

Report of Slump Data Analysis

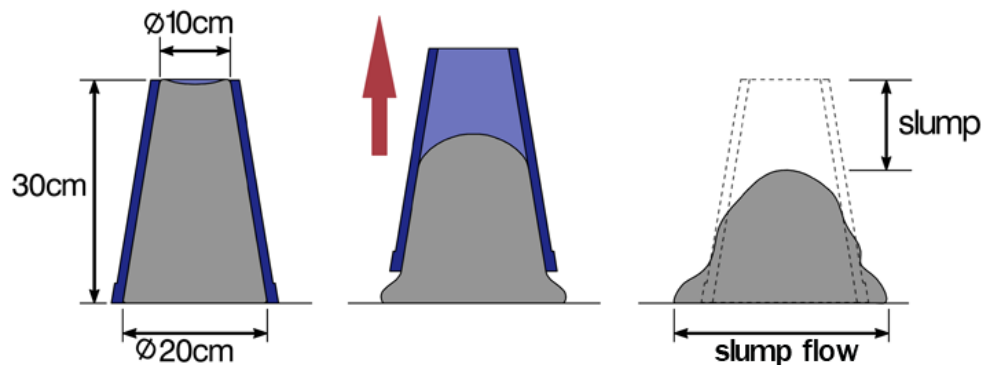
(BI 1) YAO Zeliang

(BI 2) ZHANG Meng

1. Analysis of the data set

Our group choose the “Slump” data set. This dataset contains some results about two kinds of tests executed on concrete.

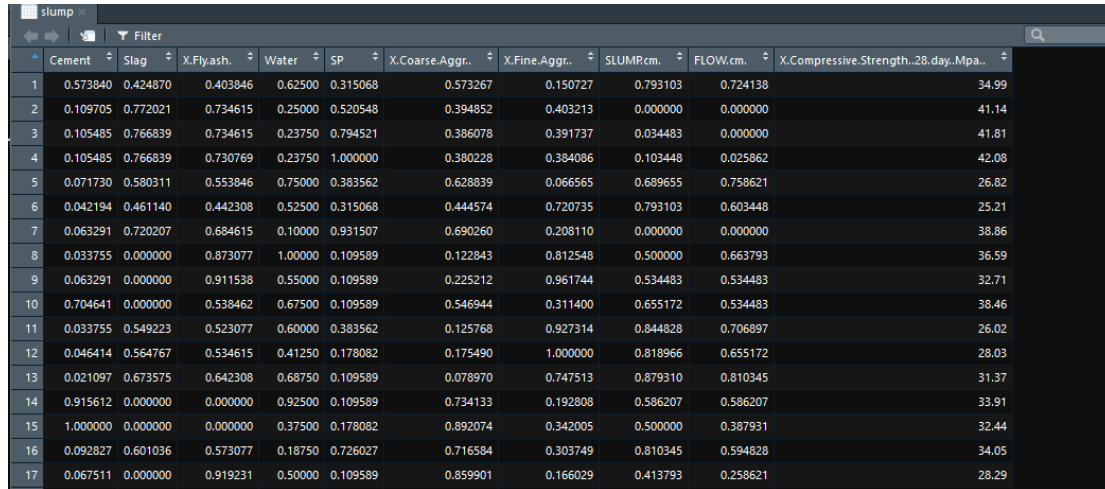
Concrete is a highly complex material. The slump flow of concrete is not only determined by the water content, but that is also influenced by other concrete ingredients. The Slump is the difference of height of the concrete mix after being placed in the Slump cone and the cone. It differs from one sample to another. Samples with lower heights are predominantly used in construction, with samples having high Slumps commonly used to construct roadway pavements. The Flowability is measured in terms of spread, hence the Flow correspond to the width of the patty.



For a mix, the Slump should be consistent. A change in Slump height would demonstrate an undesired change in the ratio of the concrete ingredients, the proportions of the ingredients are then adjusted to keep a concrete batch consistent. This homogeneity improves the quality and structural integrity of the concrete.

1.1 Components of the dataset

Slump dataset (show in R studio):



	Cement	Slag	X.Fly.ash.	Water	SP	X.Coarse.Aggr.	X.Fine.Aggr.	SLUMP.cm.	FLOW.cm.	X.Compressive.Strength..28.day..Mpa..
1	0.573840	0.424870	0.403846	0.62500	0.315068	0.573267	0.150727	0.793103	0.724138	34.99
2	0.109705	0.772021	0.734615	0.25000	0.520548	0.394852	0.403213	0.000000	0.000000	41.14
3	0.105485	0.766839	0.734615	0.23750	0.794521	0.386078	0.391737	0.034483	0.000000	41.81
4	0.105485	0.766839	0.730769	0.23750	1.000000	0.380228	0.384086	0.103448	0.025862	42.08
5	0.071730	0.580311	0.553846	0.75000	0.383562	0.628839	0.066565	0.689655	0.758621	26.82
6	0.042194	0.461140	0.442308	0.52500	0.315068	0.444574	0.720735	0.793103	0.603448	25.21
7	0.063291	0.720207	0.684615	0.10000	0.931507	0.690260	0.208110	0.000000	0.000000	38.86
8	0.033755	0.000000	0.873077	1.00000	0.109589	0.122843	0.812548	0.500000	0.663793	36.59
9	0.063291	0.000000	0.911538	0.55000	0.109589	0.225212	0.961744	0.534483	0.534483	32.71
10	0.704641	0.000000	0.538462	0.67500	0.109589	0.546944	0.311400	0.655172	0.534483	38.46
11	0.033755	0.549223	0.523077	0.60000	0.383562	0.125768	0.927314	0.844828	0.706897	26.02
12	0.046414	0.564767	0.534615	0.41250	0.178082	0.175490	1.000000	0.818966	0.655172	28.03
13	0.021097	0.673575	0.642308	0.68750	0.109589	0.078970	0.747513	0.879310	0.810345	31.37
14	0.915612	0.000000	0.000000	0.92500	0.109589	0.734133	0.192808	0.586207	0.586207	33.91
15	1.000000	0.000000	0.000000	0.37500	0.178082	0.892074	0.342005	0.500000	0.387931	32.44
16	0.092827	0.601036	0.573077	0.18750	0.726027	0.716584	0.303749	0.810345	0.594828	34.05
17	0.067511	0.000000	0.919231	0.50000	0.109589	0.859901	0.166029	0.413793	0.258621	28.29

We have a list of 10 variables of Slump data set:

	Name	Description
Input	Cement	A binder, a substance used in construction that sets and hardens and can bind other materials together.
	Slag	Used to make durable concrete structures in combination with ordinary Portland cement and/or other Pozzolanic materials.
	Fly ash	Owing to its Pozzolanic properties, fly ash is used as a replacement for some of the Portland cement content of concrete.
	Water	The content of water.
	SP	Superplasticizers: they are chemical admixtures that can be added to concrete mixtures to improve workability.
	Coarse Aggr.	Crushed Stone, ...
	Fine Aggr.	Sand, ...
Output	SLUMP (cm)	
	FLOW (cm)	
	28-day Compressive Strength (Mpa)	

The data set includes 103 observations. The initial data set included 78 data. After several years, we got 25 new data points.

Attribute Information:

Input variables (7) (component kg in one M³ concrete): Cement, Slag, Fly ash, Water, SP, Coarse Aggr, Fine Aggr.

Output variables (3): SLUMP (cm), FLOW (cm), 28-day Compressive Strength (Mpa)

We cannot determine if there is outlier by simply observing the data set, however, we can tell that there is no missing value in any variable, and the types of the data are all **numeric**.

For the further analysis , we use **R studio** as our analysis tool.

1.2 Description of the variables and observations

1) Univariate analysis (position criteria, dispersion criteria)

```
> summary(slump)
      Cement      Slag      X.Fly.ash.      Water      SP      X.Coarse.Aggr..      X.Fine.Aggr..
Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.06329   1st Qu.:0.000259  1st Qu.:0.4442  1st Qu.:0.2500  1st Qu.:0.1096  1st Qu.:0.3261  1st Qu.:0.1679
Median :0.46835   Median :0.518135  Median :0.6308  Median :0.4500  Median :0.2466  Median :0.5001  Median :0.3906
Mean   :0.39196   Mean   :0.404009  Mean   :0.5731  Mean   :0.4646  Mean   :0.2835  Mean   :0.5147  Mean   :0.3787
3rd Qu.:0.70422   3rd Qu.:0.647668  3rd Qu.:0.9075  3rd Qu.:0.6188  3rd Qu.:0.3836  3rd Qu.:0.7160  3rd Qu.:0.5639
Max.   :1.00000   Max.   :1.000000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
SLUMP.cm.      FLOW.cm.      X.Compressive.Strength..28.day..Mpa..
Min.   :0.0000   Min.   :0.0000   Min.   :17.19
1st Qu.:0.5000   1st Qu.:0.3190   1st Qu.:30.90
Median :0.7414   Median :0.5862   Median :35.52
Mean   :0.6224   Mean   :0.5105   Mean   :36.04
3rd Qu.:0.8276   3rd Qu.:0.7543   3rd Qu.:41.20
Max.   :1.0000   Max.   :1.0000   Max.   :58.53
```

And then we get the following variance list (dispersion criteria):

```
> apply(slump,2,var)
      Cement      Slag      X.Fly.ash.      Water      SP      X.Coarse.Aggr..      X.Fine.Aggr..
0.11076605   0.09813890   0.10793266   0.06683774   0.06380776   0.03697798   0.06683774
0.05871837   0.09105501   0.09175271   61.43787809
```

Since the first 7 variables are independent variables with a wide range, their variances are relatively big. So, **it doesn't make much sense if we observe the variance list alone.**

2) Bivariate analysis (Scatter plot, covariance matrix)

We often use correlation matrix to analyze the correlation between each two variables.

Correlation matrix:

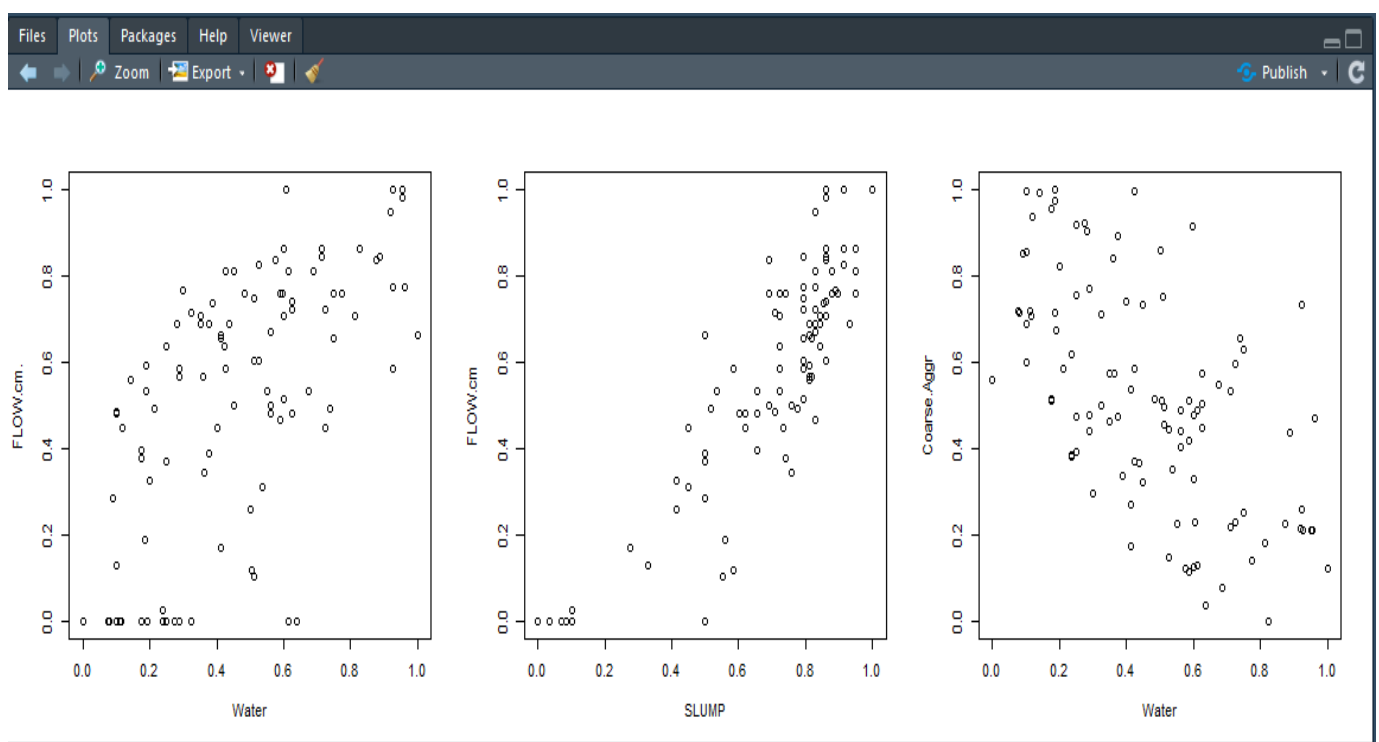
```
> res<-cor(s slump)
> round(res,2)
```

	Cement	Slag	X.Fly.ash.	Water	SP	X.Coarse.Aggr..	X.Fine.Aggr..	SLUMP.cm.	FLOW.cm.	X.Compressive.Strength..28.day..Mpa..
Cement	1.00	-0.24	-0.49	0.22	-0.11	-0.31	0.06	0.15	0.19	0.45
Slag	-0.24	1.00	-0.32	-0.03	0.31	-0.22	-0.18	-0.28	-0.33	-0.33
X.Fly.ash.	-0.49	-0.32	1.00	-0.24	-0.14	0.17	-0.28	-0.12	-0.06	0.44
Water	0.22	-0.03	-0.24	1.00	-0.16	-0.60	0.11	0.47	0.63	-0.25
SP	-0.11	0.31	-0.14	-0.16	1.00	-0.10	0.06	-0.21	-0.18	-0.04
X.Coarse.Aggr..	-0.31	-0.22	0.17	-0.60	-0.10	1.00	-0.49	-0.19	-0.33	-0.16
X.Fine.Aggr..	0.06	-0.18	-0.28	0.11	0.06	-0.49	1.00	0.20	0.19	-0.15
SLUMP.cm.	0.15	-0.28	-0.12	0.47	-0.21	-0.19	0.20	1.00	0.91	-0.22
FLOW.cm.	0.19	-0.33	-0.06	0.63	-0.18	-0.33	0.19	0.91	1.00	-0.12
X.Compressive.Strength..28.day..Mpa..	0.45	-0.33	0.44	-0.25	-0.04	-0.16	-0.15	-0.22	-0.12	1.00

In the correlation matrix, we find that most of the variance pairs have low values indicating that the correlations of each pair are not very high. The top 2 **positively relative pairs** are **Water/FLOW.cm.: 0.63**, **SLUMP.cm./FLOW.cm.: 0.91** and the **most negatively relative** pair is **Water/Coarse. Aggr.: -0.60**.

We draw the **scatter plot** of these variable pairs to get better analysis:

```
> layout(matrix(c(1,2,3),1,3,byrow = TRUE))
> plot(s slump$water,slump$FLOW.cm.,xlab = "water",ylab = "FLOW.cm.")
> plot(s slump$SLUMP.cm.,slump$FLOW.cm.,xlab = "SLUMP",ylab = "FLOW.cm")
> plot(s slump$water,slump$X.Coarse.Aggr.,xlab = "water",ylab = "Coarse.Aggr")
> |
```



From the plots, we can see that the observations distribute based on the correlation regulation, but we cannot recognize which variables are the key factors for classification only from these plots. And we can't get potential knowledge or make prediction only by existing attributes. If we want to get more information, we need to do multi-variate analysis. Thus, we need to implement some data mining methods for components analysis and knowledge discovery.

2. Visualization

2.1 Principal Components Analysis (PCA)

a) Introduction

PCA is a classic unsupervised method in data mining for linear dimension reduction and principal components analysis. In our project, while we don't have labels for observations, we can use PCA to find out the most important axes which contain most of the information in the data set in order to represent the most significant features.

b) Implementation

When we process PCA on source data set, we need to determine the variables and additional variables first. Normally we choose qualitative statistics and quantitative summary statistics as additional variables. In our data set, all the first 7 variables are quantitative and continuous. They are in the same status, each representing one of the content of the concrete.

So, we choose 7 of them as PCA variables.

And we choose the rest 3 variables ***SLUMP.cm., FLOW.cm. and Compressive.Strength..28.day..Mpa.*** as quantitative supplementary variables. After we determined the variables, we need to normalize the data and get correlation matrix of variables. The principal components are composed by the eigenvalue got from correlation matrix after diagonalization and the source data columns. The principle of choosing principal components are choosing the eigenvalue which beyond 1 or the percentage of variance which beyond the average value.

We use package ***FactoMineR*** and ***PCA*** function in R to implement PCA.

```
> install.packages("FactoMineR",dependencies = TRUE)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/FactoMineR_1.39.zip'
Content type 'application/zip' length 3029040 bytes (2.9 MB)
downloaded 2.9 MB
```

package 'FactoMineR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\E560\AppData\Local\Temp\Rtmp8cs6c9\downloaded_packages

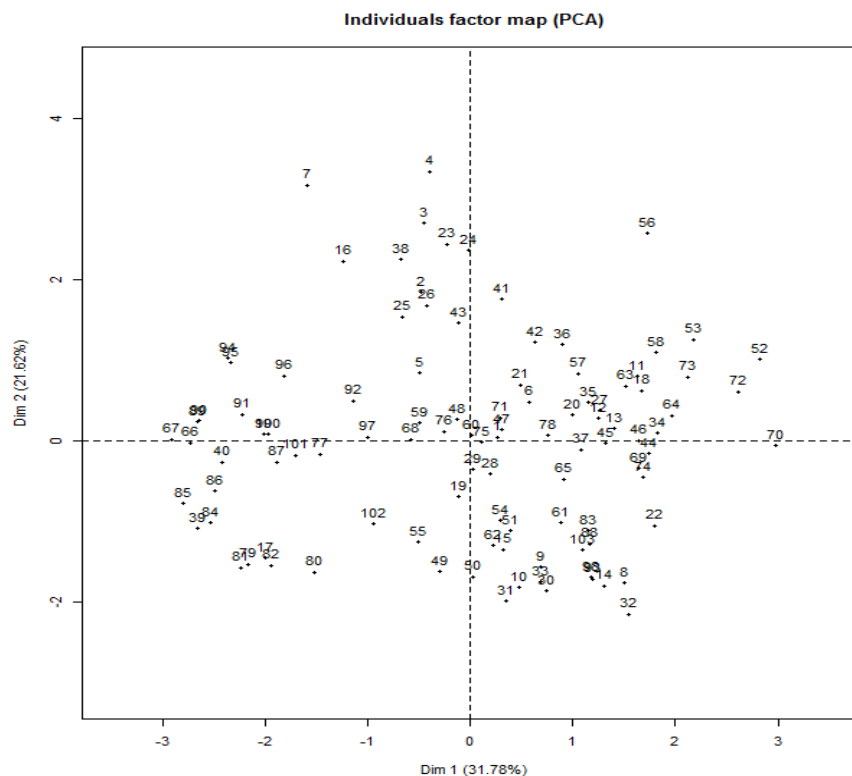
```
> |
```

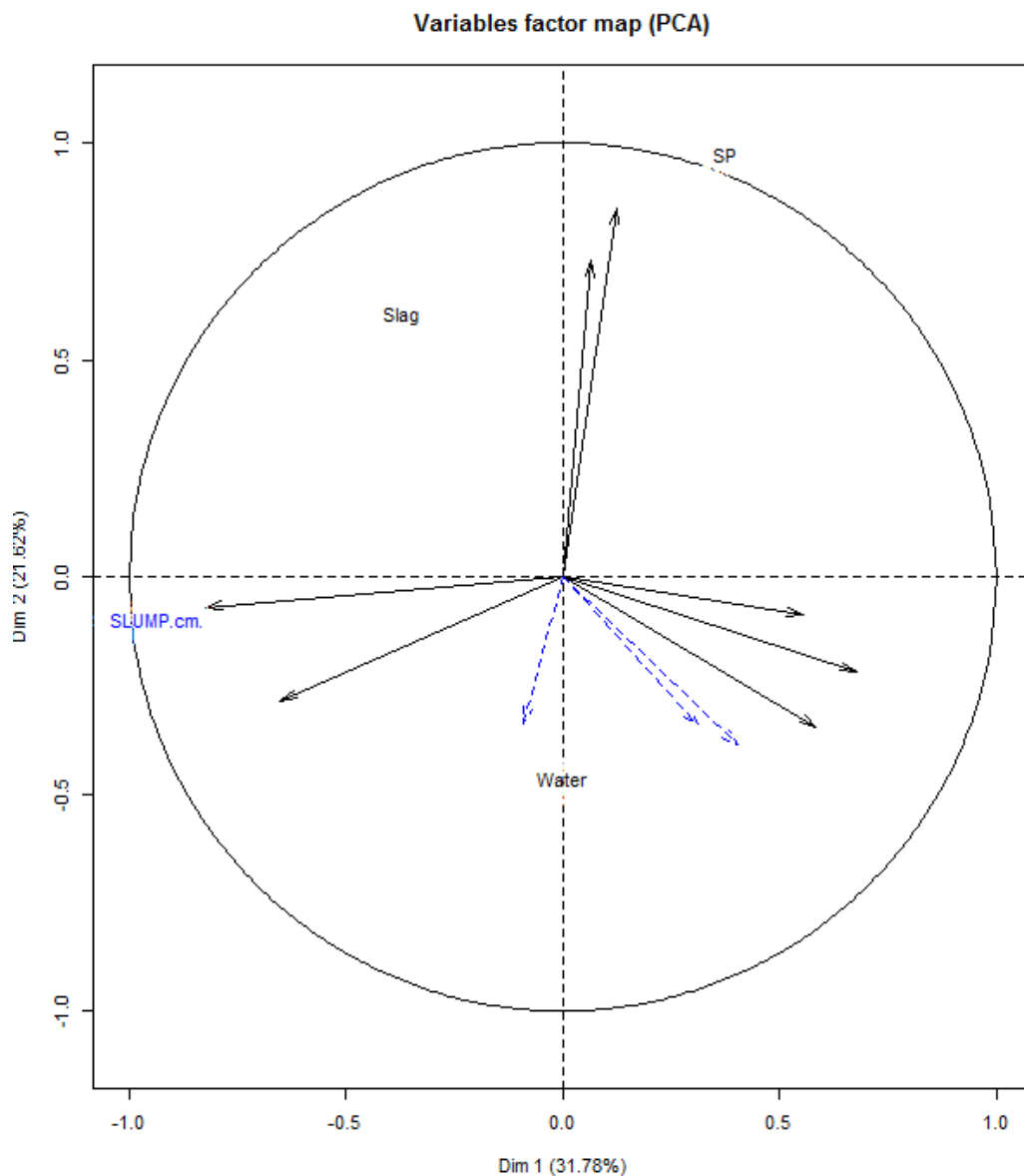
```
> library(FactoMineR)
> result<-PCA(s slump[,1:10],quanti.sup = c(8:10))
> result$eig
```

	eigenvalue	percentage of variance
comp 1	2.22456864	31.77955194
comp 2	1.51342568	21.62036692
comp 3	1.10865945	15.83799220
comp 4	1.00611536	14.37307651
comp 5	0.68127912	9.73255892
comp 6	0.46289896	6.61284222
comp 7	0.00305279	0.04361129

	cumulative percentage of variance
comp 1	31.77955
comp 2	53.39992
comp 3	69.23791
comp 4	83.61099
comp 5	93.34355
comp 6	99.95639
comp 7	100.00000

```
> |
```

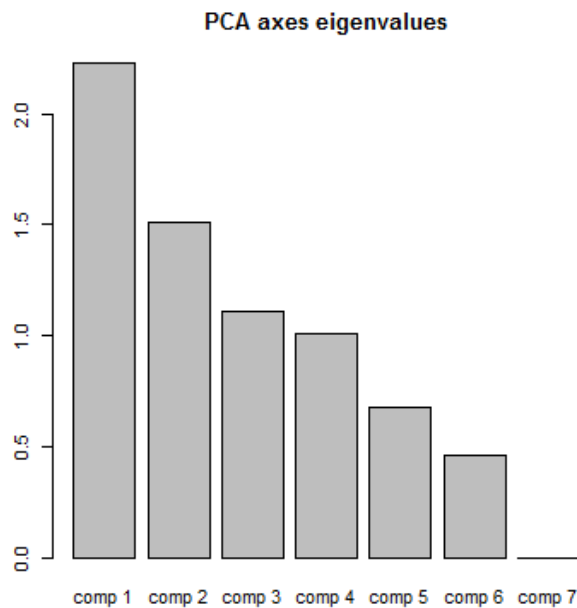




c) Result

From the variables factor map, SP seems to be highly correlated to Slag, but we knew from the correlation matrix that their correlation value is only about 0.3. As a result, we consider that the implementation of PCA by 2 axes in this case is not accurate. What also worthy to be mentioned, the cumulative percentage of variance of the first two components is only 0.53, which also tells the fact ***that PCA is not suitable.***

```
> barplot(result$eig[,1], main="PCA axes eigenvalues")
> |
```



From eigenvalue list, we can find that the first 3 axes contain 83.61% information of all the source data set.

3. Prediction:

3.1 Regression Line

a) Introduction

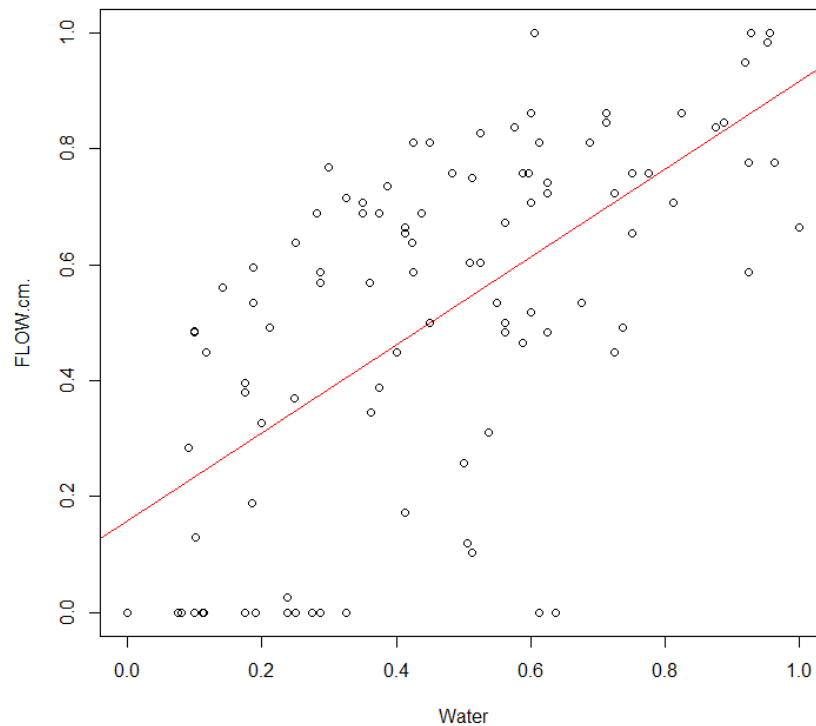
In our source data set, we have totally 103 observations and 10 variables, literally 7 independent variables and 3 dependent variables. Thus, we should have 21 independent variable / dependent variable pairs. Tracing back to the previous correlation matrix of our data set, there is only one pair of independent/ dependent variables, **Water/FLOW.cm.**, whose correlation value is comparatively larger (0.6320). In this particular case, linear regression is the method we choose for interpreting the prediction.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Most commonly, the conditional mean of y given the value of X is assumed to be an affine function of X ; less commonly, the median or some other quantile of the conditional distribution of y given X is expressed as a linear function of X .

b) Implementation

c)

```
> attach(slump)
> plot(Water, FLOW.cm., abline(lm(FLOW.cm.~Water), col="red"))
>
```



d) Result

Previously in the correlation matrix, we had confirmed that Slump is highly related to Flow. And as now we can see in the graph with linear regression line: The red line interpolates the data points and confirms a linear dependency between the two coordinates; the water content is linearly dependent to the Flow variation. Therefore, as soon as the water content is increased, consequently the Slump and Flow values are increased proportionally.

3.2 K-means

a) Introduction

K-means is the most common and classic cluster method in unsupervised learning.

The principle of K-means is to select initial group center randomly, group the nearest points, and reselect the group center, regroup and iteration until stable. The key factors for the performance is the selection of initial data centers and the times of iteration. The advantage of K-means is its low cost, simple algorithm, and a good scalability to fit large data set. But it is also sensitive to outliers and initial centers. Most of the time it can only give an approximate result.

b) Implementation

We set $k=3$. And for the initial centers, we set 100 groups randomly and choose the best result automatically. We set the iteration times 1000 for the best result.

We used package **Cluster** and **kmeans** function in R studio to do these steps.

```
> k<-kmeans(s1ump[,1:7],3,1000,100)
> k$cluster
[1] 1 3 3 3 3 3 3 3 3 1 3 3 3 1 1 3 2 1 1 1 1 1 3 3 3 3 1 1 1 1 1 1 1 3 1 1 3 2 2 3 3 3 1 1 1 1 1 1 1 1 1 1 1
[55] 1 3 1 1 3 3 1 1 1 1 1 2 2 3 1 1 1 1 1 1 3 3 2 1 2 2 2 2 3 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 1
>
```

Validation using Davies Bouldin index:

```
> install.packages("clusterSim",dependencies = TRUE)
also installing the dependencies 'httpuv', 'xtable', 'sourcetools', 'assertthat', 'htmlwidgets', 'htmltools', 'jsonlite', 'shiny', 'cli', 'crayon', 'praise', 'R6', 'rlang', 'withr', 'ade4', 'e1071', 'rgl', 'R2HTML', 'modeest', 'mlbench', 'testthat'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/httpuv_1.3.5.zip'
Content type 'application/zip' length 933416 bytes (911 KB)
downloaded 911 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/xtable_1.8-2.zip'
Content type 'application/zip' length 710238 bytes (693 KB)
downloaded 693 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/sourcetools_0.1.6.zip'
Content type 'application/zip' length 528077 bytes (515 KB)
downloaded 515 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/assertthat_0.2.0.zip'
Content type 'application/zip' length 102400 bytes (100 KB)
downloaded 100 KB

package 'modeest' successfully unpacked and MD5 sums checked
package 'mlbench' successfully unpacked and MD5 sums checked
package 'testthat' successfully unpacked and MD5 sums checked
package 'clusterSim' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\E560\AppData\Local\Temp\Rtmp8cs6C9\downloaded_packages
> library("clusterSim")
载入需要的程辑包: cluster
载入需要的程辑包: MASS

This is package 'modeest' written by P. PONCET.
For a complete list of functions, use 'library(help = "modeest")' or 'help.start()'.

> d<-dist(s1ump)
>
```

When $K = 2$, $DBI \approx 2.9$

```

> K<-kmeans(slung[,1:7],2,1000,100)
> View(K)
> print(index.DB(slung,K$cluster,d,centrotypes = "medoids"))
$DB
[1] 2.93009

$r
[1] 2.93009 2.93009

$R
      [,1]      [,2]
[1,]      Inf 2.93009
[2,] 2.93009      Inf

$d
      1      2
1 0.000000 5.074112
2 5.074112 0.000000

$s
[1] 7.413183 7.454421

$centers
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.097046 0.756477 0.723077 0.5375 0.452055 0.353905
[2,] 0.704641 0.000000 0.538462 0.6750 0.109589 0.546944
      [,7]      [,8]      [,9]      [,10]
[1,] 0.265493 0.448276 0.310345 33.51
[2,] 0.311400 0.655172 0.534483 38.46

```

When K = 3, DBI \approx 5.8

```

> K<-kmeans(slung[,1:7],3,1000,100)
> print(index.DB(slung,K$cluster,d,centrotypes = "medoids"))
$DB
[1] 5.819366

$r
[1] 4.640502 6.408799 6.408799

$R
      [,1]      [,2]      [,3]
[1,]      Inf 2.839467 4.640502
[2,] 2.839467      Inf 6.408799
[3,] 4.640502 6.408799      Inf

$d
      1      2      3
1 0.000000 5.208331 3.151917
2 5.208331 0.000000 2.289686
3 3.151917 2.289686 0.000000

$s
[1] 7.370613 7.418268 7.255865

$centers
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]      [,10]
[1,] 0.704641 0.000000 0.538462 0.67500 0.109589 0.546944 0.311400 0.655172 0.534483 38.46
[2,] 0.012658 0.663212 0.630769 0.28750 0.520548 0.476748 0.514155 0.818966 0.568966 33.38
[3,] 0.077637 0.632642 0.923077 0.24875 0.157534 0.756946 0.045524 0.500000 0.370690 35.52
>

```

When K = 4, DBI \approx 5.3

```

> K<-kmeans(slung[,1:7],4,1000,100)
> print(index.DB(slung,K$cluster,d,centrotypes = "medoids"))
$DB
[1] 5.301638

$R
[1] 1.57718 7.63763 7.63763 4.35411

$R
      [,1] [,2] [,3] [,4]
[1,]    Inf 1.24173 1.57718 1.101214
[2,] 1.241730    Inf 7.637630 4.354110
[3,] 1.577180 7.63763    Inf 3.330609
[4,] 1.101214 4.35411 3.330609    Inf

$d
      1      2      3      4
1 0.000000 9.803707 8.677807 12.639679
2 9.803707 0.000000 1.640970 2.931861
3 8.677807 1.640970 0.000000 4.287071
4 12.639679 2.931861 4.287071 0.000000

$S
[1] 6.663450 5.510103 7.023016 7.255543

$centers
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 0.894515 0.000000 0.684615 0.8750 0.109589 0.225212 0.307575 0.689655 0.836207 43.95
[2,] 0.523207 0.404145 0.384615 0.5125 0.315068 0.456274 0.460597 0.793103 0.750000 34.18
[3,] 0.092827 0.000000 0.953846 0.1875 0.520548 0.973969 0.162204 0.724138 0.534483 35.39
[4,] 0.021097 0.673575 0.642308 0.6875 0.109589 0.078970 0.747513 0.879310 0.810345 31.37

```

When K = 5, DBI \approx 7.5

```

> K<-kmeans(slung[,1:7],5,1000,100)
> print(index.DB(slung,K$cluster,d,centrotypes = "medoids"))
$DB
[1] 7.498566

$R
[1] 10.099213 7.941507 1.610632 10.099213 7.742265

$R
      [,1] [,2] [,3] [,4] [,5]
[1,]    Inf 7.941507 1.338896 10.099213 5.434477
[2,] 7.941507    Inf 1.288854 5.914689 7.742265
[3,] 1.338896 1.288854    Inf 1.059623 1.610632
[4,] 10.099213 5.914689 1.059623    Inf 4.078611
[5,] 5.434477 7.742265 1.610632 4.078611    Inf

$d
      1      2      3      4      5
1 0.000000 1.728198 11.200598 1.293617 2.772247
2 1.728198 0.000000 9.803707 1.809669 1.640970
3 11.200598 9.803707 0.000000 11.301712 8.677807
4 1.293617 1.809669 11.301712 0.000000 2.953172
5 2.772247 1.640970 8.677807 2.953172 0.000000

$S
[1] 8.042695 5.681806 6.953736 5.021823 7.023016

$centers
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 0.119409 0.741969 0.916538 0.5050 0.184932 0.512431 0.045907 0.586207 0.120690 32.84
[2,] 0.523207 0.404145 0.384615 0.5125 0.315068 0.456274 0.460597 0.793103 0.750000 34.18
[3,] 0.894515 0.000000 0.684615 0.8750 0.109589 0.225212 0.307575 0.689655 0.836207 43.95
[4,] 0.063291 0.000000 0.911538 0.5500 0.109589 0.225212 0.961744 0.534483 0.534483 32.71
[5,] 0.092827 0.000000 0.953846 0.1875 0.520548 0.973969 0.162204 0.724138 0.534483 35.39

```

c) Result

The **Davies–Bouldin index (DBI)** is a metric for evaluating clustering algorithms. This is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. The drawback is that a good value reported by this method does not imply the best information retrieval.

Definition of DBI:

Let $R_{i,j}$ be a measure of how good the clustering scheme is. This measure, by definition has to account for $M_{i,j}$, the separation between the i^{th} and the j^{th} cluster, which ideally has to be as large as possible, and S_i , the within cluster scatter for cluster i , which has to be as low as possible. Hence the Davies–Bouldin index is defined as the ratio of S_i and $M_{i,j}$ such that these properties are conserved:

1. $R_{i,j} \geq 0$.
2. $R_{i,j} = R_{j,i}$.
3. When $S_j \geq S_k$ and $M_{i,j} = M_{i,k}$ then $R_{i,j} > R_{i,k}$.
4. When $S_j = S_k$ and $M_{i,j} \leq M_{i,k}$ then $R_{i,j} > R_{i,k}$.

With this formulation, the lower the value, the better the separation of the clusters and the 'tightness' inside the clusters.

A solution that satisfies these properties is:

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

This is used to define D_i :

$$D_i \equiv \max_{j \neq i} R_{i,j}$$

If N is the number of clusters:

$$DB \equiv \frac{1}{N} \sum_{i=1}^N D_i$$

DB is called the Davies–Bouldin index. This is dependent both on the data as well as the algorithm. D_i chooses the worst-case scenario, and this value is equal to $R_{i,j}$ for the most similar cluster to cluster i . There could be many variations to this

formulation, like choosing the average of the cluster similarity, weighted average and so on.

With the way how DBI is defined, a lower value will mean that the clustering is better. (The best clustering scheme essentially minimizes the Davies–Bouldin index.)

However, considering only the average DBI of all clusters apparently is not a good idea. Increasing the number of clusters k , certainly, will always reduce the amount of DBI in the resulting clustering, to the extreme case of zero DBI if each data point is considered its own cluster (the case that each data point overlaps with its own centroid).

We therefore can solely conclude that, for our data set using K-means clustering, $k=2$ is better than $k=3$, but we cannot say it's the best number needed for clustering the data. Nevertheless, K-means method is still a choice for prediction when we have large data set and low infrastructure cost (using K-means as the basic cluster method).

4 Theoretical details

4.1 Univariate analysis

Univariate analysis explores each variable in a data set, separately. It looks at the range of values, as well as the central tendency of the values. It describes the pattern of response to the variable. It describes each variable on its own.

Descriptive statistics describe and summarize data. Univariate descriptive statistics describe individual variables.

For ordinal variables the median can be calculated as a measure of central tendency and the range (and variations of it) as a measure of dispersion. For interval level variables, the arithmetic mean (average) and standard deviation are added to the toolbox and, for ratio level variables, we add the geometric mean and harmonic mean as measures of central tendency and the coefficient of variation as a measure of dispersion.

4.2 Bivariate analysis

Bivariate analysis means the analysis of two (“bi”) variables. It is one of the simplest forms of statistical analysis, used to find out if there is a relationship between two sets of values. It usually involves the variables X and Y .

Bivariate analysis is not the same as two sample data analysis. With two sample data

analysis, the X and Y are not directly related. You can also have a different number of data values in each sample; with bivariate analysis, there is a Y value for each X.

Univariate Data	Bivariate Data
involving a single variable	involving two variables
does not deal with causes or relationships	deals with causes or relationships
the major purpose of univariate analysis is to describe	the major purpose of bivariate analysis is to explain
<ul style="list-style-type: none"> • central tendency - mean, mode, median • dispersion - range, variance, max, min, quartiles, standard deviation. • frequency distributions • bar graph, histogram, pie chart, line graph, box-and-whisker plot 	<ul style="list-style-type: none"> • analysis of two variables simultaneously • correlations • comparisons, relationships, causes, explanations • tables where one variable is contingent on the values of the other variable. • independent and dependent variables

4.3 Principal Components Analysis (PCA)

Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric.

PCA procedure:

Suppose that we have a random vector \mathbf{X} .

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

with population variance-covariance matrix

$$\text{var}(\mathbf{X}) = \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_p^2 \end{pmatrix}$$

Consider the linear combinations

$$\begin{aligned} Y_1 &= e_{11}X_1 + e_{12}X_2 + \cdots + e_{1p}X_p \\ Y_2 &= e_{21}X_1 + e_{22}X_2 + \cdots + e_{2p}X_p \\ &\vdots \\ Y_p &= e_{p1}X_1 + e_{p2}X_2 + \cdots + e_{pp}X_p \end{aligned}$$

Each of these can be thought of as a linear regression, predicting Y_i from X_1, X_2, \dots, X_p . There is no intercept, but $e_{i1}, e_{i2}, \dots, e_{ip}$ can be viewed as regression coefficients.

Note that Y_i is a function of our random data, and so is also random. Therefore, it has a population variance

$$\text{var}(Y_i) = \sum_{k=1}^p \sum_{l=1}^p e_{ik}e_{il}\sigma_{kl} = \mathbf{e}_i'\Sigma\mathbf{e}_i$$

Moreover, Y_i and Y_j will have a population covariance

$$\text{cov}(Y_i, Y_j) = \sum_{k=1}^p \sum_{l=1}^p e_{ik}e_{jl}\sigma_{kl} = \mathbf{e}_i'\Sigma\mathbf{e}_j$$

Here the coefficients e_{ij} are collected into the vector

$$\mathbf{e}_i = \begin{pmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{ip} \end{pmatrix}$$

First Principal Component (PCA1): Y_1

The *first principal component* is the linear combination of x-variables that has maximum variance (among all linear combinations), so it accounts for as much variation in the data as possible.

Specifically we will define coefficients $\mathbf{e}_{11}, \mathbf{e}_{12}, \dots, \mathbf{e}_{1p}$ for that component in such a way that its variance is maximized, subject to the constraint that the sum of the squared coefficients is equal to one. This constraint is required so that a unique answer may be obtained.

More formally, select $\mathbf{e}_{11}, \mathbf{e}_{12}, \dots, \mathbf{e}_{1p}$ that maximizes

$$\text{var}(Y_1) = \sum_{k=1}^p \sum_{l=1}^p e_{1k} e_{1l} \sigma_{kl} = \mathbf{e}_1' \Sigma \mathbf{e}_1$$

subject to the constraint that

$$\mathbf{e}_1' \mathbf{e}_1 = \sum_{j=1}^p e_{1j}^2 = 1$$

Second Principal Component (PCA2): Y_2

The *second principal component* is the linear combination of x-variables that accounts for as much of the remaining variation as possible, with the constraint that the correlation between the first and second component is 0

Select $\mathbf{e}_{21}, \mathbf{e}_{22}, \dots, \mathbf{e}_{2p}$ that maximizes the variance of this new component...

$$\text{var}(Y_2) = \sum_{k=1}^p \sum_{l=1}^p e_{2k} e_{2l} \sigma_{kl} = \mathbf{e}_2' \Sigma \mathbf{e}_2$$

subject to the constraint that the sums of squared coefficients add up to one,

$$\mathbf{e}_2' \mathbf{e}_2 = \sum_{j=1}^p e_{2j}^2 = 1$$

along with the additional constraint that these two components will be uncorrelated with one another.

$$\text{cov}(Y_1, Y_2) = \sum_{k=1}^p \sum_{l=1}^p e_{1k} e_{2l} \sigma_{kl} = \mathbf{e}_1' \Sigma \mathbf{e}_2 = 0$$

All subsequent principal components have this same property – they are linear combinations that account for as much of the remaining variation as possible and they are not correlated with the other principal components

We will do this in the same way with each additional component. For instance:

ith Principal Component (PC*i*): Y_i

We select $\mathbf{e}_{i1}, \mathbf{e}_{i2}, \dots, \mathbf{e}_{ip}$ that maximizes

$$\text{var}(Y_i) = \sum_{k=1}^p \sum_{l=1}^p e_{ik} e_{il} \sigma_{kl} = \mathbf{e}_i' \Sigma \mathbf{e}_i$$

subject to the constraint that the sums of squared coefficients add up to one...along with the additional constraint that this new component will be uncorrelated with all the previously defined components.

$$\begin{aligned} \mathbf{e}_i' \mathbf{e}_i &= \sum_{j=1}^p e_{ij}^2 = 1 \\ \text{cov}(Y_1, Y_i) &= \sum_{k=1}^p \sum_{l=1}^p e_{1k} e_{il} \sigma_{kl} = \mathbf{e}_1' \Sigma \mathbf{e}_i = 0, \\ \text{cov}(Y_2, Y_i) &= \sum_{k=1}^p \sum_{l=1}^p e_{2k} e_{il} \sigma_{kl} = \mathbf{e}_2' \Sigma \mathbf{e}_i = 0, \\ &\vdots \\ \text{cov}(Y_{i-1}, Y_i) &= \sum_{k=1}^p \sum_{l=1}^p e_{i-1,k} e_{il} \sigma_{kl} = \mathbf{e}_{i-1}' \Sigma \mathbf{e}_i = 0 \end{aligned}$$

Therefore all principal components are uncorrelated with one another.

How do we find the coefficients e_{ij} for a principal component?

The solution involves the eigenvalues and eigenvectors of the variance-covariance matrix Σ .

Solution:

We are going to let λ_1 through λ_p denote the eigenvalues of the variance-covariance matrix Σ . These are ordered so that λ_1 has the largest eigenvalue and λ_p is the smallest.

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$$

We are also going to let the vectors \mathbf{e}_1 through \mathbf{e}_p

$$\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p$$

Denote the corresponding eigenvectors. It turns out that the elements for these eigenvectors will be the coefficients of our principal components.

The variance for the i th principal component is equal to the i th eigenvalue.

$$\text{var}(Y_i) = \text{var}(e_{i1}X_1 + e_{i2}X_2 + \dots + e_{ip}X_p) = \lambda_i$$

Moreover, the principal components are uncorrelated with one another.

$$\text{cov}(Y_i, Y_j) = 0$$

The variance-covariance matrix may be written as a function of the eigenvalues and their corresponding eigenvectors. This is determined by using the Spectral Decomposition Theorem. This will become useful later when we investigate topics under factor analysis.

Spectral Decomposition Theorem

The variance-covariance matrix can be written as the sum over the p eigenvalues, multiplied by the product of the corresponding eigenvector times its transpose as shown in the first expression below:

$$\begin{aligned} \Sigma &= \sum_{i=1}^p \lambda_i \mathbf{e}_i \mathbf{e}_i' \\ &\cong \sum_{i=1}^k \lambda_i \mathbf{e}_i \mathbf{e}_i' \end{aligned}$$

The second expression is a useful approximation if $\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_p$ are small. We

might approximate Σ by

$$\sum_{i=1}^k \lambda_i \mathbf{e}_i \mathbf{e}_i'$$

Again, this will become more useful when we talk about factor analysis.

Earlier in the course we defined the total variation of \mathbf{X} as the trace of the variance-covariance matrix, or if you like, the sum of the variances of the individual variables. This is also equal to the sum of the eigenvalues as shown below:

$$\begin{aligned} \text{trace}(\Sigma) &= \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2 \\ &= \lambda_1 + \lambda_2 + \cdots + \lambda_p \end{aligned}$$

This will give us an interpretation of the components in terms of the amount of the full variation explained by each component. The proportion of variation explained by the i_{th} principal component is then going to be defined to be the eigenvalue for that component divided by the sum of the eigenvalues. In other words, the i_{th} principal component explains the following proportion of the total variation:

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

A related quantity is the proportion of variation explained by the first k principal component. This would be the sum of the first k eigenvalues divided by its total variation.

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

Naturally, if the proportion of variation explained by the first k principal components is large, then not much information is lost by considering only the first k principal components.

4.4 Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other, but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

Linear Regression procedure:

Regression Correlation Coefficient

Correlation is used when we have quantitative data that is paired. It helps us to look at the trends in the overall distribution of that paired data. Some of the paired data shows straight line pattern but practically, the data never falls in a straight line exactly.

The correlation coefficient is sometimes also called the cross-correlation coefficient. It is a quantity that helps in determining the quality of a least squares fitting to the original data given.

For linear least squares fitting, the coefficient '**b**' in

$$y = a + b \times x$$

is given as follows:

$$b = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} = \frac{SS_{xy}}{SS_{xx}}$$

And the coefficient '**b**' in

$$x = a' + b' \times y$$

is given by

$$b' = \frac{n \sum xy - \sum x \sum y}{n \sum y^2 - (\sum y)^2}$$

The coefficient of correlation 'r' is now defined as:

$$r^2 = b \times b' = \frac{(SS_{xy})^2}{SS_{xx} \times SS_{yy}}$$

The coefficient of correlation is also known as the product moment correlation coefficient or also Pearson's coefficient. The coefficient of correlation has a very important physical interpretation.

We can say that r^2 is proportional to SS_{yy} which is recorded and accounted for regression.

LINEAR REGRESSION ANALYSIS: FITTING A REGRESSION LINE TO THE DATA

When a scatter plot is indicating that there exist a strong and linear relationship between given two variables which is confirmed by high correlation coefficient, we can then fit a straight line to the given data which may also be used to predict the value of the dependent variable for a given the value of the independent variable.

The equation of a regression line (straight line) is:

$$y = a + bx; b = \frac{S_{xy}}{S_{xx}}$$

$$a = \bar{y} - b\bar{x} = \frac{\sum_i y_i - b \sum_i x_i}{n}$$

where

$$\bar{x} = \frac{\sum x}{n} = \text{mean of the } x \text{ data}$$

$$\bar{y} = \frac{\sum y}{n} = \text{mean of the } y \text{ data}$$

4.5 K-means

K-means is one of the oldest and most commonly used clustering algorithms. It is a prototype based clustering technique defining the prototype in terms of a centroid which is considered to be the mean of a group of points and is applicable to objects in a continuous n-dimensional space.

The K-means algorithm involves randomly selecting K initial centroids where K is a user defined number of desired clusters. Each point is then assigned to a closest centroid and the collection of points close to a centroid form a cluster. The centroid gets updated according to the points in the cluster and this process continues until the points stop changing their clusters.

K-means procedure:

k-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function known as squared error function given by:

$$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (\|x_i - v_j\|)^2$$

Where,

' $\|x_i - v_j\|$ ' is the Euclidean distance between x_i and v_j .

' c_i ' is the number of data points in i_{th} cluster.

' c ' is the number of cluster centers.

Algorithmic steps for k-means clustering

Let $X = \{x_1, x_2, x_3, \dots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \dots, v_c\}$ be the set of centers.

- 1) Randomly select 'c' cluster centers.
- 2) Calculate the distance between each data point and cluster centers.
- 3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
- 4) Recalculate the new cluster center using:

$$v_i = (1 / c_i) \sum_{j=1}^{c_i} x_j$$

Where, 'ci' represents the number of data points in i_{th} cluster.

- 5) Recalculate the distance between each data point and new obtained cluster centers.
- 6) If no data point was reassigned then stop, otherwise repeat from step 3).