MSc Business Analytics

Financial Modelling and Analysis

Chapter 3. Factor Models and Principal Components (Multivariate analysis)

Instructor: Roman Matkovskyy

Twitter: @matkovskyy

Outline

- Dimension Reduction
- PCA
- Factor models: economic and statistical models

Principal Components Analysis

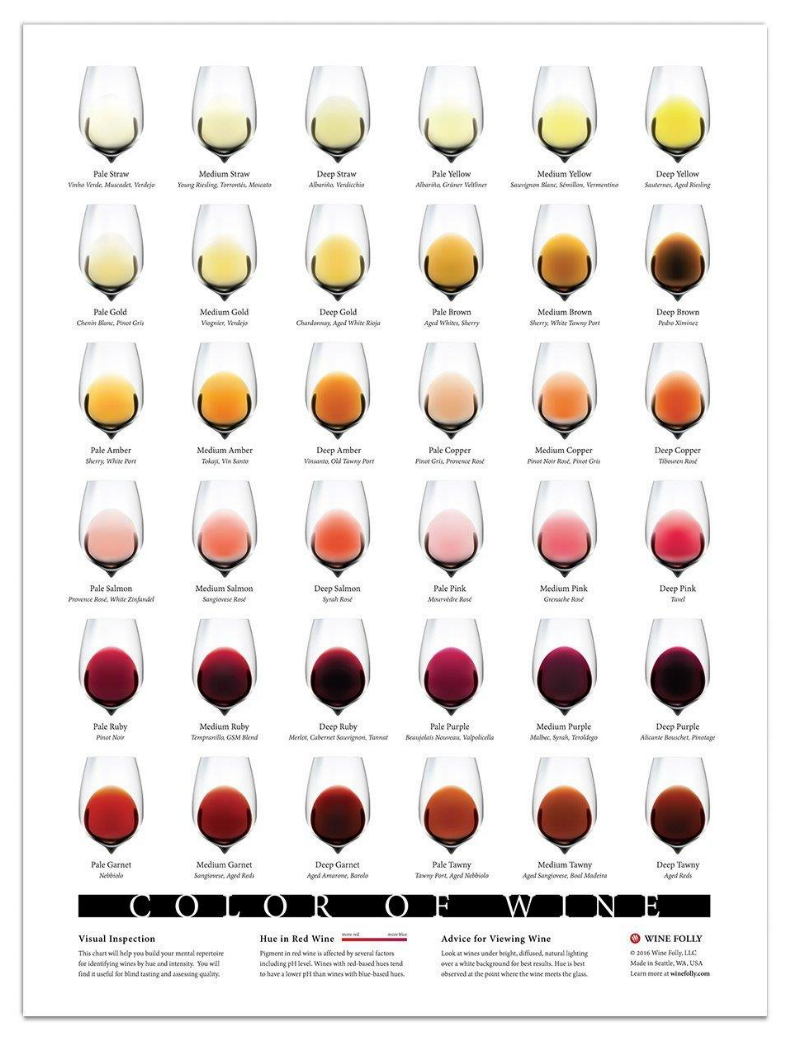
- One of the problems with a lot of sets of multivariate data is that there are simply too many variables
- Having too many variables can also cause problems for other multivariate techniques that the researcher may want to apply to the data.
  - The possible problem of too many variables is sometimes known as the *curse of dimensionality* (Bellman 1961) [hat the error increases with the increase in the number of features]

Dimension Reduction

- *High-dimensional data* can be challenging to analyse.
  - difficult to visualize,
  - need extensive computer resources,
  - often require special statistical methodology.
- Fortunately, in many practical applications, high-dimensional data have most of their variation in a lower-dimensional space that can be found using dimension reduction techniques.
- There are many methods designed for dimension reduction, and in this session we will study two closely related techniques, *factor analysis* and *principal components analysis*, often called PCA.
- **PCA finds structure in the covariance or correlation matrix and uses this structure to locate low-dimensional subspaces containing most of the variation in the data**.
- **Factor analysis explains returns with a smaller number of fundamental variables called *factors or risk factors***.
  - Factor analysis models can be classified by the types of variables used as factors, macroeconomic or fundamental, and by the estimation technique, time series regression, cross-sectional regression, or statistical factor analysis.

What is PCA in simple words

- it's just a method of summarizing some data.
- For instance, one can have some wine bottles standing on the table.
- We can describe each wine by *its colour*, by *how strong it is*, by *how old it is*, and so on.
- One can create a whole list of different characteristics of each wine.
  - But many of them will measure related properties and so will be redundant.
- If so, we should be able to summarize each wine with fewer characteristics. This is what PCA does.



Visual Inspection

This chart will help you build your mental repertoire for identifying wines by hue and intensity. You will find it useful for blind tasting and assessing quality.

Hue in Red Wine

Pigment in red wine is affected by several factors including pH level. Wines with red-based hues tend to have a lower pH than wines with blue-based hues.

Advice for Viewing Wine

Look at wines under bright, diffused, natural lighting over a white background for best results. Hue is best observed at the point where the wine meets the glass.

© 2016 Wine Folly, LLC
Made in Seattle, WA, USA
Learn more at winefolly.com

Does PCA check what characteristics are redundant and discards them?

- No, *PCA is not selecting some characteristics and discarding the others*.
  - Instead, ***it constructs some new characteristics that turn out to summarize our list of wines well*.**
  - These *new characteristics are constructed using the old ones*; for example, a new characteristic might be computed as wine age minus wine acidity level or some other combination like that (we call them *linear combinations*).
  - *PCA finds the best possible characteristics, the ones that summarize the list of wines as well as only possible* (among all conceivable linear combinations).

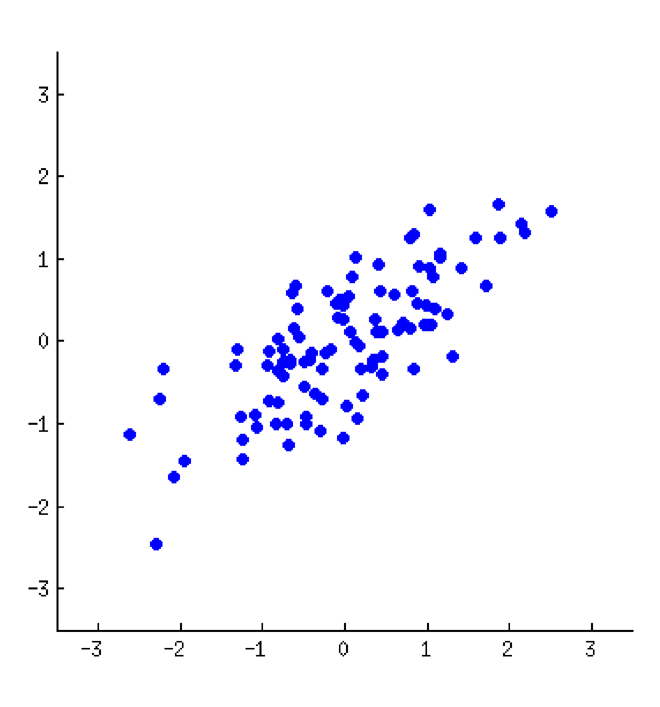
How does PCA summarise the characteristics?

- First, **you are looking for some wine properties (characteristics) that strongly differ across wines.**
  - Indeed, imagine that you come up with a property that is the same for most of the wines.
  - This would not be very useful? Wines are very different, but your new property makes them all look the same! This would certainly be a bad summary. Instead, PCA looks for properties that show as much variation across wines as possible.

- Second, **you look for the properties that would allow you to predict, or "reconstruct", the original wine characteristics**.
  - Again, imagine that you come up with a property that has no relation to the original characteristics; if you use only this new property, there is no way you could reconstruct the original ones! This, again, would be a bad summary.
  - So PCA looks for properties that allow to reconstruct the original characteristics.
- <u>These two items are actually equivalent</u>.

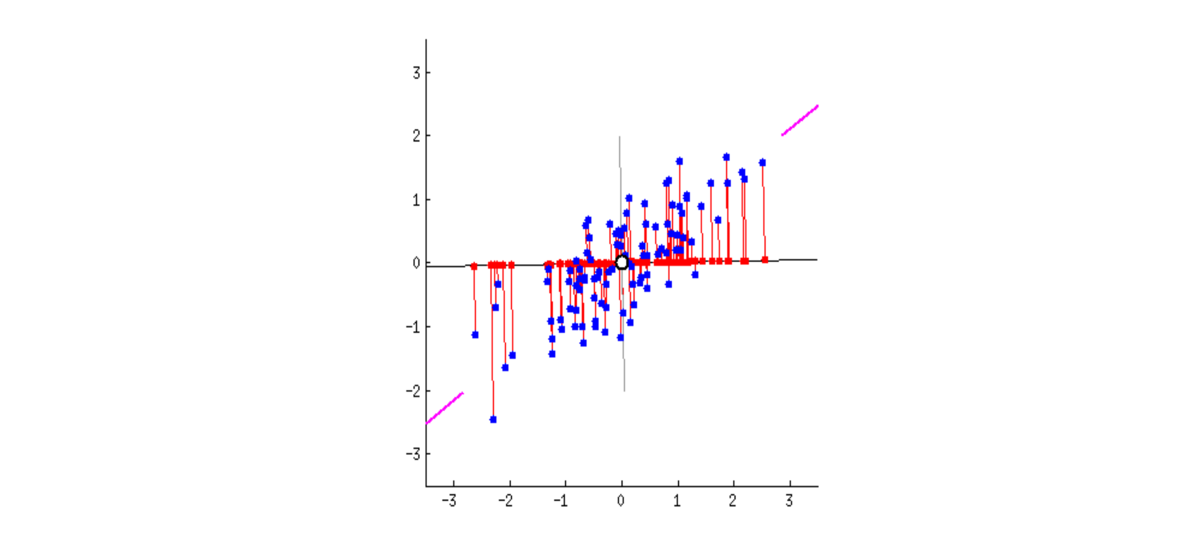
Why do they are equivalent?

- Let us pick two wine characteristics, perhaps wine darkness and alcohol content.
- Let's imagine that they are correlated.
- Here is what a scatter plot of different wines could look like this Fig.
- Each dot in this "wine cloud" shows one particular wine.
- One can see that the two properties (x and y on this figure) are correlated.
- A new property can be constructed by drawing a line through the center of this wine cloud and projecting all points onto this line.
- This new property will be given by a linear combination w1x+w2y, where each line corresponds to some particular values of w1 and w2.



Why do they are equivalent?

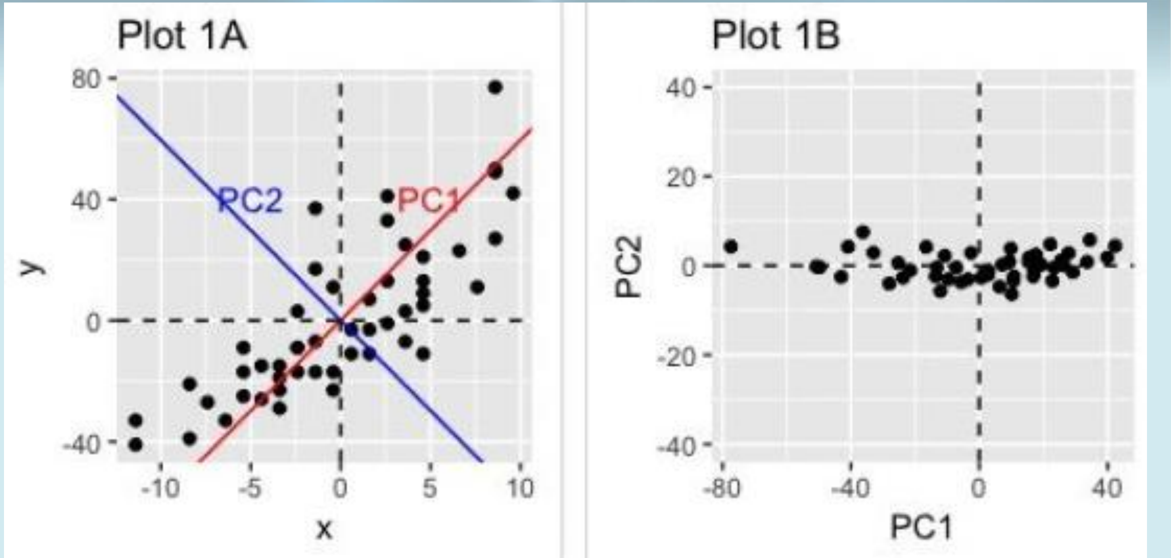
- Now look here very carefully -- here is how these projections look like for different lines (red dots are projections of the blue dots)
- *PCA will find the "best" line according to two different criteria of what is the "best".*
- **First, the variation of values along this line should be maximal.** Pay attention to how the "spread" ("variance") of the red dots changes while the line rotates; can you see when it reaches maximum?
- **Second, if we reconstruct the original two characteristics (position of a blue dot) from the new one (position of a red dot), the reconstruction error will be given by the length of the connecting red line**.
- Observe how the length of these red lines changes while the line rotates; can you see when the total length reaches minimum?



If you stare at this animation for some time, you will notice that "the maximum variance" and "the minimum error" are reached at the same time. This line corresponds to the new wine property that will be constructed by PCA

Principal Components Analysis

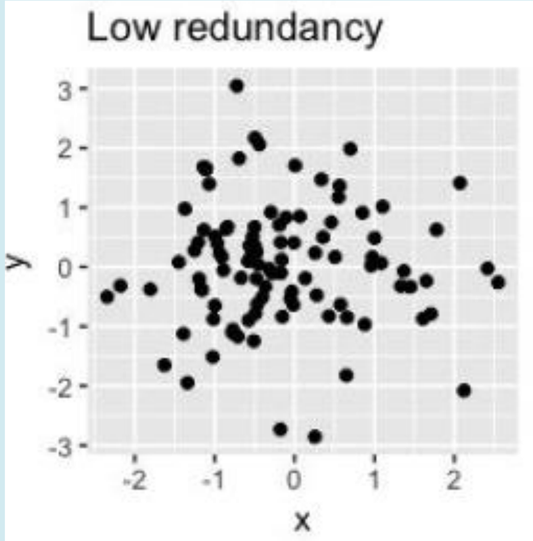
- Understanding the details of PCA requires knowledge of linear algebra. Here, we'll explain only the basics with simple graphical representation of the data.
- In the Plot 1A, the data are represented in the X-Y coordinate system.
- **The dimension reduction is achieved by identifying the principal directions, called *principal components*, in which the data varies**.
- **PCA assumes that the directions with the largest variances are the most "important" (i.e, the most principal).**
- In the figure, the *PC1 axis* is the **first principal direction** along which the samples show the largest variation.
- The *PC2 axis* is the **second most important direction** and it is **orthogonal** to the PC1 axis.
- The dimensionality of our two-dimensional data can be reduced to a single dimension by projecting each sample onto the first principal component (Plot 1B)
- Technically speaking, the amount of variance retained by each principal component is measured by the so-called **eigenvalue.**

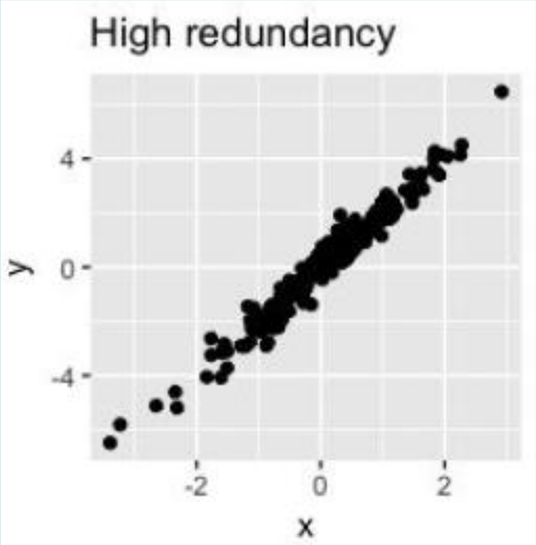


- Note that, **the PCA method is particularly useful when the variables within the data set are highly correlated**.
- Correlation can indicate that there is **redundancy in the data**.
- Due to this redundancy, PCA can be used to reduce the original variables into a smaller number of new variables (= **principal components**) explaining most of the variance in the original variables.

Principal Components Analysis

- Taken together, the main purpose of principal component analysis is to:
  - identify hidden pattern in a data set,
  - reduce the dimensionnality of the data by **removing the noise** and **redundancy** in the data,
  - identify correlated variables





Math behind Principal Components Analysis (optional)

- PCA starts with a sample $\boldsymbol{Y}\,i = (Y_{i,1}, \ldots, Y_{i,d})$, $i = 1, \ldots, n$, of d-dimensional random vectors with **mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$**.
- One goal of PCA is finding “structure” in $\boldsymbol{\Sigma}$.
- We will start with a simple example that illustrates the main idea.
- Suppose that $\boldsymbol{Y}_i = \boldsymbol{\mu} + W_i\boldsymbol{o}$, where $W_1, \ldots, W_n$ are i.i.d. (Independent and identically Distributed) mean-zero random variables and $\boldsymbol{o}$ is some fixed vector, which can be taken to have norm 1 (***1-norm for a vector is sum of absolute values; 2-norm is the usual Euclidean norm - square root of the sum of the squares of the values***)
- The $\boldsymbol{Y}_i$ lie on the line that passes through $\boldsymbol{\mu}$ and is in the direction given by $\boldsymbol{o}$, so that all variation among the mean-centered vectors $\boldsymbol{Y}i - \boldsymbol{\mu}$ is in the one-dimensional space spanned by $\boldsymbol{o}$.
- The covariance matrix of $\boldsymbol{Y}\,i$ is

$$\Sigma = E\{W_i^2 oo^T\} = \sigma_W^2 oo^T$$

- The vector $\boldsymbol{o}$ is called *the first principal axis* of $\boldsymbol{\Sigma}$ and is the only eigenvector of $\boldsymbol{\Sigma}$ with a nonzero eigenvalue, so $\boldsymbol{o}$ can be estimated by an eigen-decomposition of the estimated covariance matrix (not study in this course).

Math behind Principal Components Analysis (optional), cont.

- A slightly more realistic situation is where $\boldsymbol{Y}_i = \boldsymbol{\mu} + W_i\boldsymbol{o} + e_i$, where e_i is a random vector uncorrelated with W_i and having a “small” covariance matrix.
- Then most of the variation among the $\boldsymbol{Y}_i - \boldsymbol{\mu}$ vectors is in the space spanned by $\boldsymbol{o}$, but there is small variation in other directions due to i.
- PCA can be applied to either the sample covariance matrix or the correlation matrix.
- We will use $\boldsymbol{\Sigma}$ to represent whichever matrix is chosen. The correlation matrix is, of course, the covariance matrix of the standardized variables, so the choice between the two matrices is really a decision whether or not to standardize the variables before PCA.
- This issue will be addressed later. Even if the data have not been standardized, to keep notation simple, we assume that the mean $\boldsymbol{Y}$ has been subtracted from each $\boldsymbol{Y}i$.

Math behind Principal Components Analysis (optional), cont.

$$\boldsymbol{\Sigma} = \boldsymbol{O}\mathrm{diag}(\lambda_1, \ldots, \lambda_d)\boldsymbol{O}^{\mathrm{T}}$$

- where ***O*** is an orthogonal matrix whose columns $o_1, \ldots, o_d$ are the eigenvectors of ***Σ*** and $\lambda_1 > \ldots > \lambda_d$ are the corresponding eigenvalues.
- The columns of ***O*** have been arranged so that the eigenvalues are ordered from largest to smallest.
- This is not essential, but it is convenient. We also assume no ties among the eigenvalues, which almost certainly will be true in actual applications.
- A *normed linear combination* of Y_i(either standardized or not) is of the form $a^TY_i = \sum_{j=1}^{p} a_jY_{i,j}$ where $\|a\| = \sqrt{\sum_{j=1}^{p} a_i^2} = 1$
- The first principal component is the normed linear combination with the greatest variance.
- The variation in the direction ***α***, where ***α*** is any fixed vector with norm 1, is $Var(a^TY_i) = a^T\Sigma_a$that we maximize over ***α.***
- The maximizer is ***α*** = ***o***1, the eigenvector corresponding to the largest eigenvalue, and is called the first principal axis.
- The projections $o_1^TY_i$, *i* = 1*, . . . , n*, onto this vector are called the first principal component or principal component scores.
- After the first principal component has been found, one searches for the direction of maximum variation perpendicular to the first principal axis (eigenvector).
- This means maximizing $Var(a^TY_i) = a^T\Sigma_a$ subject to $\|a\| = 1$ and $a^To_1 = 0$.
- The maximizer, called the second principal axis, is o_2, and the second principal component is the set of projections $o_2^TY_i$, *i* = 1*, . . . , n*, onto this axis.
- The third principal component maximizes $Var(a^TY_i) = a^T\Sigma_a$ subject $\|a\| = 1$ and $a^To_1 = 0$ and $o_2 = 0$ and is $o_3^TY_i$, and so forth, so that ***o***1*, . . . ,* ***od*** are the principal axes and the set of projections $o_j^TY_i$, *i* = 1*, . . . , n*, onto the *j*th eigenvector is the *j*th principal component.

Math behind Principal Components Analysis (optional), cont.

$$\lambda_i = o_i^T \Sigma o_i$$

- is the variance of the *i*th principal component, $\lambda_i/(\lambda_1+\cdot\cdot\cdot+\lambda_d)$ is the proportion of the variance due to this principal component, and $(\lambda_1+\cdot\cdot\cdot+\lambda_i)/(\lambda_1+\cdot\cdot\cdot+\lambda_d)$ is the proportion of the variance due to the first *i* principal components.
- The principal components are mutually uncorrelated.
- Let $Y = \begin{pmatrix} Y_1^T \\ \vdots \\ Y_n^T \end{pmatrix}$, be the original data and let $S = \begin{pmatrix} o_1^T Y_1 & \dots & o_d^T Y_1 \\ \vdots & \ddots & \vdots \\ o_1^T Y_n & \dots & o_d^T Y_n \end{pmatrix}$ be the matrix of principal components. Then *S*=*YO*.
- Postmultiplication of ***Y*** by ***O*** to obtain ***S*** is an orthogonal rotation of the data. For this reason, the eigenvectors are sometimes called the *rotations*, e.g., in output from R's pca() function

Original or the standardized variables.

- **If the components of *Yi* are comparable, e.g., are all daily returns on equities or all are yields on bonds, then working with the original variables should cause no problems.**
- **However, if the variables are not comparable, e.g., one is an unemployment rate and another is the GDP in dollars, then some variables may be many orders of magnitude larger than the others.**
- In such cases, the large variables could completely dominate the PCA, so that the first principal component is in the direction of the variable with the largest standard deviation.
- To eliminate this problem, one should **standardize the variables**.
- Standardize value of **xi = (xi-mean of x)/std deviation** (it is done automatically by the software)

Example: Principal components analysis of yield curves

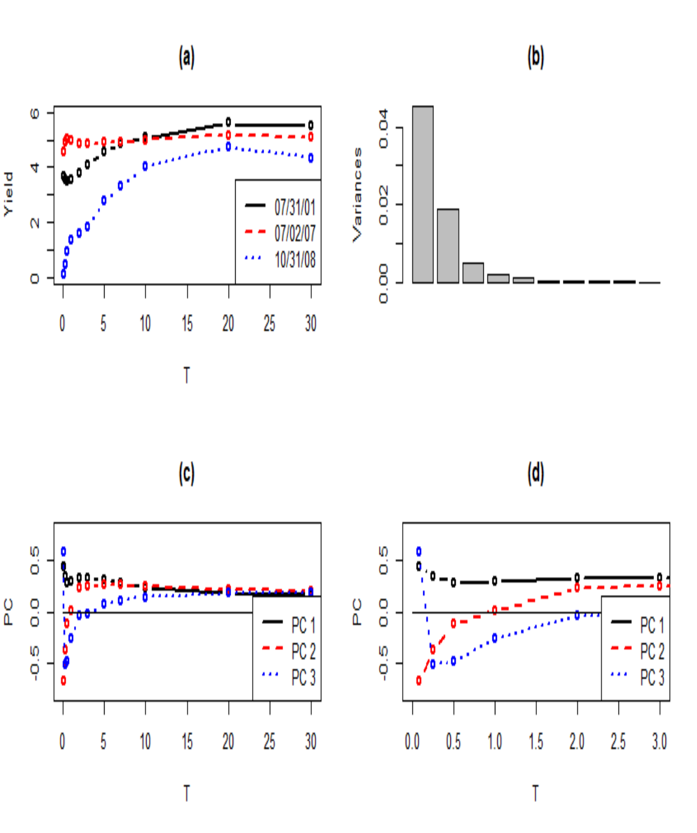
This example uses yields on Treasury bonds at 11 maturities, T = 1, 3, and 6 months and 1, 2, 3, 5, 7, 10, 20, and 30 years. Daily yields were taken from a U.S. Treasury website for the time period January 2, 1990, to October 31, 2008.

First, we will look at the 11 eigenvalues using R’s function prcomp(). The code is:

```
datNoOmit = read.table("treasury_yields.txt",header=T) # import the dataset
diffdatNoOmit = diff(as.matrix(datNoOmit[,2:12])) # diff the data
dat=na.omit(datNoOmit) # omit all NAs in initial dataset
diffdat = na.omit(diffdatNoOmit) # omit all NAs in diff dataset
n = dim(diffdat)[1] # a number of observations

pca = prcomp(diffdat) # Performs a principal components analysis on the given data matrix and returns the results as an object of class prcomp.
summary(pca) # return the statistics
> summary(pca) # return the statistics

Importance of components:
                         PC1   PC2    PC3    PC4    PC5     PC6     PC7     PC8     PC9    PC10    PC11
Standard deviation     0.213 0.136 0.0715 0.0449 0.0333 0.01726 0.01400 0.01078 0.00924 0.00789 0.00610
Proportion of Variance 0.622 0.255 0.0699 0.0275 0.0152 0.00408 0.00268 0.00159 0.00117 0.00085 0.00051
Cumulative Proportion  0.622 0.876 0.9464 0.9739 0.9891 0.99320 0.99588 0.99747 0.99864 0.99949 1.00000
>
```

Example: Principal components analysis of yield curves , cont.

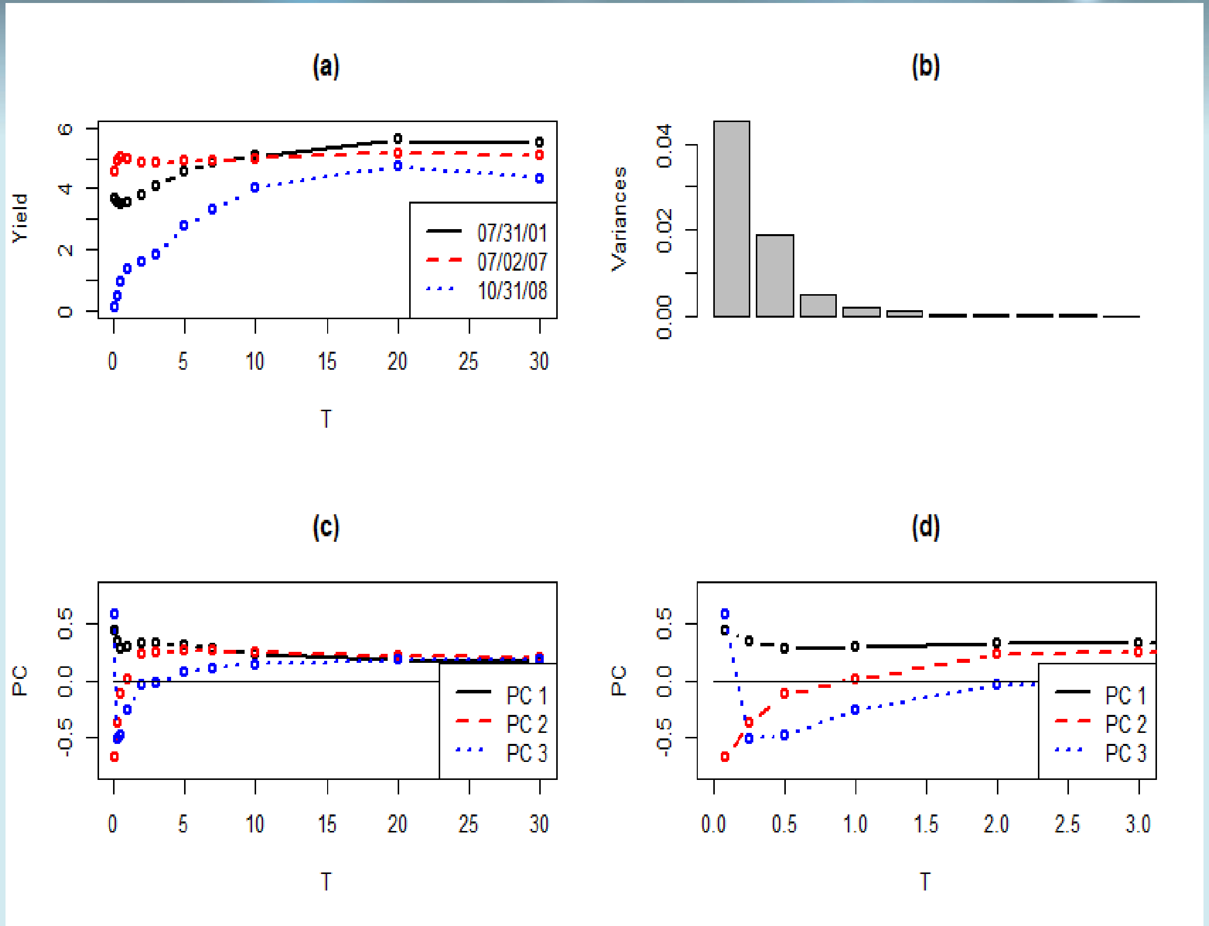
```
The results are:
Importance of components:
                         PC1   PC2    PC3    PC4    PC5     PC6     PC7     PC8     PC9    PC10    PC11
Standard deviation     0.213 0.136 0.0715 0.0449 0.0333 0.01726 0.01400 0.01078 0.00924 0.00789 0.00610
Proportion of Variance 0.622 0.255 0.0699 0.0275 0.0152 0.00408 0.00268 0.00159 0.00117 0.00085 0.00051
Cumulative Proportion  0.622 0.876 0.9464 0.9739 0.9891 0.99320 0.99588 0.99747 0.99864 0.99949 1.00000
```



- The first row gives the values of $\sqrt{\lambda_i}$, a square root of the variance of the ith principal component, the second row the values of **$\lambda_i/(\lambda_1+\cdots+\lambda_d)$ –** the proportion of variance due to this principal component**,** and the third row the values of **$(\lambda_1+\cdots+\lambda_i)/(\lambda_1+\cdots+\lambda_d)$** for $i = 1, \ldots, 11$ is the proportion of the variance due to the first *i* principal components.
- One can see, for example, that the standard deviation of the first principal component is 0.21 and represents **62%** of the total variance.
- Also, the first three principal components have **94.6%** of the variation, and this increases to **97.4%** for the first four principal components and to **98.9%** for the first five. The variances (the squares of the first row) are plotted in Fig. b. This type of plot is called a “scree plot” since it looks like scree, fallen rocks that have accumulated at the base of a mountain.
- We will concentrate on the first three principal components since approximately 95% of the variation in the changes in yields is in the space they span.
- The eigenvectors, labeled “PC,” are plotted in the Fig. c and d, the latter showing detail in the range $T \leq 3$. The eigenvectors have interesting interpretations.
- The first, has all positive values. A change in this direction either increases all yields or decreases all yields, and by roughly the same amounts.

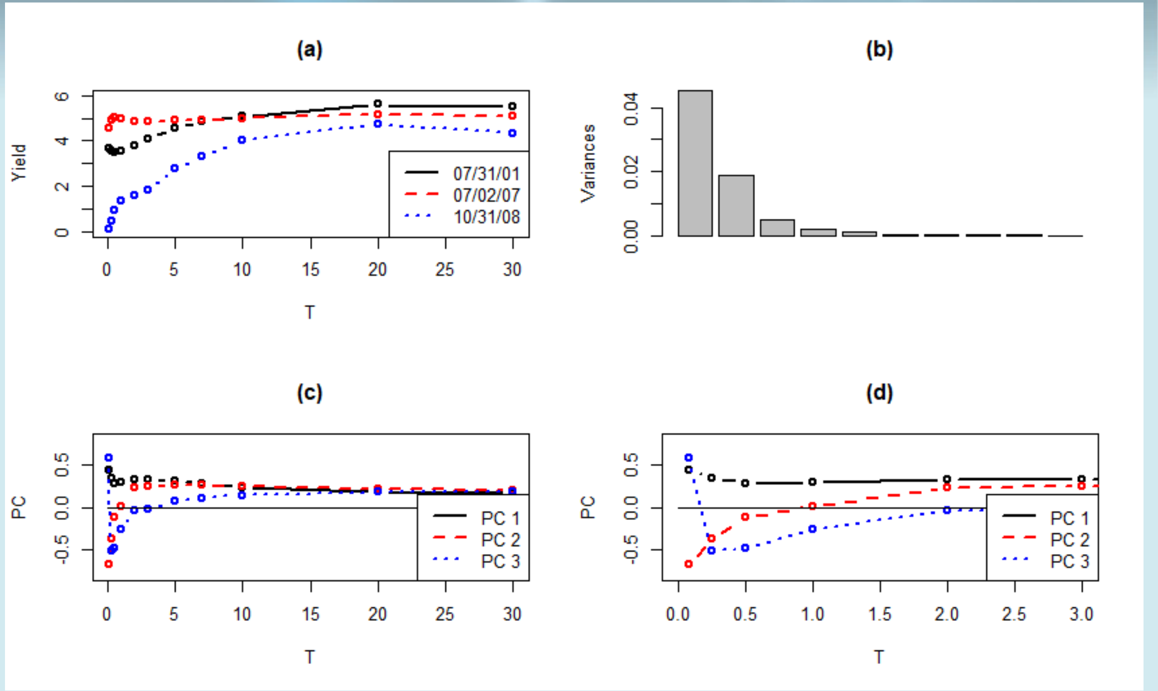
Example: Principal components analysis of yield curves, cont.

- The yield curves are shown in Fig. for three different dates.
- Notice that the yield curves can have a variety of shapes. In this example, we will use PCA to study how the curves change from day to day.



Example: Principal components analysis of yield curves , cont.

- To analyze daily changes in yields, all 11 time series were differenced.
- Daily yields were missing from some values of *T.*
- Differencing caused a few additional days to have missing values.
- In the analysis, all days with missing values of the differenced data were omitted.
- This left 819 days of data starting on July 31, 2001, when the one-month series started and ending on October 31, 2008, with the exclusion of the period February 19, 2002 to February 2, 2006 when the 30-year Treasury was discontinued.
- The covariance matrix, not the correlation matrix, was used, because in this example the variables are comparable and in the same units.



(a) Treasury yields on three dates. (b) Scree plot for the changes in Treasury yields. Note that the first three principal components have most of the variation, and the first five have virtually all of it. (c) The first three eigenvectors for changes in the Treasury yields. (d) The first three eigenvectors for changes in the Treasury yields in the range $0 \le T \le 3$

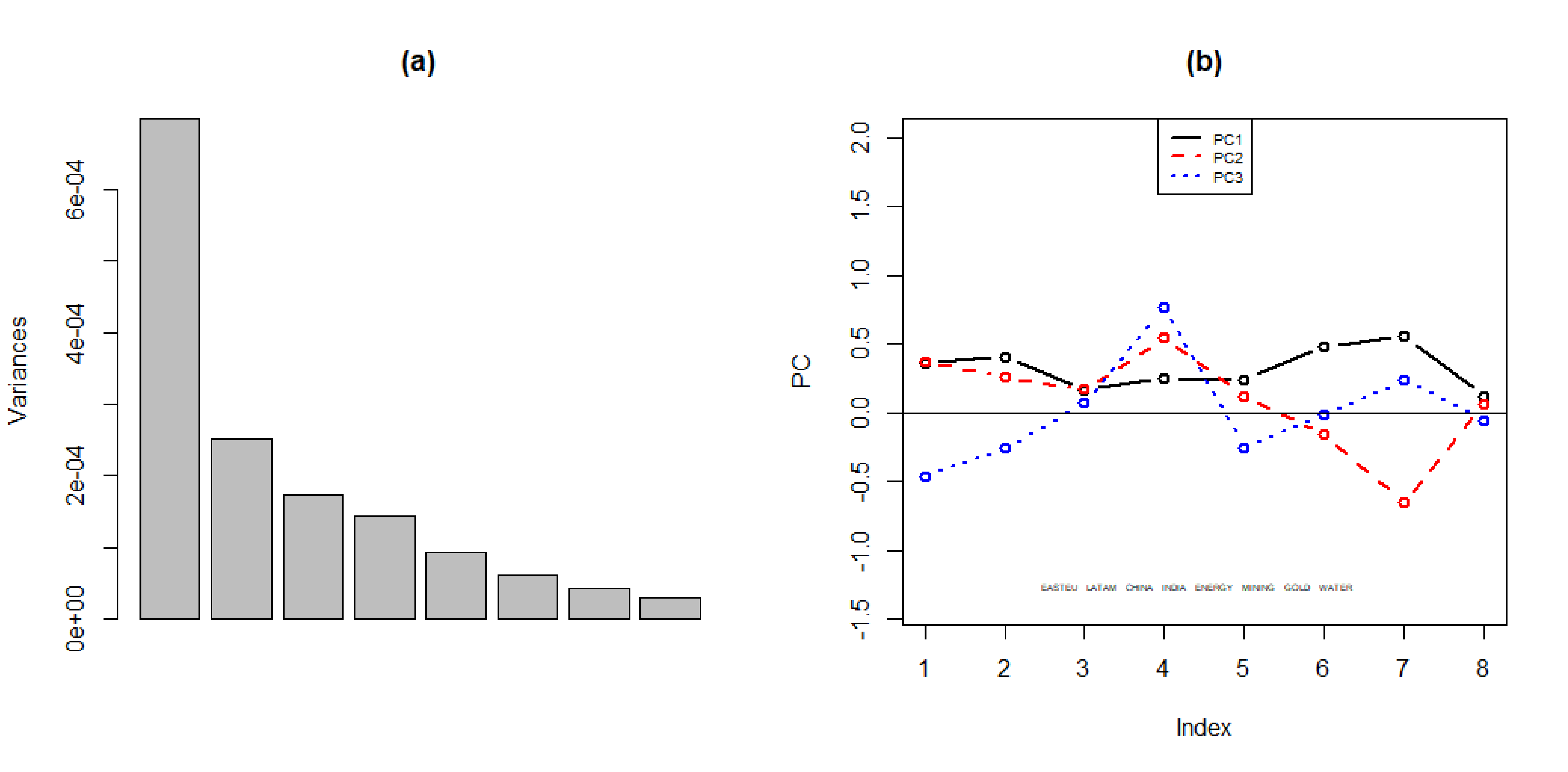
Example: Principal components analysis of equity funds

- This example uses the data set equityFunds.csv. The variables are daily returns from January 1, 2002 to May 31, 2007 on eight equity funds: EASTEU (1), LATAM (2), CHINA (3), INDIA (4), ENERGY (5), MINING (6), GOLD (7), and WATER (8). The following code was run:

```
equityFunds = read.csv("equityFunds.csv")
pcaEq = prcomp(equityFunds[ , 2:9])
summary(pcaEq)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8
Standard deviation     0.0264 0.0158 0.0132 0.0120 0.00969 0.00786 0.00647 0.00548
Proportion of Variance 0.4670 0.1676 0.1165 0.0968 0.06271 0.04129 0.02797 0.02008
Cumulative Proportion  0.4670 0.6346 0.7511 0.8480 0.91067 0.95196 0.97992 1.00000
```

- The results in this example are different than those for the changes in yields, because in this example the variation is less concentrated in the first few principal components.
- For example, the first three principal components have only 75% of the variance, compared to 95% for the yield changes. For the equity funds, one needs six principal components to get 95 %.

Example: Principal components analysis of equity funds

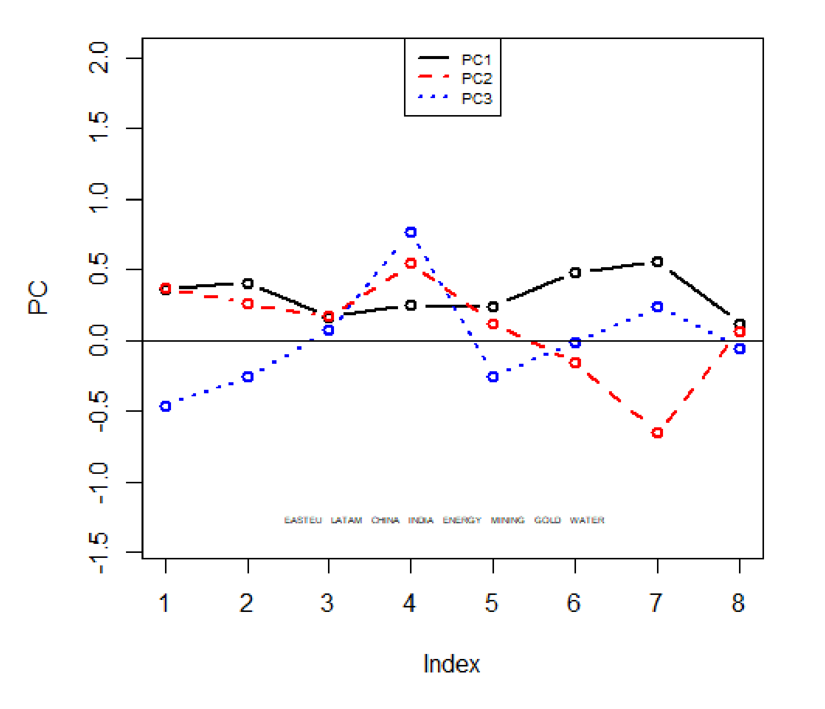


Example: Principal components analysis of equity funds

Importance of components:

```
                         PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8
Standard deviation     0.0264 0.0158 0.0132 0.0120 0.00969 0.00786 0.00647 0.00548
Proportion of Variance 0.4670 0.1676 0.1165 0.0968 0.06271 0.04129 0.02797 0.02008
Cumulative Proportion  0.4670 0.6346 0.7511 0.8480 0.91067 0.95196 0.97992 1.00000
```

- The first three eigenvectors are plotted in Fig.
- The first eigenvector has only positive values, and returns in this direction are either positive for all of the funds or negative for all of them.
- The second eigenvector is negative for mining and gold (funds 6 and 7) and positive for the other funds. Variation along this eigenvector has mining and gold moving in the opposite direction of the other funds.
- Gold and mining stock moving counter to the rest of the stock market is a common occurrence and, in fact, these types of stock often have negative betas, so it is not surprising that the second principal component has 17% of the variation. (*A negative beta simply means that the stock is inversely correlated with the market.*)
- The third principal component is less easy to interpret, but its loading on India (fund 4) is higher than on the other funds, which might indicate that there is something different about Indian equities.



Exercise: *Principal components analysis of the Dow Jones 30 – do it yourself*

- Analyse returns on the 30 stocks on the Dow Jones average.
- The data are in the data set DowJone30.csv and cover the period from January 2, 1991 to January 2, 2002
- Interpret the results
- How many principal components do explain more 95% of variance?
- Why are the Dow Jones stocks behaving differently compared to the equity funds?

Exercise: *Principal components analysis of the Dow Jones 30. Solution*

- Analyse returns on the 30 stocks on the Dow Jones average.
- The data are in the data set DowJone30.csv and cover the period from January 2, 1991 to January 2, 2002

```
##Exercise: the Dow Jones 30
DowJones30 = read.csv("DowJones30.csv")
pcaDJ = prcomp(DowJones30[,2:31])
summary(pcaDJ)
```

- In contrast to the analysis of the equity funds where six principal components were needed to obtain 98% of the variance, here the first three principal components have over 95% of the variance.
- Why are the Dow Jones stocks behaving differently compared to the equity funds?
  - The Dow Jones stocks are similar to each other since they are all large companies in the United States.
  - Thus, we can expect that their returns will be highly correlated with each other and a few principal components will explain most of the variation

Factor model

Factor Models

- We will start with a factor model for excess equity returns*

$$R_{j,t} = \beta_{0,j} + \beta_{1,j}F_{1,j} + \cdots + \beta_{p,j}F_{p,j} + \varepsilon_{j,t}$$

- where $R_{j,t}$ is either the return or the excess return* on the *j*th asset at time *t*, $F_{1,j}, \ldots, F_{p,j}$ are variables, called *factors* or *risk factors*, that represent the "state of the financial markets and world economy" at time *t*,
- $\varepsilon_{1,t}, \ldots, \varepsilon_{n,t}$ are uncorrelated, mean-zero random variables called the ***unique risks*** of the individual stocks.
- *The assumption that unique risks are uncorrelated means that all cross-correlation between the returns is due to the factors*.
- Notice that the factors do not depend on *j* since they are common to all returns.
- The parameter $\boldsymbol{\beta_{i,j}}$ **is *called a factor loading* and specifies the sensitivity of the *j*th return to the *i*th factor.**
- Depending on the type of factor model, either the loadings, the factors, or both the factors and the loadings are unknown and must be estimated.

**For example, if the one year Treasury has returned 2.0% and the technology stock Facebook has returned 15% then the excess return achieved for investing in Facebook is 13%*

Factor Models, cont

- A *factor* can be any variable thought to affect asset returns. Examples of factors include:

1. returns on the market portfolio;
2. growth rate of the GDP;
3. interest rate on short term Treasury bills or changes in this rate;
4. inflation rate or changes in this rate;
5. interest rate spreads, for example, the difference between long-term Treasury bonds and long-term corporate bonds;
6. return on some portfolio of stocks, for example, all U.S. stocks or all stocks with a high ratio of book equity to market equity — this ratio is called BE/ME in Fama and French (1992, 1995, 1996);
7. the difference between the returns on two portfolios, for example, the difference between returns on stocks with high BE/ME values and stocks with low BE/ME values.

   - *The ratio compares a firm's book value to its market value*. A company's book value is calculated by looking at the company's historical cost, or accounting value. *A high book-to-market ratio might mean that the market is valuing the company's equity cheaply compared to its book value.*

- **With enough factors, most, and perhaps all, commonalities between assets should be accounted for in the model. Then the $\varepsilon_{j,t}$ should represent factors truly unique to the individual assets and therefore should be uncorrelated across *j* (across assets), as is being assumed.**
- Factor models that use macroeconomic variables such as 1–5 as factors are called *macroeconomic factor models*.
- *Fundamental factor models* use observable asset characteristics (fundamentals) such as 6 and 7 as factors. Both types of factor models can be fit by time series regression or cross-sectional regression.

Fitting Factor Models by Time Series Regression. *A macroeconomic factor model*

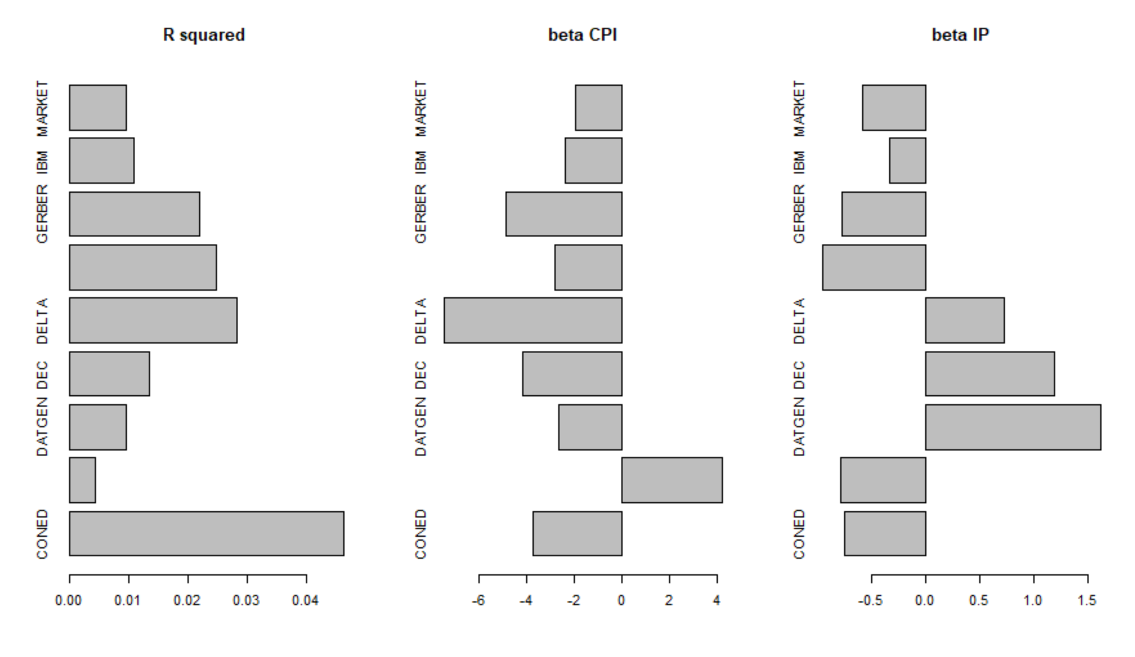
- **The efficient market hypothesis implies that stock prices change because of new information.**
- Although there is considerable debate about the extent to which markets are efficient, *one still can expect that stock returns will be influenced by unpredictable changes in macroeconomic variables*.
- Accordingly, ***the factors in a macroeconomic model are not the macroeconomic variables themselves, but rather the residuals*** *when changes in the macroeconomic variables are predicted from past data by a time series model.*

Fitting Factor Models by Time Series Regression. *A macroeconomic factor model*

- The macroeconomic variables in this example are changes in the logs of CPI (Consumer Price Index) and IP (Industrial Production).
- Monthly returns on nine stocks were taken from the berndtInvest.csv data set. The returns are from January 1978 to December 1987.
- The CPI and IP series from July 1977 to December 1987 were used, but the month of July 1977 was lost through differencing. This left enough data (the five months August 1977 to December 1977) for forecasting CPI and IP beginning January 1978 when the return series started.
- A bivariate AR model was fit. We will use the residuals from the AR(5) model as the factors.

```
berndt = as.matrix(berndtInvest[,-1]) #1978-01-01 to 1987-12-01
CPI.dat = read.csv("CPI.dat.csv")
IP.dat = read.csv("IP.dat.csv")
berndt = as.matrix(berndtInvest[,-1])   #  1978-01-01 to 1987-12-01
CPI2 = as.matrix(CPI.dat$CPI[775:900])# 1977-07-30  to 1987-12-31
CPI = as.data.frame(diff(log(CPI2)))
names(CPI)[1]="CPI" # name the first column
IP2 = as.matrix(IP.dat$IP)[703:828,] #1977-07-28 to 1987-12-28
IP = as.data.frame(diff(log(IP2)))
names(IP)[1] = "IP“ # name the first column
CPI_IP = cbind(CPI,IP)

arFit = ar(cbind(CPI,IP)) # autoregressive model
res = arFit$resid[6:125,] # residuals
lmfit = lm(berndt[,2:10]~res[,1]+res[,2])
slmfit = summary(lmfit)
rsq=rep(0,9) #create a variable rsq with nine 0 values
for (i in 1:9){
     rsq[i]= slmfit[[i]][[8]]
} # we extract the values of R2 from each of 9 models

beta_CPI = lmfit$coef[2,] # extract all CPI betas (b1)
beta_IP = lmfit$coef[3,] # extract all IP betas (b2)
```

Fitting Factor Models by Time Series Regression. *A macroeconomic factor model*

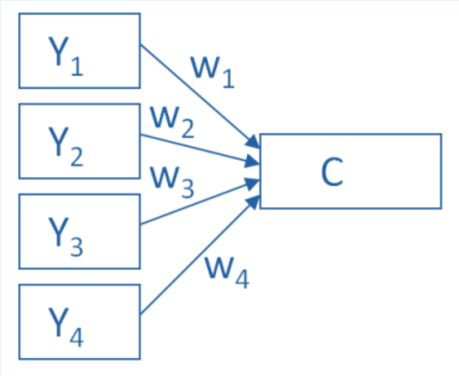


```
res = arFit$resid[6:125,] # residuals of the AR (5) modle (the
interpretations of the residuals is that they are unexpected shocks)

lmfit = lm(berndt[,2:10]~res[,1]+res[,2]) #fit a regression Y is
a set of teh stock returns and X1 and X2 - residuals from the AR modles

slmfit = summary(lmfit) #summary of the regression

rsq = rep(0,9) #create a variable rsq with nine 0 values

for (i in 1:9){
          rsq[i]= slmfit[[i]][[8]]
} # we extract the values of R2 from each of 9 models

beta_CPI = lmfit$coef[2,] # extract all CPI betas (b1)
beta_IP = lmfit$coef[3,] # extract all IP betas (b2)

par(mfrow=c(1,3)) # building three graphs/bars in a row
barplot(rsq,horiz=T,names=names(beta_CPI),main="R squared")
barplot(beta_CPI,hori=T,main="beta CPI")
barplot(beta_IP,hori=T,main="beta IP")
```

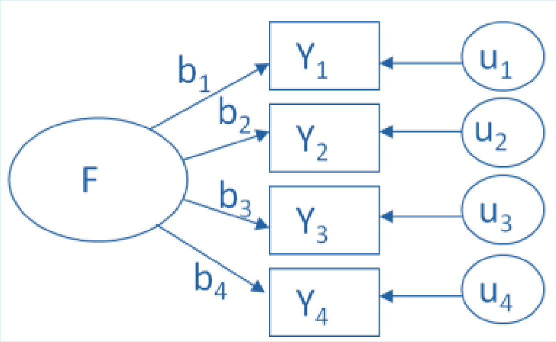
- R_2 and the slopes for the regressions of the stock returns on the CPI residuals and the IP residuals are plotted in Fig. for each of the 9 stocks.
- the R_2-values are very small, so the macroeconomic factors have little explanatory power (that is common for this type of the models)
- For this reason, fundamental factor models are more widely used than macroeconomic models.

PCA vs Statistical factor model

- **PCA's approach to data reduction is to create one or more index variables from a larger set of measured variables.**
- It does this *using a linear combination* (basically a weighted average) of a set of variables.
- The created index variables are *called components.*
- The whole point of the PCA is to figure out how to do this in an optimal way: the optimal number of components, the optimal choice of measured variables for each component, and the optimal weights. This model can be set up as a simple equation: C = w1(Y1) + w2(Y2) + w3(Y3) + w4(Y4)



- **A Factor Analysis - it is a model of the measurement of a latent variable.**
- **This latent variable cannot be directly measured with a single variable** (think: intelligence, social anxiety). Instead, it is seen through the relationships it causes in a set of Y variables.
- The measurement model for a simple, one-factor model looks like the diagram below.
- F, the latent Factor, is causing the responses on the four measured Y variables. So the arrows go in the opposite direction from PCA.
- Just like in PCA, the relationships between F and each Y are weighted, and the factor analysis is figuring out the optimal weights.
- In this model we have is a set of error terms. These are designated by the u's. This is the variance in each Y that is unexplained by the factor.



Statistical Factor Models

- In a statistical factor model, neither the factor values nor the loadings are directly observable.
- All that is available is the sample $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ or, perhaps, only the sample covariance matrix.
- This is the same type of data available for PCA and we will see that statistical factor analysis and PCA have some common characteristics.
- As with PCA, one can work with either the standardized or unstandardized variables.
- R's `factanal()` function automatically standardizes the variables.

Statistical Factor Models (optional explanation)

- We start with the multifactor model in matrix notation and the return covariance matrix which for convenience will be $R_t = \beta_0 + \beta^T F_t + \epsilon_t$ and $\Sigma_R = \beta^T \Sigma_F \beta + \Sigma_\epsilon$
- Here β^T is *d*×*p* where *d* is the dimension of *Rt* and *p* is the number of factors.
- The only component of $\Sigma_R = \beta^T \Sigma_F \beta + \Sigma_\epsilon$ that can be estimated directly from the data is Σ_R. One can use this estimate to find estimates of β, Σ_F, and Σ_ϵ
- However, it is too much to ask that all three of these matrices be identified from Σ_R alone.

Statistical Factor Models (optional explanation)

Explanation (optional)

- Here is the problem: Let ***A*** be any *p×p* invertible matrix. Then the returns vector R_t in $R_t = \beta_0 + \beta^T F_t + \epsilon_t$ is unchanged if $\beta^T$is replaced by $\beta^T A^{-1}$ and F_t is replaced by AF_t.
- Therefore, the returns only determine β and F_t up to a nonsingular linear transformation, and consequently a set of constraints is needed to identify the parameters.
- The usual constraints are the factors are uncorrelated and standardized, so that $\Sigma_F = I$ where ***I*** is the *p×p* identity matrix.
- With these constraints, $\Sigma_R = \beta^T \Sigma_F \beta + \Sigma_\epsilon$ simplifies to the statistical factor model $\Sigma_R = \beta^T \beta + \Sigma_\epsilon$
- However, even with this simplification, β, is only determined up to a rotation, that is, by multiplication by an orthogonal matrix.
- To appreciate why this is so, let ***P*** be any orthogonal matrix, so that P^T = P^{-1}.
- Then $\Sigma_R = \beta^T \beta + \Sigma_\epsilon$ is unchanged if β is replaced by $\mathrm{P}\beta$ since $(P\beta)^T(P\beta) = P^T \beta^T P\beta = \beta^T P^{-1} P\beta = \beta^T \beta$
- Therefore, to determine β a further set of constraints is needed. One possible set of constraints is that $\beta \Sigma_\epsilon^{-1} \beta^T$ is diagonal (Mardia et al., 1979, p. 258).
- Output from R's function factanal() satisfies this constraint when the argument rotation is set to "none".
- If the main purpose of the statistical factor model is to estimate ΣR, then the choice of constraint is irrelevant since all constraints lead to the same product $\beta^T \beta$
- In particular, rotation of β does not change the estimate of ΣR.

Example : Factor analysis of equity funds

- This example continues the analysis of the equity funds data set that was used previously to illustrate PCA. The code for fitting a 4-factor model (p = 4) using factanal() is:

```
equityFunds = read.csv("equityFunds.csv")
fa_none = factanal(equityFunds[ , 2:9], 4, rotation = "none")
print(fa_none,cutoff = 0.1)
Call:
factanal(x = equityFunds[, 2:9], factors = 4, rotation = "none")

Uniquenesses:
EASTEU  LATAM  CHINA  INDIA ENERGY MINING   GOLD  WATER
 0.735  0.368  0.683  0.015  0.005  0.129  0.005  0.778

Loadings:
       Factor1 Factor2 Factor3 Factor4
EASTEU  0.387   0.169   0.293
LATAM   0.511   0.167   0.579
CHINA   0.310   0.298   0.362
INDIA   0.281   0.951
ENERGY  0.784                   0.614
MINING  0.786           0.425  -0.258
GOLD    0.798                  -0.596
WATER   0.340           0.298   0.109

               Factor1 Factor2 Factor3 Factor4
SS loadings      2.571   1.069   0.823   0.819
Proportion Var   0.321   0.134   0.103   0.102
Cumulative Var   0.321   0.455   0.558   0.660

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 17.02 on 2 degrees of freedom.
The p-value is 0.000201
```

- **The “loadings” are the estimates $\widehat{\boldsymbol{\beta}}^{T}$.** By convention, any loading with an absolute value less than the parameter cutoff is not printed, and the default value of cutoff is 0.1
- **Uniquenesses** are the variance in each item that is not explained by the four factors (The uniquenesses are the diagonal elements of the estimate $\hat{\Sigma}_{\epsilon}$).
- Since there are eight funds and four factors, the loadings are in an 8 ×4 matrix fa_none$loadings.
- Because all its loadings have the same sign, the first factor is an overall index of the eight funds.
- The second factor has large loadings on the four regional funds (EASTEU, LATAM, CHINA, INDIA) and small loadings on the four industry section funds (ENERGY, MINING, GOLD, WATER). **The four regions are all emerging markets, so the second factor might be interpreted as an emerging markets factor.** The fourth factor is a contrast of MINING and GOLD with ENERGY and WATER, and mimics a hedge portfolio that is long on ENERGY and WATER and short on GOLD and MINING.
- The third factor is less interpretable.
- **The chi-square statistic and p-value in factanal are testing the hypothesis that the model fits the data perfectly. When the p value is low, as it is here, we can reject this hypothesis - so in this case, the 4-factor model does not fit the data perfectly**

Varimax Rotation of the Factors

- As discussed earlier, the estimate of the covariance matrix is unchanged if the loadings β are rotated by multiplication by an orthogonal matrix.
- **“Rotation” might increase the interpretability of the loadings.**
- In some applications, it is desirable for each loading to be either close to 0 or large, so that a variable will load only on a few factors, or even on only one factor.
- **Varimax rotation attempts to make each loading either small or large by maximizing the sum of the variances of the squared loadings**.
- Varimax rotation is the default with R’s factanal() function, but this can be changed as in the previous Example where no rotation was used.
- In finance, having variables loading on only one or a few factors is not that important, and may even be undesirable, so varimax rotation may not advantageous.
- We repeat again for emphasis that the estimate of ΣR is not changed by rotation. The uniquenesses are also unchanged. Only the loadings change.

Example: Factor analysis of equity funds: Varimax rotation

```
fa_vari = factanal(equityFunds[,2:9],4,rotation="varimax") #factor
model with rotation

Call:
factanal(x = equityFunds[, 2:9], factors = 4, rotation = "varimax")

Uniquenesses:
EASTEU  LATAM  CHINA  INDIA ENERGY MINING   GOLD  WATER 
 0.735  0.368  0.683  0.015  0.005  0.129  0.005  0.778 

Loadings:
       Factor1 Factor2 Factor3 Factor4
EASTEU 0.436   0.175   0.148   0.148  
LATAM  0.748   0.174           0.180  
CHINA  0.494           0.247          
INDIA  0.243           0.959          
ENERGY 0.327   0.118           0.934  
MINING 0.655   0.637           0.168  
GOLD   0.202   0.971                  
WATER  0.418                   0.188  

               Factor1 Factor2 Factor3 Factor4
SS loadings      1.804   1.445   1.028   1.003
Proportion Var   0.226   0.181   0.129   0.125
Cumulative Var   0.226   0.406   0.535   0.660

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 17 on 2 degrees of freedom.
The p-value is 0.000201
```

- The most notable change compared to the nonrotated loadings is that now all loadings with an absolute value above 0.1 are positive.
- Therefore, the factors all represent long positions, whereas before some were more like hedge portfolios.
- However, the rotated factors seem less interpretable compared to the unrotated factors, so a financial analyst might prefer the unrotated factors.

Going into other details + other ways to estimate

- we should evaluate the “factorability” of our data.
- In other words, “are there meaningful latent factors to be found within the data?” We can check two things:
  - (1) Bartlett’s test of sphericity; and
  - (2) the Kaiser-Meyer-Olkin measure of sampling adequacy.

Bartlett's Test of Sphericity

- The most liberal test is *Bartlett's test of sphericity* - **this evaluates whether or not the variables intercorrelate at all, by evaluating the observed correlation matrix against an "identity matrix"** (a matrix with ones along the principal diagonal, and zeroes everywhere else).
- **If this test is not statistically significant, you should not employ a factor analysis.**

```
#Assessing the Factorability of the Data
cortest.bartlett(equityFunds[,2:9])

> cortest.bartlett(equityFunds[,2:9])
R was not square, finding R from data
$chisq
[1] 3870.341

$p.value
[1] 0

$df
[1] 28
```

Bartlett's test was statistically significant, suggesting that the observed correlation matrix among the items is not an identity matrix.

This really isn't a particularly powerful indication that you have a factorable dataset, though - all it really tells you is that at least some of the variables are correlated with each other.

(Bartlett, M. S. (1951). The Effect of Standardization on a χ^2 Approximation in Factor Analysis. *Biometrika 38*(3/4), 337--344.)

KMO

- **The Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy** is a better measure of factorability.
- **The KMO tests if the partial correlations within your data are not close enough to zero to suggest that there is at least one latent factor underlying your variables.**
- **The minimum acceptable value is 0.50, but most authors recommend a value of at 0.60 before undertaking a factor analysis.**
- The KMO function in the psych package produces an overall Measure of Sampling Adequacy (MSA, as its labelled in the output), and an MSA for each item.
- Theoretically, if your overall MSA is too low, you could look at the item MSA's and drop items that are too low.
- This should be done with caution, of course, as is the case with any atheoretical, empirical method of item selection.

```
#KMO
KMO(equities)
> KMO(equities)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = equities)
Overall MSA =  0.76
MSA for each item = 
EASTEU  LATAM  CHINA  INDIA ENERGY MINING   GOLD  WATER 
  0.92   0.79   0.84   0.85   0.89   0.67   0.59   0.89 
```

The overall KMO for our data is 0.76 which is excellent - this suggests that we can go ahead with our planned factor analysis.

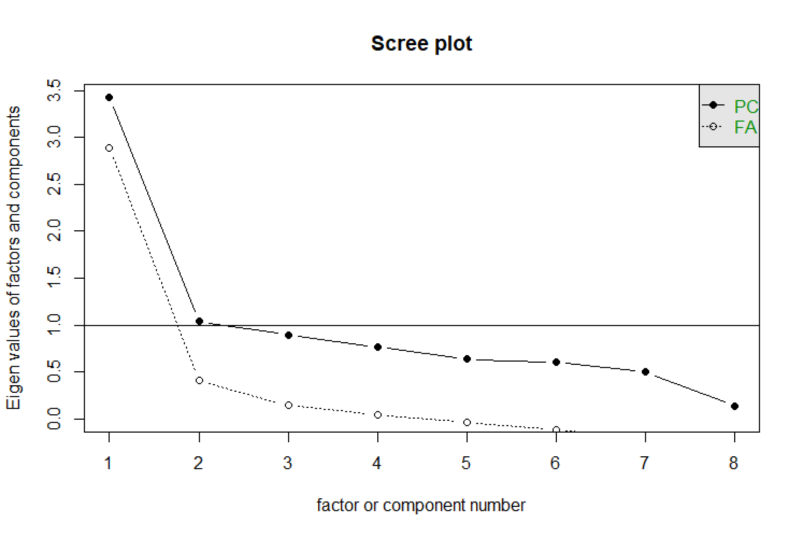
Determining the number of factors to extract

- The first decision that we will face in our factor analysis is the decision as to the number of factors that we will need to extract, in order to achieve the most parsimonious (but still interpretatable) factor structure.
- There are several methods that we could use, but the two most commonly employed methods are the scree plot, and parallel analysis. The simplest technique is the scree plot.

Scree plot

- **Eigenvalues are a measure of the amount of variance accounted for by a factor, and so they can be useful in determining the number of factors that we need to extract.**
- In a scree plot, we simply plot the eigenvalues for all of our factors, and then look to see where they drop off sharply.
- The scree plot technique involves drawing a straight line through the plotted eigenvalues, starting with the largest one.
- The last point to fall on this line represents the last factor that you extract, with the idea being that beyond this, the amount of additional variance explained is non-meaningful.
- In fact, the word "scree" refers to the loose stone that lies around the base of the mountain.
- A "scree plot" is effectively looking to help you differentiate between the points that represent "mountain", and the points that represent "scree."
- Regardless of whether you are using a principal components or a principal axis factor extraction, however, there is a very large first factor in this data.
- It is hard to be sure how many factors can be used. I would rather say 3 maximum.

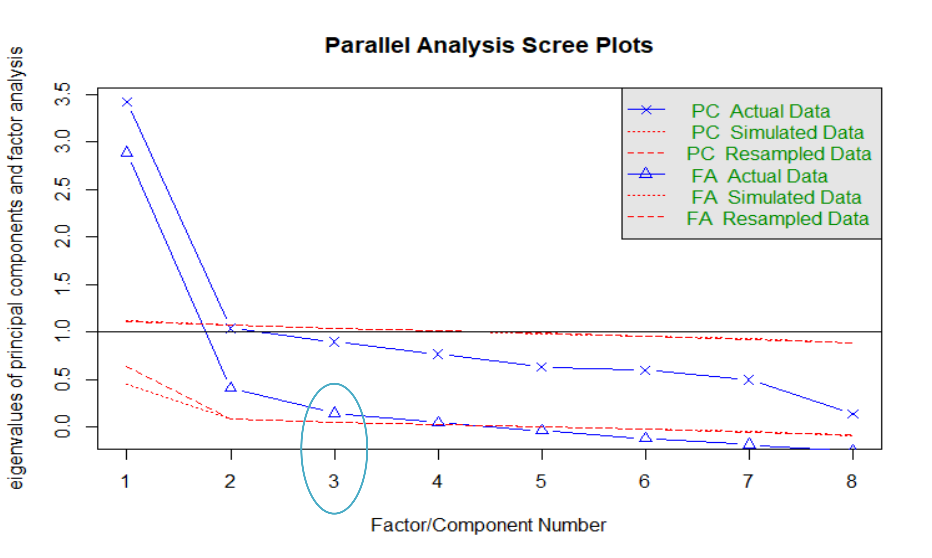
```
scree(equities)
```



Parallel Analysis

- A better method for evaluating the scree plot is within a parallel analysis.
- In addition to plotting the eigenvalues from our factor analysis (whether it's based on principal axis or principal components extraction), a parallel analysis involves generating random correlation matrices and after factor analyzing them, comparing the resulting eigenvalues to the eigenvalues of the observed data.
- *The idea behind this method is that observed eigenvalues that are higher than their corresponding random eigenvalues are more likely to be from "meaningful factors" than observed eigenvalues that are below their corresponding random eigenvalue.*
- When looking at the parallel analysis scree plots, there are two places to look depending on which type of factor analysis you're looking to run.
- **The two blue lines show you the observed eigenvalues - they should look identical to the scree plots drawn by the scree function.**
- The red dotted lines show you the random eigenvalues or the simulated data line.
- **Each point on the blue line that lies above the corresponding simulated data line is a factor or component to extract.** In this analysis, you can see that 3 factors in the "Factor Analysis" parallel analysis lie above the corresponding simulated data line and 1 components in the "Principal Components" parallel analysis lie above the corresponding simulated data line.
- These two indicators are a little bit different. *Components* means that there are unidimensional construct but it has some components. *Factors* means that there are non-correlated factors.
- In our case, however, we should probably compare the 3 factor and 2 factor solutions, to see which one is most interpretable.

```
> fa.parallel (equities)

Parallel analysis suggests that
the number of factors =  4  and
the number of components =  1
```



Conducting the Factor Analysis

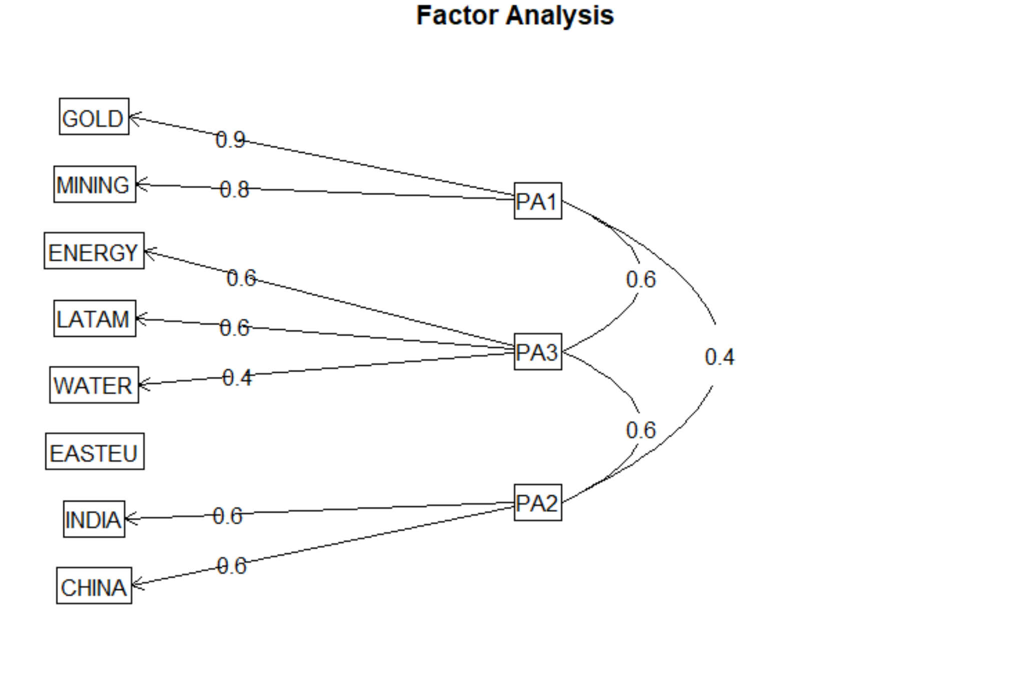
- We already have a good idea as to how many factors we should extract in our analysis.
- Now we need to decide whether we will use “common factor” analysis, or “principal components” analysis.
- **In a very broad sense, “common factor” analysis (or “principal axis factoring”) is used when we want to identify the latent variables that are underlying a set of variables, while “principal components” analysis is used to reduce a set of variables to a smaller set of factors (i.e., the “principal components” of the data).**
- In other words, common factor analysis is used when you want to evaluate a theoretical model with a set of variables, and principal components analysis is used for data reduction (as you saw before).

fa function

- The `fa` function takes the following parameters when it is called:
  - the variables to be used within the factor analysis
  - the number of factors we want to extract (3)
  - the type of factor analysis we want to use “ols” – OLS analysis)
  - the number of iterations or attempts to use when identifying the “best”" solution (50 is the default, but we have changed it to 100)
  - the type of rotation we want to use - **oblimin** (**Oblimin is a type of oblique rotation which means it allows the factors to correlate**. **Varimax** is an orthogonal rotation and **does not allow correlation** - factors are 90 degrees)

Results

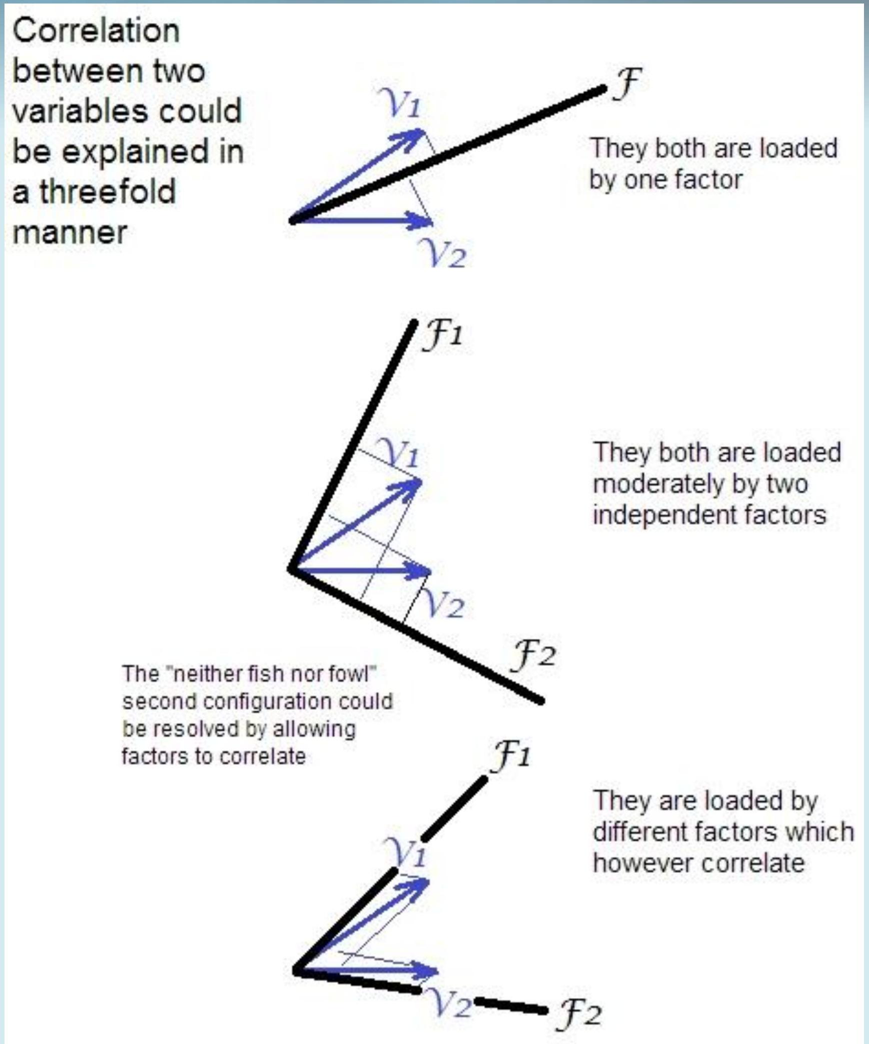
```
# estimation factor model
factor.model <- fa(equities, nfactors
= 3, fm="ols", max.iter = 100, rotate
= "oblimin")

fa.diagram(factor.model)
```



Back to rotations again

- **Reason for rotation.** *Rotations are done for the sake of interpretation of the extracted factors in factor analysis* (or components in PCA, if want to use PCA as a factor analytic technique).
- It is when different factors tend to load different variables. You then interpret a factor as the meaning which lies on the intersection of the meaning of the variables which are loaded enough by the factor; thus, to receive different meaning, factors should load variables differentially.
- Sometimes a rule of thumb is applied - a factor should load decently at least 3 variables.
- **Rotation does not change the position of variables relative to each other in the space of the factors, i.e. correlations between variables are being preserved**.
- What are changed are the coordinates of the variable vectors' end-points onto the factor axes - the loadings. After an orthogonal rotation of the loading matrix, factor variances get changed, but factors remain uncorrelated and variable communalities are preserved.
- **Choice**. There are many forms of orthogonal and oblique rotations. Why?
- For example, **varimax** - the most popular orthogonal method - tries to maximize variance among the squared values of loadings of each factor; the sometimes used orthogonal method **quartimax** minimizes the number of factors needed to explain a variable, and often produces the so called "general factor".
- **Orthogonal factors are easier to interpret and the whole factor model is statistically simpler. But, how can you be sure that the factors are not correlated?**
- **Oblique rotation methods (albeit each having their own inclinations) allow, but don't force, factors to correlate, and are thus less restrictive.**
- Two most important oblique methods **are promax and oblimin. Promax is the oblique enhancement of varimax:** the varimax-based structure is then loosed in order to meet "simple structure" to a greater degree. It is often used in confirmatory FA. **Oblimin** is very flexible due to its parameter gamma which, when set to 0, makes oblimin the quartimin method yielding most oblique solutions. A gamma of 1 yields the least oblique solutions, the covarimin, which is yet another varimax-based oblique method alternative to promax.
- **If oblique rotation shows that factors are only weakly correlated, you may be confident that "in reality" it is so, and then you may turn to orthogonal rotation.**

Correlation between two variables



Correlation between two variables could be explained in a threefold manner

They both are loaded by one factor

They both are loaded moderately by two independent factors

The "neither fish nor fowl" second configuration could be resolved by allowing factors to correlate

They are loaded by different factors which however correlate

Communalities

- The communality for each variable is the percentage of variance that can be explained by the retained factors.
- It is a common variance that ranges between 0 and 1.
- **Values closer to 1 suggest that extracted factors explain more of the variance of an individual item.**
  - **It’s best if the retained factors explain more of the variance in each variable.**

```
> factor.model$communality
   EASTEU     LATAM     CHINA     INDIA    ENERGY    MINING      GOLD     WATER
0.2722540 0.5764581 0.3783584 0.4123569 0.3718171 0.9967473 0.6709421 0.2586096
```

Percentage of variance accounted for

- Eigenvalues: The eigenvalues derived in the extracted factor solution are stored within e.values. These are the eigenvalues that were plotted in the scree plots that we looked at near the beginning of this process.
- The eigenvalue is a measure of how much of the variance of the observed variables a factor explains.
- If eigenvalues are greater than zero, then it’s a good sign.
- **Eigenvalues close to zero imply there is item multicollinearity, since all the variance can be taken up by the first component.**

```
> factor.model$e.values
[1] 3.4277598 1.0381969 0.8972720 0.7667112 0.6337270 0.6011694 0.4969426 0.1382211
```

- Any factor with an eigenvalue ≥1 explains more variance than a single observed variable.

- We can use the eigenvalues to calculate the percentage of variance accounted for by each of the factors.
- Given that the maximum sum of the eigenvalues will always be equal to the total number of variables in the analysis, we can calculate the percentage of variance accounted for by dividing each eigenvalue by the total number of variables in the analysis. In our example this is 8

```
100*factor.model$e.values/length(factor.model$e.values)
[1] 42.846998 12.977461 11.215900  9.583890  7.921587  7.514618  6.211783  1.727764
```

The structure matrix

- We can also look at **the structure matrix** - this is just the pattern matrix multiplied by the factor intercorrelation matrix.
- The result is that these **values represent the correlations between the variables and the factors** - which may be more intuitive to interpret.

```
print(factor.model$Structure, cutoff=0, digits=3)
Loadings:
       [,1]  [,2]  [,3] 
EASTEU 0.384 0.476 0.417
LATAM  0.524 0.741 0.521
CHINA  0.322 0.474 0.588
INDIA  0.235 0.302 0.639
ENERGY 0.380 0.607 0.295
MINING 0.978 0.733 0.508
GOLD   0.811 0.380 0.271
WATER  0.293 0.506 0.317
```