

HW4 Alternative

FUNDAMENTALS OF COMPUTATIONAL MATHEMATICS

Zachary Palmore

2/12/2021

Directions

With the attached data file, build and visualize eigenimageries that accounts for 80% of the variability. Provide full R code and discussion.

Preparation

To perform a principal component analysis (PCA) it is important to have and understand the data in the analysis. In this section, the data, which in this case is images, are read into our R environment. There are several considerations to make, such as how to extract the files and import them, what format we want the files in for the PCA, whether to scale the data to the same size, and more. This analysis begins by reading the images as a raster array.

The following packages were used during this analysis and should be installed prior to replication.

```
library(jpeg)
library(EBImage) # Requires BiocManager::install('EBImage')
library(RCurl)
library(ggpubr)
library(gridExtra)
library(tidyverse)
```

Image Arrays

Using an

```
n <- 17 # number of images present in folder
locale <- "C:/data/images/jpg" # location on computer
imgs <- list.files(locale, pattern="\\.jpg")[1:n] # list of image names
imgs
```

```
## [1] "RC_2500x1200_2014_us_53446.jpg" "RC_2500x1200_2014_us_53455.jpg"
## [3] "RC_2500x1200_2014_us_53469.jpg" "RC_2500x1200_2014_us_53626.jpg"
## [5] "RC_2500x1200_2014_us_53632.jpg" "RC_2500x1200_2014_us_53649.jpg"
## [7] "RC_2500x1200_2014_us_53655.jpg" "RC_2500x1200_2014_us_53663.jpg"
## [9] "RC_2500x1200_2014_us_53697.jpg" "RC_2500x1200_2014_us_54018.jpg"
## [11] "RC_2500x1200_2014_us_54067.jpg" "RC_2500x1200_2014_us_54106.jpg"
## [13] "RC_2500x1200_2014_us_54130.jpg" "RC_2500x1200_2014_us_54148.jpg"
```

```
## [15] "RC_2500x1200_2014_us_54157.jpg" "RC_2500x1200_2014_us_54165.jpg"
## [17] "RC_2500x1200_2014_us_54172.jpg"
```

Create an array to store the image files in when they are read in. At the same time, the dimensions of the array are set to be equivalent to that of the scaled images.

The height and width were given in the names of each image. They are all the same.

```
h <- 1200
w <- 2500
s <- 20
ra <- array(rep(0, length(imgs)*h/s*w/s*3), # creates placeholder array elements
            dim = c(length(imgs), h/s, w/s, 3)) # sets array dimensions
path <- "C:/data/images/jpg/"

for (i in 1:n){
  rs <- resize(readJPEG(paste0(path, imgs[i])), h/s, w/s) # locates each file and adjusts h and w by the
  ra[i,,,] <- array(rs, dim = c(1, h/s, w/s, 3)) # Adds the image values to the placeholder array
}

# Example function
plot_jpeg = function(path, add=FALSE)
{ jpg = readJPEG(path, native=T) # read the file
  res = dim(jpg)[2:1] # get the resolution, [x, y]
  if (!add) # initialize an empty plot area if add==FALSE
    plot(1,1,xlim=c(1,res[1]),ylim=c(1,res[2]),asp=1,type='n',xaxs='i',yaxs='i',xaxt='n',yaxt='n',xlab=
rasterImage(jpg,1,1,res[1],res[2])
}
par(mfrow=c(3,3))
par(mai=c(.3,.3,.3,.3))
for (i in 1:n){ # adjust to number of files present in folder
  plot_jpeg(writeJPEG(ra[i,,,])) # view shoes function applied here; plots the raster data
}
```





```

vecs <- matrix(0, n, prod(dim(ra)))
for (i in 1:n) {
  r <- as.vector(ra[i,,,1])
  g <- as.vector(ra[i,,,2])
  b <- as.vector(ra[i,,,3])
  vecs[i,] <- t(c(r, g, b))
}
pre <- data.frame(t(vecs))

pre.cov <- cov(pre) # calculate covariance matrix for each
pre.eigens <- eigen(pre.cov) # find eigenvalues and eigenvectors of each matrix
str(pre.eigens)

## List of 2
## $ values : num [1:17] 0.9347 0.1246 0.0458 0.0304 0.0249 ...
## $ vectors: num [1:17, 1:17] -0.177 -0.326 -0.12 -0.295 -0.307 ...
## - attr(*, "class")= chr "eigen"

phi <- -(pre.eigens$vectors[,1:2]) # Inverse of loadings for PC1 and PC2
colnames(phi) <- c("PC1", "PC2")
PC1 <- as.matrix(pre) %*% phi[,1]
PC2 <- as.matrix(pre) %*% phi[,2]
PCs <- data.frame(row.names = NULL, PC1, PC2)
head(PCs)

```

```

##          PC1          PC2
## 1 3.934566 0.9676124
## 2 3.934566 0.9676124
## 3 3.934566 0.9676124
## 4 3.934566 0.9676124
## 5 3.934566 0.9676124
## 6 3.934566 0.9676124

PC1hist <- ggplot(PCs, aes(PC1)) + geom_histogram(binwidth = .1) + xlim(0, 4) + xlab(NULL)
PC2hist <- ggplot(PCs, aes(PC2)) + geom_histogram(binwidth = .1) + xlim(0, 4) + xlab(NULL)
PC2pnt <- ggplot(PCs, aes(PC2, PC1)) + geom_point()
PC1pnt <- ggplot(PCs, aes(PC1, PC2)) + geom_point()
ggarrange(PC1hist, PC2hist, PC1pnt, PC2pnt)

```

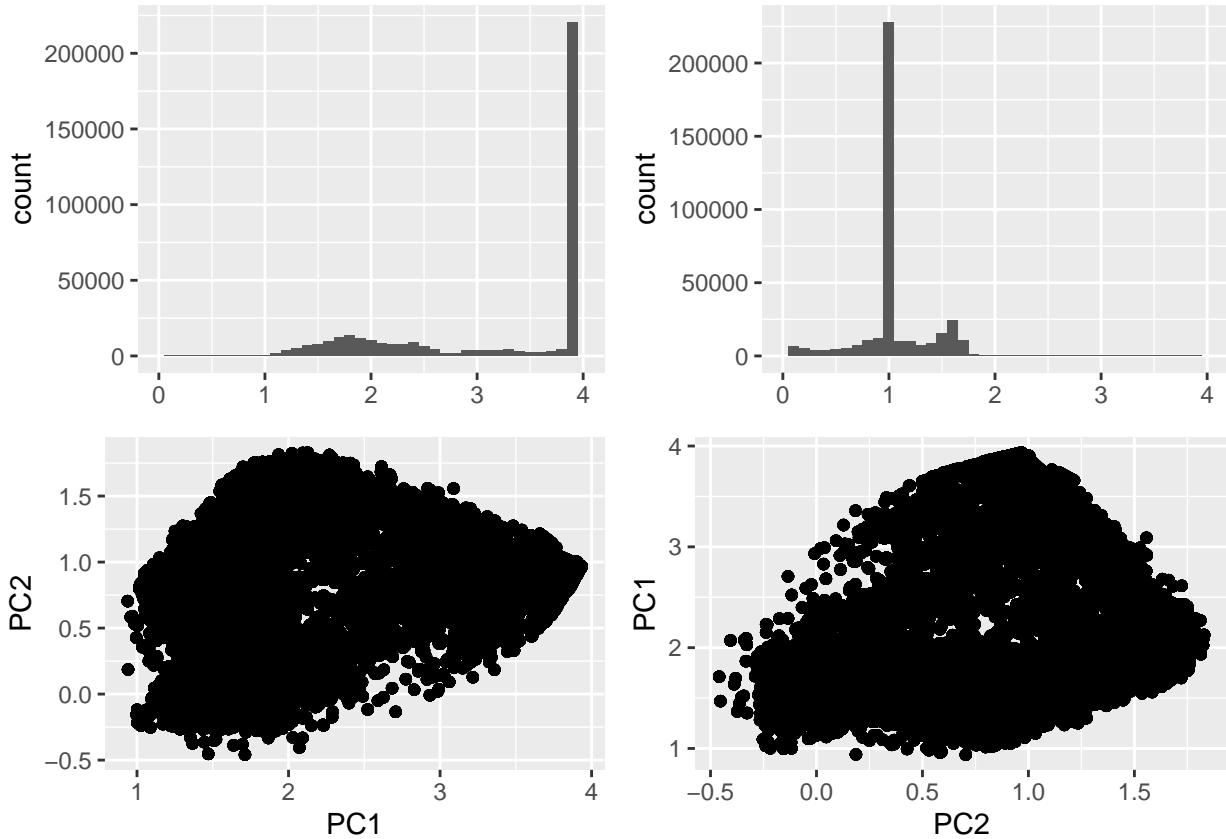
```

## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 7803 rows containing non-finite values (stat_bin).

## Warning: Removed 2 rows containing missing values (geom_bar).

```



```

PVE <- pre.eigens$values / sum(pre.eigens$values)
PVE

```

```

## [1] 0.724729452 0.096632022 0.035529397 0.023597983 0.019287649 0.014874858
## [7] 0.012147092 0.011931203 0.010493340 0.008991208 0.008542160 0.007289829
## [13] 0.006524755 0.006279693 0.005639438 0.004971436 0.002538483

```

```

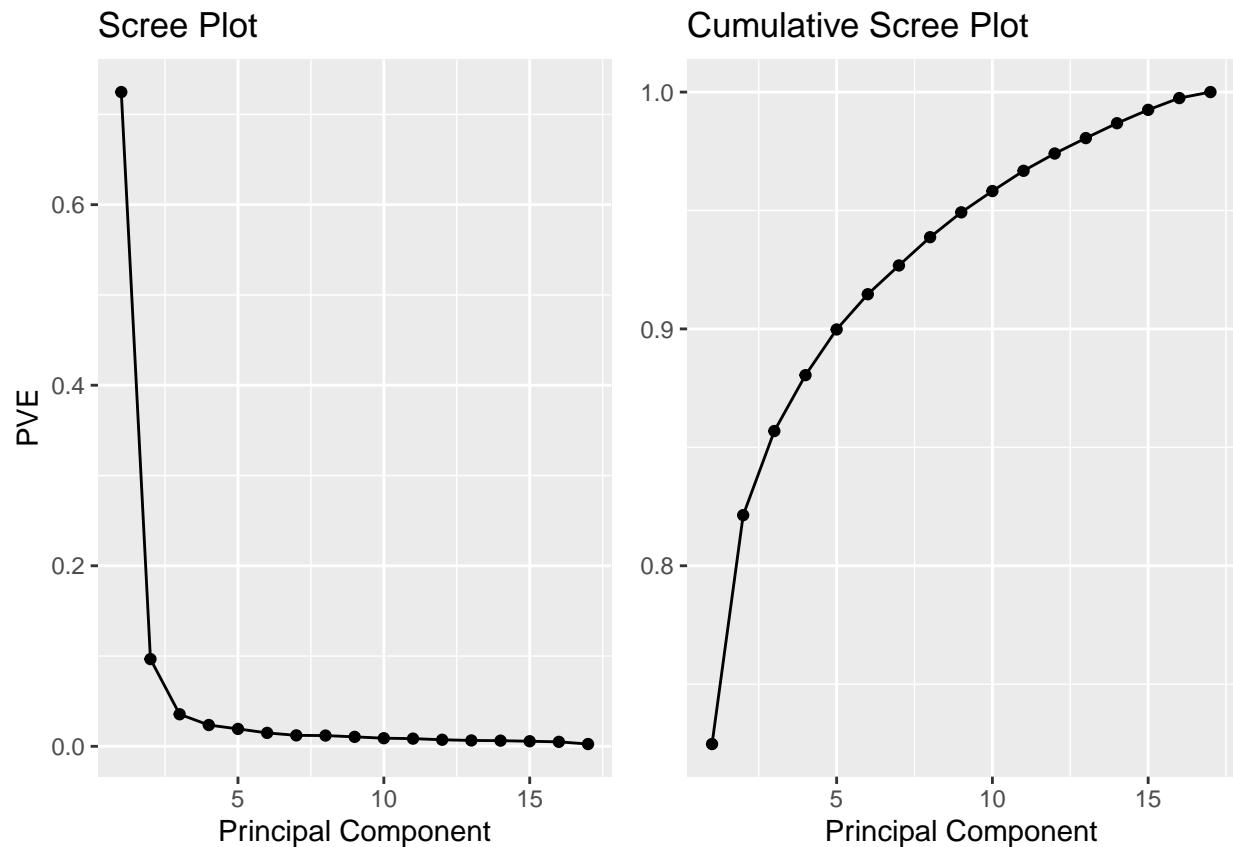
# qplot(PVE, c(1:17)) + geom_line() + coord_flip()
# qplot(c(1:17), cumsum(PVE)) + geom_line()

# PVE (aka scree) plot
PVEplot <- qplot(c(1:17), PVE) +
  geom_line() +
  xlab("Principal Component") +
  ylab("PVE") +
  ggtitle("Scree Plot")

# Cumulative PVE plot
cumPVE <- qplot(c(1:17), cumsum(PVE)) +
  geom_line() +
  xlab("Principal Component") +
  ylab(NULL) +
  ggtitle("Cumulative Scree Plot")

grid.arrange(PVEplot, cumPVE, ncol = 2)

```



```

post <- prcomp(pre, scale = TRUE)
post$rotation <- -post$rotation
post$x <- -post$x
# post_biplot <- biplot(post, scale=0)

```

```

newdata=ra
dim(newdata)=c(length(imgs),h*w*3/s^2)
mypca=princomp(t(as.matrix(newdata)), scores=TRUE, cor=TRUE)

```

```

mypca2=t(mypca$scores)
dim(mypca2)=c(length(imgs), h/s, w/s,3)
par(mfrow=c(5,5))
par(mai=c(.001,.001,.001,.001))
for (i in 1:n){#plot the first 25 Eigenshoes only
plot_jpeg(writeJPEG(mypca2[i,,,], bg="white")) #complete without reduction
}

```



```

va <- ra
dim(va) <- c(n, h*w*3/s^2)
prcomp_results <- princomp(t(as.matrix(va)))
head(prcomp_results$scores, 3)

```

```

##          Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6
## [1,] 0.73413 -0.0214839 0.007451134 0.01476149 -0.004353643 0.00450454
## [2,] 0.73413 -0.0214839 0.007451134 0.01476149 -0.004353643 0.00450454
## [3,] 0.73413 -0.0214839 0.007451134 0.01476149 -0.004353643 0.00450454
##          Comp.7      Comp.8      Comp.9      Comp.10     Comp.11
## [1,] 0.005811962 0.0001790381 0.0005485279 0.000511427 -0.002022676
## [2,] 0.005811962 0.0001790381 0.0005485279 0.000511427 -0.002022676

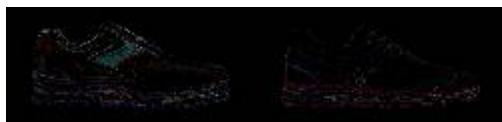
```

```

## [3,] 0.005811962 0.0001790381 0.0005485279 0.000511427 -0.002022676
##          Comp.12      Comp.13      Comp.14      Comp.15      Comp.16
## [1,] -0.001399854 0.003551336 -0.001887749 0.0008538379 -0.0001753731
## [2,] -0.001399854 0.003551336 -0.001887749 0.0008538379 -0.0001753731
## [3,] -0.001399854 0.003551336 -0.001887749 0.0008538379 -0.0001753731
##          Comp.17
## [1,] 0.001010229
## [2,] 0.001010229
## [3,] 0.001010229

vispca_results <- t(prcomp_results$scores)
dim(vispca_results)=c(n, h/s, w/s, 3)
par(mfrow=c(5,5))
par(mai=c(.001,.001,.001,.001))
for (i in 1:n){#plot the first 25 Eigenshoes only
  plot_jpeg(writeJPEG(vispca_results[i,,,])) #complete without reduction
}

```



```

# pcarray <- im
# dim(pcarray) <- c(length(imgs), h*w*3/s^2)
postx <- (as.matrix(post$x))
# dim(posta) <- c(length(imgs), h/s, w/s, 3)
# par(mfrow <- c(3,3))

```

```
pcaresults <- array(rep(0, length(imgs)*h/s*w/s*3), # creates placeholder array elements
                      dim = c(length(imgs), h/s, w/s, 3)) # sets array dimensions

for (i in 1:n){
  vs <- resize(readJPEG(paste0(path, imgs[i])), h/s, w/s) # locates each file and adjusts h and w by the
  pcaresults[i,,,] <- array(postx, dim = c(1, h/s, w/s, 3)) # Adds the image values to the placeholder
}

par(mfrow=c(3,3))
par(mai=c(.3,.3,.3,.3))
for (i in 1:n){ # adjust to number of files present in folder
  plot_jpeg(writeJPEG(pcaresults[i,,,], bg = "white")) # view shoes function applied here; plots the raster
```

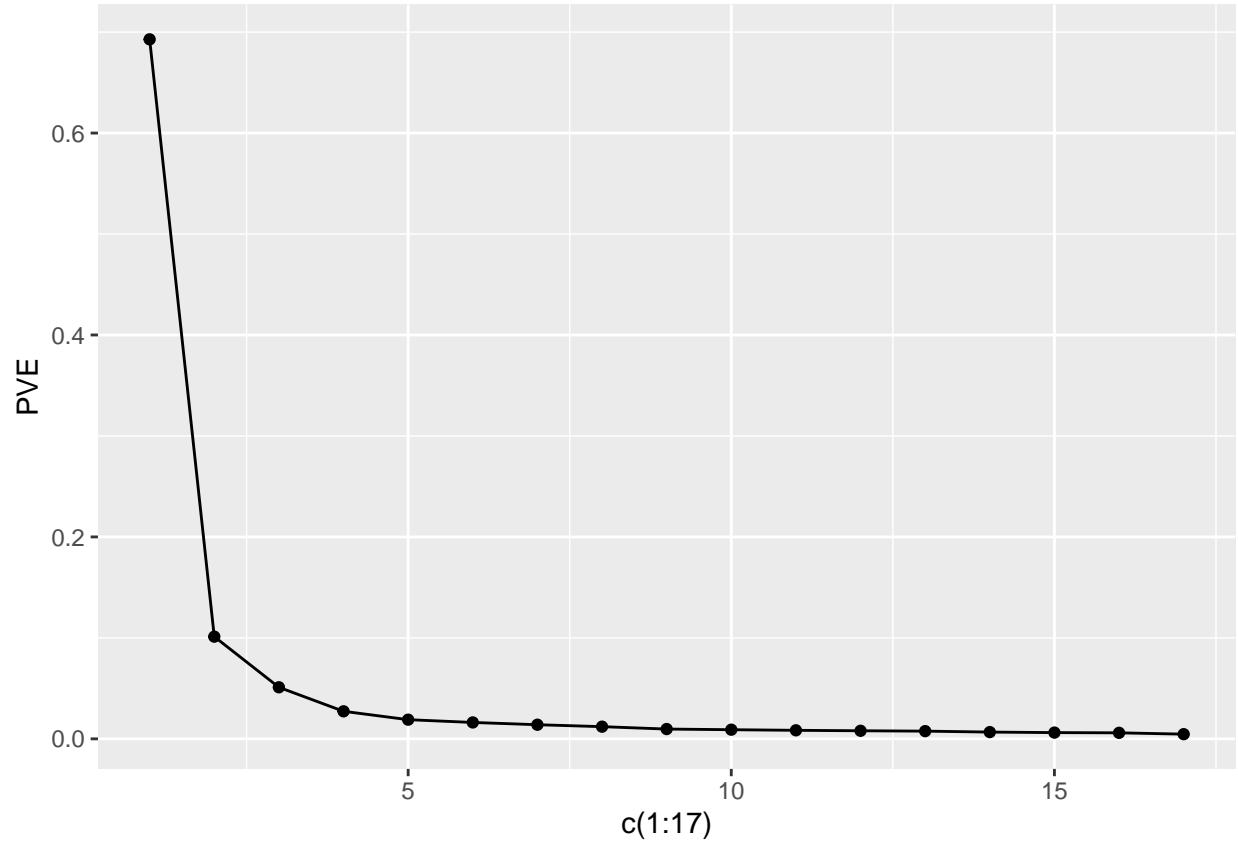




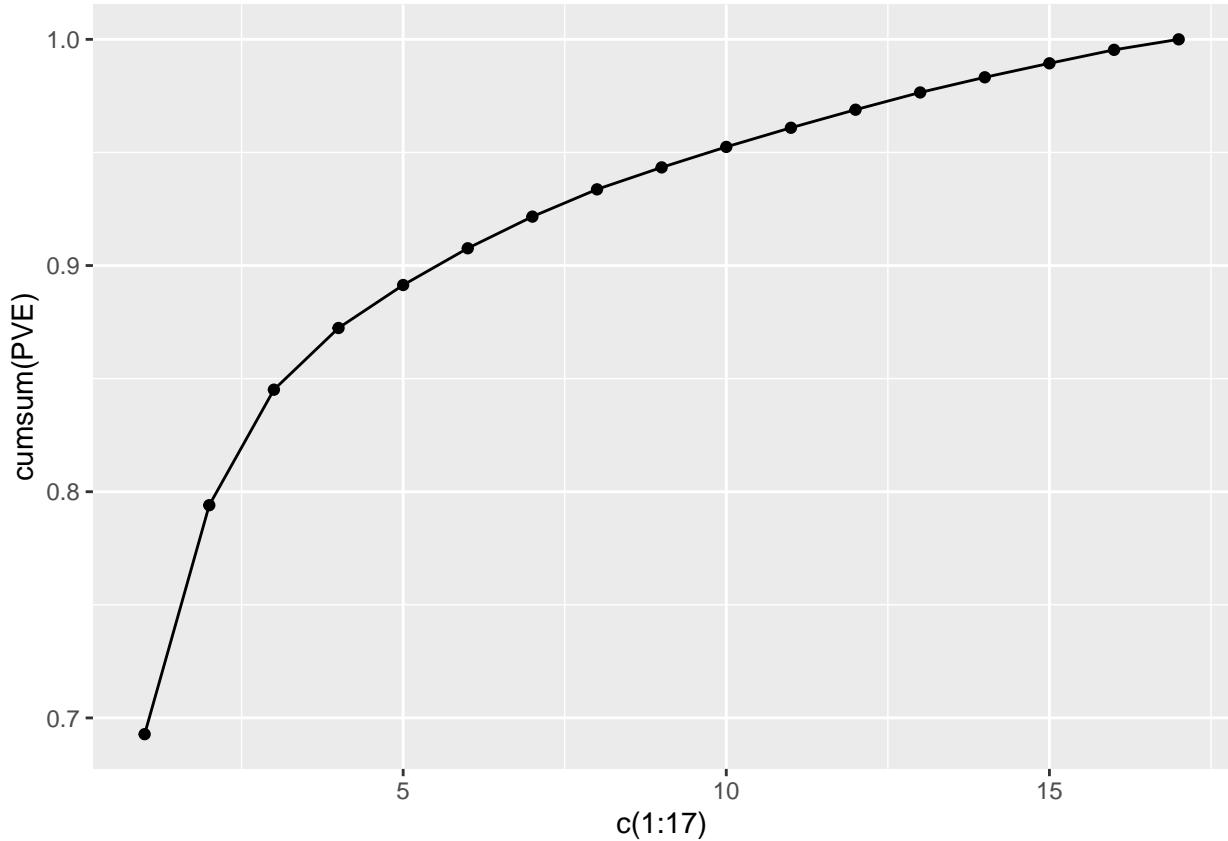
```
VE <- post$sdev^2  
PVE <- VE / sum(VE)  
round(PVE, 2)
```

```
## [1] 0.69 0.10 0.05 0.03 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01  
## [16] 0.01 0.00
```

```
qplot(PVE, c(1:17)) + geom_line() + coord_flip()
```



```
qplot(c(1:17), cumsum(PVE)) + geom_line()
```



```

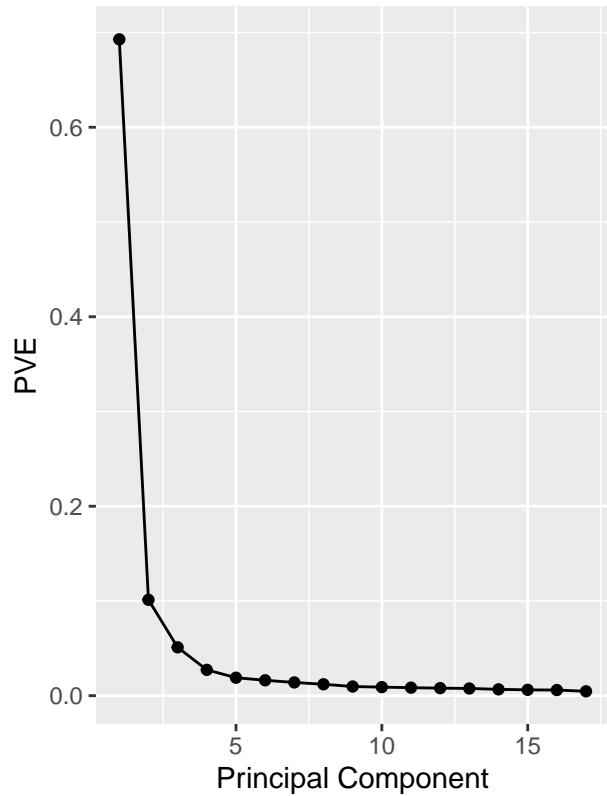
# PVE (aka scree) plot
PVEplot2 <- qplot(c(1:17), PVE) +
  geom_line() +
  xlab("Principal Component") +
  ylab("PVE") +
  ggtitle("Scree Plot Post PCA")

# Cumulative PVE plot
cumPVE2 <- qplot(c(1:17), cumsum(PVE)) +
  geom_line() +
  xlab("Principal Component") +
  ylab(NULL) +
  ggtitle("Cumulative Scree Plot Post PCA")

grid.arrange(PVEplot2, cumPVE2, ncol = 2)

```

Scree Plot Post PCA



Cumulative Scree Plot Post PCA

