Contents

Dimensionality Reduction

- PRELIMINARIES
  - If we know that the vector x satisfies an eigen equation $Sx = \lambda_1 x$ for some λ_1
    - S a [known] p×p matrix, λ_1 unknown

    then we **know** that x must be one of the p eigenvectors of S (and λ_1 one of the p eigenvalues): — **only** need to decide which eigenvector it is
  - To see this compare roots of equations…..

- PRELIMINARIES (continued)

  Polynomial Equations:-
  - If we know x satisfies the equation $ax^2+bx+c=0$
    - a, b, and c known

    then we **know** x is one of the two roots of the quadratic:— **only** need to decide which root is needed

- PRELIMINARIES (continued)
  - If we know x satisfies the polynomial equation $ax^p+bx^{p-1}+\ldots = 0$ then we know x is one of the p roots of the polynomial
  - Standard computer packages produce roots of polynomials
    - `polyroot(.)` in **R**
    - Need to decide which root is required
  - Standard computer packages provide eigenanalyses of matrices
    - `eigen(.)` in **R**
    - Need to decide which eigenvalue is required

A Procedure for Maximization

- 1: Introduce some constraint
- 2: Introduce a Lagrange multiplier & define a new objective function
- 3: Differentiate w.r.t. x and set =0
- 4: Recognise this is an *eigenequation* with the Lagrange multiplier as *eigenvalue*
- 5: Deduce that there are **ONLY** a limited number of possible values for this eigenvalue (all of which can be calculated numerically)
- 6: Use some extra step to determine which eigenvalue gives the maximum (typically use the constraint somewhere)

- **Setup:-**
  - n observations x_{ij} on p variables $X_1,X_2\ldots,X_p$
    - i=1,…,n and j=1,…,p
    - $X_1,X_2\ldots,X_p$ are **n-vectors** of n observations
- Objective:-

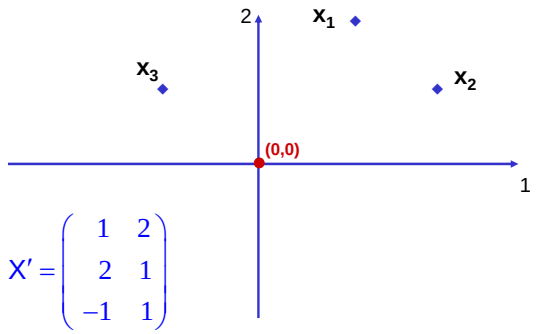
  Create new variables $Y_1, Y_2,\ldots,Y_p$
  - which are *combinations* of original variables $X_1,X_2\ldots,X_p$
  - such that the first *few* contain 'most' of the information in the data

- Then we can use the first ***few*** new variables $Y_1, Y_2, \ldots$ instead of all p of the original X_i's without loosing too much information
  - If p is large and '**few**' is small & not too much information is lost then this is worthwhile

Aside: projections onto vectors

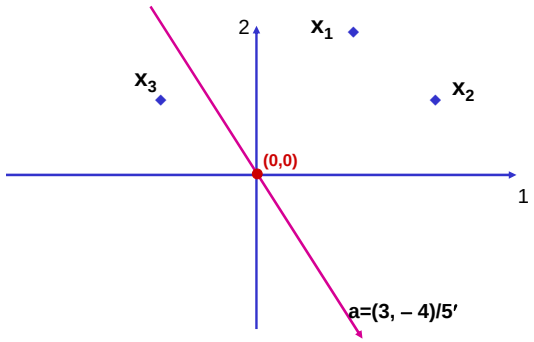
- If **X′** is an n×p data matrix and **a** a p×1 vector then **Y′= X′a** is the projection of **X′** onto **a**
  - Values of **Y′** give the coordinates of each observation along the vector **a**
- Example: $x_1=(1, 2)'$, $x_2=(2, 1)'$, $x_3=(-1,1)'$ $a=(3, -4)/5$,
  - So $X' = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -1 & 1 \end{pmatrix}$ $Y' = \begin{pmatrix} -1 \\ \frac{2}{5} \\ -\frac{7}{5} \end{pmatrix}$

Aside: projections onto vectors

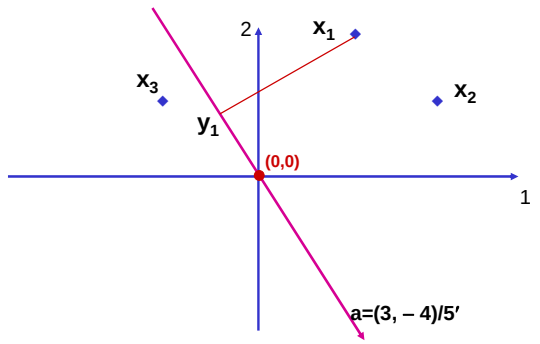


$$X' = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -1 & 1 \end{pmatrix}$$

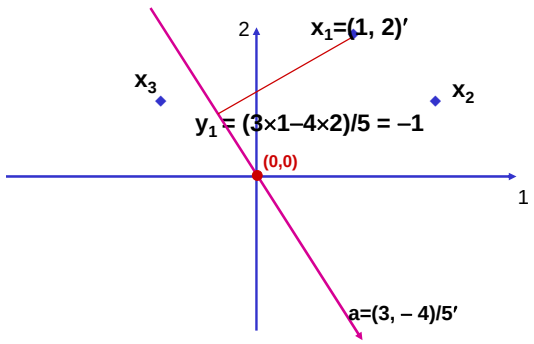
Aside: projections onto vectors



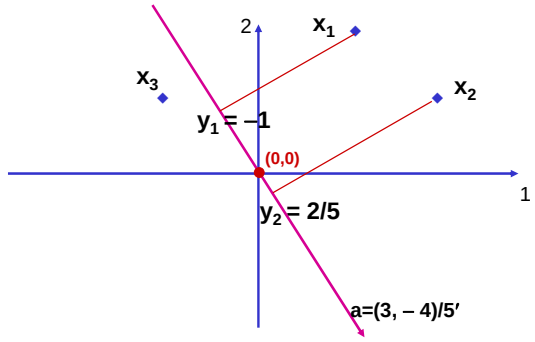
Aside: projections onto vectors



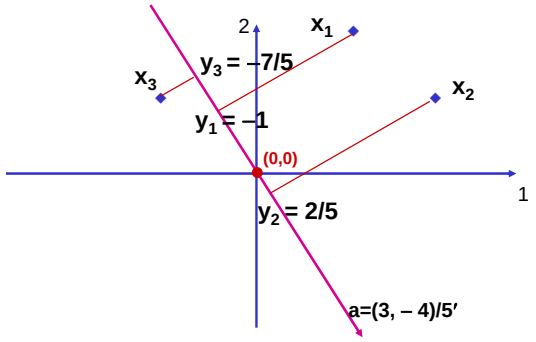
Aside: projections onto vectors



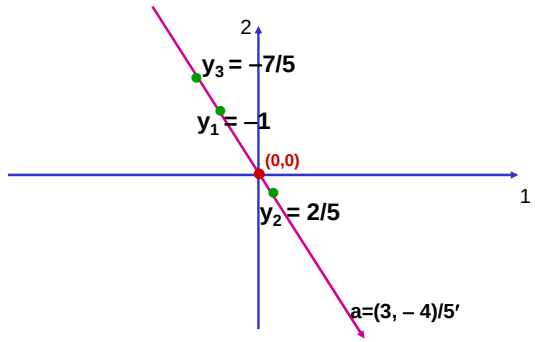
Aside: projections onto vectors



Aside: projections onto vectors



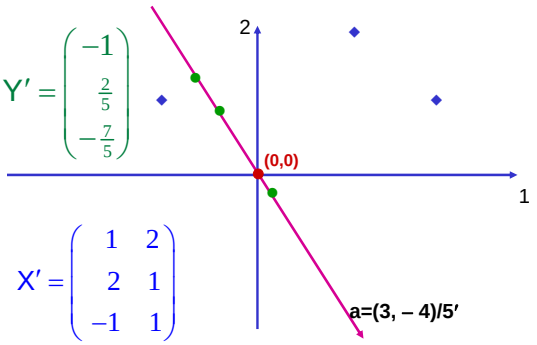
Aside: projections onto vectors



Aside: projections onto vectors

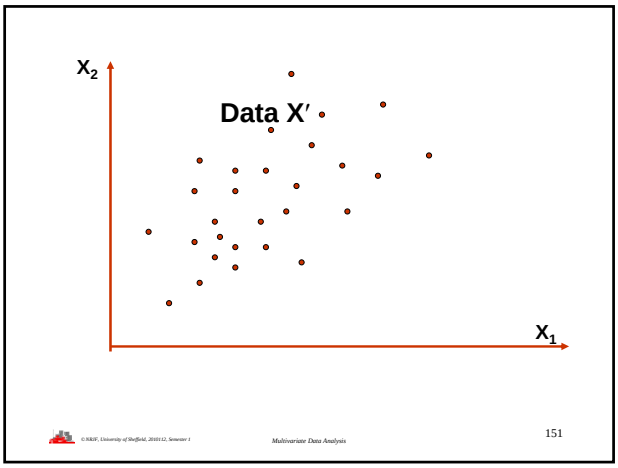
$$Y' = \begin{pmatrix} -1 \\ \frac{2}{5} \\ -\frac{7}{5} \end{pmatrix}$$

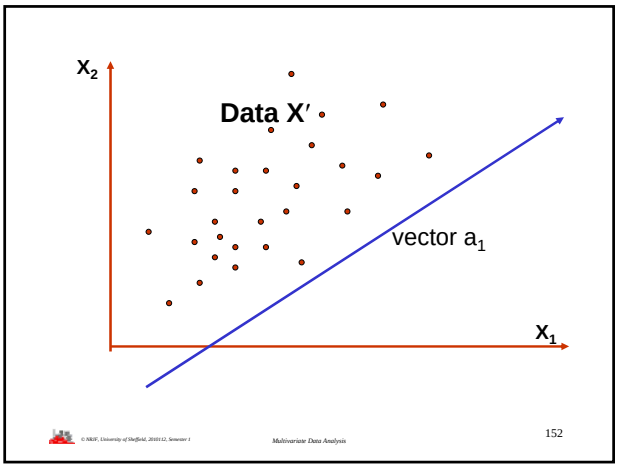
$$X' = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -1 & 1 \end{pmatrix}$$

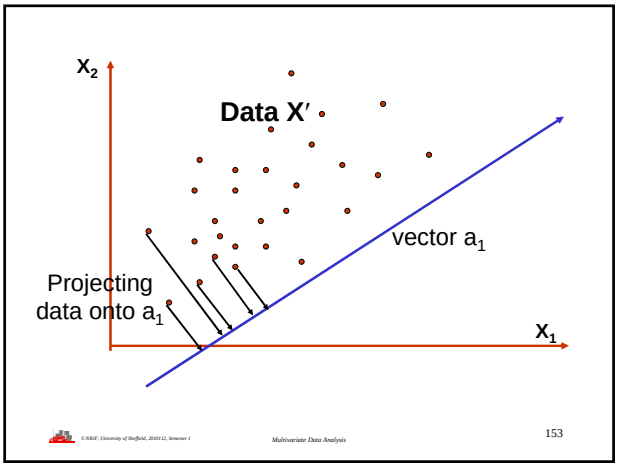


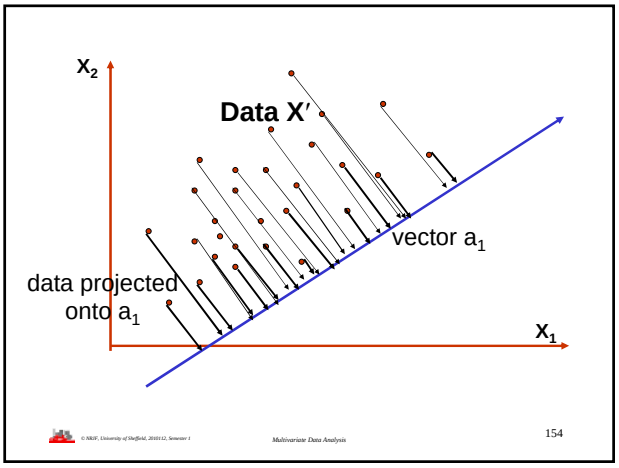
- Principal Component Analysis (**PCA**)
  - Note spelling …….**al** (not …….**le**)
  - ♦ **Information ≡ variance**
    - If variance is small then all observations are nearly the same so little information in the data
    - Large variance ⇒ more information
  - ♦ Combination = ***linear*** *combination*
- Other dimensionality reduction techniques take different starting points
  - Projection pursuit, LDA, scaling methods,……..

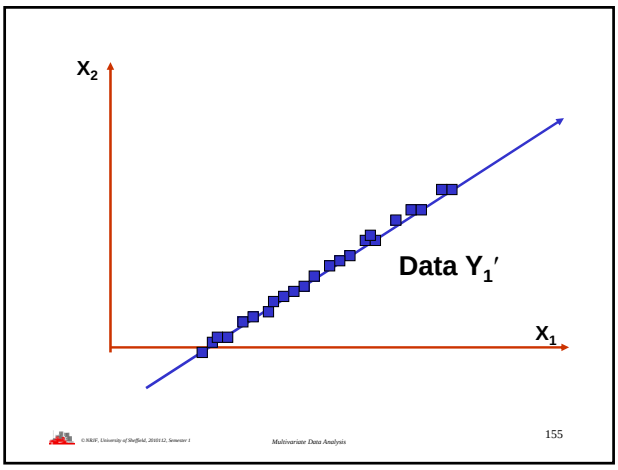
- ♦ Few = 2,3,4,….
- ♦ Combination = *linear* combination of X_1, X_2….
  - $Y_1 = a_{11}X_1 + a_{12}X_2 + \ldots + a_{1p}X_p$
  - $Y_2 = a_{21}X_1 + a_{22}X_2 + \ldots + a_{2p}X_p$
  - ……
  - $Y_p = a_{p1}X_1 + a_{p2}X_2 + \ldots + a_{pp}X_p$
- ♦ So $Y_1 = a_1' X$ where $a_1' = (a_{11}, a_{12}, \ldots, a_{1p})$
  or $Y_1' = X' a_1$
  - X' is the n × p data matrix
- ♦ & $Y_2 = a_2' X$ where $a_2' = (a_{21}, a_{22}, \ldots, a_{2p})$
  or $Y_2' = X' a_2$……
- ♦ & $Y = A'X$ where $A = (a_1, a_2, \ldots, a_p)$ (or $Y' = X'A$)

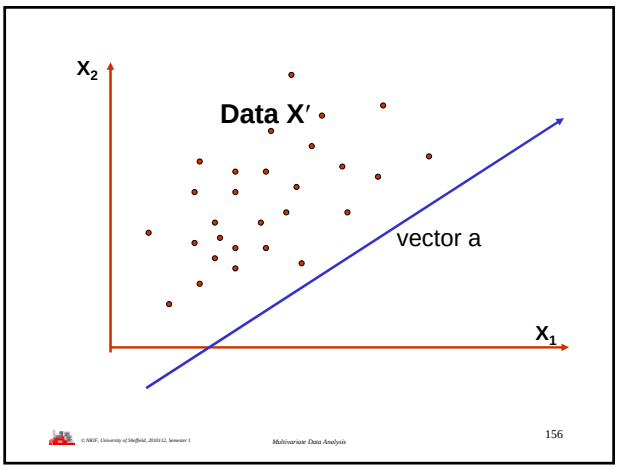

X2
Data X′
X1
151


X2
Data X′
vector a1
X1
152


X2
Data X′
vector a1
Projecting
data onto a1
X1
153


X2
Data X′
vector a1
data projected
onto a1
X1
154


X2
Data Y1′
X1
155


X2
Data X′
vector a
X1
156

- $Y_1=a_{11}X_1+a_{12}X_2+\ldots+a_{1p}X_p= a_1' X$
  - X_i is a n-vector of observations on p^{th} variable
  - a_1' is $1 \times p$, X is $p \times n$ so Y_1 is $1 \times p \times p \times n = 1 \times n$
  - i.e. Y_1 is a set of n [univariate] observations
- So, want to choose a_1 to maximize the variance of the n observations Y_1
  - i.e. choose the p elements of a_1 to maximize var(Y_1')
  - NB var(Y_1') is a **scalar**
- Problem insoluble as it stands……

- If all elements of a_1 are multiplied by a konstant **k** then var(Y_1') multiplied by $\mathbf{k^2}$
  - i.e. problem is unbounded without constraint on the magnitude of the elements a_{1j}
- Convenient to restrict attention to vectors a_1 s.t. $a_1'a_1=1$
- i.e s.t. $\sum_{j=1}^{p} a_{1j}^2 = 1$
- i.e. we need to impose the ***scale constraint*** $\mathbf{a_1'a_1=1}$ and maximize var(Y_1') subject to this

- ASIDES: level of understanding needed:—
  - Awareness that the underlying mathematics can ***justify*** all this
  - The mathematics explains ***why*** the interpretations work
    - i.e why it is of practical use
  - The mathematics can be applied in other situations

- ASIDES:
  - Var(Y_1') = var($X'a_1$) = a_1'var(X')a_1 = $a_1'Sa_1$
    - See Task Sheet 1
  - To maximize $a_1'Sa_1$ *subject to* $a_1'a_1=1$
    - Introduce a Lagrange Multiplier λ and define a new objective function (which is an ordinary ***scalar***)
      $\mathbf{\Omega_1 = a_1'Sa_1 - \lambda_1(a_1'a_1-1)}$
      and maximize wrt both a_1 and λ_1 (See Appendix 0)
  - Derivative of $a_1'Sa_1$ wrt a_1 is $2Sa_1$ (a p-vector)
    - See Appendix 0

- ASIDES (continued)
  - If we know that the vector x satisfies an eigen equation $Sx = \lambda_1 x$ for some λ_1
    - S a [known] p×p matrix, λ_1 unknown

    then we **know** that x must be one of the p eigenvectors of S (and λ_1 one of the p eigenvalues): — **only** need to decide which eigenvector it is
  - To see this compare roots of equations…..

- ASIDES (continued)

  Polynomial Equations:-
  - If we know x satisfies the equation
    $ax^2+bx+c=0$
    - a, b, and c known

    then we **know** x is one of the two roots of the quadratic:— **only** need to decide which root is needed

- ASIDES (continued)
  - If we know x satisfies the polynomial equation $ax^p+bx^{p-1}+\ldots. = 0$ then we know x is one of the p roots of the polynomial
  - Standard computer packages produce roots of polynomials
    - Need to decide which root is required
  - Standard computer packages provide eigenanalyses of matrices
    - Need to decide which eigenvalue is required

- To maximize $\Omega_1 = a_1'Sa_1 - \lambda_1(a_1'a_1 - 1)$
  - (with respect to a_1 & λ_1)
- $2Sa_1 - 2\lambda_1 a_1 = 0$
  - Differentiating wrt a_1 and set = 0
- $a_1'a_1 - 1 = 0$
  - Differentiating wrt λ_1 and set = 0
- $Sa_1 = \lambda_1 a_1$
- $\Rightarrow a_1$ must be an eigenvector of S
  - (so problem is nearly solved)

- $Sa_1 = \lambda_1 a_1 \Rightarrow a_1'Sa_1 = \lambda_1 a_1'a_1 = \lambda_1$
  - (since $a_1'a_1 = 1$)
- But $var(Y_1') = var(X'a_1) = a_1'Sa_1 = \lambda_1$
- i.e. whichever of the p eigenvectors of S we choose for a_1 then $var(Y_1')$ = corresponding eigenvalue
- So, choose λ_1 to be largest eigenvalue & a_1 to be the corresponding eigenvector

- Next linear combination: $Y_2' = X'a_2$
  - Want to choose a_2 to maximize $var(Y_2') = var(X'a_2) = a_2'Sa_2$
  - Need scale constraint on a_2 as before: $a_2'a_2 = 1$
  - Also need to ensure a_2 is different from a_1 and convenient to have a_2 orthogonal to a_1:
    $a_2'a_1 = 0$
  - Define $\Omega_2 = a_2'Sa_2 - \lambda_2(a_2'a_2 - 1) - \mu a_2'a_1$

- $2Sa_2 - 2\lambda_2 a_2 - \mu a_1 = 0$
  - Differentiating wrt a_2 and set = 0
- $a_2'a_2 - 1 = 0$
  - Differentiating wrt λ_2 and set = 0
- $a_2'a_1 = 0$
  - Differentiating wrt μ and set = 0
- Premultiplying 1st eq by a_1' and a_2' in turn shews $\mu = 0$ giving
  $Sa_2 = \lambda_2 a_2$ — problem nearly solved

- premultiplying by a_2' gives $a_2'Sa_2 = \lambda_2 a_2'a_2 = \lambda_2$
- i.e. $var(X'a_2) = a_2'Sa_2 = \lambda_2$
- i.e. whichever of the p eigenvectors of S we choose for a_2 then $var(Y_2')$ = corresponding eigenvalue
- So, choose λ_2 to be largest ***available*** eigenvalue and a_2 the corresponding eigenvector — can't be largest since that gives a_1 so take λ_2 as ***second*** largest eigenvalue

A Procedure for Maximization

- 1: Introduce some constraint
- 2: Introduce a Lagrange multiplier & define a new objective function
- 3: Differentiate w.r.t. x and set =0
- 4: Recognise this is an *eigenequation* with the Lagrange multiplier as *eigenvalue*
- 5: Deduce that there are **ONLY** a limited number of possible values for this eigenvalue (all of which can be calculated numerically)
- 6: Use some extra step to determine **which eigenvalue** gives the maximum (typically use the constraint somewhere)

- Definitions:
  - The first Principal Component of the data X′ is the vector a_1 such that the projection of the data X′ onto a_1, i.e. X′a_1, has maximal variance, subject to the normalizing constraint $a_1'a_1=1$
  - subsequent Principal Components defined recursively as maximising variance of projection of X′ subject to orthogonality with preceding ones

- **Theorem:**
  - The p principal components of data X′ are the p eigenvectors $a_1,a_2,....,a_p$ corresponding to the p ordered eigenvalues $\lambda_1\geq\lambda_2\geq...\geq\lambda_p$ of the variance of X′, S.
  - The variances of the data projected onto the principal components are equal to the eigenvalue corresponding to that eigenvector

- **Notes**
  - Really require a formal proof by induction
  - Could define PCs as eigenvectors and then deduce as a theorem that they maximize variances
    - (some authors do this)
  - Here they are defined by the required property and the theorem gives a method for obtaining them

- **Notes(Ct^{d}.)**:-

  Theoretical difficulty if
  - $\lambda_i=\lambda_{i+1}$
    - since then a_i and a_{i+1} not determined
    - can chosen in arbitrarily many ways
      - anywhere orthogonal to a_{i-1} & each other
  - $\lambda_p=0$
    - since a_p not uniquely defined

- Not a practical problem with real data
  - Unless linear dependency between variables
    - e.g. X_3 = weight @ start, X_4 = weight @ end & X_5 = weight gain = $X_4 - X_3$
      - don't need X_5 in analysis as well as X_3 & X_4
  - Then $\lambda_{k+1}=\lambda_{k+2}=\ldots=\lambda_p=0$ if there are p – k dependencies
    - With real data we may have $\lambda_p\approx 0$ but $\neq 0$ because of rounding errors

- To achieve dimensionality reduction we actually want $\lambda_{k+1} = \lambda_{k+2} = \ldots = \lambda_p = 0$ for $k \approx 2 / 3 / 4 \ldots$
  - & so '*discard*' k dimensions without loss of information
  - If $\lambda_j = 0$ then the j^{th} PC has zero variance $\Rightarrow$ **no information on j^{th} pc**
- If $\lambda_j \approx 0$ for $i > k$ then last $p - k$ PCs have '*small*' variance & so '*little*' information
  - '*small*' & '*little*' **comparative**

Information Conservation in PCs

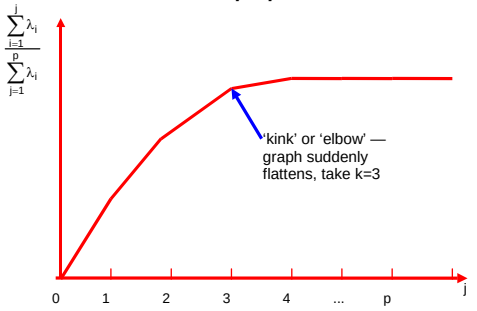
- 'total' variation/information in original data is $s_{11}+s_{22}+\ldots+s_{pp} = \text{tr}(S)$
  - sum of variances of original components
- 'total' variation/information of PCs is sum of variance of PCs = $\Sigma\lambda_i$
- but $\Sigma\lambda_i = \text{tr}(S)$
  - property of eigenvalues of S (see Appendix 0)
- 'total' variation / information is conserved

Measuring Information in PCs

- $\lambda_k / \Sigma\lambda_i$ = prop$^{n.}$ of total variance on k^{th} PC
  - amount of information 'contained' in k^{th} PC
- $(\lambda_1 + \lambda_2 + \ldots + \lambda_k) / \Sigma\lambda_i$ = proportion information in first k PCs
- Want this '*large*' **but** also want k '*small*'
- Trade-off dimensionality & information
- Assess **informally & graphically**
  - does another PC give worthwhile extra info?

Scree Graph:

partial relative cumulative proportion vs dimension



Transformation to PCs

- Transformation $X' \rightarrow Y'$ is given by $Y' = X'A$
  - **A** is matrix of [normalized] eigenvectors of S
    - (or eigenvectors of R, the correlation matrix)
  - $A = (a_1, a_2, \ldots, a_p)$ — a_i eigenvectors of S ordered by magnitude of eigenvalues λ_i, $[\lambda_1 > \lambda_2 > \ldots > \lambda_p]$
  - Note that $a_i' a_j = 0$ if $i \neq j$
    - (orthogonality of eigenvectors of real symmetric matrices)
  - $a_i' a_i = 1$
    - (normalizing constraint imposed without loss of generality)
- **SO**, **A** is orthogonal and $\mathbf{A'A = I_p}$
- i.e. transformation $X' \rightarrow Y'$ is a rotation/reflection
- $\Rightarrow$ no statistical information is altered/lost/gained

SO:-

- We can use the transformed data Y′ instead of X′ & still keep the same statistical information — advantage is that the first few components of Y′ will have 'most' of the interesting information
  - e.g. in scatter plots of first components,
    - i.e. the *scores* on the first few PCs
    - i.e. "plots of the first few PCs"
  - e.g. again: use just the first few PCs
    - (≡ first few components of Y′)

  in later statistical analysis (i.e. tests etc)

- **NB** Computer Package Implementation:
  - Most packages (Minitab / S-PLUS etc) produce a different form of 'scree-graph' in 'ready-made' menu for PCA
    - e.g. bar charts of eigenvalues in S-PLUS
    - e.g. eigenvalue *vs* dimension in Minitab
  - Easier to see 'kinks' in partial cumulants
    - change in 2nd derivative vs change in magnitude
  - Easy to produce **real** scree graphs
    - from a written function in S-PLUS
    - from command line in Minitab (see later)

- Trivial Example (p only = 3)

$$S = \begin{pmatrix} 451.39 & 271.17 & 168.70 \\ * & 171.73 & 103.29 \\ * & * & 66.65 \end{pmatrix}$$

  - tr(S)=689.77
  - $\lambda_1 = 680.4 \quad \lambda_2 = 6.5 \quad \lambda_3 = 2.86$
  - Check: $\Sigma\lambda_j = 689.76$ ✓
    - ***Note: not much info on PCs 2 & 3***

- Eigenvectors:-

	Principal Components		
	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$
1==length	.8126	–.5454	–.2054
2==width	.4955	.8321	–.2491
3=height	.3068	.1006	.9465
variance=λ_j	680.4	6.5	2.86

✓ Check that a_i satisfy eigenequations ✓

- 1st component: 99% total variance
  - typical of data on ***sizes***
  - reflects variation in overall turtles sizes
- value for any particular turtle
  - (or the *score* on the 1st p.c.)

is Y_1=.81 × length + .50 × width +. 31 × height

— weighted average of all dimensions

  - Y_1 is big if length, width & height are all big
  - Y_1 is small if length, width & height are all small
  - i.e. Y_1 reflects general ***size***.

- 2nd p.c. is

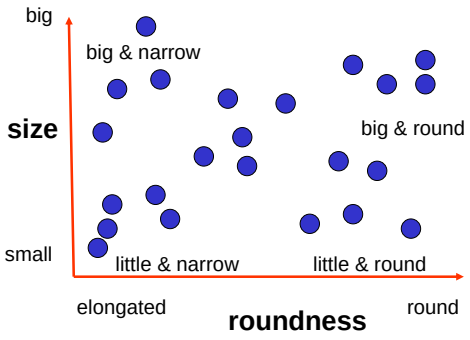
Y_2= –.55 × length + .83 × width + .10 × height

large small

- Y_2 is length *vs* width
  - Y_2 is small for long thin shells
  - & large for short wide ones
- Y_2 measures how ***round*** shells are

- Y_1 measures **size** — Y_2 **roundness**
- & independent properties
  - (with these measures)

- SO:–
  - Plots of Y_1 *vs* Y_2 will have
    - big round turtles in top right hand corner
    - little round turtles in bottom RH corner
    - big elongated in top LH
    - little elongated in bottom LH.



- Interpretation of coefficients
  - *See also §2.1.5*
  - Another example: ***white leghorn chickens***
  - measures of skulls and of length of legs & wings

- Values rounded to 2 digits

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6
x_1 skull l.	0.35	0.53	0.76	–0.04	0.02	0.00
x_2 skull b.	0.33	0.70	–0.64	0.00	0.00	0.03
x_3 humerus	0.44	–0.19	–0.05	0.53	0.18	0.67
x_4 ulna	0.44	–0.25	0.02	0.48	–0.15	–0.71
x_5 femur	0.44	–0.28	–0.06	–0.50	0.65	–0.13
x_6 tibia	0.44	–0.22	–0.05	–0.48	–0.69	0.17

All positive and approx equal

- Values rounded to 1 digits

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6	
x_1 skull l.	0.4	0.6	0.7	0	0	0	skull
x_2 skull b.	0.4	0.6	–0.7	0	0	0	
x_3 humerus	0.4	–0.2	0	0.5	0	0.7	wing
x_4 ulna	0.4	–0.2	0	0.5	0	–0.7	
x_5 femur	0.4	–0.2	0	–0.5	0.6	0	leg
x_6 tibia	0.4	–0.2	0	–0.5	–0.6	0	

- Values rounded to **0 digits**

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6	
x_1 skull l.	+	+	+				skull
x_2 skull b.	+	+	–				
x_3 humerus	+	–		+		+	wing
x_4 ulna	+	–		+		–	
x_5 femur	+	–		–	+		leg
x_6 tibia	+	–		–	–		

- 1st PC — sum of all components
  - Measures **size**
- 2nd PC — (skull) *vs* (wing & leg)
  - Measures **overall body shape**
    - (big heads & little bodies) *vs* (little heads & big bodies)
- 3rd PC — skull length *vs* width
  - Measures **shape of skull**
- 4th PC — wings *vs* legs
  - Measures **body shape**
- 5th & 6th measure wing & leg shape

- 1st PC — sum of all components
  - » Measures **size**
- 2nd PC — (skull) *vs* (wing & leg)
  - » Measures **overall body shape**
  - » (big heads & little bodies) *vs* (little heads & big bodies)
- 3rd PC — skull length *vs* width
  - » Measures **shape of skull**
- 4th PC — wings *vs* legs
  - » Measures **body shape**
- 5th & 6th measure wing & leg shape

- SO:–
  what do plots of PC1 *vs* PC2 look like?
  P2*vs* PC3 etc…..?

- Comments (*see also* §2.1.5)
  - Interpretation of *loadings* and amounts of information on each PC strictly relies on all data in ***same units***
  - same units
    - ⇒ similar scales of measurement
    - ⇒ all measures in e.g. mm or all in kg

AND

similar standard deviations

- Even if data all in same 'units' (e.g. mm) but some variables have a very different s.d. then difficult to interpret a linear combination of variables
- Different s.d.

  ⇒ different scale of measurement

- But real data may be on different scales

SO

- Standardize all variables
  - i.e. subtract mean & divide by s.d.
  - then data are dimensionless
- Covariance matrix of standardized data
  ≡ correlation matrix of raw data

- **SO**, if data on different scales
  - i.e. different units **OR** have differing s.d.s

  do eigenanalysis on correlation
  not on covariance matrix

- Exploratory analysis of data will shew if correlation or covariance matrix is best
  - e.g. descriptive statistics of individual variables
- If largest variance > ~4 × smallest then use correlation, otherwise use covariance
  - max(s.d.) > 2 × min(s.d.)
- If only one variable has a very large variance then 1st PC is dominated by this (obviously)

- Analysis of covariance matrix is more natural to interpret
- Use of correlation matrix is pragmatic
  - Some times useful to look at both analyses
- Same degree of dimensionality reduction, similar interpretation of loadings or coeffs in identifying PCs as 'underlying factors'
  - may not have formal statistical hypothesis tests
    - » but these are not very useful anyway

PCA of leghorn on correlation matrix

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6
x_1 skull l.	-0.35	-0.40	-0.85	-0.05	0.01	0.03
x_2 skull b.	-0.29	-0.81	0.50	-0.02	0.01	0.04
x_3 humerus	-0.44	0.26	0.11	-0.50	0.60	0.33
x_4 ulna	-0.45	0.20	0.10	-0.47	-0.60	-0.41
x_5 femur	-0.45	0.16	0.10	0.50	0.37	-0.58
x_6 tibia	-0.45	0.21	0.07	0.47	-0.38	0.62
eigenvalue	4.46	0.78	0.46	0.17	0.08	0.05

- Correlation matrix: values rounded to **0 digits**

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6	
x_1 skull l.	–	–	–				skull
x_2 skull b.	–	–	+				
x_3 humerus	–	+		–	–	+	wing
x_4 ulna	–	+		–	+	–	
x_5 femur	–	+		+	+	–	leg
x_6 tibia	–	+		+	–	+	

- Covariance matrix: values rounded to **0 digits**

Original variable	Principal Components a_1	a_2	a_3	a_4	a_5	a_6	
x_1 skull l.	+	+	+				skull
x_2 skull b.	+	+	–				
x_3 humerus	+	–		+		+	wing
x_4 ulna	+	–		+		–	
x_5 femur	+	–		–	+		leg
x_6 tibia	+	–		–	–		

- Most PCs same in structure
  - Except low order ones
- Some have changed sign
  - signs are arbitrary
  - can multiply all coefficients by –1
  - relative +/ – signs are important for contrasts
- Sum of all eigenvalues is 6 (= p)
  - diagonal terms of correlation matrix are all = 1

- **S-PLUS:-**
  - **Covariance** matrix is default
- **Minitab:-**
  - Stat>Multivariate>Principal Components
  - Analysis of **correlation** matrix is default
  - Storage menu keeps *coefficients* and *scores*
  - Coefficients ≡ elements of eigenvectors
  - Scores ≡ coordinates of data points on PCs

- Why look at plots on principal components?
  - (i.e. ***score plots***)
- Reveals structure which inflates variance
  - Subgroups
  - Outliers
- c.f. univariate histograms reveals bimodality
- Scree plots of eigenvalues help decide on dimensionality reduction

- **Notes** (*see also §2.1.10*)
  - Strictly PCA is for continuous data
    - (calculated a covariance/correlation matrix)
  - Experience is PCA 'works' if some variables are binary and most are continuous
    - or lots of binary and no continuous
  - Categorical variables with k >2 levels should be coded into dummy binary variables
  - Should plot data on PCs with equal scalings on axes
    - need MASS library in **R** (or S-plus) [`eqscplot(.)`]

- **Miscellaneous comments** ***(see also §2.1.11)***
  - *supplementary data*
    - Superimpose further data on same plot on PCAs
  - Interpretation of loadings is only viable for small numbers of variables
    - may need to examine coefficients graphically
  - Outliers:-
    - may be revealed on first few PCs or last few or on ***cut-off*** PCs
  - Generalizations to other techniques
    - Projection pursuit / non-linear PCA / &c. &c.

- **Misc coms** ***(contd.)***
  - ***Data Visualisation Methods***
    - Generative Topographic Mapping (GTM), Hierarchical GTM (HGTM), Self-Organising Maps (SOM), Sammon Mapping, GTM with Feature Selection (GTM-FS), probabilistic PCA, …, …,
  - All of these are good for '**Data Visualisation**'
    - i.e. for looking at multivariate data in 2 or 3 dimensions and detecting groups structure ('features') or relationships between individual points but they are ***not statistical***
    - they do not ***analyse variability***
    - they do not partition variance into components related to the original variables

- Typically first PC reflects overall size or amount or level
  - i.e. objects often vary most in overall size
  - or there may be large variation in base level of response (e.g. in ratings data from questionnaires).
  - Suggestion: consider ignoring the first p.c. if it only reveals the obvious (i.e. look at proportions of variance explained excluding the first PC)

or

use correlation matrix

or

both

¿ Why is it ***typical*** that first PC reflects ***overall*** size or amount or level**?**

- **Perron-Frobenius Theorem:**
  - If A is a symmetric matrix with positive elements, then all of the coefficients of the first eigenvector of A have the same sign
  - So, if all pairwise correlations between variables are positive then first PC is a weighted average of all variables
  - If all measures are size dimensions then likely to be mutually positively correlated
    - If a small number of correlations are negative then often case that 2nd or 3rd (or…..) PC is size measure

- **R** implementation
  - Two functions:–
    - `princomp(.)`
    - `prcomp()`
  - `princomp(.)` is identical to S-Plus function
    - uses `eigen(.)`
  - `prcomp(.)` works differently but like other similar analyses
    - uses `svd(.)` & is numerically more stable
  - `predict(.,.)` used for both to rotate supplementary data to same coordinates

R implementation

- `princomp(.)`
  - loadings are in `$loadings`
  - scores in `$scores`
  - analyse with correlation with `cor=TRUE`
- example: –
  - `body.pc<-princomp(bodysize,cor=T)`
  - `body.pc$loadings`
    - use for interpretation of PCs
  - `body.pc$scores`
    - use for plotting original data rotated onto PCs

R implementation

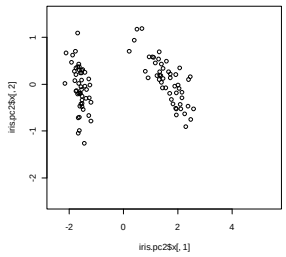
- `prcomp(.)`
  - loadings are in `$rotation`
  - scores in `$x`
  - analyse with correlation with `scale=TRUE`
- example: –
  - `body.pc2<-princomp(bodysize,scale=T)`
  - `body.pc2$rotation`
    - use for interpretation of PCs
  - `body.pc$x`
    - use for plotting original data rotated onto PCs

R implementation

- Use of `predict(.,.)`:–
  - `predict(PCAobject,newdata)`
  - `prcomp()`
- example: –
  - `iris.pc2<-prcomp(irisnf[1:100,-5])`
    - calculate PCs just on first hundred observations (first two groups)
  - `irisnew<-predict(iris.pc2,irisnf[101:150,-5])`
    - rotate observations 101 to 150 to same coordinate system
  - `plot(iris.pc2$x[,1],iris.pc2$x[,2], xlim=c(-2.5,5.5),ylim=c(-2.5,1.5))`
  - `points(irisnew[,1],irisnew[,2],col=2)`
  - plot first hundred points on their PCs followed by last fifty as supplementary points

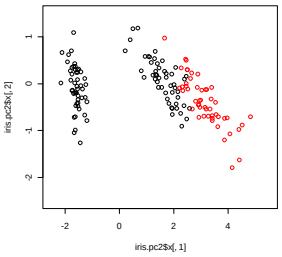
R implementation

```
> iris.pc2<-prcomp(irisnf[1:100,-5])
> irisnew<-predict(iris.pc2,irisnf[101:150,-5])
> plot(iris.pc2$x[,1],iris.pc2$x[,2],
+ xlim=c(-2.5,5.5),ylim=c(-2.5,1.5))
```



R implementation

```
> iris.pc2<-prcomp(irisnf[1:100,-5])
> irisnew<-predict(iris.pc2,irisnf[101:150,-5])
> plot(iris.pc2$x[,1],iris.pc2$x[,2],
+ xlim=c(-2.5,5.5),ylim=c(-2.5,1.5))
> points(irisnew[,1],irisnew[,2],col=2)
```



- **Summary of PCA**
  - original variables ⇨ new ones
    - uncorrelated
    - decreasing proportions of variance
  - New variables ***linear*** combinations of old
  - Transformation is a rotation/reflection of data
    ⇒ statistical information is **conserved**
  - scree plots &c shew relative importance of individual new components
    - assess 'how many are needed'

- Scatterplots of data on first few components contain almost all information so may reveal features such as group structure, outliers, …
  - i.e. scatterplots of ***scores*** on PCs
- Can assess the importance of original variables (examination of *loadings*)
- May need to examine loadings graphically if large number of variables

- Many other techniques follow **'by analogy'**:
  - rotate/transform original data to new variables
  - assess importance of new variables
  - interpret loadings of old variables: includes aspects of
    - projection pursuit
    - discriminant analysis
    - canonical correlation analysis
    - correspondence analysis,……