Lecture 9 - Data Mining Basics

Yingda Zhai (Dr.)

School of Computing, NUS

October 11, 2022

BT1101 Roadmap: Predictive (7-10), Prescriptive (11-12)

Week 1 - 6	Descriptive Analytics
Week 7	Linear Regression
Week 8	Logistic Reg & Time Series
Week 9	Data Mining Basics
Oct 18	*Online Assessment*
Week 11	Linear Optimization
Week 12	Integer Optimization & Summary
Week 13	Tutorials and Consultation
Exam Wk	*Final Exam*

1 Model Selection

- Feature Selection
- Data Visualization: Panel of Scatterplots
- "Restricted" vs. "Unrestricted": F-Test
- Stepwise Model Selection
- Best Practice for Model Building

2 Data Dimensionality Reduction

- Information is About Variations in Data
- Dimensionality Reduction
- Principal Component Analysis
- Intuition of Principal Component Analysis
- Principal Component Analysis in R
- Visualization of PCA
- Model Selection vs. Data Dimensionality Reduction

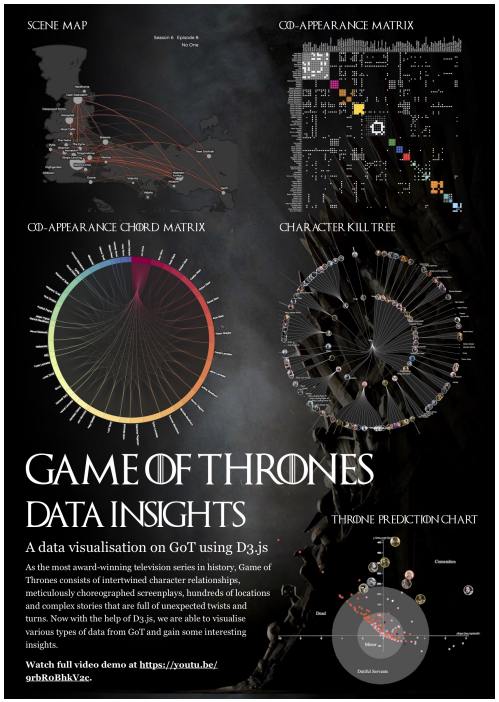
3 Clustering and Classification

- Unsupervised Clustering: k-Means
- How k-Means Works?
- Visualization of Clustering
- Supervised Classification: Logistic Regression
- Evaluating Classifiers: Classification Matrix

NUS National University of Singapore

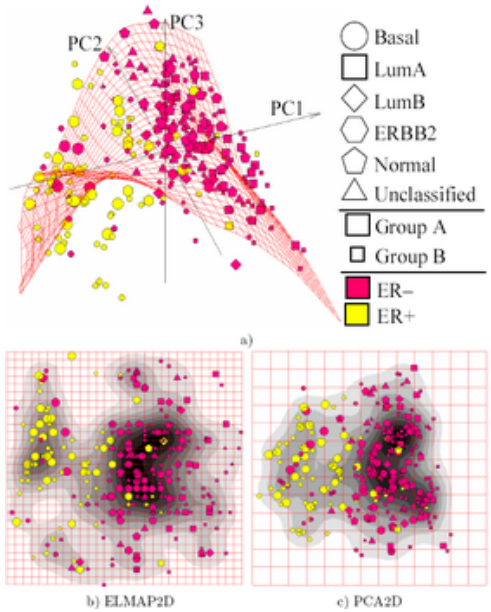
Learn Wisdom from Data

Data Visualization



Lords&kings in Westeros

Data Reduction



Data dimensionality reduction

Machine Classification

Train your machine to swipe Tinder for you

Learn Wisdom from Data

- Data mining is an approach to uncover hidden patterns and extract information/knowledge from large data set. (*without prior theory*)
- Data analysis is to use data to **test** and **validate** theories, models and hypotheses. (*with prior theory*)
- Today's world of big data makes data mining as well as machine learning algorithms very popular and attractive.
- But be cautious of problems of generalization and overfitting.

NUS National University of Singapore

Learning Objectives

- Understand the concept and best practice of model selection; Be able to conduct F-test on linear combination of regression coefficients and forward/backward stepwise model selection.
- Understand the concept of data dimensionality reduction; Be able to apply principal component analysis to summarize information in a large dataset. Understand the similarity and difference between model selection and data dimensionality reduction.
- Understand basic unsupervised clustering technique such as k-mean and supervised classification such as logistic regression and be able to assess the quality of classifier using classification matrix.
- Be able to visualize (plot) the output of various techniques above.

Model Selection

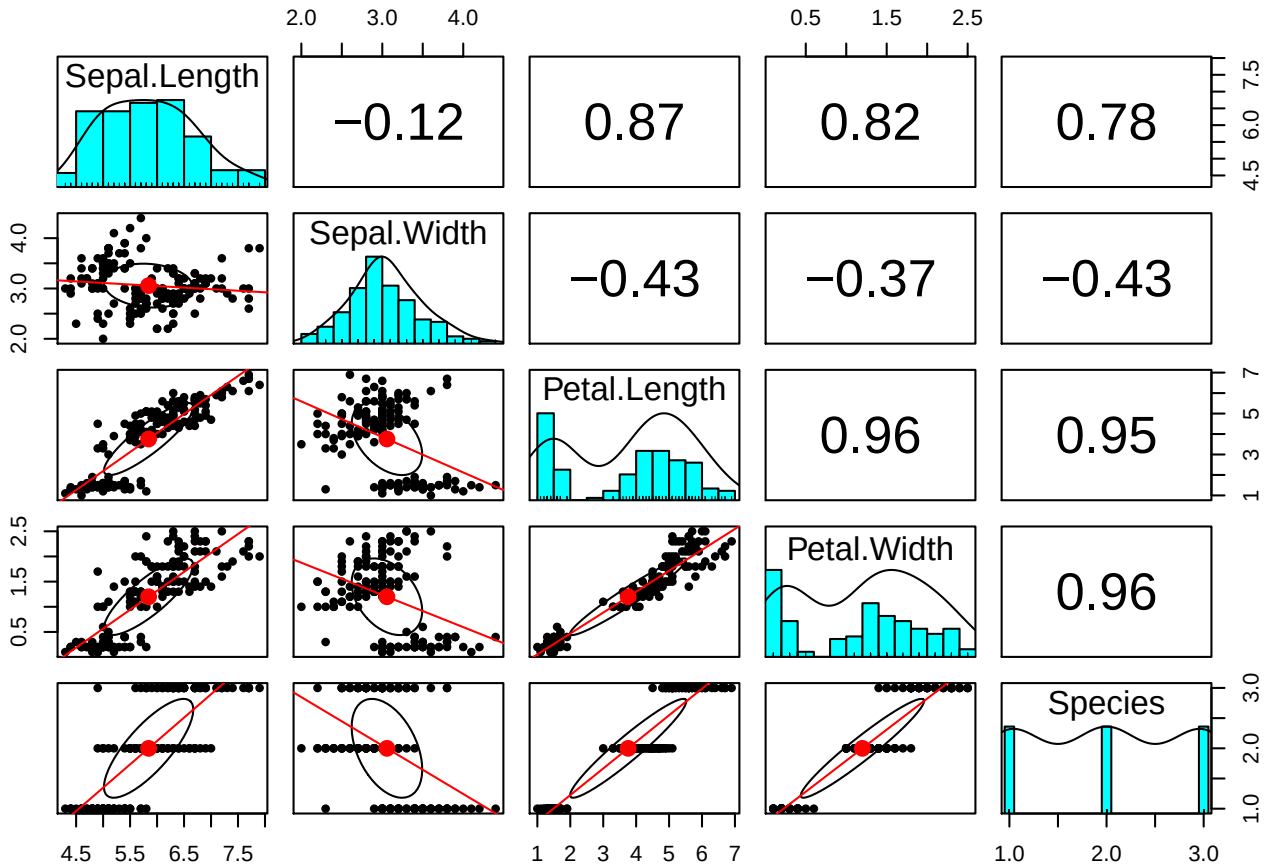
Model Selection

- Model selection is a procedure of selecting the "best" statistical model among candidate ones, given data and selection rule.
- Model selection is often about selecting features that best explain *the response variable*, thus a.k.a feature selection.
- Given a large cross-sectional data set (i.e. many observations and many variables), how do we select a regression model to explain Y? Or put in another way, which $\boldsymbol{X}$'s should be included in the model?

Data Visualization

- Data visualization is simply a graphic representation of data. In model selection, a panel of scatterplots is helpful for us to visualize the relationships in variables. `psych::pairs.panels` is one way to do this.

```
# use 'iris' data in R as illustrative example
data(iris); pairs.panels(iris, lm=TRUE)
```



Model Selection: F-Test (Chow Test)

- Consider a "kitchen-sink" or an "unrestricted" model including **all $\boldsymbol{X}$**'s:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i, \text{ for } i = 1, \ldots, n. \quad (1)$$

- Consider a "restricted" model with the restriction being $\beta_1 = 0$:

$$y_i = \beta_0 + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i, \text{ for } i = 1, \ldots, n. \quad (2)$$

- A F-test (a.k.a Chow test) can be conducted to test: **"Is the explanatory power of unrestricted model is significantly better than that of the restricted model (in which X_1 is excluded)?"**
- A *large* F-statistic or a *small* p-value of such F-test shows evidence to **reject the null hypothesis**, (H_0 : unrestricted model is not significantly better than restricted one in terms of explanatory power for Y.)

RM:
1. F-statistic $= \frac{(\text{SSR}_r - \text{SSR}_u)}{\text{SSR}_r/(n-k+1)}$ where SSR_u and SSR_r are **sum of squares residuals** for unrestricted and restricted model, respectively. Fact: $\text{SSR}_r \geq \text{SSR}_u$.
2. The restriction can actually be any linear combination of β's, e.g. $\beta_1 = \beta_2 = 0$, $2\beta_3 + 5\beta_9 = 7$, etc. F-test in ANOVA (back in L7) to test **all slopes $\beta = 0$** is a special case of Chow test.

Model Selection: *F*-Test in R

- Use `mroz` as an example, restricted model excludes `hours`:

```
# fit "restricted" and "unrestricted" models with OLS
fit_restricted = lm(wage ~ educ + age + faminc + unem + city + exper +
    expersq, mroz)
fit_unrestricted = lm(wage ~ hours + educ + age + faminc + unem + city
    + exper + expersq, mroz)
# use 'anova(mod1, mod2)' to conduct the F-test
anova(fit_restricted, fit_unrestricted)
```

```
Model 1: wage ~       educ+age+faminc+unem+city+exper+expersq
Model 2: wage ~ hours+educ+age+faminc+unem+city+exper+expersq
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    745 6413.5
2    744 5754.5  1    659.05 85.209 < 2.2e-16 ***
```

RM:
1. It's not coincidence that SSR if greater for restricted model.
2. Small p-value of F-test shows strong evidence to reject the null. In this case, slope for `hours` is statistically non-zero and unrestricted model is significantly better in terms of explanatory power (by including `hours`).
3. F-test by excluding only one variable is called *partial* F-test.

Stepwise Model Selections

- Since we can "trial-and-error" to select (i.e. include/exclude) predictors $\boldsymbol{X}$'s, there are generally two directions to do this *systematically*:
  - Backward stepwise selection: start with "full" model, conduct (patrial) F-tests for all included predictors and exclude the predictor with **lowest** F-statistics; repeat with updated model until endpoint.
  - Forward stepwise selection: start with intercept-only model, conduct F-tests for all *potential* predictors and add the one with **largest** F-statistics; repeat with updated model until endpoint.
- We can manually conduct the backward and forward stepwise model selection by using `drop1()` and `add1()` in RscriptL9.R.
- Instead of such **F-based** model selection, sometimes a more efficient approach is a stepwise model selection based on **lower AIC** score. In R, `step()` automates such process for you.

RM: Such "endpoint" could be all remaining predictors have p-values less than 0.05 in their partial F-tests.

Stepwise Model Selection in R: `step('backward')`

```
# automated backward stepwise selection based on AIC score
step(model_full, direction = 'backward')
```

```
Start:  AIC=1549.35
wage ~ hours + age + educ + faminc + unem + city + exper + expersq
          Df Sum of Sq    RSS    AIC
- unem     1      0.29 5754.8 1547.4
- city     1      0.67 5755.2 1547.4
- age      1      6.30 5760.8 1548.2
- expersq  1     12.84 5767.3 1549.0
<none>                 5754.5 1549.3
- exper    1     44.57 5799.1 1553.2
- faminc   1     63.40 5817.9 1555.6
- educ     1    342.66 6097.2 1590.9
- hours    1    659.05 6413.5 1629.0
... (other steps omitted)
Final Step:  AIC=1544.28
wage ~ hours + educ + faminc + exper + expersq
```

RM: First step of backward stepwise selection is shown above, each row shows the result of partial F-test by excluding (thus "-" sign) corresponding predictor.

Stepwise Model Selection in R: `step('forward')`

```
step(model_intercept, scope = ~ hours + age + educ + faminc + unem +
    city + exper + expersq, direction = 'forward')
```

```
Start:  AIC=1772.26
wage ~ 1
          Df Sum of Sq    RSS    AIC
+ hours    1   1413.72 6489.4 1625.9
+ educ     1    801.10 7102.0 1693.8
+ exper    1    496.24 7406.9 1725.4
+ faminc   1    422.36 7480.8 1732.9
+ expersq  1    294.80 7608.3 1745.6
+ city     1     33.81 7869.3 1771.0
<none>                7903.1 1772.3
+ age      1      9.44 7893.7 1773.4
+ unem     1      0.00 7903.1 1774.3
... (other steps omitted)
Final Step: AIC=1544.28
wage ~ hours + educ + exper + faminc + expersq
```

RM:
1. In forward selection, a list of potential predictors need to be specified in `scope`.
2. Note that backward and forward stepwise selection agree on the final model, which is *not* always the case.

Best Practice for Model Building

1. **Write down hypothesis** with pre-selected predictors using theory, intuition, experience, etc.
2. **Check data and assumptions** for relationships, distributional assumption, missingness, etc.
3. **Model Selection** with a systematic approach such as stepwise selection.
4. **Evaluate and interpret model** with RMSE, AIC/BIC and meaningful interpretation in English!

RM:
1. Remember:*principle of parsimony*: simpler model often performs better, all things equal; correlation $\neq$ causality!
2. Top-bottom: data analysis; bottom-top: data mining.
3. Think model selection as a journey to discover your data.

NUS National University of Singapore

Data Dimensionality Reduction

Data Dimensionality Reduction

- Recall in `mroz`, we introduced variations of all our predictors $\boldsymbol{X}$ such as `hours`, `educ`, etc. to explain observed variation of response var `wage` (Y).

person	wage	hours	educ	motheduc	fatheduc	faminc	...
Mary	\$3354	1610h	12	12	7	\$16310	...
Susan	\$1389	874h	11	7	7	\$23762	...
Erica	\$4545	1598h	14	12	14	\$956	...

- For big data, we potentially have hundreds of predictors (i.e. high-dimensional). Including all predictors in regression model have unwanted consequences, such as over-fitting, low power in hypothesis testing, multicollinearity, etc.

Question:

How could we extract most amount of information (variations) from all our data and at the same time not include all predictors $\boldsymbol{X}$ in the model?

Data Dimensionality Reduction

- Simply construct a new *indexing* predictor combining several predictors! e.g. $\texttt{fameduc} = 0.5\texttt{motheduc} + 0.5\texttt{fatheduc}$:

person	wage	hours	educ	motheduc	fatheduc	fameduc	...
Mary	$3354	1610h	12	12	7	9.5	...
Susan	$1389	874h	11	7	7	7	...
Erica	$4545	1598h	14	12	14	13	...

- Instead of using two predictors, we could only include fameduc, which summarizes information (variations) in motheduc and fatheduc.
- A systematic approach to find a smaller set of *principal predictors* which extract and summarize information of high-dimensional data is called data dimensionality reduction.

Data Dimensionality Reduction: Principal Component Analysis (PCA)

- One idea is to identify such smaller a set of principal components (PC's) such that along their dimensions, variations of all predictors are explained the most by them.

person	wage	pc1	pc2	pc3	...	hours	educ	$\boldsymbol{X}$...
1. Mary	\$3354	$\boldsymbol{X}_1\phi_1$	$\boldsymbol{X}_1\phi_2$	$\boldsymbol{X}_1\phi_3$	...	1610h	12	$\boldsymbol{X}_{1k}$...
2. Cath	\$1389	$\boldsymbol{X}_2\phi_1$	$\boldsymbol{X}_2\phi_2$	$\boldsymbol{X}_2\phi_3$	...	874h	11	$\boldsymbol{X}_{2k}$...
3. Erica	\$4545	$\boldsymbol{X}_3\phi_1$	$\boldsymbol{X}_3\phi_2$	$\boldsymbol{X}_3\phi_3$	...	1598h	14	$\boldsymbol{X}_{3k}$...

- We would like a set of principal components such that:
  1. pc1 explains the most variation of all $\boldsymbol{X}$; pc2 is **orthogonal** to pc1 and explains second most variation and so forth.
  2. Each principal component is a normalized **linear combination** of all predictors $\boldsymbol{X}$, e.g. for pc1 (similar for other pc's)

$$\text{pc1} = \phi_{11}X_1 + \phi_{12}X_2 + \cdots + \phi_{1k}X_k, \text{ s.t. } \sum_k \phi_{1k}^2 = 1. \quad (3)$$

- Such approach of finding principal components are called principal component analysis (PCA).

Principal Component Analysis (PCA)

- Given total k predictors, PCA generates total k **orthogonal** principal components, ranked by percentage of variation of all predictors $\boldsymbol{X}$ explained. For p-th principal component, $p = 1, 2, \ldots, k$:

$$\text{pc}[p] = \phi_{p1} X_1 + \phi_{p2} X_2 + \cdots + \phi_{pk} X_k, \text{ s.t. } \sum_{i=1}^{k} \phi_{pi}^2 = 1; \tag{4}$$

RM:

1. The linear weights of predictors on $\text{pc}[p]$, $(\phi_{p1}, \phi_{p2}, \cdots, \phi_{pk})$, is called loading of pc's. It describes how $\text{pc}[p]$ is composed of all k predictors.
2. Note that each PC "borrows" information from **all predictors $\boldsymbol{X}$**. No predictor is excluded from PCA.
3. As for dimensionality reduction, we choose several **top ranked PC's**.
4. The very definition (4) of principal components make them **difficult to interpret** since they are rescaled linear indices of predictors: the price we pay for dimensionality reduction.

Principal Component Analysis: Intuition

- Imagine the "data cloud" formed by each data point $(x_1, x_2, \ldots, x_k)$. `pc1` is chosen to be along the "longest" dimension of the data cloud where:
  1. The projected data points onto the dimension of `pc1` have largest variations.
  2. The distance between the projected data points onto the dimension of `pc1` and data point themselves are smallest.
- *Given* `pc1` is fixed, `pc2` is chosen to be along the second "longest" dimension of the data cloud and have be **orthogonal** to `pc1`; and so on.

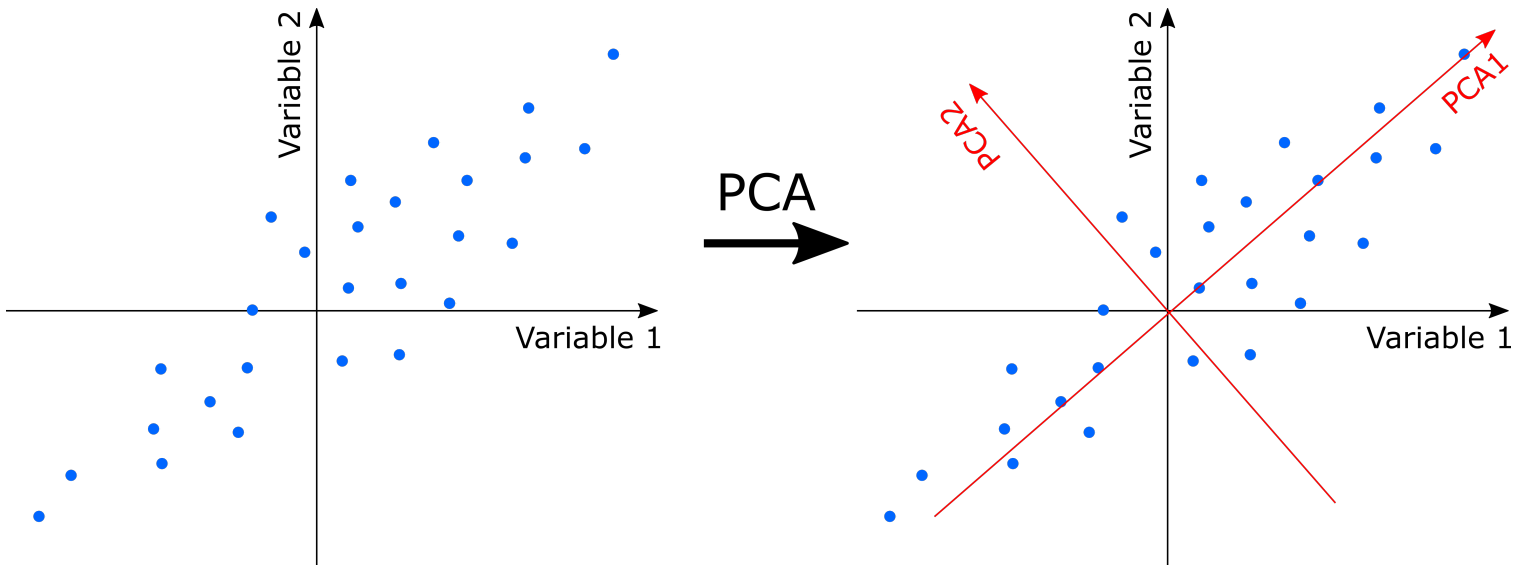


Figure: An example of PCA in 2D Euclidean space

Principal Component Analysis: Intuition

- Equivalently, pc's identify an alternative coordinates in which:
  1. The projected data points onto the dimension of pc1 have largest variations.
  2. The distance between the projected data points onto the dimension of pc1 and data point themselves are smallest.
- *Given* pc1 is fixed, pc2 is chosen to be along the second "longest" dimension of the data cloud and be **orthogonal** to pc1; and so on.

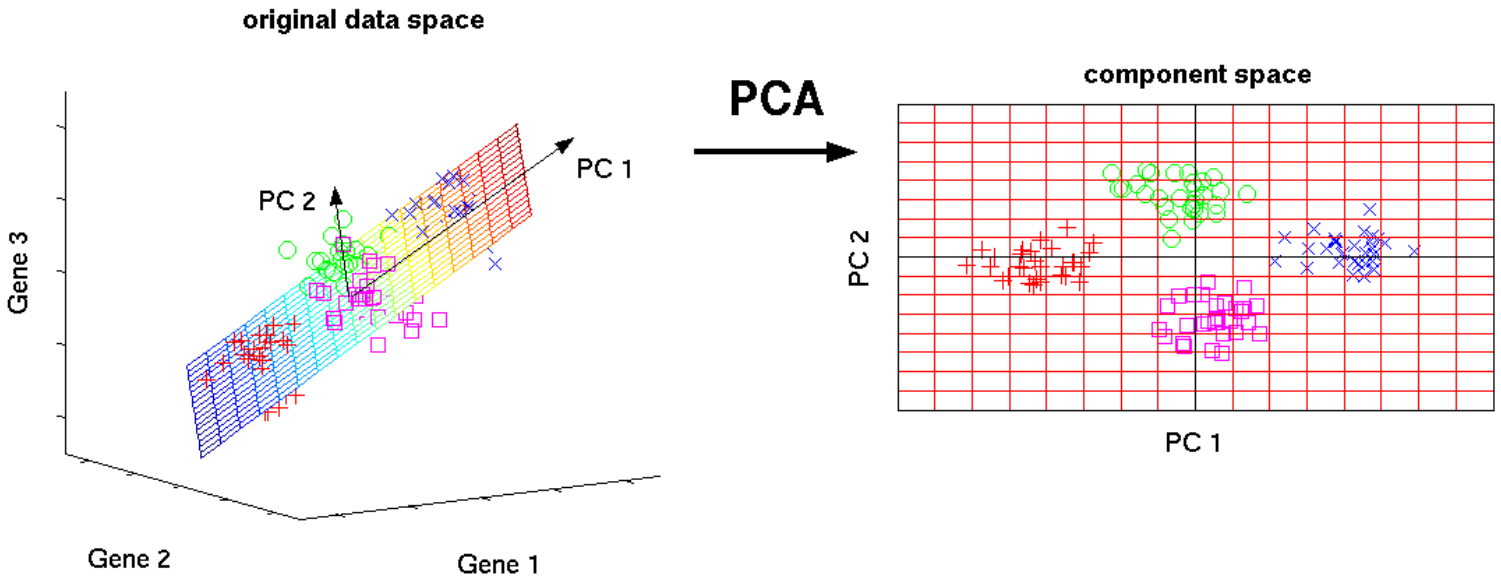
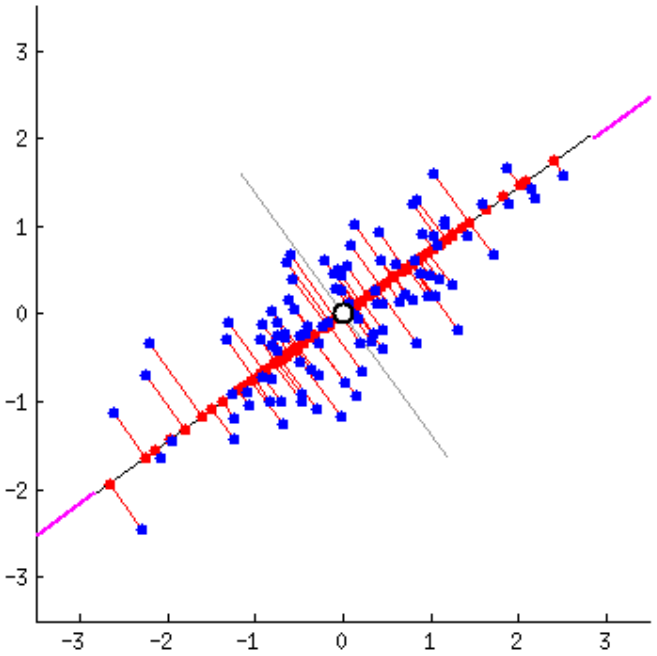


Figure: An example of PCA in 3D Euclidean space (source: nlpca.org)

Principal Component Analysis: Intuition

- Equivalently, pc's identify an alternative coordinates in which:
  1. The projected data points onto the dimension of pc1 have largest variations.
  2. The distance between the projected data points onto the dimension of pc1 and data point themselves are smallest. (pca visualization)



RM:
- Projected distances: red lines (projected data points: red dots).
- At PCA position, 1 and 2 must hold true simultaneously. (can you see why? hint: Pythagoras theorem.)

Running Principal Component Analysis in R

- In `mroz`, we can use PCA to summarize information from **all predictors** except for `wage` and `city` (total 7 predictors).
  - The former, `wage`, is our dependent variable which we try to explain;
  - The latter, `city`, is a categorical variable which won't be handled in baseline PCA model.

```
# call 'rcomp()' with 'scale = TRUE'.
pca_mroz = prcomp(formula = ~ . -wage -city, data = mroz, center = TRUE, scale = TRUE)
summary(pca_mroz)
```

```
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7
Standard deviation     1.5265 1.1991 1.0548 0.9358 0.8477 0.68376 0.23993
Proportion of Variance 0.3329 0.2054 0.1590 0.1251 0.1027 0.06679 0.00822
Cumulative Proportion  0.3329 0.5383 0.6972 0.8223 0.9250 0.99178 1.00000
```

RM:
1. Data reduction techniques with categorical variables won't be covered.
2. pc's are ranked by their proportion of variation explained along their dimensions. It is natural to see top ranked pc's also have largest standard deviations. Top 3 pc's account for about 70% percent of variation in $\boldsymbol{X}$.
3. `scale = TRUE` is recommended since it standardizes all predictors, making them comparable.

Running Principal Component Analysis in R

- Recall $(\phi_{11}, \phi_{12}, \ldots, \phi_{1k})$ in (3) is called the **loading** of pc1:
$$\text{pc1} = \phi_{11}X_1 + \phi_{12}X_2 + \cdots + \phi_{1k}X_k, \text{ s.t. } \sum_i \phi_{1i}^2 = 1.$$
- The loading $(\phi_{11}, \phi_{12}, \ldots, \phi_{1k})$ not only describes how pc1 is composed of all 7 predictors $\boldsymbol{X}$ but also points to the direction of coordinates rotation in Euclidean space.

```
# examine the loading of PC1 and PC2 in first two columns of 'rotation'.
rbind(pca_mroz$rotation[,1],pca_mroz$rotation[,2])
```

```
> rbind(pca_mroz$rotation[,"PC1"],pca_mroz$rotation[,"PC2"])
          hours        age        educ      faminc        unem       exper     expersq
PC1  -0.3456316 -0.3219658 -0.04915844 -0.03124984 0.001989362 -0.62445188 -0.61930578
PC2   0.2707647 -0.2092560  0.66505204  0.64363668 0.130403688 -0.03808345 -0.08877245
> sum(pca_mroz$rotation[,1]^2)
[1] 1
```

RM: 1 To check if our result is correct by using the normalization condition $\sum_k \phi_{1k}^2 = 1$ for PC1, »sum(pca_mroz$rotation[,1]^2) gives the result which is exactly equal to one.

2 `pca_mroz$rotation` contains the loadings for all 7 PC's.

Running Principal Component Analysis in R

- Now let's run a linear regression of wage on our top 3 ranked PC's since they summarize almost 70% of variation in all predictor $\boldsymbol{X}$.

```
# extracting top 3 PC's to run a linear regression of 'wage ~ pc1 + pc2 + pc3'
mroz_pca = mroz
mroz_pca$pc1 = pca_mroz$x[,"PC1"]
mroz_pca$pc2 = pca_mroz$x[,"PC2"]
mroz_pca$pc3 = pca_mroz$x[,"PC3"]
pcafit = lm(wage ~ pc1 + pc2 + pc3, mroz_pca)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.37457    0.10262  23.139  < 2e-16 ***
pc1         -0.60385    0.06727  -8.976  < 2e-16 ***
pc2          1.02724    0.08564  11.995  < 2e-16 ***
pc3          0.46880    0.09735   4.816 1.78e-06 ***
---
Residual standard error: 2.816 on 749 degrees of freedom
Multiple R-squared:  0.2485,  Adjusted R-squared:  0.2455
F-statistic: 82.54 on 3 and 749 DF,  p-value: < 2.2e-16
```

RM:
1. Note that all 3 principal components are strongly significant. The $R^2 = 0.2485$ is decently high and large F-stat indicates a powerfully explanatory linear model.
2. However, the coefficient estimates of principal components are difficult to interpret since they are just some constructed indices.

Data Visualization: PCA Biplot

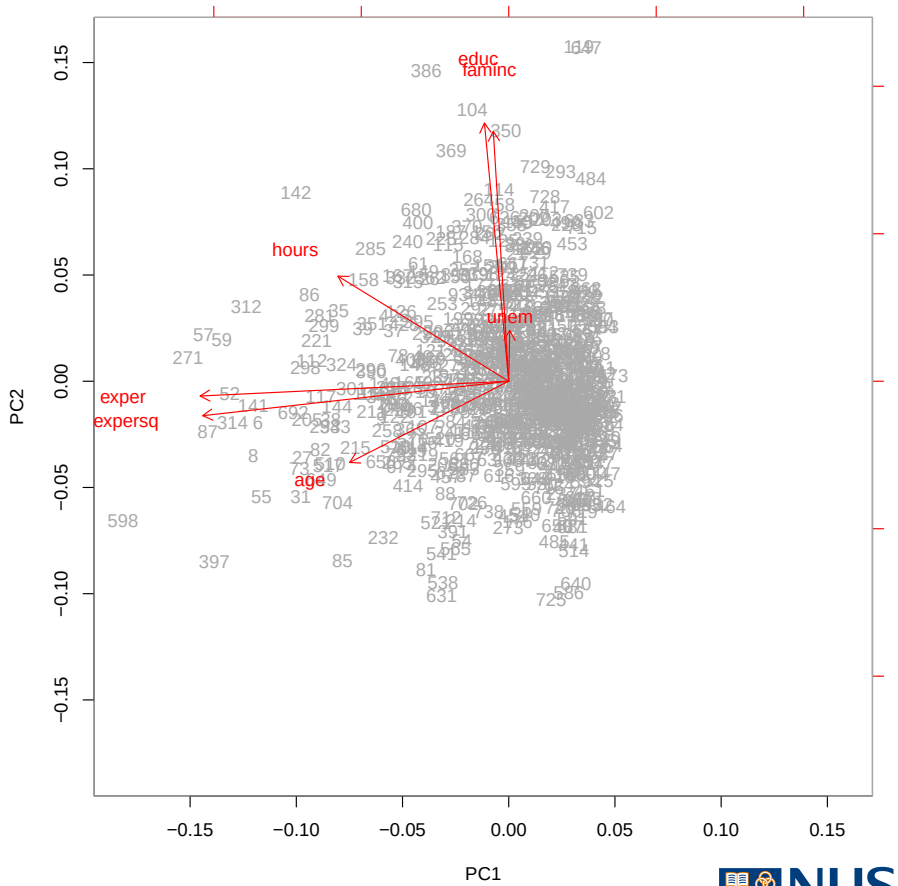
```
> rbind(pca_mroz$rotation[,"PC1"],pca_mroz$rotation[,"PC2"])     # loadings of pc1 and pc2
          hours        age        educ      faminc        unem       exper     expersq
PC1  -0.3456316 -0.3219658 -0.04915844 -0.03124984 0.001989362 -0.62445188 -0.61930578
PC2   0.2707647 -0.2092560  0.66505204  0.64363668 0.130403688 -0.03808345 -0.08877245
```

- We can use the PCA biplot to show the compositions of **top 2 PC's** with respect to all predictors.

```
# call 'biplot()' for 'prcomp' obj
biplot(pca_mroz)
```

RM:

1. You can see how data points (grey) look like in the "plane" formed by pc1 and pc2 (where most variation should be).
2. The red arrows in biplot are pointing in the direction of the original predictors, as projected into the 2D plane of the biplot.
3. For instance, the arrow direction of hours is given by (-0.346, 0.271) in the coordinate set up by pc1 and pc2. Similar to other arrows.



PCA biplot of PC1 and PC2

Model Selection and Data Dimensionality Reduction

- Both methods are used for **reducing** the number of features.
  1. Model selection which is also called feature selection, is often about selecting features that best explain the response variable.
  2. Data dimensionality reduction is about finding fewer principal predictors that summarize information of high-dimensional data set.
- However, model selection is **selecting and excluding** available features without changing them while data dimensionality reduction first summarize information in **all features by transformation** and then select fewer transformed features to represent original high-dimensional data.
  1. Model selection: loss of information but more interpretable.
  2. Data dimensionality reduction: preserve information better but reduced interpreability.
- No free lunch: always a price to pay!

Clustering and Classification

Clustering Analysis

- Model selection and dimensionality reduction are about selecting and "compressing" features (columns), any insight from the observations (rows)?
- We could identify clusters of observations who share similarity in their features.

person	wage	hours	educ	motheduc	fatheduc	faminc	...
Mary	$3354	1610h	12	12	7	$16310	...
Susan	$1389	874h	11	7	7	$23762	...
Erica	$4545	1598h	14	12	14	$956	...
Audrey	$1179	692h	6	8	6	19438	...

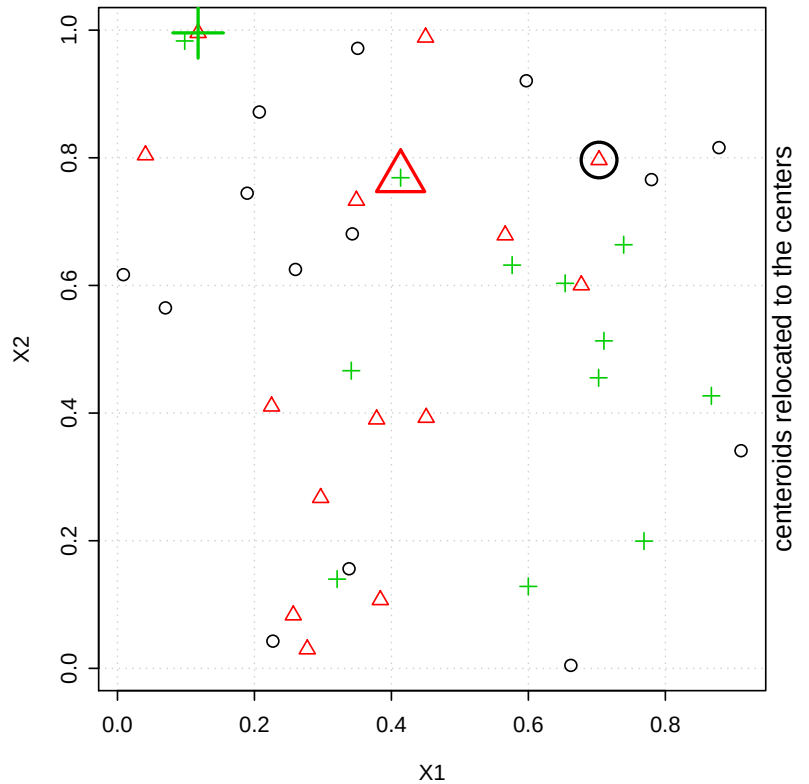
- k-means partitions n observations into k clusters in which each observation is assigned to the nearest centroid (mean) and **within-cluster distance** is minimized.
- k-means is a typical example of unsupervised learning which uncovers patterns in data **without pre-labeled data by human**.

How k-Means Algorithm Works?

G: Each data point is assigned to a cluster such that the sum of squared distance from all datapoints in a cluster to the cluster center, is minimized.

- We need to decide the number of cluster k before algorithm starts.
- k-means clustering algorithm mainly consists of two steps, which are repeated over and over again until convergence.

1 **Initialization Step**: Place the centroids of k clusters on k randomly chosen datapoints.

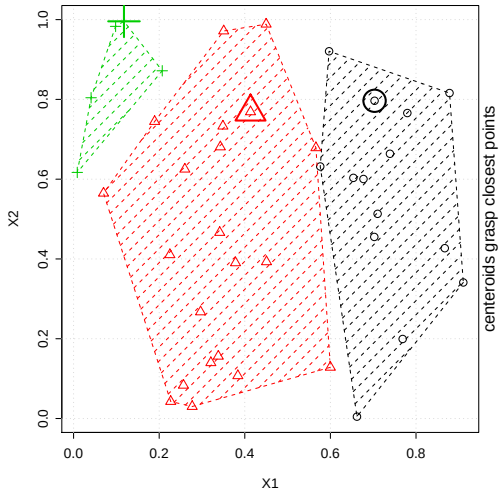


$k = 3$ centroid's locations are randomly initialized.

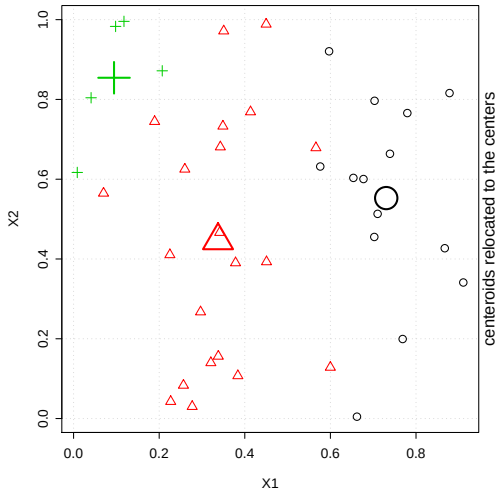
How k-Means Algorithm Works?

2. **Assignment Step**: Distance from each datapoint to all centroids are computed such that datapoints are "assigned" to the cluster with the closest centroid.

3. **Update Step**: Update the centroid position to be the mean of all points assigned to that cluster.



(a) Each data point is assigned to the closest centroid.

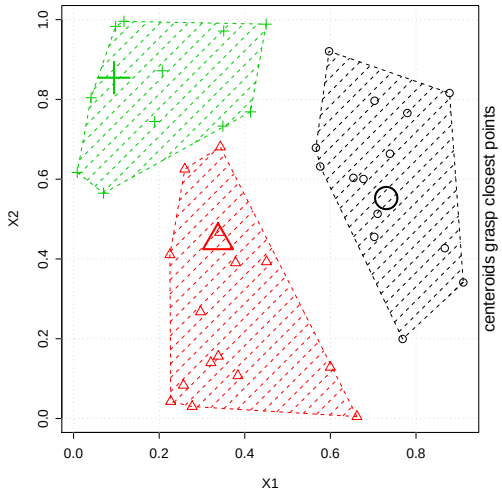


(b) If centroid is not in the center, move them to the center.

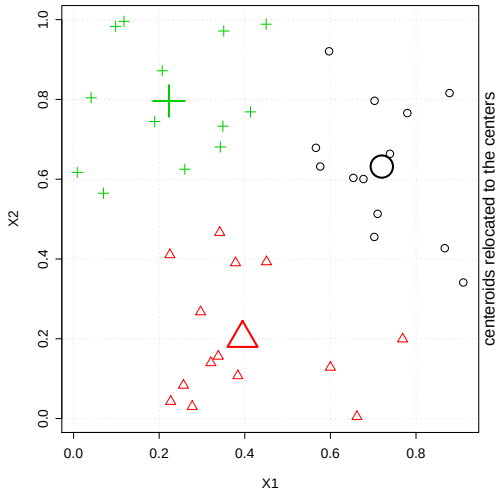
How k-Means Algorithm Works?

2. **Assignment Step**: Distance from each datapoint to all centroids are computed such that datapoints are "assigned" to the cluster with the closest centroid.

3. **Update Step**: Update the centroid position to be the mean of all points assigned to that cluster, until no more repositioning is needed.



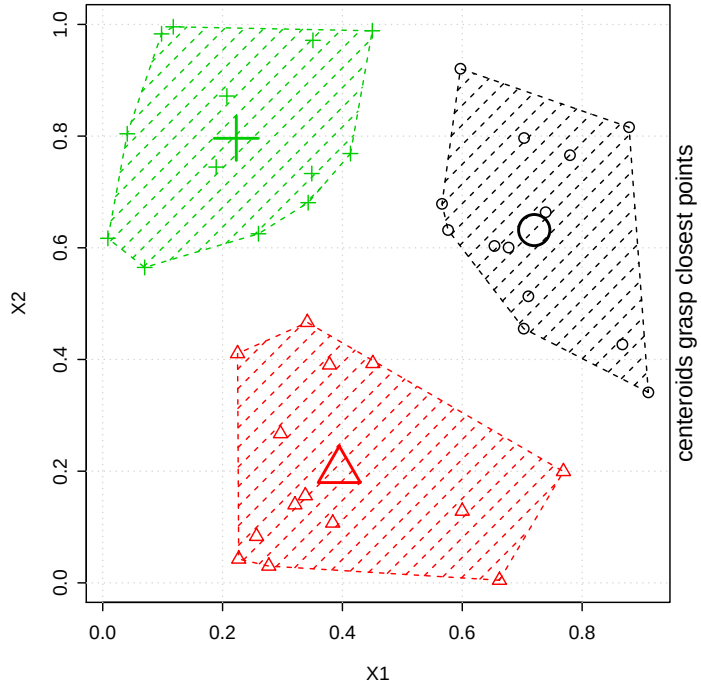
(a) Each data point is assigned to the closest centroid.



(b) If centroid is not in the center, move them to the center.

How k-Means Algorithm Works?

- k-means algorithm stops when there no more changes of centroid's positions or pre-specified maximum number of iterations is reached.
- Note that there is no guarantee that k-means will reach the global optimum, i.e. each run of k-means algorithm may depend on the random initialization.

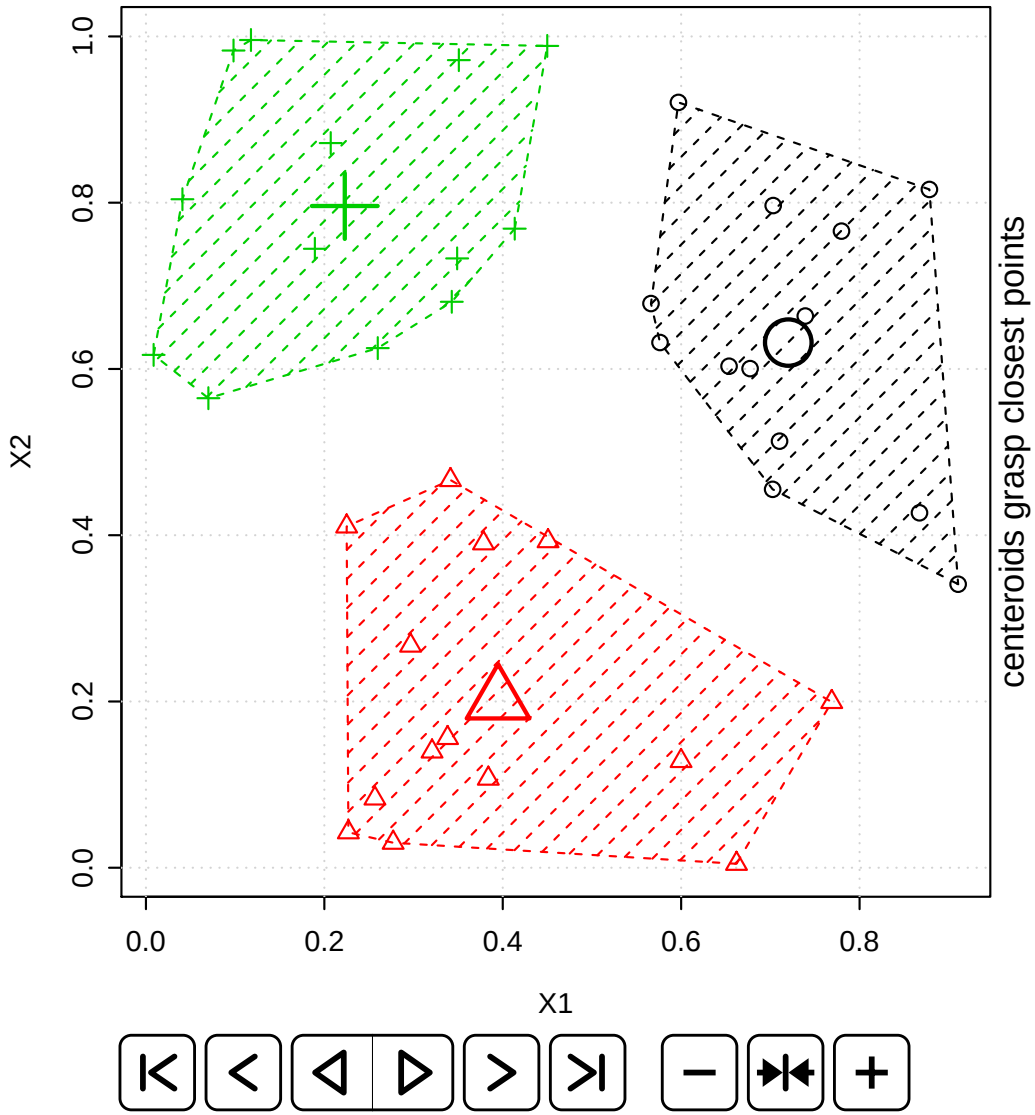


Clusters are stablized with no more relocation of centroids.

```
# run a k-means algorithm using 'kmeans()' function
km_mroz = kmeans(mroz, centers = 3, nstart = 10)
```

RM: `nstart` specifies number of initialization. R repeats `kmeans()` with 10 random initializations and report the best one with *the lowest within-cluster sum of squared distance*.

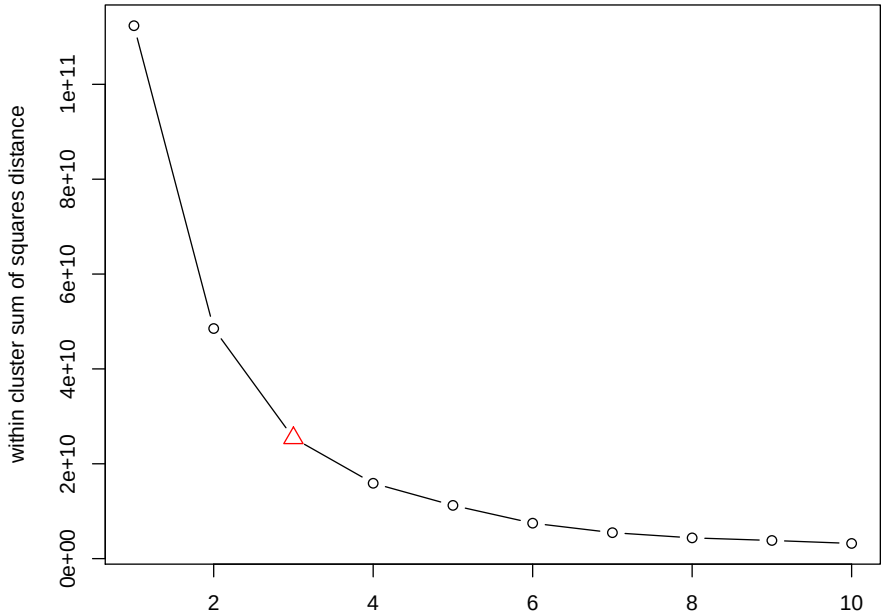
How k-Means Algorithm Works?



How To Determine Number of Cluster k?

- Since k needs to be specified beforehand, how to determine the best number of clusters k?

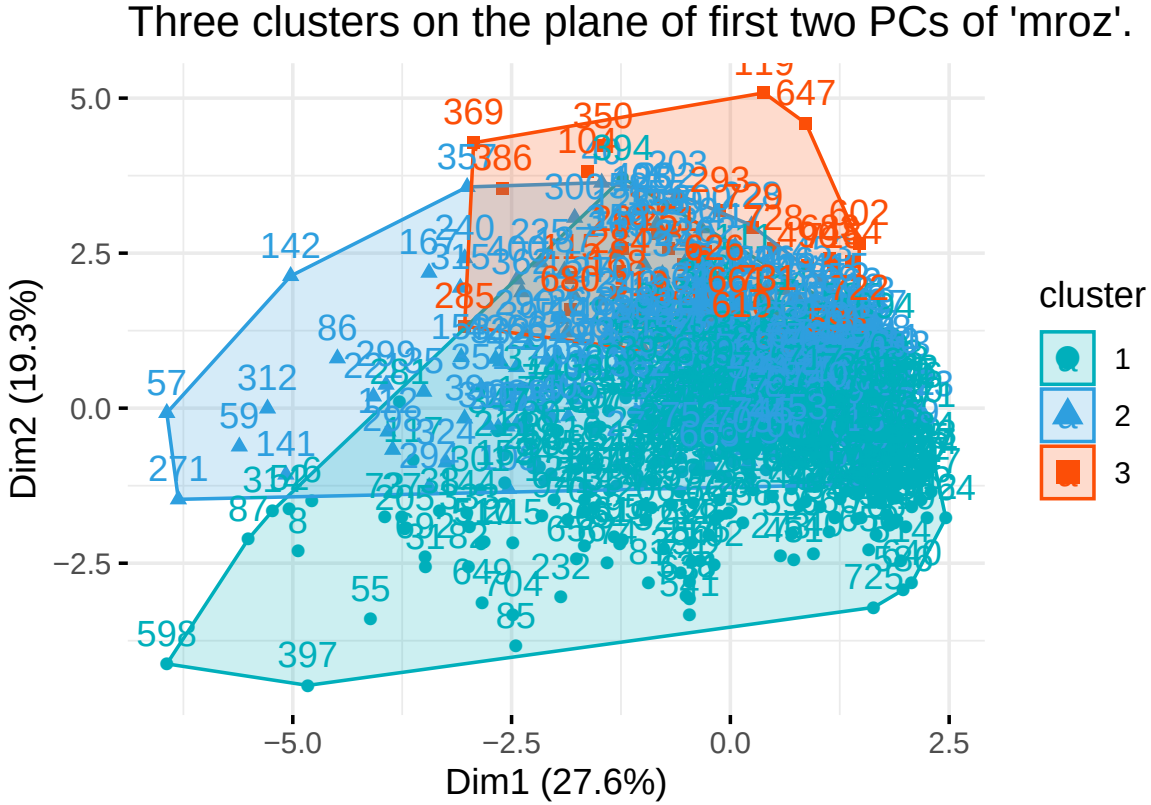
RM: When choosing the best number of clusters, look for the "elbow"! In this case, either $k = 3$ or $k = 4$.




```
# plot 'within-cluster sum of squared distance' as a function of #clusters
wss = rep(NA, 10)
for (k in c(1:10)){
  wss[k] = kmeans(mroz, k, nstart = 10)$tot.withinss
}
cols = rep('black', length(wss))
```

Plot k-Means in `mroz`

```
km_mroz = kmeans(mroz, centers = 3, nstart = 10)
# k-means plot using 'fviz_cluster' in 'factoextra' package;
fviz_cluster(km_mroz, data = mroz,
             palette = c("#00AFBB","#2E9FDF", "#FC4E07"),
             ggtheme = theme_minimal(),
             main = "Three clusters on the plane of first two PCs of 'mroz'.")
```



RM: `fviz_cluster()` applies PCA first and plots the k-means clustering of observations that are projected onto the "plane" of top two PC's.

Classification: Logistic Regression

- Classification, which is the mostly used machine learning technique, identifies which category a new observation belongs to, based on a trained **classifier** on a training set of data in which membership of observations are already labeled.
- It is supervised learning technique since such classifer has to be trained on pre-labeled data by human.

Customer	Previous Spending	Marital Status	#Ads Displayed	*Purchased*
Andy	\$476	Married	3	Yes
Charlie	\$169	Single	2	No
Ashley	\$23	Married	6	No

- There are many classifiers: naive Bayes, k-nearest neighbor, various neural networks (which you can learn in future courses) and logistic regression!

$$\text{logit}(p) = \log p/(1-p) \sim \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

RM: p is exactly the probability of **a label for an event being positive**, e.g. purchase in Amazon, ads-click, like in Instagram, etc.

Classification (Confusion) Matrix

- Let's focus on the more fundamental: how to evaluate performance of our classifers? Classification matrix or confusion matrix (you shall see why...)

	Actual: Yes	Actual: No
Predicted: Yes	true positive (TP)	false positive (FP)
Predicted: No	false negative (FN)	true negative (TN)

Sensitivity	$TP/(TP+FN)$	proportion of predicted yes out of actual yes
Specificity	$TN/(FP+TN)$	proportion of predicted no out of actual no
Precision	$TP/(TP+FP)$	proportion of actual yes out of predicted yes
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$	proportion of correction classification
F1-Score	$\frac{2*Precision*Sensitivity}{Precision+Sensitivity}$	balanced index between precision and sensitivity

RM:
1. High precision: lower chance of misclassifying healthy individuals (causing distress and wasted testkits).
2. High sensitivity: improves detecting true sick cases (otherwise the undetected go untreated).

Classification Matrix in Hypothesis Testing: Type I and Type II Errors

- Classification matrix in hypothesis testing give us **type I and type II errors**:

1. Type I error (α): the probability that we reject the null, given the null is true, i.e. reject a true null, i.e. Pr(reject $H_0|H_0$ is true) $= \alpha$.
2. Type II error (β): the probability that we fail to reject the null, given the null is false, i.e. fail to reject a false null, i.e. Pr(don't reject $H_0|H_0$ is false) $= \beta$.

Error Types Table	H_0 is False	H_0 is True
Reject H_0	correct rejection $(1-\beta)$ true positive	type I error (α) false positive
Don't Reject H_0	type II error (β) false negative	correct no rejection $(1-\alpha)$ true negative

RM:
1. Recall in hypothesis testing, we control type I error as our specified "significance level", e.g. $\alpha = 0.05$. In above matrix, read "H_0 is False" as "Actual:Yes" event.
2. β is the probability of type II error and $1-\beta$ is called the power of test, i.e. the probability that we (successfully) reject the nul given null is false.

Logit Classifer and Classification Matrix in R

- In RscriptL9.R, a logit classifier is used on a subset of `titanic.csv` to classify survival of each passenger. `confusionMatrix()` in `caret` package is then used to produce classification matrix.

```
# use 'glm()' with specified parameter 'family = binomial' for logistic
    regression
fit_surv = glm(survived ~ sex + age + sibsp + parch + fare + embarked, family =
     binomial, data = titanic, control = list(maxit = 50))
# predict the survival probability using fitted logistic regression
predprob_surv = predict(fit_surv, type = 'response')
# define survived = 1 when predicted probability >= 0.5; 0 otherwise
pred_surv = ifelse(predprob_surv >= 0.5, 1, 0)
# using 'confusionMatrix()' in 'caret' package
cm = confusionMatrix(pred_surv, titanic$survived, positive = 'Survived')
```

```
              Reference
Prediction     Not Survived Survived
  Not Survived          474      112
  Survived               75      228
```

RM:
1. Watch out that Survived is specified as "positive" event.
2. Sensitivity $= 228/(112 + 228) = 0.67$; Precision $= 228/(228 + 75) = 0.75$; Accuracy $= (474 + 228)/(474 + 75 + 112 + 228) = 0.79$.

Summary

- Model selection is selecting and excluding available features to identify a model with best explanatory power for outcome variable. Both Chow test and some information criteria (AIC/BIC) can be used in stepwise selection.
- Data dimensionality reduction is identifying a smaller set of principal predictors that transform and summarize all variables. Principal components in PCA are ranked by proportion of variations explained in data.
- k-means algorithm can be used to find k clusters of data points based on distances of their features. Logisitic regression gives us a commonly used logit classifier to label binary outcomes and classification matrix is used to evaluate classifiers. Don't be confused in confusion matrix!

Our History of Learning