Statistical Methods for Big Data

Worksheet 12 Foods

Load the Foods.txt file from Blackboard. The Foods data set was collected to investigate how food consumption patterns varied between different European countries. The relative consumption of 20 different food items was compiled for 16 countries. The values show the percentage of households that normally use each item. Before analysing the data set with PCA let’s explore the data set.

To convert the imported data set into a data frame we use the command:

Foods <- as.data.frame(Foods)

The PCA functions we will be using do not expect the data set to contain categorical variables like the first column of the data set, Foods$Country, however each observation (each row) does need an identifier. We can specify this identifier using the row.names function:

row.names(Foods)<-Foods$Country

this specifies the row names of the Foods data frame

Next we remove the Foods$Country variable from the Foods data table:

Foods$Country<- NULL

First summarize the data set, and then plot histograms to visualize the distribution of each of the 20 variables. Finally, plot scatter graphs to visualize the relationships between some variables of interest (it is too time consuming to plot scatter graphs for every pair of variables)

There are 20 variables so to speed this process along we can use the following code:

windows(20,16)

par(mfrow=c(4,5))

for(i in seq(1, length(Foods), 1)) hist(Foods[,i], xlab = names(Foods[i]), main = names(Foods[i]))

Some of the variables are skewed; PCA assumes that the variables are distributed normally so we may have to apply some transformations. The variable Gr\_Coffe has a strong negative skew. An effective transformation for this variable is to reflect the data so that it becomes positively skewed and then take a log transformation.

To reflect the data you add one to the maximum value of the variable then subtract each value of the variable. Let C denote the maximum value of that variable then the reflected data is given by (C+1 - *xi* )

Foods$Gr\_Coffe<-(-log(max(Foods$Gr\_Coffe)+ 1-Foods$Gr\_Coffe)))

hist(Foods$Gr\_Coffe)

Transform any variables that (severely) violate the assumption of normality.

Now we can use the prcomp function to run a PCA on our data:

pca <- prcomp(Foods, scale. = T, center = T)

summary(pca)

We get the following output (just the first 5 PCs shown here):

Importance of components:

PC1 PC2 PC3 PC4 PC5

Standard deviation 2.5430 1.9833 1.7564 1.19952 1.13646

Proportion of Variance 0.3233 0.1967 0.1542 0.07194 0.06458

The standard deviation is the square root of the eigenvalue associated with each PC and gives a measure of the amount of variation explained by each PC. The proportion and cumulative proportion of variation explained by each PC is also given.

The prcomp function returns an object with the following components:

|  |  |
| --- | --- |
| sdev | the standard deviations of the principal components (i.e., the square roots of the eigenvalues of the covariance/correlation matrix). |
| rotation | the matrix of variable loadings (i.e., a matrix whose columns contain the eigenvectors). |
| x | the value of the rotated data - the centred (and scaled if requested) data multiplied by the rotation matrix is returned. These are the **scores**. |
| center, scale | the centering and scaling used, TRUE or FALSE. |

pca$rotation

The output below just shows the first 4 PCs and all 20 variables.

PC1 PC2 PC3 PC4

Gr\_Coffe -0.14625427 -0.21801357 -0.29352478 -0.40137837

Inst\_Coffe -0.15496740 0.38384686 0.01161348 0.27440930

Tea -0.27448349 0.05393536 0.34685610 0.07412305

Sweetner -0.31149355 -0.15402246 -0.11478088 0.11523641

Biscuits -0.24208323 0.16415845 0.07522063 -0.33140412

Pa\_Soup -0.17733941 0.31709088 0.08458654 0.24655988

Ti\_Soup -0.35730163 0.06602491 0.05965781 -0.04771117

In\_Potat -0.18682314 -0.11953113 -0.26761101 0.21909703

Fro\_Fish -0.18189306 -0.36415168 -0.17319599 0.18487141

Fro\_Veg -0.27722691 -0.28243462 -0.12390608 0.14903635

Apples -0.18575475 0.28650275 -0.28927042 -0.15056751

Oranges -0.20588780 0.12918843 -0.36356578 0.03215752

Ti\_Fruit -0.34855179 0.14987998 -0.04380406 0.11008851

Jam -0.25999726 0.03746992 0.30817822 0.18663200

Garlic 0.24484618 0.18614899 -0.35529232 0.12243865

Butter -0.15498217 0.14223407 0.19305057 -0.40938661

Margarine -0.12509588 -0.05880623 -0.17611574 -0.29820321

Olive\_Oil 0.15622208 0.11299764 -0.18675210 0.32435541

Youghurt -0.05938587 0.30759874 -0.32343064 -0.08426999

Crisp\_Bread -0.16165697 -0.36278820 0.01164160 0.11425573

So PC1 can be written as:

PC1 = -0.15*x*1 – 0.15*x*2 – 0.27*x*3 - 0.31*x*4 + . . . - 0.16*x*20

Let’s examine the scores.

pca$x

The output gives the scores for each of the twenty observations for the first for principal components (remember that the scores represent the position of the observation in relation to the principal components).

PC1 PC2 PC3 PC4 Ger -1.23557520 0.2944933 -0.76003424 0.41004933

Ita 3.88192567 0.2047540 0.52002625 -0.00938182

Fra 0.41217013 3.1144855 -1.68606419 -1.28431848

Hol -2.86832061 1.3725755 -0.33579511 -0.07068100

Bel 0.56876119 1.0230010 -0.86575558 -0.90510351

Lux -2.00110806 1.7759352 -2.72730677 -0.43220676

Eng -3.65907553 1.4248426 1.90527710 0.72462170

Port 4.74385302 -0.6847860 0.02450925 0.55771266

Aus 3.47971668 -0.8381829 0.65876367 0.27088059

Swi -0.18225262 1.2533860 -0.70537701 2.10598309

Swe -2.70382968 -4.0976290 -0.08813597 1.58826481

Den -2.57686513 -2.6751580 -0.66902740 -0.81672248

Nor 0.09758385 -2.2729652 0.40405109 -0.19900437

Fin 0.29895870 -2.2548419 0.39744824 -2.91876534

Spain 2.29951141 0.3627131 -1.29453864 1.25815406

Ire -0.55545380 1.9973770 5.22195933 -0.27948247

We can plot the scores – be sure to open up a new graphics window or your histograms will be overwritten

plot(pca$x, xlim=c(-4.5, 4.5), ylim=c(-4.5,4.5))

however the plot is not particularly informative without labels, try:

text(pca$x, labels=rownames(Foods), cex=0.7)

This is better but the labels are placed directly over the markers, try:

text(pca$x, labels=rownames(Foods), adj=c(0.3, -0.5), cex=0.7)

abline(h=0, v=0)

The adj(0.3, -0.5)argument moves the labels (you can move them into alternative positions by altering the values) .



The scores plot shows the position of each observation (each country) relative to the new principal component axes. Countries near to each other on the scores plot can be considered to have similar eating habits, whereas countries far from each have different eating habits.

To find out which food items are important in determining a county’s position relative to PC1 and PC2 we can look at a plot of the loadings. The loadings plot shows how each variable (each food item) is correlated with the principal components, i.e. which variables contributed to the variation described by each principal component.

Before we plot the loadings, let’s look at them again in a table (first 4 shown below):

> pca$rotation

PC1 PC2 PC3 PC4

Gr\_Coffe -0.14625427 -0.21801357 -0.29352478 -0.40137837

Inst\_Coffe -0.15496740 0.38384686 0.01161348 0.27440930

Tea -0.27448349 0.05393536 0.34685610 0.07412305

Sweetner -0.31149355 -0.15402246 -0.11478088 0.11523641

Biscuits -0.24208323 0.16415845 0.07522063 -0.33140412

Pa\_Soup -0.17733941 0.31709088 0.08458654 0.24655988

Ti\_Soup -0.35730163 0.06602491 0.05965781 -0.04771117

In\_Potat -0.18682314 -0.11953113 -0.26761101 0.21909703

Fro\_Fish -0.18189306 -0.36415168 -0.17319599 0.18487141

Fro\_Veg -0.27722691 -0.28243462 -0.12390608 0.14903635

Apples -0.18575475 0.28650275 -0.28927042 -0.15056751

Oranges -0.20588780 0.12918843 -0.36356578 0.03215752

Ti\_Fruit -0.34855179 0.14987998 -0.04380406 0.11008851

Jam -0.25999726 0.03746992 0.30817822 0.18663200

Garlic 0.24484618 0.18614899 -0.35529232 0.12243865

Butter -0.15498217 0.14223407 0.19305057 -0.40938661

Margarine -0.12509588 -0.05880623 -0.17611574 -0.29820321

Olive\_Oil 0.15622208 0.11299764 -0.18675210 0.32435541

Youghurt -0.05938587 0.30759874 -0.32343064 -0.08426999

Crisp\_Bread -0.16165697 -0.36278820 0.01164160 0.11425573

If we look at the loadings for PC1 we see that the variable Ti\_Fruit has the largest magnitude (-0.350) so we know this variable is important for determining PC1. Similarly, Sweetener and Ti\_Soup are influential. Note that all the food items except Olive\_Oil and Garlic are negatively correlated with PC1 which tells us that Olive\_Oil and Garlic are negatively correlated with the other food items (i.e. countries with a high consumption of olive oil and garlic are likely to have a low consumption of tinned fruit, sweetener, tinned soup, etc). The value for Gr\_Coffe is small, which tells us that consumption of ground coffee is not important for determining a country’s position relative to PC1.

We can look at the loadings for each PC separately:

> loadings <- pca$rotation

> sorted\_loadings <- loadings[order(loadings[,1]),1]

> plot(sorted\_loadings,main = "Loadings Plot for PC1", ylim = c(-0.4, 0.3), xlab="Variable Loadings")

> text(sorted\_loadings, labels = names(sorted\_loadings), adj = c(0.3, -0.5), cex = 0.7)



Look at the loadings for PC2 and decide which variables are influential, then plot them.

To create a loadings plot showing PC1 and PC2:

> plot(pca$rotation[,1], pca$rotation[,2], xlab=”PC1”, ylab = “PC2”, xlim = c(-0.5, 0.5), ylim=c(-0.5,0.5))

> text(pca$rotation[,1], pca$rotation[,2], labels= colnames(Foods), adj = c(0.3, -0.5), cex = 0.7)



We can also use the function loadingplot from the package pls

Tools…> Install packages>….pls

> library(pls)

> ?loadingplot

> loadingplot(pca, scatter=T, xlim=c(-0.5, 0.5), ylim = c(-0.5, 0.5), labels = names(Foods))

Note that the pls package can also plot the scores.

So far we have only considered the first two principal components but it is possible that we need more than two PCs to describe the data satisfactorily.

To decide how many PC we need we look at the eigenvalues and the proportions of variance:

> summary(pca)

to create a scree plot:

> screeplot(pca)

> screeplot(pca, type=”lines”)

How many PCs should we retain if we follow the Kaiser criterion?

How many PCs should we retain if we only retain PCs that account for at least 5% of the variance?

How many PCs should we retain if we wish to account for 90% of the variance?

We can examine score plots and loading plots of different combinations of principal components:

> scoreplot(pca, comps =c(1,3), labels = “names”, xlim = c(-4,4), ylim = c(-4,4))

> abline(v=0, h=0)



The scores plot of the first and third principal component shows that the position of Ireland and England relative to the third principal component separates them from the other countries. To find out which foods are driving this we look at the loadings plot.

> loadingplot(pca, scatter = TRUE, comps =c(1,3), labels = “names”, xlim = c(-4,4), ylim = c(-0.5,0.5))

> abline(v=0, h=0)



We see that the variables, Jam, Tea , Oranges, and Gr\_Coffe are correlated with PC3, indicating that Ireland and England have higher tea and jam consumption and lower orange and ground coffee consumption than the other countries.

To check the data set for outliers we can plot a confidence ellipse around the scores. The dataEllipse function in the car package can be used to do this.

dataEllipse(pca$x[,1], pca$x[,2], levels =0.95, xlim =c(-7,7), ylim=c(-6,6), xlab=”PC1”, ylab=”PC2”)

abline(h=0, v=0)

Finally we will create bar charts to illustrate how well each variable is explained by a PCA model with

1. 1 principal component ()
2. 2 principal components )
3. 5 principal components )
4. First we calculate the values predicted by using a model with just 1 principal component using the identity X = TPT + E:

We can calculate the matrix containing the standardised data, X, using the scale function:

> X<- scale(Foods)

Remember that when using prcomp the scores, T, are given by pca$x and the loadings, P, are given by pca$rotation.

> X\_pred1 <- pca$x[,1]%\*%t(pca$rotation[,1])

Here pca$x[,1], gives just the first column of the scores and pca$rotation[,1] gives just the first column of the loadings (since we are using just 1 PC).

The symbols %\*% tell R that we wish to perform matrix multiplication, as opposed to element wise multiplication.

The function t() transposes a matrix.

The residual matrix E is given by:

> E<- X – X\_pred1

The squared residuals are given by:

> E\_sq<-E\*E

(Note the element wise multiplication)

To calculate we use the formula

R2X1 <-1 – colSums(E\_sq)/colSums(X\*X)

We can plot these in a bar chart

> barplot(R2X1, las=2, ylab=”R2X1”, ylim = c(0,1), cex.names = 0.8)

The argument las=2 places the labels perpendicular to the x-axis and the argument cex.names = 0.8 specifies the size of the labels.

Repeat this process for cases (b) and (c).

Hint: to calculate the values predicted by using a model with 2 principal components use:

> X\_pred2 <- pca$x[,1:2]%\*%t(pca$rotation[,1:2])

since this will included PC1 and PC2 in the model. If you wish to find out how well each variable is explained by PC2 only (it can be useful to find out which variables are explained by which components) then use:

> X\_pred2\_only <- pca$x[,2]%\*%t(pca$rotation[,2])