DA410_Assignment7_MattGraham

Principal Component Analysis (PCA) Unsupervised ML technique that seeks linear combinations (principals) of original predictors that explain a large portion of variation in dataset.

Goal: Explain most variability in dataset w/ fewer variables vs. the original dataset.

Allows to better visualize variation present in a dataset w/ many variables. Most helpful w/ wide datasets when you have many variables for each sample. W/ many variables present, it’s hard to plot data to explain trends.

PCA allows us to see overall shape of data and explaing which variables are very similar and different.

High level process summary:

- Get dataset w/ many variables
- Simplify dataset down to principal components (multiple linear combinations)

These are underlying structure of data that shows where data has most variance and most spread out. Meaning we find straight line that best spreads that data out when it is projected upon it. First principal componant shows most substantial variation in data. Then second, third, etc.

```
library(nnspat)  # used for dist2full()
library("dplyr")  # used to select numeric datatypes
library("ggplot2")
library(reshape)  # used for melting matricies
library(klaR)
library(ggvis)
library(class)
library(gmodels)
library(MASS)
library(readxl)
library(psych)
```

Get data

```
probe <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/T3_6_PROBE.DAT", header
=FALSE)
colnames(probe) <- c('ind', 'pos_1', 'pos_2', 'pos_3', 'pos_4', 'pos_5')
probe <- probe[, 2:6]
probe
```

pos_1 <int>	pos_2 <int>	pos_3 <int>	pos_4 <int>	pos_5 <int>
51	36	50	35	42
27	20	26	17	27
37	22	41	37	30
42	36	32	34	27

pos_1 <int>	pos_2 <int>	pos_3 <int>	pos_4 <int>	pos_5 <int>
27	18	33	14	29
43	32	43	35	40
41	22	36	25	38
38	21	31	20	16
36	23	27	25	28
26	31	31	32	36

1-10 of 11 rows Previous **1** 2 Next

a. Get S and R

```
S = cov(probe)
R = cor(probe)
```

S

```
as.data.frame(S)
```

	pos_1 <dbl>	pos_2 <dbl>	pos_3 <dbl>	pos_4 <dbl>	pos_5 <dbl>
pos_1	65.09091	33.64545	47.59091	36.77273	25.42727
pos_2	33.64545	46.07273	28.94545	40.33636	28.36364
pos_3	47.59091	28.94545	60.69091	37.37273	41.12727
pos_4	36.77273	40.33636	37.37273	62.81818	31.68182
pos_5	25.42727	28.36364	41.12727	31.68182	58.21818

5 rows

R

```
as.data.frame(R)
```

	pos_1 <dbl>	pos_2 <dbl>	pos_3 <dbl>	pos_4 <dbl>	pos_5 <dbl>
pos_1	1.0000000	0.6143902	0.7571850	0.5750730	0.4130573
pos_2	0.6143902	1.0000000	0.5473897	0.7497770	0.5476595
pos_3	0.7571850	0.5473897	1.0000000	0.6052716	0.6918927

	pos_1 <dbl>	pos_2 <dbl>	pos_3 <dbl>	pos_4 <dbl>	pos_5 <dbl>
pos_4	0.5750730	0.7497770	0.6052716	1.0000000	0.5238876
pos_5	0.4130573	0.5476595	0.6918927	0.5238876	1.0000000

5 rows

b. Get eigenvalues and eigenvectors of S and R

```
eig.S <- eigen(S)
eig.R <- eigen(R)
```

Eigen S

```
eig.S
```

```
## eigen() decomposition
## $values
## [1] 200.375372  36.090806  34.072122  14.967285   7.385324
##
## $vectors
##           [,1]        [,2]        [,3]       [,4]       [,5]
## [1,] 0.4727831 -0.57631826  0.41685181  0.2285789  0.4672469
## [2,] 0.3918187 -0.10826396 -0.45239805  0.6558114 -0.4472185
## [3,] 0.4875471  0.09600807  0.47945493 -0.3689607 -0.6221505
## [4,] 0.4677199 -0.12056819 -0.61953744 -0.5803451  0.2146494
## [5,] 0.4080320  0.79522446  0.08869553  0.2114962  0.3853964
```

Eigen R

```
eig.R
```

```
## eigen() decomposition
## $values
## [1] 3.4164933 0.6144313 0.5722740 0.2712115 0.1255899
##
## $vectors
##            [,1]       [,2]       [,3]       [,4]       [,5]
## [1,] -0.4418394 -0.2006104  0.6786078  0.2125365  0.5087760
## [2,] -0.4535595 -0.4280646 -0.3491277  0.6055405 -0.3499642
## [3,] -0.4727808  0.3678765  0.3754368 -0.2581448 -0.6584479
## [4,] -0.4536224 -0.3934629 -0.3345386 -0.7010073  0.1899641
## [5,] -0.4120276  0.6974023 -0.4058723  0.1734903  0.3860467
```

c. Percent variance explained and plot S and R

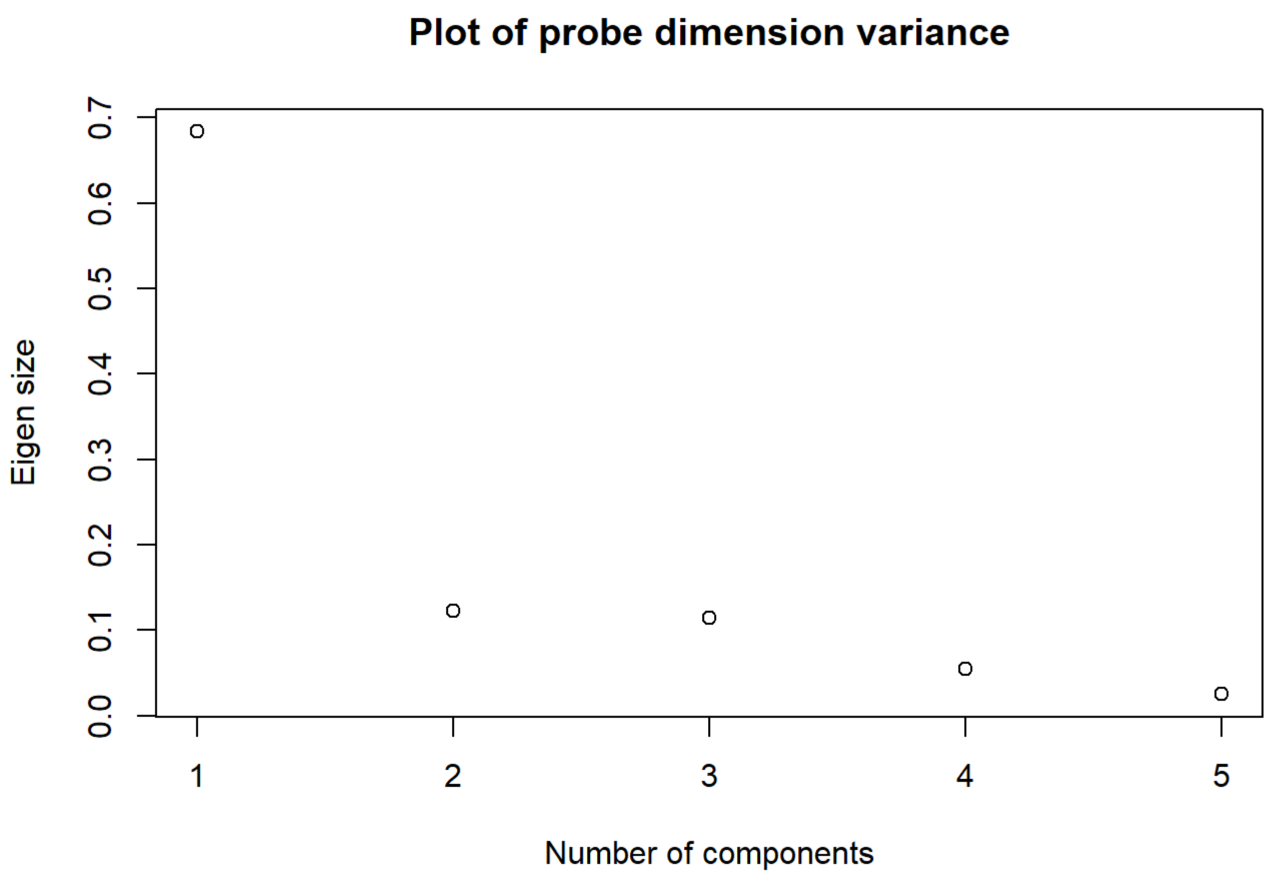
```r
for (r in eig.R$values) {
  print(r/sum(eig.R$values))
}
```

```
## [1] 0.6832987
## [1] 0.1228863
## [1] 0.1144548
## [1] 0.0542423
## [1] 0.02511798
```

We can see that of our eigen values, ~65.5% of our variance is explained with just one dimension, while ~26% is explained with 2 and 3 dimensions. This can be seen below.

Plot

```r
plot(eig.R$values/sum(eig.R$values), xlab = 'Number of components', ylab='Eigen size', main='
Plot of probe dimension variance')
```



d. Decide compontent retention and show reasoning

Based on the percentage of total variance, it makes sense to keep the first 3 components as components 4 and 5 make up less over 10% of all explained variance.

e. Interpretation

Over the duration of subjects' time, as we model our data, we can conclude that most of our variation happens within the first 3 time windows as compared to subsequent ones. This makes sense as the longer it takes people to complete things reduces in probability.

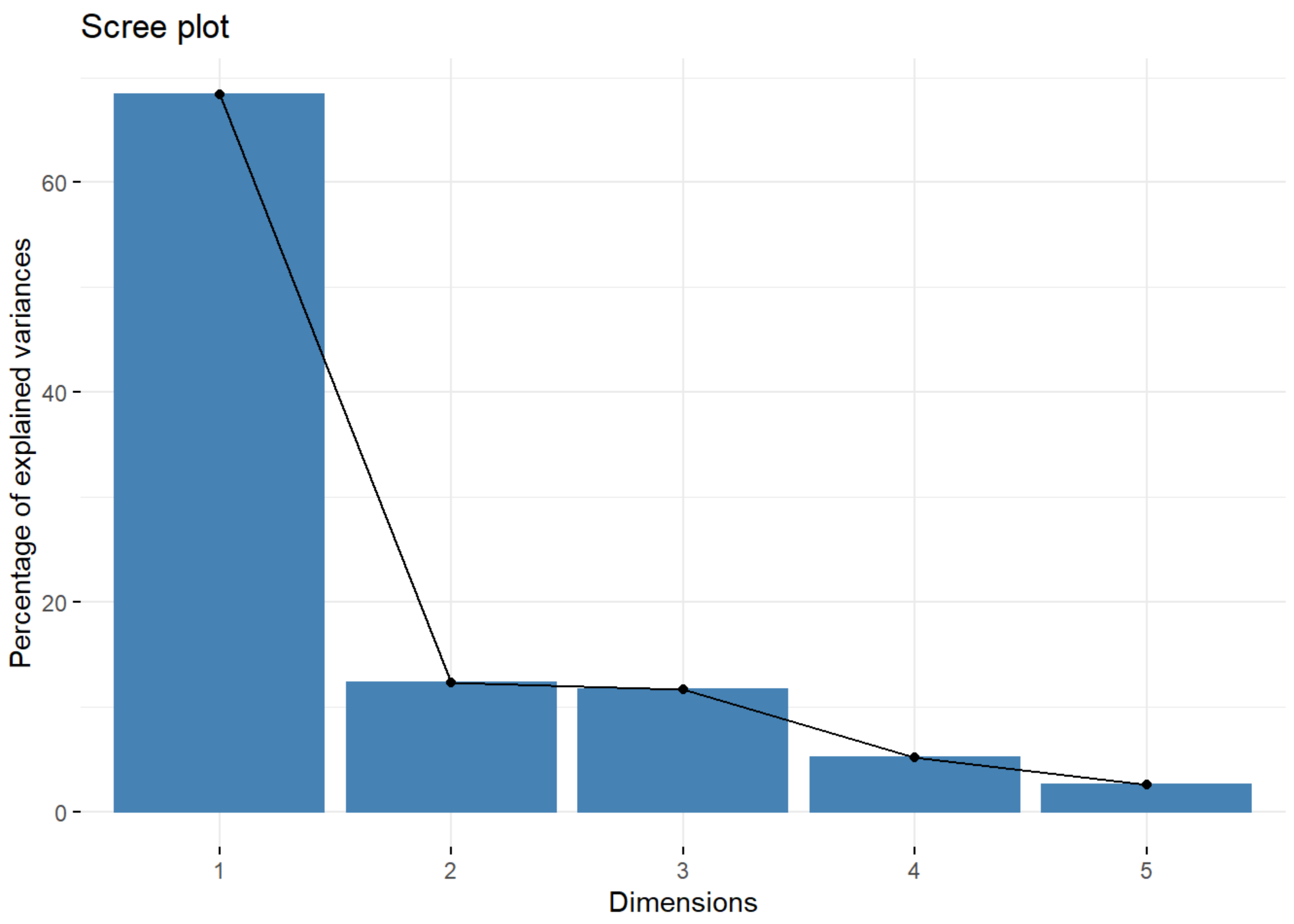
Example for discussion

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
probe.pca <- prcomp(probe)
fviz_eig(probe.pca)
```


Scree plot
Percentage of explained variances
Dimensions