2. Perform the below given activities:

a. Apply PCA to the dataset and show proportion of variance

b. Perform PCA using SVD approach

c. Show the graphs of PCA components

ds=read.table("C:\Users\VED PRAKASH\Desktop\table data\epi\_r.csv",sep=",",header=TRUE)

str(ds)

ds<- ds %>% mutate(y = (G1+G2+G3)/3) %>% select(-G1, -G2, -G3)

tds <- data.frame(model.matrix( ~ .- 1, data=ds))

cor\_tds <- cor(tds, tds, method = "pearson")

cor\_df<- data.frame(cor=cor\_tds[1:40,41], varn = names(cor\_tds[1:40,41]))

cor\_df<- cor\_df%>%mutate(cor\_abs = abs(cor)) %>% arrange(desc(cor\_abs))

plot(cor\_df$cor\_abs, type="l")

list\_varn <- cor\_df %>% filter(cor\_abs>0.2)

filter\_df <- data.frame(tds) %>% select(y,one\_of(as.character(list\_varn$varn)))

head(filter\_df)

corrgram(filter\_df,lower.panel=panel.cor,upper.panel=panel.pie, cor.method = "pearson")

summary(lm(data = filter\_df, y ~ .))

xv <- filter\_df %>% select(-y)

pca = prcomp(xv, scale. = T, center = T)

plot(pca, type="l")

summary(pca)

spca = summary(pca)

plot(spca$importance[3,], type="l")

pca\_df <- data.frame(pca$x)

pca\_df <- pca\_df %>% select(-PC7,-PC8)

pca\_df$y = filter\_df$y

corrgram(pca\_df,lower.panel=panel.cor,upper.panel=panel.pie)

model <- lm(data = pca\_df, y ~ .)

summary(model)

PCbiplot <- **function**(PC, x="PC1", y="PC2") {

data <- data.frame( PC$x)

plot <- ggplot(data, aes\_string(x=x, y=y))

datapc <- data.frame(varnames=row.names(PC$rotation), PC$rotation)

mult <- min(

(max(data[,y]) - min(data[,y])/(max(datapc[,y])-min(datapc[,y]))),

(max(data[,x]) - min(data[,x])/(max(datapc[,x])-min(datapc[,x])))

)

datapc <- transform(datapc,

v1 = .7 \* mult \* (get(x)),

v2 = .7 \* mult \* (get(y))

)

plot <- plot + coord\_equal() + geom\_text(data=datapc, aes(x=v1, y=v2, label=varnames), size = 3, vjust=1, color="darkred")

plot <- plot + geom\_segment(data=datapc, aes(x=0, y=0, xend=v1, yend=v2), arrow=arrow(length=unit(0.2,"cm")), alpha=0.5, color="black")

plot

}

PCbiplot(pca)

PCA is a powerful tool to reduce dimensionality or to get a different perspective on your data. At the same time the interpretation of results is more diffcult, but possible e.g. with the biplot. With PCA we do not lose prediction power, but we are able to eliminate collinearity.

1. Quadratic Forms

Let $A = \begin{bmatrix} 4 && 1.5 \\ 1.5 && 7 \end{bmatrix} $ (a symmetric matrix). A quadratic form $x^TAx = 4x\_1^2 + 7x\_2^2 + 3x\_1 x\_2$

For **diagonal matricies**, the resulting equation does not include cross-product terms and is thus, much easier to work with. **Example:**

Let $A = \begin{bmatrix} 4 && 0 \\ 0 && 7 \end{bmatrix} $. A quadratic form $x^TAx = 4x\_1^2 + 7x\_2^2$. We thus want our matrix to be **diagonal**. We can change it to be that way!

Change of Variable

First define a change of variable: $x = Py$ where $P$ is an orthogonal matrix with **orthonormal columns**. We will now see why this is very beneficial and simplifies the problem alot! Now rewrite the equation as:

$x^TAx = (Py)^TA(Py) = y^TP^TA(Py) = y^T(P^TAP)y$

Now it is obvious why a change of variable with $P$ having **orthonormal columns** is nice: We now have $P^TAP = D$ because $A = PDP^T$, where $PDP^T$ is an **eigenvalue decomposition** of $A$.

Now we have $y^TDy$ and the matrix is diagonal! We can get to x again with $x = Py$.

2. Constrained Optimization

I will now describe a very simple constrained optimization problem where we want to maximize the quadratic form $ x^TAx $ subj. to $\lVert x \lVert=1$. This problem has a very interesting solution.

So since $A$ is symmetric, it is **orthogonally diagonizable**. We know from sec. 1: $y^TDy$. So just **create an eigenvalue decomposition**, and let $D$ be the diagonal matrix of eigenvalues, arranged in descending order, with the eigenvectors in $P$ according to $D$ and normalized.

How is $y^TDy$ maximized?

**Example:**

Let $D = \begin{bmatrix} 9 && 0 \\ 0 && 6 \end{bmatrix} $. So $y^TDy = 9y\_1^2 + 6y\_2^2$. The maximum value of the function is reached if $y=\begin{bmatrix}1 \\ 0\end{bmatrix}$. It is 9, the largest eigenvalue of $A$, corresponding to its eigenvector $Py$. So the $x$ that maximizes $x^TAx$ is the eigenvector of the largest eigenvalue of $A$.

**Now we are ready to discuss the SVD and PCA!**

3. The Singular Value Decomposition

Every mxn matrix $A$ can be described in the form $AV=U\Sigma$. So $A=U\Sigma V^T$. I will briefly describe the decomposition.

Singular Values

We want to find vector $v$ of magnitude 1, that maximizes $\lVert Av \lVert$. Since the same vector maximizes $\lVert Av \lVert^2$, and this problem is much easier, we will maximize $\lVert Av \lVert^2$.

**=>** $argmax\_v(\lVert Av \lVert) = argmax\_v(\lVert Av \lVert^2)$

$\lVert Av \lVert^2 = (Av)^T(Av) = v^T A^T Av$

This is a **Quadradic Form** since $A^TA$ is **symmetric**. We know that its unit **eigenvector** maximizes $v^T A^T Av$, with its corresponding **eigenvalue** as the maximum. Therefore $max(\lVert Av \lVert) = \sqrt{max(\lVert Av \lVert^2)}$, which is called **singular value** $\sigma\_1$ of $A$ with the unit eigenvector $v\_1$ from $A^TA$. $A$ has as many singular values as linearly independent columns ($range(A)$).

Finally we have: $\lVert Av\_i \lVert=\sigma\_i$

Now construct $U\Sigma$, which must equal to $AV$.

$dim(A)=$mxn

$dim(U)=$mxm, called left singular vectors.

$dim(\Sigma)=$mxn, diagonal matrix with singular values $\sigma\_i$ of $A$ in descending order.

$dim(V)=$nxn, called right singular vectors (Orthonormal eigenvectors of $A^TA$).

We define the columns of $U$ [$u\_1$,...,$u\_m$] as $u\_i = \frac{Av\_i}{\sigma\_i}$. We define $\Sigma$ to be a **diagonal matrix** with the **singular values** $\sigma$ of A, in descending order, and the eigenvectors in $V$ arranged accordingly.

Therefore $AV = U\Sigma$ and $A = U\Sigma V^T$

4. Relationship between SVD and PCA

I will briefly describe how the **covariance matrix** of $X$ ($C\_x$) is related its SVD. For simplicity, define $X$ to already be in **mean-deviation form**.

$COV(X) = \frac{XX^T}{n-1} = C\_x$

$COV(X) = (U\Sigma V^T)(U\Sigma V^T)^T/(n-1) = U\Sigma V^T V\Sigma^T U^T / (n-1) = U\frac{\Sigma^2}{n-1}U^T = C\_x$

Recall that the goal of PCA is to find a change of variable $X=PY$ for which the new covariance matrix is diagonal. Because $P^TX=Y$:

$COV(Y) = (P^TX)(P^TX)^T/(n-1) = P^T(C\_x(n-1))P/(n-1) = P^TC\_xP = D$

So we have $C\_x = PDP^T$

From the SVD we have $C\_x = U\frac{\Sigma^2}{n-1}U^T$

Both $P$ and $U$ are **orthonormal eigenvectors** of $XX^T$. **Both decompositions differ just by the scaling factor** $n-1$ in the **diagonal matricies** (since the columns of $P$ and $U$ have same length of 1). The eigenvalues $\lambda\_i$ of $D$ are related to the singular values from the **SVD** $\sigma\_i$ via $\lambda\_i = \frac{\sigma\_i}{n-1}$.