UCLA Anderson
SCHOOL of MANAGEMENT

MGMTMFE 431:

Data Analytics and Machine Learning

Topic 6:
Clustering and Unsupervised Learning:
Part I

Spring 2019

Professor Lars A. Lochstoer

UCLA Anderson
SCHOOL of MANAGEMENT

Agenda

a. Intro: Unsupervised learning and clustering overview

b. (Asymptotic) Principle Components Analysis

c. Other techniques
   i. K-means clustering (find clusters)
   ii. Hierarchical clustering (determine number of clusters)

a. Unsupervised learning and clustering

In ***unsupervised learning***, we don’t know the dependent variable (the outcome we are trying to predict).

- For instance, we may want to classify users into main ‘groups,’ without knowing what those groups would be, ex ante

Examples include:

1. Principal components analysis of asset returns
2. Topic modeling in text
3. Market segment analysis

In all of these cases, we are looking for ***common patterns***

- That is, common movement that describes the main features of the data
  - **Example**: in PCA we were interested in the eigenvectors corresponding to the largest eigenvalues: the main drivers of stock returns

UCLA Anderson
SCHOOL of MANAGEMENT

b. Review: Principal Component Analysis

Consider a panel of N stocks over T dates.

Denote the sample N by N covariance matrix Σ

- Recall the ***Spectral Decomposition:***
  - from lecture 4, Empirical Methods in Finance:

A real, symmetric $m \times m$ matrix $\mathbf{B}$ has a spectral decomposition given by

$$\mathbf{B} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'$$

where

- $\boldsymbol{\Lambda}$ is a diagonal matrix with eigenvalues $\boldsymbol{\lambda}$ on the diagonal that are all real and positive
- $\mathbf{P}$ is an $m \times m$ **orthogonal** matrix consisting of the m eigenvectors

UCLA Anderson SCHOOL of MANAGEMENT

b. Review: Principal Component Analysis

Recall that when we apply PCA to Σ, each eigenvector defines the portfolio weights in a portfolio with variance equal to the corresponding eigenvalue

- I.e., the factor that explains most of the variance in the panel (i.e., most of the covariance between stocks; factor 1) is given by

$$F_{1,t} = \sum_{n=1}^{N} P(n,1) R_{n,t}^{e}$$

where P(n,1) refers to the n'th row in column 1 of matrix P

- Also recall, all factors obtained from the PCA are uncorrelated

Principle Components Analysis is an unsupervised machine learning technique

b. Asymptotic Principal Component Analysis

A really cool and useful technique is called ***Asymptotic Principal Components Analysis***

- I've used this myself with good results in both practitioner and research settings
- Original work: Chamberlain and Rothschild (1983)
- Finance-related: Connor and Korajczyk (1986, 1993), Jones (2001) (see CCLE)

Works well when *N* is large and *T* is small

- That's exactly our problem in portfolio choice
- With large N, we get a covariance matrix with N*(N+1)/2 entries that typically entails so many elements that:

1. The estimates are very noisy (to the point of being useless when you try to invert a matrix)
2. Could even be more elements than we have observations (N*T), so the matrix will not be full rank and can't even be inverted

b. Asymptotic Principal Component Analysis

You can read the theory behind APCA in the papers I have posted on CCLE

- Here I will only give the way to implement
- Assumption: returns follow a *K*-factor model

1. Take a relevant sample of stock returns (perhaps last year of daily data)
2. Let Rt denote the T by N matrix of stock returns in this sample
3. Let Ω = Rt * Rt'
4. Get eigenvectors and eigenvalues of Ω
5. The eigenvectors corresponding to the *K* largest eigenvalues are the *T* returns to the *K* factors of the underlying factor model (up to a constant of proportionality).

As N goes to infinity and T > K, we can estimate the factor returns with arbitrary degree of accuracy (if the factor model is 'strict')

UCLA Anderson
SCHOOL of MANAGEMENT

b. Asymptotic Principal Component Analysis

Connor and Korajczyk (1993) provide a test for how many *pervasive* factors there are

- I.e., what is *K*?

The cool thing is that you can perform this analysis rolling through the data and get a sense of

- To what extent is the factor structure time-varying
- What are the current factors driving most of the variation in stock returns
- This can help you construct better forward-looking hedge portfolios
- Perhaps this is informative for which risks are priced?
    - Presumably, if a factor is ‘small’ (has low volatility) it is not associated with a high risk premium, and vice versa

UCLA Anderson
SCHOOL of MANAGEMENT

b. APCA: An example

The dataset French_Factors_2012_2016.dta on CCLE has monthly returns over 5 years for 138 of the factors on Ken French's webpage

We will load the data and estimate the two main principal components using ACPA, as well as stocks' loadings on the two extracted factors

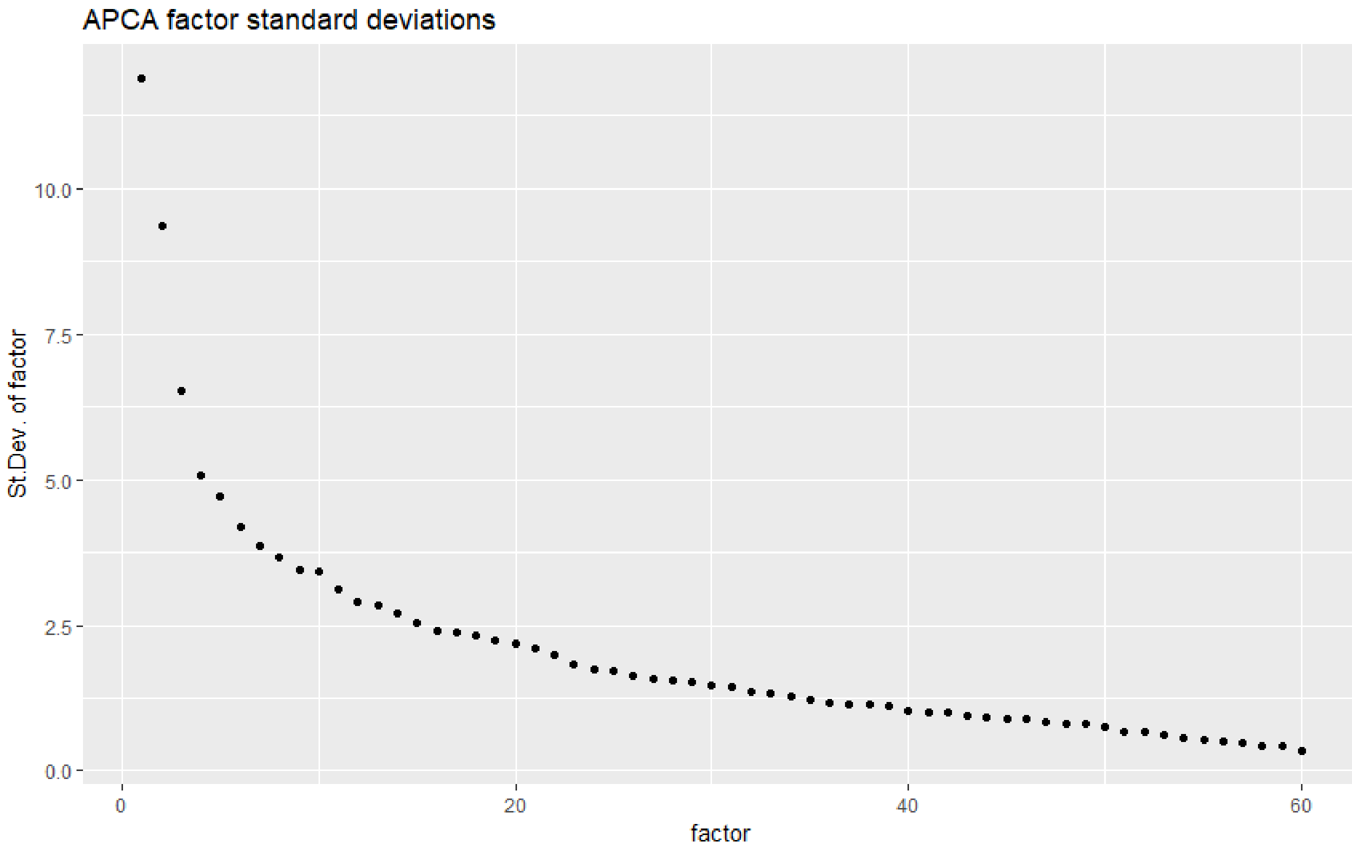
- Note: easy to extend to more factors
- There are 60 time series observations, 138 cross-sectional observations
  - Cannot estimate the covariance matrix of the 138 return series with 60 time observations of each series
  - But, going back more than 5 years may use data that is too old, not relevant…
  - This is where ACPA shines!

b. APCA: An example

Use the MTS (Multivariate time series) package. I recommend it!

```
➢ # Download data and set as data.table
➢ Factors_DT <- as.data.table(read.dta("French_Factor_Data_2012_2016.dta"))
➢ # run asymptotic pca
➢ French_Factors = apca(Factors_DT[, !"date"],2)
```

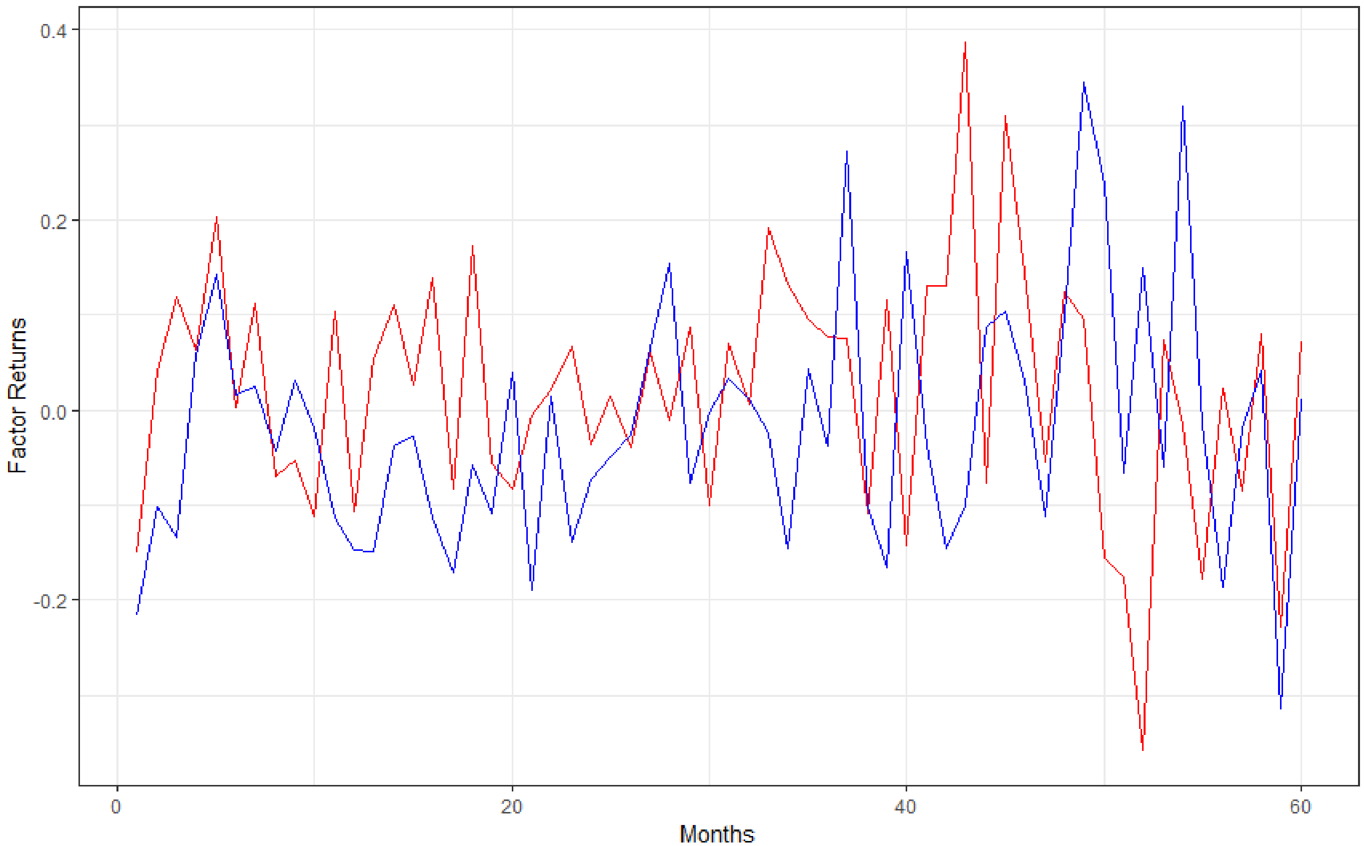
```
> qplot(1:60,French_Factors$sdev, xlab = "factor", ylab="St.Dev. of factor",main="APCA factor standard deviations")
```



b. APCA: An example

Extract the two factors (in APCA, these are the two first PCs, remember)

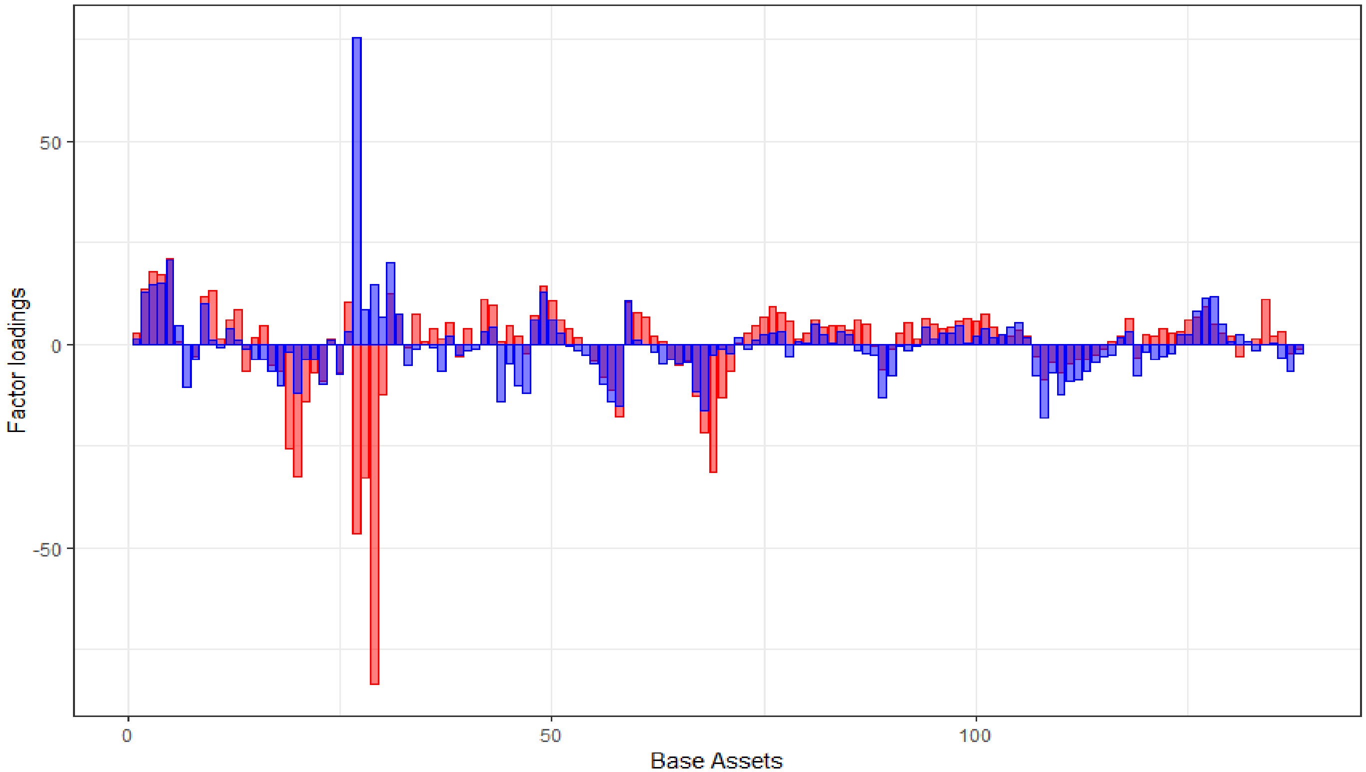
```
➢ qplot(1:60,French_Factors$sdev, xlab = "factor", ylab="St.Dev. of factor",main="APCA factor standard deviations")
➢ # return factors extracted from apca
➢ apca_factors <- data.table(date = 1:60, factor1 = French_Factors$factors[,1], factor2 = French_Factors$factors[,2])
➢ ggplot(data = apca_factors, aes(x = date, y = factor1)) + geom_line(color = "red") +
  geom_line(data = apca_factors, aes(x = date, y = factor2), color = "blue") + theme_bw() +
  xlab('Months') + ylab('Factor Returns')
```



b. APCA: An example

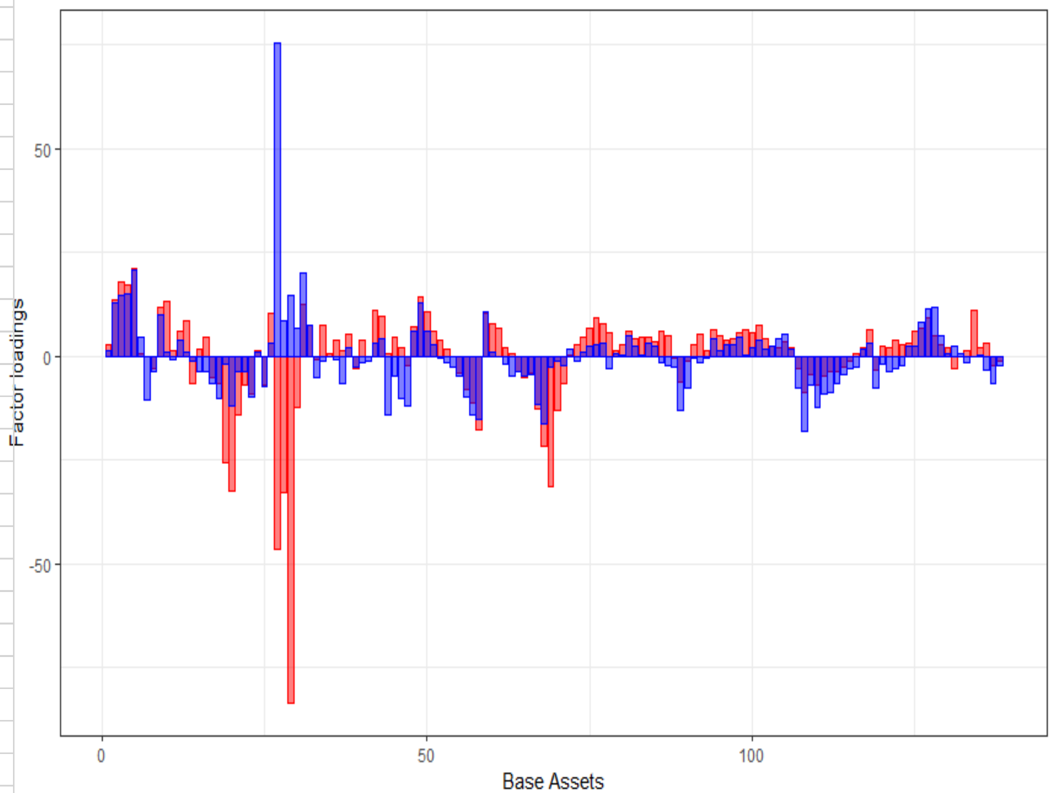
Extract the loadings required to replicate the factors:

```
➢ # plot factor loadings
➢ apca_loadings <- data.table(BaseAsset = 1:138, loadings1 = French_Factors$loadings[,1], +
       loadings2 = French_Factors$loadings[,2])
➢ ggplot(data = apca_loadings, aes(x = BaseAsset, y = loadings1)) +
  geom_bar(stat = "identity", alpha = 0.5, fill = "red", color = "red") +
  geom_bar(data = apca_loadings, aes(x = BaseAsset, y = loadings2), +
              stat = "identity", fill = "blue", alpha = 0.5, color = "blue") +
  theme_bw() + xlab('Base Assets') + ylab('Factor loadings')
```



b. APCA: Link the loadings to the portfolios for intuition

1	Agric	35	Comps	69	Mom1	104	BM6
2	Food	36	Chips	70	Mom2	105	BM7
3	Soda	37	LabEq	71	Mom3	106	BM8
4	Beer	38	Paper	72	Mom4	107	BM9
5	Smoke	39	Boxes	73	Mom5	108	BM10
6	Toys	40	Trans	74	Mom6	109	Size1
7	Fun	41	Whlsl	75	Mom7	110	Size2
8	Books	42	Rtail	76	Mom8	111	Size3
9	Hshld	43	Meals	77	Mom9	112	Size4
10	Clths	44	Banks	78	Mom10	113	Size5
11	Hlth	45	Insur	79	Invest1	114	Size6
12	MedEq	46	RlEst	80	Invest2	115	Size7
13	Drugs	47	Fin	81	Invest3	116	Size8
14	Chems	48	Other	82	Invest4	117	Size9
15	Rubbr	49	MktBeta1	83	Invest5	118	Size10
16	Txtls	50	MktBeta2	84	Invest6	119	DP1
17	BldMt	51	MktBeta3	85	Invest7	120	DP2
18	Cnstr	52	MktBeta4	86	Invest8	121	DP3
19	Steel	53	MktBeta5	87	Invest9	122	DP4
20	FabPr	54	MktBeta6	88	Invest10	123	DP5
21	Mach	55	MktBeta7	89	Prof1	124	DP6
22	ElcEq	56	MktBeta8	90	Prof2	125	DP7
23	Autos	57	MktBeta9	91	Prof3	126	DP8
24	Aero	58	MktBeta1(	92	Prof4	127	DP9
25	Ships	59	Variance1	93	Prof5	128	DP10
26	Guns	60	Variance2	94	Prof6	129	Issue1
27	Gold	61	Variance3	95	Prof7	130	Issue2
28	Mines	62	Variance4	96	Prof8	131	Issue3
29	Coal	63	Variance5	97	Prof9	132	Issue4
30	Oil	64	Variance6	98	Prof10	133	Issue5
31	Util	65	Variance7	99	BM1	134	Issue6
32	Telcm	66	Variance8	100	BM2	135	Issue7
33	PerSv	67	Variance9	101	BM3	136	Issue8
34	BusSv	68	Variance1(	102	BM4	137	Issue9
				103	BM5	138	Issue10



UCLAAnderson
SCHOOL of MANAGEMENT

b. APCA: Epilogue

Use factors to hedge movements in asset values not related to your strategy.

E.g., your signal for each stock and time is Xit (e.g., bm_it).

- Run Fama-MacBeth regression at each *t* including both your signal and the factor loadings of the APCA's to control for these 'large' factors in the cross-section of stock returns!

c. K-means clustering

- Principal components analysis is a linear clustering tool
  - Particularly useful for asset returns as portfolio returns are linear combinations of the underlying individual asset returns
- But, we may be interested in non-linear clustering techniques as well
- And, what about unstructured data like text? Can we think about factors for such data as well?
- We will start with perhaps the simplest technique: K-means clustering

c. K-means clustering

- Elegant method for partitioning data into *K* distinct, non-overlapping clusters.
  - Method needs you to specify number of clusters
  - Hierarchical clustering can help find this number (we don't have time to cover)

- Let $C_1, C_2, \ldots, C_K$ denote sets containing indices of the observations in each cluster. Let there be *n* obs. These sets satisfy the properties

1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1,2,\ldots,n\}$. In words, each observation belongs to at least one of the *K* clusters
2. $C_k \cap C_{k'} = \emptyset \;\; for\ all\ k \neq k'$. In words, the clusters are non-overlapping: no observation belongs to more than one cluster

UCLA Anderson
SCHOOL of MANAGEMENT

c. K-means clustering: main idea

- A *good* cluster is one where the within-cluster variation is as small as possible
  - Let $W(C_k)$ denote the within-cluster variation of cluster *k*
  - Then, the objective function is:

$$\underbrace{min}_{C_1,\dots,C_K} \sum_{k=1}^{K} W(C_k)$$

- A common choice for variation metric is the squared Euclidean distance, where $|C_k|$ denotes number of observations in *k'th* cluster

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2$$

c. K-means clustering: graphic intuition

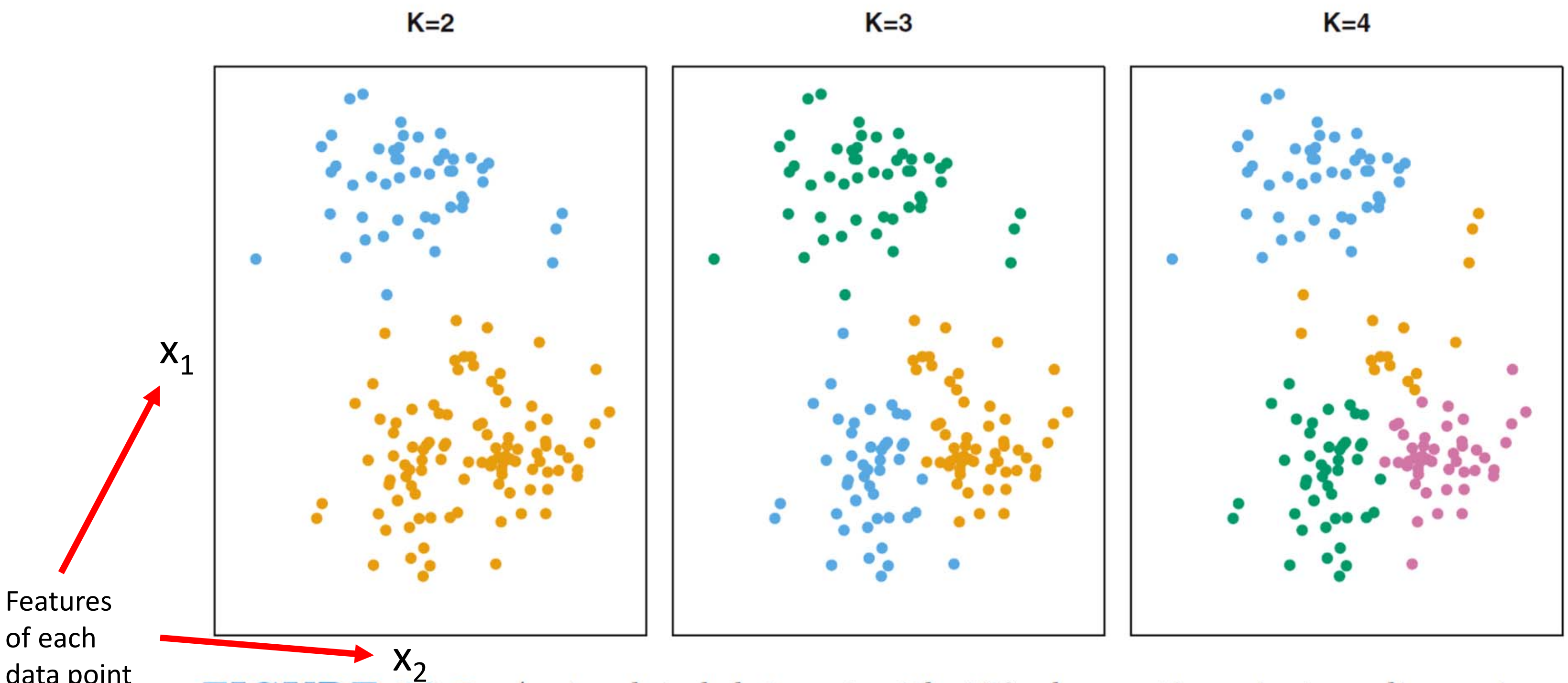


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

c. K-means clustering: usage

- K-Means is very popular in a variety of domains.
  - In biology it is often used to find structure in DNA-related data or subgroups of similar tissue samples to identify cancer cohorts.
  - In marketing, K-Means is often used to create market/customer/product segments.
  - In finance, a potential use-area is nonlinear volatility modeling, though more sophisticated methods are available (e.g., EGARCH)
    - But, can get sense of typical market *regimes/patterns* in, e.g., returns, valuations, etc.

UCLA Anderson
SCHOOL of MANAGEMENT

c. K-means clustering: Example

- Download S&P500 daily returns data (use ‘spider’, SPY)

```
> # Clean workspace
> rm(list = ls())
>
> # K-means clusters for understanding data
> require(quantmod)
> require(ggplot2)
> Sys.setenv(TZ="GMT")
> getSymbols('SPY',from='2000-01-01')
[1] "SPY"
>
> # plot density of daily SPY returns
> x=remove_missing(data.frame(d=index(Cl(SPY)),return=as.numeric(Delt(Cl(SPY)))))
Warning message:
Removed 1 rows containing missing values.
> ggplot(x,aes(return))+stat_density(colour="steelblue", size=2, fill=NA)+xlab(label='Daily returns')
>
> nasa=tail(cbind(Delt(Op(SPY),Hi(SPY)),Delt(Op(SPY),Lo(SPY)),Delt(Op(SPY),Cl(SPY))),-1)
> colnames(nasa)=c('High','Low','Close')
```

c. K-means clustering: Example

- Estimate assuming K = 5, using high, low and open as characteristics of a day

```
> # choose 5 clusters
> kmeanObject=kmeans(nasa,5,iter.max=10)
> kmeanObject$centers
         High          Low         Close
1 0.003554887 -0.014106806 -0.0099620223
2 0.003873982 -0.004265301  0.0002807315
3 0.012535822 -0.003610078  0.0091455341
4 0.005925872 -0.034786211 -0.0278744645
5 0.036411040 -0.008680281  0.0293011401
```

```
> # show autocorrelations as transition counts
> xtabs(~autocorrelation[,1]+(autocorrelation[,2]))
                   autocorrelation[, 2]
autocorrelation[, 1]    1    2    3    4    5
                   1  236  381  209   28   25
                   2  321 1574  484   18   23
                   3  265  412  207   32   37
                   4   29   36   31   24   14
                   5   28   16   22   33   13
```

Transition matrix (autocorrelation of state/cluster) in percentage

```
                   autocorrelation[, 2]
autocorrelation[, 1]    1    2    3    4    5
                   1 0.27 0.43 0.24 0.03 0.03
                   2 0.13 0.65 0.20 0.01 0.01
                   3 0.28 0.43 0.22 0.03 0.04
                   4 0.22 0.27 0.23 0.18 0.10
                   5 0.25 0.14 0.20 0.29 0.12
```

c. Optimal *K*

- Natural idea: look at average within-cluster variation
  - Code in Code Snippets