

IST772 Summary Template: Chapter 12 – Dealing with Too Many Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

Name: Hendi Kushta
Date: 16/04/2023

****Important** Copying and/or pasting anything from the textbook will not be acceptable for your chapter notes submissions. You must write your notes in your own words and generate your own code, results, and graphs in R. This is what forces your brain to process the material that you read.**

INTRODUCTION

In actual data analysis scenarios, several measurements of the same thing are frequently available. In previous chapters, analyses were performed on individual variables. A multi-item rating scale that is frequently used in surveys is one example. Researchers can utilize principle components analysis, a tool that reorganizes the underlying correlation structure among a group of measurements and shows it in a compact manner, to integrate these various measurements into a single composite measurement. As a result of principal component analysis, primary components—new variables that reflect the input variables and capture their shared variance—are created.

This methodology, along with other exploratory factor analysis methods like factor analysis and independent components analysis, is crucial for dimension reduction in big data applications since it makes the study simpler by merging variables into composites. The simplest of these exploratory factor analysis methods is addressed in this chapter, principal components analysis.

The statistical method of principle components analysis (PCA) is investigated using the iris data set. Four measurements of iris flowers from a total of 150 plants are included in the data set. If various measurements are variations of the same underlying source of variance, such as overall plant size, PCA can assist in identifying this source of variance. If so, it is possible to combine the measurements to provide a more accurate metric of plant size. Horticulturists researching the effects of soil, water, and light on blossom growth, for example, can use this composite index as a dependent variable in future analyses or model construction.

IST772 Summary Template: Chapter 12 – Dealing with Too Many Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

```
##{r}
str(iris)
##
```

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

The iris data set comprises information on the iris flower's petal length, petal width, sepal length, and sepal width measurements in addition to a factor designator with three levels named "Species". The factor won't be taken into account for the principle components analysis (PCA) that is being done. Since the PCA process only accepts numeric variables, a copy of the data frame will be made without the factor to ensure compatibility.

```
##{r}
irisN <- subset(iris,select=-Species) # Remove the species designator
str(irisN)
##
```

```
'data.frame':  150 obs. of  4 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

Before beginning the principal components analysis, a full correlation matrix for the variables in the iris data set will be examined. The correlations have been rounded to three significant digits to enhance visual interpretability.

```
##{r}
round(cor(irisN,digits=3) # Show correlation matrix for the iris data
##
```


	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.000	-0.118	0.872	0.818
Sepal.Width	-0.118	1.000	-0.428	-0.366
Petal.Length	0.872	-0.428	1.000	0.963
Petal.Width	0.818	-0.366	0.963	1.000

The iris data set's variables' correlation matrices are evaluated, paying close attention to the direction and strength of each correlation. Petal.Length and Petal.Width are highly and positively connected with Sepal.Length, however Sepal.Width is only somewhat correlated with these two dimensions. Petal.Length and Petal.Width exhibit a very high degree of correlation. The relationship between Sepal.Width and the other factors seems weaker. The "principal()"

IST772 Summary Template: Chapter 12 – Dealing with Too Many Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

technique from the "psych" package will be used to do the principal components analysis because it supports the needed features. A new data object called irisNout will contain the analysis's output.

```
{r}
# install.packages("psych")
library(psych)
irisNout <- principal(irisN)
irisNout
```



The image shows two side-by-side windows from an R environment. The left window is titled 'R Console' and contains the R code used for the principal components analysis. The right window is titled 'data.frame' and shows the output of the 'principal()' function, which is a 4x4 data frame.

Principal Components Analysis
Call: principal(r = irisN)
Standardized loadings (pattern matrix) based upon correlation matrix

	PC1
SS loadings	2.92
Proportion Var	0.73

Mean item complexity = 1
Test of the hypothesis that 1 component is sufficient.

The root mean square of the residuals (RMSR) is 0.13
with the empirical chi square 28.19 with prob < 7.6e-07


Fit based upon off diagonal values = 0.97

The results of the principle components analysis (PCA) show how the input variables are rearranged in terms of variance and covariance. The first principal component, or PCA algorithm, like "principal()" in R, aims to synthesis a new variable from the input variables that accounts for their shared variance. Each input variable's standardized loading in the PC1 column indicates how strongly it is related to the first principal component; larger loadings denote stronger relationships. There is further information regarding the contributions of each input variable in other columns, such as h2 (communality), u2 (uniqueness), and com (complexity index).

The loadings and the percentage of variance that is accounted for by the principal component (Proportion Var), which is comparable to an R-squared value, are often the factors that analysts focus on the most. Some crucial diagnostics include the goodness of fit test and the sum of squares (SS Loadings). With the "nfactors=2" parameter in the R call to the "principal()" procedure, a two-component solution is investigated because, in the example given, the one-component solution does not enough account for the input variables' variation to be regarded as a satisfactory fit.

IST772 Summary Template: Chapter 12 – Dealing with Too Many Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

```
{r}
irisNout <- principal(irisN, nfactors=2)
irisNout
```



The image shows two small windows from an R environment. The 'R Console' window displays the command `irisNout <- principal(irisN, nfactors=2)` and the output `irisNout`. The `data.frame` window shows a 4 x 5 matrix of standardized loadings for the four iris variables across two principal components (RC1 and RC2).

Principal Components Analysis
Call: `principal(r = irisN, nfactors = 2)`
Standardized loadings (pattern matrix) based upon correlation matrix

	RC1	RC2
SS loadings	2.70	1.13
Proportion Var	0.68	0.28
Cumulative Var	0.68	0.96
Proportion Explained	0.71	0.29
Cumulative Proportion	0.71	1.00

Mean item complexity = 1.1
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0.03
with the empirical chi square 1.72 with prob < NA

Fit based upon off diagonal values = 1

The findings show that a two-component solution is adequate, with two principal components accounting for almost all of the variance in the original four variables. The first component is responsible for 68% of the variance, and the second component is responsible for another 28%. According to the first and second component loadings, Sepal.Length, Petal.Length, and Petal.Width all heavily influence the first component, although Sepal.Width has a negative first component loading and a positive second component loading. Small loadings are also present for Petal.Length and Petal.Width on the second component. Sepal.Length, Petal.Length, and Petal.Width can be merged into a single composite indicator of floral size to achieve dimension reduction. Before merging them, it is crucial to take into account the scales on which each item is assessed, as items measured on several scales cannot simply be added or averaged. A composite can be produced by averaging the items in the case of a multi-item rating scale with the same minimum and maximum values, such as a 5-point rating scale, and comparable standard deviations. The things can also be added together, however this is less typical. Before aggregating the items in the iris data set using the `summary()` command, the ranges of the elements should be verified.

IST772 Summary Template: Chapter 12 – Dealing with Too Many Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

```
##{r}  
summary(irisN)  
##
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300
Median :5.800	Median :3.000	Median :4.350	Median :1.300
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500

The summary() command reveals that each item has a substantially different scale range, with Sepal.Length ranging from 4.3 to 7.9, and Petal.Width ranging from 0.1 to 2.5. To overcome this issue, standardization can be applied to each variable.

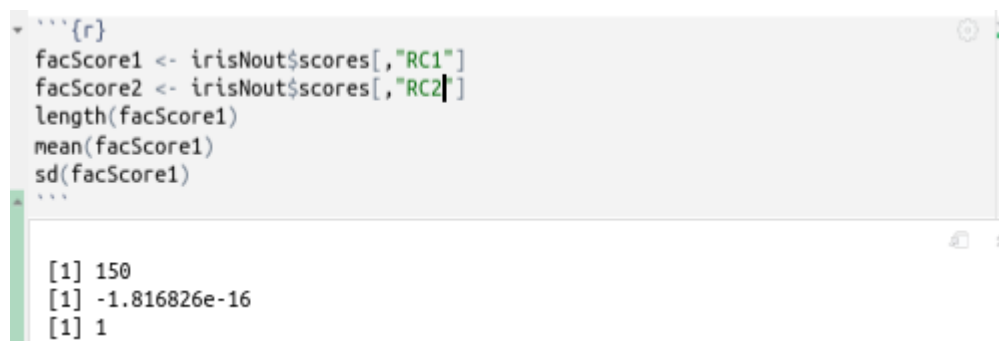
```
##{r}  
irisNS <- scale(irisN) # standardize each variable  
flowerSize <- (irisNS[,1]+ irisNS[,3]+ irisNS[,4])/3 # All except Sepal.Width  
length(flowerSize) # Check the vector  
mean(flowerSize) # Examine the mean  
sd(flowerSize) # And the standard deviation  
##
```

```
[1] 150  
[1] -1.782016e-16  
[1] 0.9606202
```

By averaging the standardized values of the three aforementioned variables, a new composite variable named "flowerSize" has been created. According to the output of the length() command, "flowerSize" contains the same number of observations (n = 150) as the original data set. When calculating the mean of standardized variables, it is assumed that the composite variable's mean will be very close to zero, as shown by the mean() statement. Similar to this, the sd() statement reveals that the composite variable's standard deviation is nearly 1. You can try linking "flowerSize" to other factors like Sepal as an exercise. Length to evaluate how strong the connections are. Think about if you anticipate a high or low correlation between the composite variable and the other factors before proceeding.

BOX ON P.281: MEAN COMPOSITIES VERSUS FACTOR SCORES

Principal components analysis involves calculating component/factor scores, which can be obtained using the loadings of the items onto the principal components. The terms "component score" and "factor score" are often used interchangeably, although "component score" is more commonly used in the context of principal components analysis, while "factor score" is more commonly used in other forms of factor analysis. With the iris data set containing $n = 150$ observations, we can use the output of the principal components command to obtain two vectors of $n = 150$ scores each, one for principal component one and one for principal component two.



```
##{r}
facScore1 <- irisNout$scores[, "RC1"]
facScore2 <- irisNout$scores[, "RC2"]
length(facScore1)
mean(facScore1)
sd(facScore1)
##
```

```
[1] 150
[1] -1.816826e-16
[1] 1
```

In summary, component/factor scores in principal component analysis are standardized values resulting from the loadings of items onto principle components. When the input variables have similar loadings, averaging their standardized values might produce a composite score, like the "f lowerSize" in this example. However, because it generalizes well to new samples and eliminates influence from disregarded input factors, adopting equal weights (by taking the mean of items) is frequently advised for generating composite scales for survey questions.

Component/factor scores are also a preferable option when the loadings of the input variables fluctuate significantly, such as when each variable contributes differently to the component/factor score. Such circumstances may lead to an unjust weighting of variables when a mean composite is created.

Component/factor scores can also be a good option if generalizability to new samples is not a problem.

When determining whether to create a composite score or use component/factor scores in statistical analyses, it is crucial to take the specific characteristics of the data and the current research issue into account.

Originality Assertion: By submitting this file you affirm that this writing is your own.

```
## {r}
cor(facScore1, flowerSize)
cor(facScore2, flowerSize)|
##
[1] 0.9840672
[1] -0.1776343
```

The component scores for the first principal component, `facScore1`, and the mean composite, `f lowerSize`, have a very high correlation ($r = 0.98$), which suggests that in this instance, the two approaches are yielding results that are almost equal. The second factor score, on the other hand, is mostly influenced by `Sepal.Width`, which was not taken into account while calculating the mean composite, hence the correlation between the second factor score and the mean composite is minimal ($r = -0.18$).

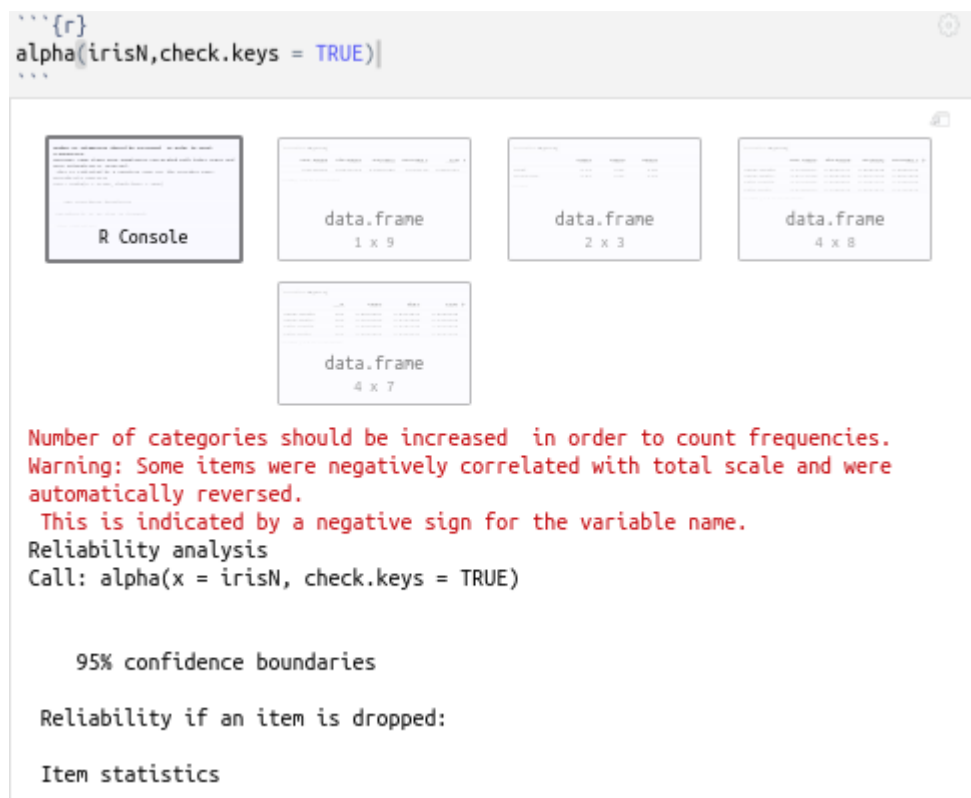
Based on these findings, the mean composite would be a good option if a combination of `Sepal.Length`, `Petal.Length`, and `Petal.Width` is required for a future study. Nonetheless, `facScore1` might be adequate if the combination was only required once. When choosing between employing component scores or a mean composite in statistical analyses, it's crucial to take the unique research topic and environment into account.

INTERNAL CONSISTENCY RELIABILITY

For multi-item scales, Cronbach's alpha reliability is a regularly used indicator of internal consistency that evaluates the coherence of a set of items in relation to their intercorrelations. More internal consistency is indicated by a higher Cronbach's alpha, which also signals that future investigations of the composite made up of the items are likely to be helpful. An alpha value of 0.70 is regarded as the minimum acceptable level in the social sciences, with 0.80 or higher being highly favored. The permissible alpha value, however, may differ depending on the scale length, with shorter scales necessitating higher item-item correlations to attain acceptable alpha values.

The `alpha()` function from the `psych` package can be used to conduct an alpha analysis and evaluate the internal consistency of the variables in the context of the iris data. The reliability of the `Sepal.Width` variable and its contribution to the composite measure can be understood through this analysis.

Originality Assertion: By submitting this file you affirm that this writing is your own.

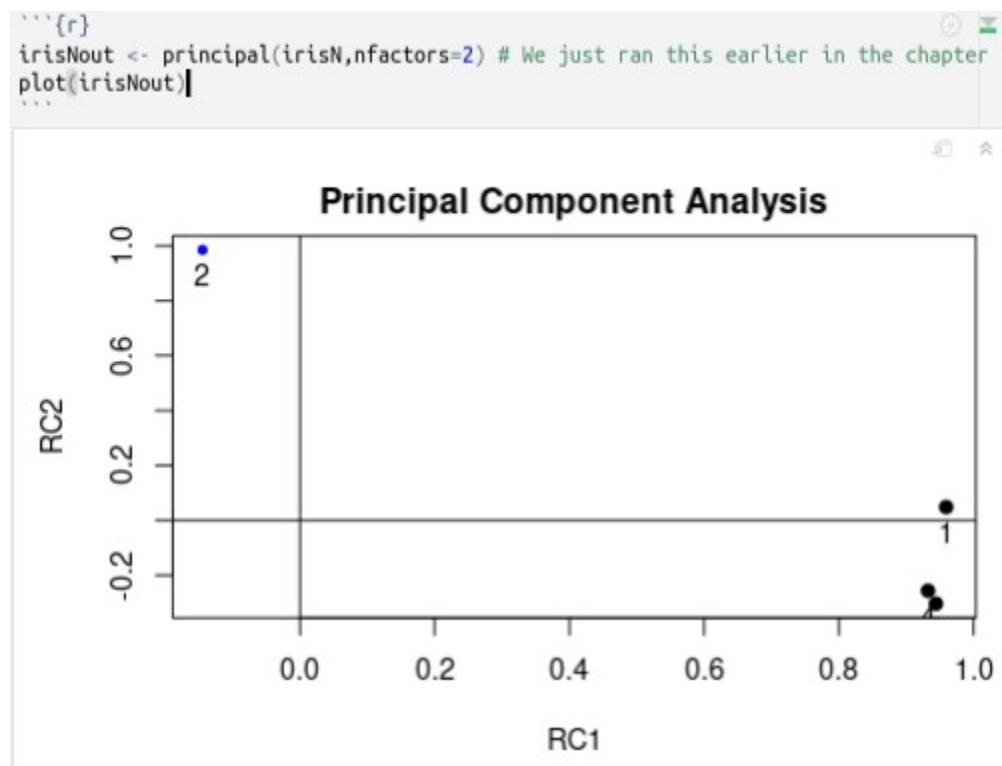


The error notice emphasizes how crucial the `alpha()` function is for addressing negatively correlated elements when generating composite scores. All four iris elements were included despite prior knowledge from main components analysis, and `alpha()` automatically flipped the sign of Sepal. width for a precise evaluation. Despite losing Sepal, overall alpha reliability was 0.81. It went up to 0.88 with width, which supported the exclusion. Principal() and `alpha()` are both exploratory approaches, and decision-making is required for scale generation and dimension reduction. Testing assumptions is advised even for scales that have already been published, and goodness-of-fit tests and confidence intervals can help with decision-making.

ROTATION

The `main()` approach produced loadings for the iris dataset that were simple to understand. One input variable and three input variables each had significant loadings on the first major component. Cross-loadings, a measure of how heavily variables are loaded across several components, were minimal. A plot command can help to visualize these results and make them easier to grasp.

Originality Assertion: By submitting this file you affirm that this writing is your own.

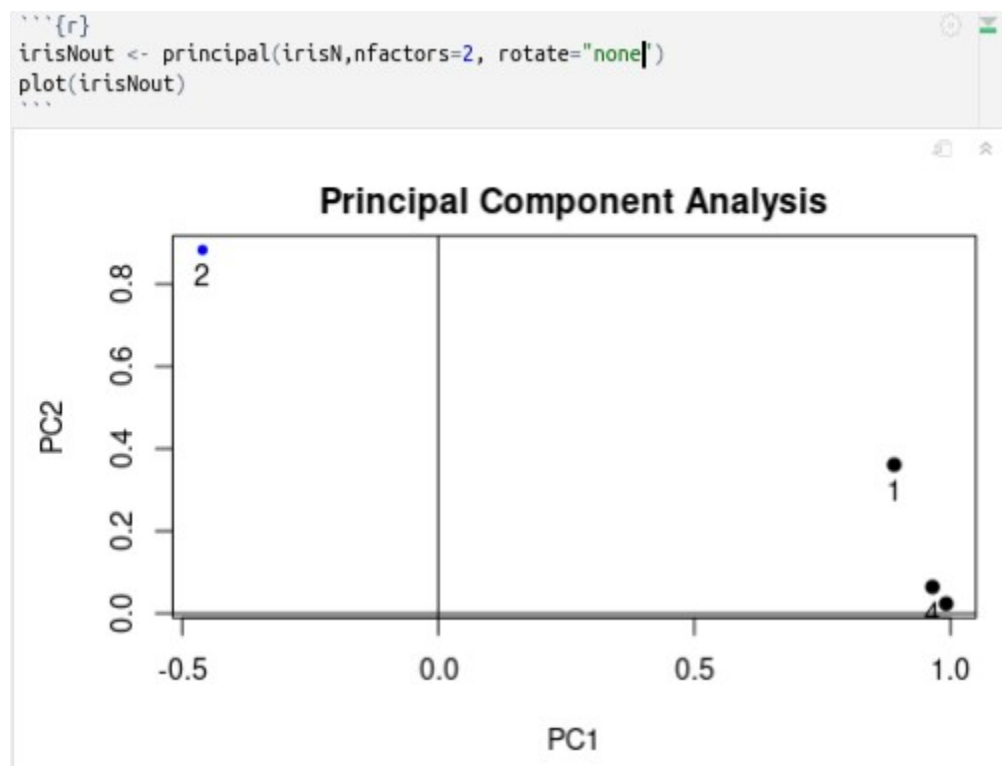


The loading plot for the iris dataset employing a two-component solution is shown in Figure. According to the plot, Sepal.Length, Petal.Length, and Petal.Width have large loadings on the first main component (X-axis), while the second component has low loadings (Y-axis). A similar pattern can be seen in Sepal.Width, which has a strong loading on the second component and a low loading on the first. From an interpretive perspective, the plot is deemed optimal because it presents a distinct picture of the two components without any significant cross-loadings.

Rotation—more precisely, varimax rotation, which is the primary() command's default rotation technique—is employed to get the desired results. Maximizing the variety in the item loadings allows for Varimax rotation, which modifies the loadings to make them as comprehensible as feasible. By stretching out the X and Y axes sideways from the body to define them, and then bending sideways at the waist to see how the coordinates change while keeping the axes at right angles to one another, the idea of rotating axes can be shown. Similar modifications are made via varimax rotation to achieve the perfect alignment of the loading coefficients.

When running the main operation, one can specify rotate="none" to view the original coordinates before rotation.

Originality Assertion: By submitting this file you affirm that this writing is your own.



In conclusion, Figure displays a loading plot for the iris data set with loadings for Sepal using a two-component solution. Petal length. Size and Petal. A clean picture of the two components without any cross-loadings is indicated by width that is close to 1 on the X-axis and close to 0 on the Y-axis. Varimax rotation, a technique for modifying the loadings to make them as comprehensible as feasible, achieves this.

In order to maximize the variance in the item loadings, Varimax rotates the axes, producing clear and understandable solutions. Due to its ease of use and interpretability, the Varimax rotation, which keeps the axes at right angles to one another, is frequently employed in exploratory factor analysis. Alternative rotation techniques, like oblimin, allow for non-orthogonal axes but can make interpretation more difficult. Most of the time, varimax rotation meets the needs of analysts.