a different color.

To use the html commenting format:

```
<!--
This text is now commented out and will not be part of the rendered output.
-->
```

To change the color you need to have Latex working:

```
\textcolor{magenta}{This is a magenta block of text. }
```

And the rendered version in pdf:

This is a magenta block of text.

Points

All subsections of the problems count equally for 10 points:

- 400 level
- 500 level

Some setup

Change the directory path below for your PC/Mac

```
setwd("~/School/MATH498/HW5/")
suppressMessages(library( fields))
```

```
## Warning: package 'fields' was built under R version 4.1.3
```

```
## Warning: package 'spam' was built under R version 4.1.3
```

Problem 1

For this problem revisit the Boulder Bolder 10K split times and the male and female 30 year age category. See the code below to setup the data for these subsets. Note that the first step in the wrangling is to transpose the times so that columns index miles (split times) and rows index individual.

```
load("BB10K.rda")

splitTimes<-  t( as.matrix(BB10K[,4 + 1:6]))
    splitTimes <- t( splitTimes)
    dim( splitTimes)
```

```
## [1] 24816     6
```

```
  # Now rows index the splits,  columns index the runners
    indM<- BB10K$DIV == "M30"
    indF<- BB10K$DIV == "F30"
  # select only female 30 years old
    split30F<- splitTimes[indF,]
    dim( split30F)
```

```
## [1] 360   6
```

```
  # total race time
    timeF30<- BB10K$TIME[indF]
  # the guys ...
  split30M<- splitTimes[indM,]
  dim( split30M)
```

```
## [1] 346   6
```

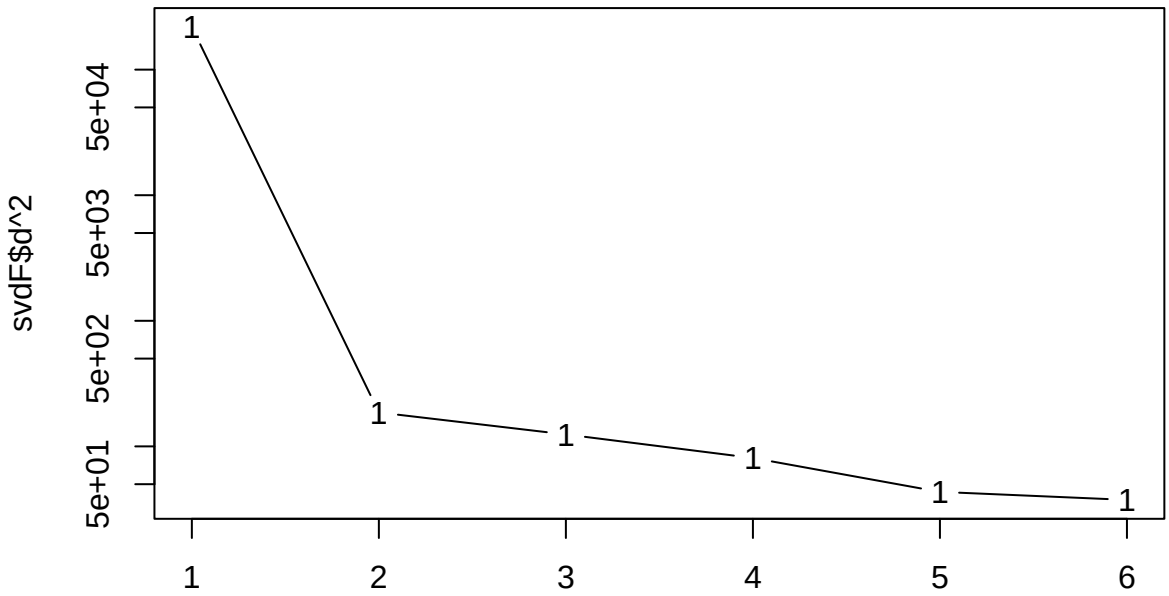
```
  timeM30<- BB10K$TIME[indM]
  split30Combined<- rbind( split30F,split30M)
  time30Combined <- c( timeF30, timeM30)
  FMInd<- c( rep( 1, length( timeF30)),
             rep( 2,length( timeM30))
                  )
```

1(a)

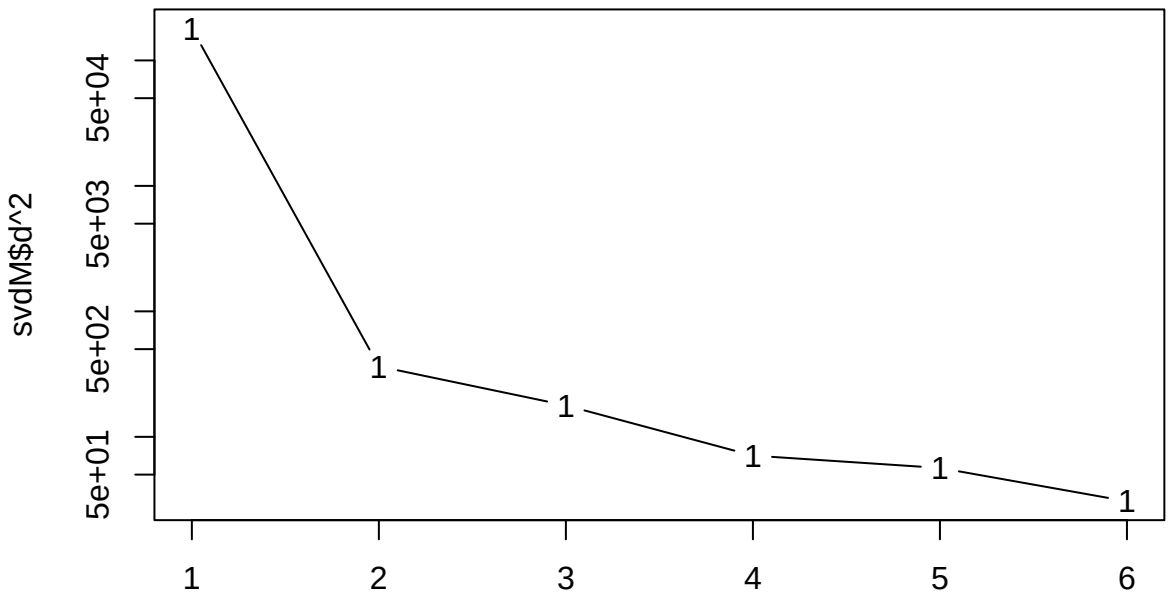
First consider both data sets without subtracting off the column means. Plot the square of the singular values (i.e. the `d` component) for the SVD applied to each M and F data matrix separately
and for combined data sets as function of the index (`1:6`). A perfect dimension reduction based on these singular values is for the first few to be large and then the remaining ones to be close to zero – creating a knee shape. For each of these data matrices is there an obvious "knee" where the singular values fall off more quickly.

For plotting you might use the **matplot** function with `type="b"` and `log="y"`.

```
svdF=svd(split30F)
matplot(svdF$d^2,type="b",log="y")
```



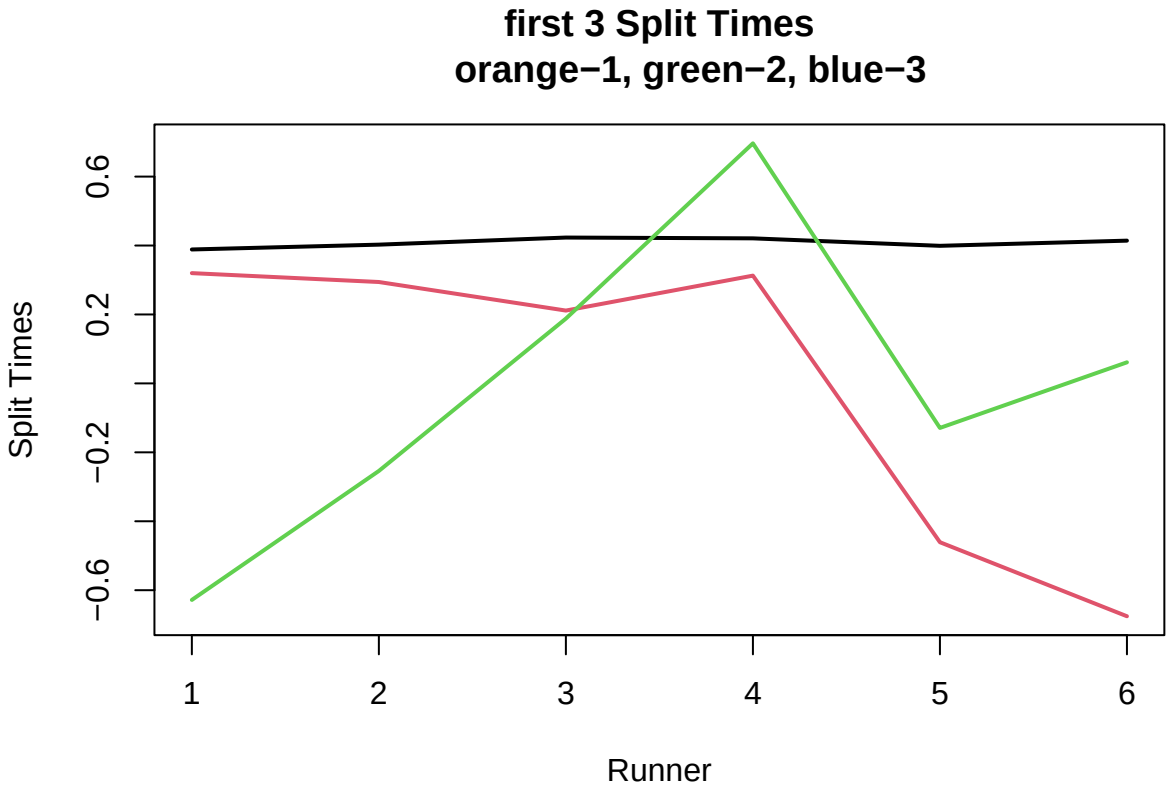

```
svdM=svd(split30M)
matplot(svdM$d^2,type="b",log="y")
```



1(b)

- For the female runners make a plot of the first three basis functions (columns of V).

```
V<- svdF$v
Dsvd<- diag(svdF$d )
V[,1]<- -1*V[,1]
coef<- svdF$u%*%Dsvd
coef[,1]<- -1* coef[,1]
matplot(V[,1:3],
         type="l", lty=1, lwd=2, xlab="Runner",
         ylab="Split Times")
title(" first 3 Split Times
      orange-1, green-2, blue-3")
```



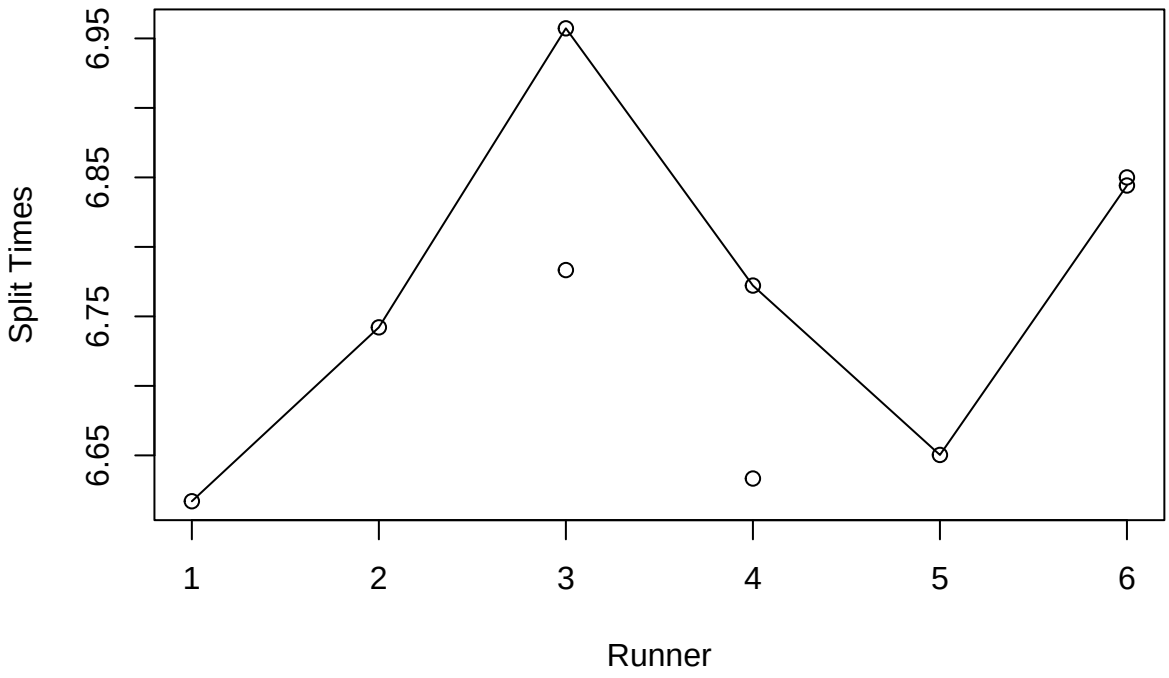
- Interpret these basis functions in terms how they effect the split times Earlier in the race split times are more flat but as the race progresses they start to go higher.

1(c)

Use just these 3 basis functions to reconstruct (approximate) the **split30F** data matrix.

- Plot the split times for the runners 5, 180, 331, and 350 and also the split times from the approximation with 3 basis functions.

```
coef<- svdF$u%*%Dsvd
coef[,1]<- -1* coef[,1]
mDay<- 6
#
F30Approx<- V[,1]*coef[mDay,1] +
  V[,2]*coef[mDay,2] +
  V[,3]*coef[mDay,3]
plot(F30Approx,xlab = "Runner",ylab="Split Times")
lines(F30Approx)
points(split30F[5,])
points(split30F[180,])
points(split30F[331,])
points(split30F[350,])
```



- If **F30Approx** is your approximate matrix then find the discrepancies

```
SSE <- rowSums((F30Approx - split30F )^2)
```

for each runner.

Verify that

```
sum( SSE)
```

```
## [1] 26392.85
```

```
D = svdF$d
sum( D[4:6]^2 )
```

```
## [1] 161.7397
```

where `D` is the vector of singular values from the svd.

1(d)

Is there a relationship between the values of **SSE** and the total time (**timeF30**)? Use one or more plots to support your answer.

```
cor(SSE,timeF30)
```

```
## [1] 0.9521395
```

1(e)

For the svd of the female runners if SVDF is the svd of the data matrix

Scale the "U" columns by the singular values to get basis coefficients:

```
U<- svdF$u
V<- svdF$v
D<- diag( svdF$d)
CM<- U%*%D
```

- Verify that `CM %*% t(V)` recovers the data matrix. Now change the sign of the first columns of U and V (see below) and again verify you recover the data matrix exactly.

```
U[,1]<- -1*U[,1]
V[,1]<- -1*V[,1]
D<- diag( svdF$d)
CM2<- U%*%D
```

- Give a reason for making this sign change even though it does not change the decomposition of the data matrix. The relationship between SSE and total time is negative.
- Based on a scatterplot of the first two columns of **CM** interpret the relationship between these two coefficients taking into account the shape of the first two basis functions. They are porportional.

Problem 2

For the combined data set above, center the data by finding the mean split time for each mile and subtracting this from each for each runner's individual split time. There are several ways to do this but here is one that works directly with the data matrix

```
splitMean<- colMeans( split30Combined)
N<- nrow( split30Combined)
split30Centered<- split30Combined -
   matrix(splitMean, nrow=N, ncol=6, byrow=TRUE )
```

2(a)

How does the svd change for the centered data compared to the uncentered version? It's the same. ## 2(b) Based on a scatterplot of the first two columns of U for the uncentered data identify two obvious outliers. What is unusual about these cases? They are low.