# Principal component analysis

## Modern multivariate statistical techniques

Miguel Rodo

2024-04-04

# Context

- Multivariate analysis helps us uncover patterns and relationships within datasets containing multiple variables.
- **Example**: Meteorologists analyze temperature, humidity, wind speed, and more to understand weather systems.
  - Possible questions:
    - What sets of variable(s) are correlated?
    - Can we predict the likelihood of rain based on these variables?
    - What variables most distinguish between different weather patterns?

# Basis of approach

- **Linear combinations**:

$$\mathbf{x}_1\mathbf{u}_1 + \mathbf{x}_2\mathbf{u}_2 + ... + \mathbf{x}_p\mathbf{u}_p$$

- Why linear combinations?
  - Simplicity » strong theoretical results » robustness, speed and interpretability

# Types of multivariate analysis

- $\mathbf{X} \sim \mathbf{X}$: PCA, factor analysis, correspondence analysis (count data)
  - **Examples**: identify correlates sets of variables
- $\mathbf{Y} \sim \mathbf{X}$: Multiple/multivariate regression, canonical correlation analysis, discrimination and classification

# Principal component analysis

- Principal components analysis (PCA) identifies the directions of greatest variation, allowing you to:
    - Represent data more compactly, and
    - Interpret relationships between variables.
- PCA is a linear technique with an algebraic solution, so it is fast to fit.

# Example applications

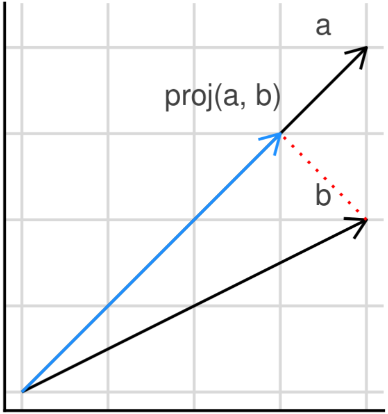
- **Market analysis**: Identify correlated stocks to ensure a diversified portfolio
- **Image compression**: Represent images with fewer pixels while preserving accuracy
- **Bioinformatics**: Identify sets of co-expressed genes responsible for different functions in the body

# Projection onto a vector ("line")

- Suppose that we wish to project the vector $\mathbf{b}$ onto the vector $\mathbf{a}$.
    - The projection is the nearest point to $\mathbf{b}$ on the line spanned by $\mathbf{a}$.
- The projection of $\mathbf{b}$ onto $\mathbf{a}$ is given by:

$$\text{proj}_{\mathbf{a}}(\mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}\mathbf{a}$$

- Here is a diagram of such a projection. Suppose that we want to project the vector $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ onto the vector $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.



# Projecting onto a unit vector

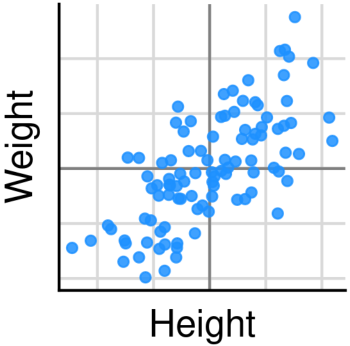
- If $\mathbf{a}$ is a unit vector, then the projection of $\mathbf{b}$ onto $\mathbf{a}$ is given by:

$$\text{proj}_{\mathbf{a}}(\mathbf{b}) = (\mathbf{a}^T\mathbf{b})\mathbf{a}.$$

- This means that the length of the projection is the dot product of $\mathbf{a}$ and $\mathbf{b}$.
  - This is a linear combination of the elements of $\mathbf{b}$: $\sum_{i=1}^{p} a_i b_i$.
- Note that the length of $\mathbf{a}$ does not affect the projection.

# Two-variable example

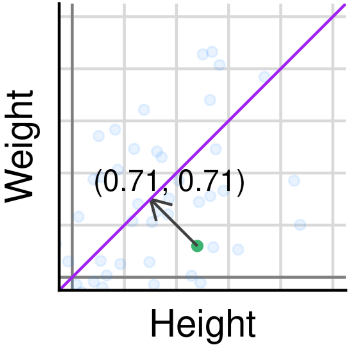
- Suppose we have a dataset with two correlated variables, perhaps weight and height.
- Then, the scatterplot of these may look something like this:



- We could combine these two variables into a single variable, representing size: $a_1 \times$ height $+ a_2 \times$ weight.
  - If we restrict $a_1^2 + a_2^2 = 1$, then this equivalent to projecting the height and weight vector ($\mathbf{b} = [\text{height}, \text{weight}]$) onto the line spanned by the vector $\mathbf{a} = [a_1, a_2]$.
  - Since the projection is independent of the length of whatever you're projecting onto, this is equivalent to projecting onto any vector in the same direction as $[a_1, a_2]$.

# Two-variable example (cont.)

- The new variable is then the *length of the projection*.
- For example, suppose we have the point $[1.2, 0.3]$:



- If we set $\mathbf{a} = [1/\sqrt{2}, 1/\sqrt{2}]$, then the new variable is $1/\sqrt{2} \times 1.2 + 1/\sqrt{2} \times 0.3$.
- We go from the matrix of vairables

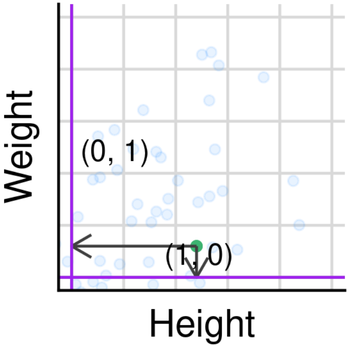
```
  height weight
1    1.2    0.3
```

- to

```
     size
1 1.06066
```

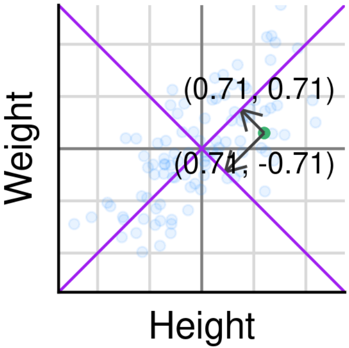
# Projections and axes

- If we set $\mathbf{a} = [1, 0]$, then the "new" variable is the height.
- If we set $\mathbf{a} = [0, 1]$, then the "new" variable is the weight.
- The point of these last two is that original coordinates are, in a sense, themselves projections.
- So, if we consider $[1, 0]$ and $[0, 1]$ as axes, then we can consider $[1/\sqrt{2}, 1/\sqrt{2}]$ as a new axis.



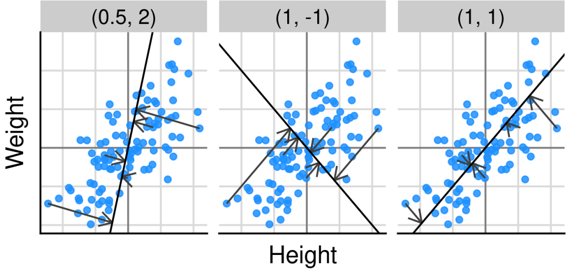
# Alternate axes example

- Clearly, if we just have one axis (e.g. $[1/\sqrt{2}, -1/\sqrt{2}]$) and two original variables (e.g. height and weight), then we cannot represent the data exactly.
- An additional linearly independent coordinate axis (e.g. $[1/\sqrt{2}, -1/\sqrt{2}]$) would then provide us two values for each point, and be an alternative way to represent the data exactly.



# How do we choose $a_1$ and $a_2$?

- Consider projecting onto the lines $[0.5, 2]$, $[1, -1]$ and $[1, 1]$:



- Height and weight increase together, so $a_1$ and $a_2$ should both be positive.
- Beyond that, we want to choose $a_1$ and $a_2$ so that the projection captures the most variation in the data.

- The variability of the (lengths of the) projections is 1.3 (0.5, 2), 0.3 (1, -1) and 1.7 (1, 1), respectively. -The projection onto $[1, 1]$ captures the most variation.

# Further motivation for capturing variation

- On its own grounds, imagine that you could retain only one of two variables: one has zero variance (all the observations are the same) and the other has some positive variance. Clearly, you would choose the latter.

# Further motivation for capturing variation

- On its own grounds, imagine that you could retain only one of two variables: one has zero variance (all the observations are the same) and the other has some positive variance. Clearly, you would choose the latter.

- Later, we will see that capturing maximal variation:
    - Minimises the distance from the original data to the projection.
    - Allows us to reconstruct the original variables with minimal error.

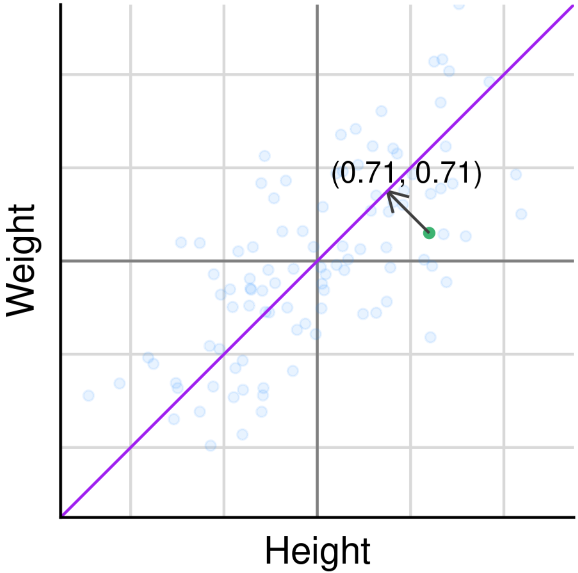
# Objective function: the first principal component

- Assume that our data $\mathbf{X} \in \mathcal{R}^p$ are distributed with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
- We seek to find the linear combination of the variables that captures the most variation.
- For the first principal component, we seek to find $\mathbf{a}_1$ such that

$$\mathsf{Var}(\mathbf{X}\mathbf{a}_1) = \mathbf{a}_1^T \boldsymbol{\Sigma} \mathbf{a}_1$$

- is maximized, subject to the constraint that $||\mathbf{a}|| = 1$.
  - This is necessary as otherwise we could meaninglessly increase the variation without bound by making $\mathbf{a}_1$ longer.
- Applied to the previous example, the first principal component would be a linear combination of height and weight that captures the most variation.

# First principal component for health and weight data



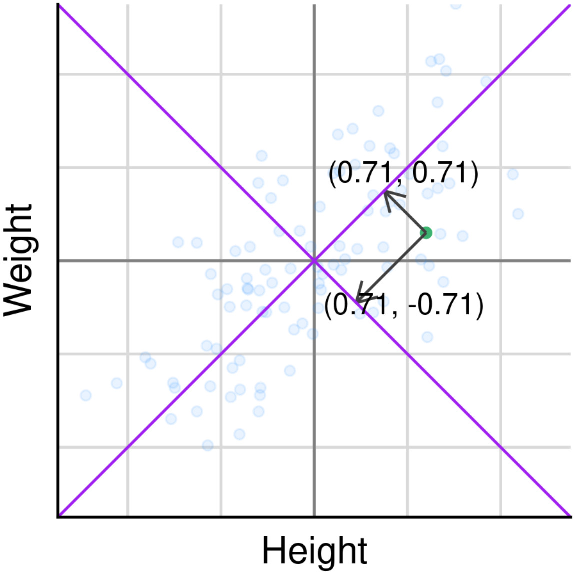
# Objective function: the second principal component

- The second principal component is the linear combination of the variables that captures the most variation, subject to the constraint that it is orthogonal to the first principal component.
- In other words, for the second principal component we seek to find $\mathbf{a}_2$ such that

$$\mathsf{Var}(\mathbf{X}\mathbf{a}_2) = \mathbf{a}_2^T \mathbf{\Sigma} \mathbf{a}_2$$

- is maximized, subject to the constraints that $||\mathbf{a}_2|| = 1$ and $\mathbf{a}_2^T \mathbf{a}_1 = 0$.

# First principal component for health and weight data



# Objective function for all principal components

- For the $k$-th principal component (for $k \leq p$), we seek to find $\mathbf{a}_k$ such that

$$\mathsf{Var}(\mathbf{X}\mathbf{a}_k) = \mathbf{a}_k^T \boldsymbol{\Sigma} \mathbf{a}_k$$

- is maximized, subject to the constraints that $||\mathbf{a}_k|| = 1$ and $\mathbf{a}_k^T \mathbf{a}_j = 0$ for $j < k$.

# Objective function for all principal components

- For the $k$-th principal component (for $k \leq p$), we seek to find $\mathbf{a}_k$ such that

$$\text{Var}(\mathbf{X}\mathbf{a}_k) = \mathbf{a}_k^T \boldsymbol{\Sigma} \mathbf{a}_k$$

- is maximized, subject to the constraints that $||\mathbf{a}_k|| = 1$ and $\mathbf{a}_k^T \mathbf{a}_j = 0$ for $j < k$.
- Up until this point, we've explored various options for $\mathbf{a}_1$ and $\mathbf{a}_2$ without any method. We'll now discuss how to compute these principal components.
- First, we'll discuss a more general results.

# Theorem 1: Maximisation of quadratic forms for points on the unit sphere

Let $\mathbf{B} : p \times p$ be a positive semi-definite matrix with the $i$-th largest eigenvalue $\lambda_i$ and associated eigenvector $\mathbf{e}_i$. Then we have that

$$\max_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \lambda_1 \text{ (attained when } \mathbf{x} = \mathbf{e}_1\text{)}$$

and that

$$\max_{\mathbf{x}\neq\mathbf{0}, \mathbf{x}\perp\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{i-1}} \frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \lambda_i \text{ (attained when } \mathbf{x} = \mathbf{e}_i\text{)}$$

for $i \in \{2, 3, \ldots, p\}$.

# Proof of Theorem 1

Let $\mathbf{P} : p \times p$ be the orthogonal matrix whose $i$-th column is the $i$-th eigenvector and $\Lambda$ be the diagonal matrix with ordered eigenvalues along the diagonal. Let $\mathbf{B}^{1/2} = \mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}'$ and $\mathbf{y} = \mathbf{P}'\mathbf{x}$.

First, we show that the quadratic form can never be larger than $\lambda_1$:

$$\begin{aligned}\frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \frac{\mathbf{x}'\mathbf{B}^{1/2}\mathbf{B}^{1/2}\mathbf{x}}{\mathbf{x}'\mathbf{P}\mathbf{P}'\mathbf{x}} = \frac{\mathbf{x}'\mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}'\mathbf{P}\boldsymbol{\Lambda}^{1/2}\mathbf{P}'\mathbf{x}}{\mathbf{y}'\mathbf{y}} = \frac{\mathbf{y}'\boldsymbol{\Lambda}\mathbf{y}}{\mathbf{y}'\mathbf{y}} \\ = \frac{\sum_{i=1}^{p}\lambda_i y_i^2}{\sum_{i=1}^{p} y_i^2} \leq \lambda_1 \frac{\sum_{i=1}^{p} y_i^2}{\sum_{i=1}^{p} y_i^2} = \lambda_1\end{aligned}$$

# Proof of Theorem 1 (cont.)

Now we show that this is actually attained for $\mathbf{x} = \mathbf{e}_1$. Since eigenvectors are by convention length 1, we consider $\mathbf{e}_1'\mathbf{B}\mathbf{e}_1$.

First, let $\mathbf{c}_i$ be the unit vector with a 1 in the $i$-th position (and 0's everywhere else).

Expanding $\mathbf{B}$ by the eigen decomposition, we have that

$$\mathbf{e}_1'\mathbf{B}\mathbf{e}_1 = \mathbf{e}_1'\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'\mathbf{e}_1.$$

Since $B$ is symmetric, the eigenvectors can be chosen orthogonal. We do so, which implies that $\mathbf{e}_i'P = \mathbf{c}_i'$, where $\mathbf{c}_i$ is the unit vector with a 1 in the $i$-th position (and 0 everwhere else). Consequently,

$$\mathbf{e}_i'\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'\mathbf{e}_i = \mathbf{c}_i'\boldsymbol{\Lambda}\mathbf{c}_i = \lambda_i,$$

and so $\mathbf{e}_1'\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'\mathbf{e}_1 = \lambda_1$.

# Proof of Theorem 1 (cont.)

Now, we consider the case where $\mathbf{x}$ is orthogonal to $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{i-1}$.

Each component in the vector $\mathbf{y}$ is the dot product of $x$ and an eigenvector $\mathbf{e}_i$. Since we choose $x$ orthogonal to the first $i-1$ eigenvectors, the first $i-1$ entries of $\mathbf{y}$ are zero.

Returning to considering the quadratic form, we have that

$$\frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \frac{\sum_{j=1}^{p} \lambda_j y_j^2}{\sum_{j=1}^{p} y_j^2} \tag{1}$$

$$= \frac{\sum_{j=i}^{p} \lambda_j y_j^2}{\sum_{j=i}^{p} y_j^2} \leq \lambda_i \frac{\sum_{j=i}^{p} y_j^2}{\sum_{j=i}^{p} y_j^2} = \lambda_i \tag{2}$$

Using the same argument as before, we can show that this is actually attained for $\mathbf{x} = \mathbf{e}_i$.

# Connection with principal components

- Theorem 1 has already done the heavy lifting, as we can simply set $\mathbf{B} = \mathbf{\Sigma}$.
- All that's left to do is place Theorem 1 in a probabilistic context, and relate it to the variance and covariance of linear combinations of a random vector.

# Theorem 2: Selecting principal components

Let $\mathbf{X} : \mathbf{p} \times 1$ be the random vector with variance-covariance matirx $\mathbf{\Sigma} : p \times p$, which has the $i$-th largest eigenvalue as $\lambda_i$ and associated eigenvector $\mathbf{e}_i$.

Then the $i$-th principal component (defined as before) is given by

$$Y_i = \mathbf{e}_i'\mathbf{X},$$

implying that

$$\begin{aligned} \mathrm{Var}[Y_i] &= \lambda_i \; \forall \; i \in \{1, 2, \dots, p\}, \text{ and} \\ \mathrm{Cov}[Y_i, Y_j] &= 0 \text{ for } i \neq j. \end{aligned}$$

# Proof of Theorem 2

From Theorem 1, we know that

$$\max_{\mathbf{x}} \frac{\mathbf{x}'\mathbf{\Sigma}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \lambda_1,$$

which we achieve by setting $\mathbf{x} = \mathbf{e}_1$.

Since $||\mathbf{e}_1|| = 1$, $\frac{\mathbf{e}_1'\mathbf{\Sigma}\mathbf{e}_1}{\mathbf{e}_1'\mathbf{e}_1} = \mathbf{e}_1'\mathbf{\Sigma}\mathbf{e}_1$, which is equal to $\mathrm{Var}[\mathbf{a}'\mathrm{X}] = \mathrm{Var}[Y_1]$ if we set $\mathbf{a} = \mathbf{e}_1$. This implies that $\mathbf{a} = \mathbf{e}_1$ maximises the variance, which is $\lambda_1$.

Analagous reasoning shows that $\mathrm{Var}[\mathbf{a}_i'\mathbf{X}]$ has a maximum value $\lambda_i$ for $i \in \{2, 3, \ldots, p\}$ that is attained when we set $\mathbf{a}_i = \mathbf{e}_i$, under the restriction $\mathbf{a}_i$ to be orthogonal to $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{i-1}$.

For $i \neq j$, $\mathrm{Cov}[\mathbf{e}_i'\mathbf{X}, \mathbf{e}_j'\mathbf{X}] = \mathbf{e}_i'\mathbf{\Sigma}\mathbf{e}_j = 0$.

# A note on terminology

- Principal components (the $\mathbf{Y}_i$'s) may also be referred to as "scores".
- The coefficient vectors $\mathbf{a}_i$ may also be referred to as "loadings" (thinking algebraically) or as "principal axes"/"principal directions" (thinking geometrically).

# Theorem three: Total variance

Let $X' = [X_1, X_2, \dots, X_p]$ be a data matrix with covariance matrix $\Sigma$ and eigenvalue-eigenvector pairs $(\lambda_1, e_1), \dots, (\lambda_p, e_p)$, ordered such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$. Define the principal components $Y_i = e_i' X'$ for $i = 1, \dots, p$. Then, the sum of variances of the original variables is equal to the sum of the eigenvalues of $\Sigma$, which is also equal to the sum of variances of the principal components:

$$\sum_{i=1}^{p} \sigma_{ii} = \sum_{i=1}^{p} \text{Var}(X_i) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \text{Var}(Y_i).$$

# Proof of theorem three

The trace of the covariance matrix $\Sigma$, which is the sum of its diagonal elements, equals the sum of its eigenvalues due to the properties of the trace operator and the orthogonality of eigenvectors:

$$\text{trace}(\Sigma) = \sum_{i=1}^{p} \sigma_{ii} = \sum_{i=1}^{p} \lambda_i.$$

Since $\Sigma$ can be decomposed as $\Sigma = PDP'$, where $D$ is the diagonal matrix of eigenvalues and $P$ is the matrix of eigenvectors, we have:

$$\text{trace}(\Sigma) = \text{trace}(PDP') = \text{trace}(D) = \sum_{i=1}^{p} \lambda_i.$$

Therefore, the total variance accounted for by the $k$-th principal component is given by the proportion:

# Theorem 4: Correlation with original variables

If $Y_1 = e_1'X, Y_2 = e_2'X, \dots, Y_p = e_p'X$ are the principal components obtained from the covariance matrix $\Sigma$, then $\rho_{Y_i X_k} = \frac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}, \quad i, k = 1, 2, \dots, p$ are the correlation coefficients between the components $Y_i$ and variable $X_k$, where $(\lambda_1, e_1), \dots, (\lambda_p, e_p)$ are eigenvalue-eigenvector pairs of $\Sigma$.

**Proof:** Let $a_k' = [0, \dots, 0, 1, 0, \dots, 0]$ so that $X_k = a_k'X$ and Cov$(X_k, Y_i) =$ Cov$(a_k'e_i', e_i') = a_k'e_i'e_i'$. Since $\Sigma e_i = \lambda_i e_i$ it follows that

$$\text{Cov}(X_k, Y_i) = a_k'e_i' = a_k'\lambda_i e_i = e_{ik}.$$

We know that Var$(Y_i) = \lambda_i$ and Var$(X_k) = \sigma_{kk}$ so

$$\rho_{Y_i X_k} = \frac{\text{Cov}(Y_i, X_k)}{\sqrt{\text{Var}(Y_i)}\sqrt{\text{Var}(X_k)}} = \frac{\lambda_i e_{ik}}{e_{ik}\sqrt{\lambda_i}\sqrt{\sigma_{kk}}}$$

# Example

Suppose three random variables $X_1$, $X_2$, and $X_3$ are thus correlated:

```
sigma_mat <- matrix(
  c(1, -2, 0,
    -2, 5, 0,
    0, 0, 2),
  byrow = TRUE,
  nrow = 3
)
sigma_mat
```

```
     [,1] [,2] [,3]
[1,]    1   -2    0
[2,]   -2    5    0
[3,]    0    0    2
```

We apply the eigen decomposition to the covariance matrix:

```
eig_obj <- eigen(sigma_mat)
```

to obtain the eigenvalues:

```
[1] 5.830 2.000 0.172
```

and the eigenvectors:

```
eig_obj$vectors |> signif(3)
```

```
       [,1] [,2]  [,3]
[1,] -0.383    0 0.924
[2,]  0.924    0 0.383
[3,]  0.000    1 0.000
```

# Constant-density contours I

By the definition of the MVN PDF, the density of $X$ is constant on the ellipsoid defined by the equation

$$(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)=c^2.$$

This ellipsoid's shape, size and orientation is determined by the eigendecomposition of $\Sigma$.

WLOG, we assume $\mu=0$. Then, the ellipsoid is defined by

$$c^2=x'\Sigma^{-1}x=\frac{1}{\lambda_1}(e_1'x)^2+\frac{1}{\lambda_2}(e_2'x)^2+...+\frac{1}{\lambda_p}(e_p'x)^2,$$

by orthogonality of the eigenvectors.

# Constant-density contours II

As the eigenvectors are orthogonal, the axes are given by

$$\pm c\sqrt{\lambda_i}\mathbf{e}_i, \quad i = 1, 2, \dots, p.$$

The eigenvectors therefore define the direction and the eigenvalues the length of the axes of the ellipsoid.

Furthermore, if we set $y_i = \mathbf{e}_i'\mathbf{x}$ (the $i$-th principal componet for $\mathbf{x}$), we have

$$c^2 = \frac{1}{\lambda_1}y_1^2 + \dots + \frac{1}{\lambda_p}y_p^2.$$

The equation is therefore satisfied by the principal components of $\mathbf{x}$.

# Variance standardisation

- Before applying PCA, one can standardise the variances to have unit variance.
- This may be appropriate when:
  - the variances have different units (e.g. weight in kg vs annual income in rands), or
  - the variances have different scales (e.g. some genes are extremely rare whereas others are abundantly expressed), or
  - we wish to eliminate any effect of differences in variance on downstream analyses, e.g. when applying penalised regression.
- Similar comments go for logging the data, or applying other transformations.
- All results go through in exactly the same way, with adjustments made in interpreting exactly what variables the principal components capture.

# Standardisation example

We have the following covariance matrix:

```
     [,1] [,2]
[1,]    1    4
[2,]    4  100
```

which produces the following eigendecomposition:

```
$values
[1] 100.00   0.84

$vectors
     [,1]  [,2]
[1,] 0.04 -1.00
[2,] 1.00  0.04
```

The associated correlation matrix:

```
     [,1] [,2]
[1,]  1.0  0.4
[2,]  0.4  1.0
```

has the following eigendecomposition:

```
$values
[1] 1.4 0.6

$vectors
     [,1]  [,2]
[1,] 0.71 -0.71
[2,] 0.71  0.71
```

The eigenvectors are in very different directions.

# Sample principal components I

In practice, we do not know $\mathbf{\Sigma}$ and must estimate it by the sample covariance matrix $\mathbf{S} = n^{-1}\mathbf{X}'\mathbf{X}$.

For example, the eigen-decomposition of the sample covariance matrix of the height and weight data is:

```
$values
[1] 1.70 0.31


$vectors
     [,1]  [,2]
[1,] 0.71 -0.71
[2,] 0.71  0.71
```

So, the first principal component is given by projections onto $[1, 1]$.

If $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, then we have that that

# Approximating a matrix $\mathbf{X}$

Suppose we have a matrix $\mathbf{X}$ of size $n \times p$ with rank $t$ and we wish to approximate it with a matrix $\hat{\mathbf{X}}$ of rank $s < t$.

In other words, we wish to find a matrix

$$\hat{\mathbf{X}} = \mathbf{AB},$$

where $\mathbf{A}$ is of size $n \times s$ and $\mathbf{B}$ is of size $s \times p$, such that

$$\mathbf{X} \approx \hat{\mathbf{X}}.$$

We want a rank $s$ approximation to the matrix $\mathbf{X}$.

# Approximation example

In the height and weight dataset, we obtained a size variable (the first principal component) that was an equally-weighted combination of weight and height, as follows:

$$\text{size} = \frac{1}{\sqrt{2}}\text{height} + \frac{1}{\sqrt{2}}\text{weight}. = \begin{pmatrix}\text{height} & \text{weight}\end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

We'll show that can approximate the original height and weight variables using the size variable:

$$[\text{height}, \text{weight}] \approx \text{size} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2}\text{size} & 1/\sqrt{2}\text{size} \end{pmatrix}.$$

We therefore have a "rule" for mapping size onto weight and height.

# Approximation example cont.

Suppose that a given observation has height 0.7 and weight 0.5.

This implies a first principal component (size) of

$$\text{size} = \frac{1}{\sqrt{2}} \times 0.7 + \frac{1}{\sqrt{2}} \times 0.5 = 0.85.$$

We can approximate the height and weight as follows:

$$\begin{pmatrix} \widehat{\text{height}} & \widehat{\text{weight}} \end{pmatrix} \approx \begin{pmatrix} 1/\sqrt{2} \times 0.85 & 1/\sqrt{2} \times 0.85 \end{pmatrix} = \begin{pmatrix} 0.6 & 0.6 \end{pmatrix}.$$

Considering the standard deviation is (by construction) 1, the error (0.1 in both cases) is small.

If weight and height were very different, the error would be larger (why?).

Whether or not PCA is a good approximation matters because a) we may actually want to reconstitute the original data after compression (e.g. image compression) and b) we want to know we're not losing too much information.

# Minimising the least squares error of the approximation

We wish to find a matrix $\hat{\mathbf{X}}$ of rank $s$ that that minimises the least squares error:

$$\text{min}_{\hat{\mathbf{X}}} \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \hat{x}_{ij})^2 = \text{min}_{\hat{\mathbf{X}}} \text{trace}((\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})').$$

We are going to

1. prove that SVD can be used to find $\hat{\mathbf{X}}$, and
2. that the PCA and SVD are equivalent in this context.

# Theorem 5a: SVD approximation

Let $\mathbf{X}$ be a matrix of size $n \times p$ with rank $t$ and SVD $\mathbf{X} = \mathbf{UDV}'$. Then the best rank $s$ approximation to $\mathbf{X}$ is given by

$$\hat{\mathbf{X}} = \mathbf{UDJ}_s\mathbf{V}',$$

where $\mathbf{J}_s$ is the matrix of size $p \times p$ with the first $s$ diagonal elements equal to 1 and the rest equal to 0.

This is equivalent to

$$\hat{\mathbf{X}} = \sum_{i=1}^{s} d_i \mathbf{u}_i \mathbf{v}_i',$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the left- and right-singular vectors and $d_i$ is the $i$-th singular value of $\mathbf{X}$.

# Proof of Theorem 5a

We use $UU^* = I_m$ and $VV^* = I_k$ to write the sum of squares as

$\text{tr}[(A-B)(A-B)^*] = \text{tr}[U^*(A-B)V^*(A-B)VU]$

$= \text{tr}[U^*(A-BV)(A-BV)^*U]$

$= \text{tr}[U^*(A-BV)U(U^*A-BV^*)]$

$= \text{tr}[(UA-BV)(UA-BV)^*]$

If we let $A' = UA$ and $C' = BV$

$= \text{tr}[(C'-C')(C'^*-C')]$

$$= \sum_{i,j=1}^{m}(a_{ij}-c_{ij})^2$$

which will be a minimum when $c_{ij} = 0$ unless $i = j$ and $i \leq s$

$$\Rightarrow UB'V^* = \Lambda_s \text{ or } B \Rightarrow \sum_{i=1}^{s}\lambda_i u_i v_i^*.$$

# Theorem 5b: PCA leads to same approximation as the SVD

The PCA approximation to $\mathbf{X}$ is given by $\tilde{\mathbf{X}} = \mathbf{Y}\mathbf{J}_k\mathbf{P}'$, where $\mathbf{J}_k$ is the matrix of size $p \times p$ with the first $k$ diagonal elements equal to 1 and the rest equal to 0.

This is the same as the SVD approximation, i.e.

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{D}\mathbf{J}_k\mathbf{V}' = \tilde{\mathbf{X}} = \mathbf{Y}\mathbf{J}_k\mathbf{P}'.$$

# Theorom 5b: proof

By definition, $\mathbf{P} = \mathbf{V}$ as $\mathbf{P}$ and $\mathbf{V}$ both have the ordered eigenvectors of $\mathbf{X}'\mathbf{X}$ along the columns.

Therefore,

$$\mathbf{Y} = \mathbf{XP} = \mathbf{XV},$$

we have that

$$\mathbf{Y} = \mathbf{UDV}'\mathbf{V} = \mathbf{UD} = \hat{\mathbf{X}}.$$

# Contribution per component

Theorem 2 showed that

$$\begin{aligned} tr(\Sigma) &= tr(P\Lambda P') = tr(\Lambda P P') = tr(\Lambda) \\ &= \lambda_1 + \lambda_2 + ... + \lambda_p = \sum Var(Y_i). \end{aligned}$$

This implies that the proportion of the total variance due to the $k^{th}$ principal component is

$$\frac{\lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_p}, k = 1, ...p,$$

and the proportion of variance accounted for by the first $r$ principal components is given by

$$\frac{\lambda_1 + \lambda_2 + ... + \lambda_r}{\lambda_1 + \lambda_2 + ... + \lambda_p}.$$

# Scree plot

A scree plot of eigenvalues $(\lambda_i)$ against indices $i$ can be used to determine the number of principal components to retain.

The point at which the plot levels off is the number of principal components to retain. This point is typically referred to as an “elbow”.

# Final example

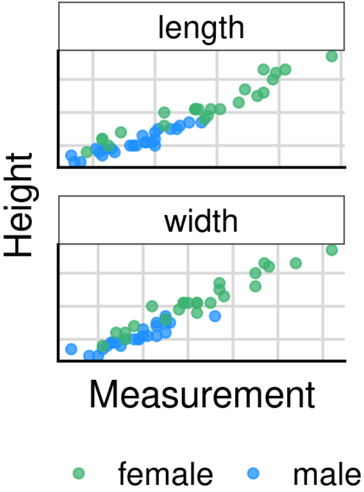
We read in data of measurements Painted Turtles (Jolicoeur and Mosimann 1960):

```
if (!requireNamespace(
  "remotes", quietly = TRUE
  )) {
  install.packages("remotes")
}
"MiguelRodo/DataTidy23RodoHonsMult@2024" |>
  remotes::install_github()
```

```
data(
  "data_tidy_turtle",
  package = "DataTidy23RodoHonsMu
)
data_tidy_turtle
```

```
# A tibble: 49 x 4
   gender length width height
   <chr>   <dbl> <dbl>  <dbl>
 1 male       93    74     37
 2 male       94    78     35
 3 male       96    80     35
 4 male      101    84     39
 5 male      102    85     38
 6 male      103    81     37
```

# Measurements highly correlated



- Due to the correlation, we may likely summarise the data very well using principal components.
- We'll use the `prcomp` function in R, and do it from first principles.

# Using the prcomp function I

```r
pr_obj <- prcomp(
  ~log(length) + log(width) + log(height),
  data = data_tidy_turtle,
  retx = TRUE
)
pr_obj
```

```
Standard deviations (1, .., p=3):
[1] 0.25969403 0.03573218 0.02104418

Rotation (n x k) = (3 x 3):
                  PC1        PC2         PC3
log(length) 0.6097413 -0.5595404  0.56136451
log(width)  0.4824691 -0.2999000 -0.82297239
log(height) 0.6288395  0.7726413  0.08709957
```

- The first principal component has a markedly higher standard deviation.
- It is weighted roughly evenly across the variables, indicating it is a general “size” variable.

# Using the prcomp function II

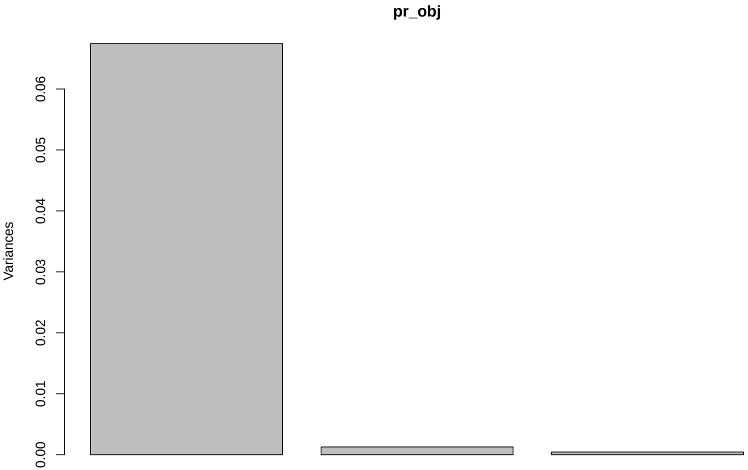
```
summary(pr_obj)
```

```
Importance of components:
                          PC1     PC2     PC3
Standard deviation     0.2597 0.03573 0.02104
Proportion of Variance 0.9751 0.01846 0.00640
Cumulative Proportion  0.9751 0.99360 1.00000
```

- The `summary` function provides the per-component standard deviation and variance, as well as the cumulative variance.

# Using the `prcomp` function III

```
plot(pr_obj)
```



- The `plot` command provides the scree plot.
- Clearly, unless we're specifically interested in non-size variables we should only retain the first principal component.

# Using the prcomp function IV

Here are the principal components:

```r
pr_obj$x |> head() |> signif(2)
```

```
    PC1     PC2    PC3
1 -0.42  0.0690  0.027
2 -0.42  0.0044 -0.015
3 -0.40 -0.0150 -0.024
4 -0.28  0.0260 -0.026
5 -0.28 -0.0036 -0.032
6 -0.31 -0.0150  0.010
```

# From first principles I

Using the SVD, we can obtain the same results as the `prcomp` function.

Removing gender, logging and centering the data:

```
data_turtle_mat <- data_tidy_turtle |>
  dplyr::select(-gender) |>
  dplyr::mutate(across(everything(), log)) |>
  dplyr::mutate(across(everything(), function(x) x - mean(x))) |>
  as.matrix()
```

We then calculate the SVD:

```
svd_turtle <- svd(data_turtle_mat)
```

# From first principles II

The eigenvalues are the squares of the singular values:

```
svd_turtle$d^2 |> signif(3)
```

```
[1] 3.2400 0.0613 0.0213
```

The right-singular vectors are the eigenvectors of the covariance matrix, and hence are the loading vectors, implying the principal components are given by:

```
(data_turtle_mat %*% svd_turtle$v) |> head() |> signif(2)
```

```
      [,1]    [,2]   [,3]
[1,] -0.42  0.0690  0.027
[2,] -0.42  0.0044 -0.015
[3,] -0.40 -0.0150 -0.024
[4,] -0.28  0.0260 -0.026
[5,] -0.28 -0.0036 -0.032
```

# Biplots

- The axes of principal components are linear combinations of the original variables.
- This makes interpreting the principal components in terms of the original points more difficult.
- To alleviate this problem, we can plot the principal components and the original variables on the same plot.
- This is called a biplot.

# Definition of a biplot

Suppose we have a rank $s$ matrix $\hat{\mathbf{X}} : n \times p$. Then we write

$$\hat{\mathbf{X}} = \mathbf{A}\mathbf{B},$$

where $\mathbf{A}$ is of size $n \times s$ and $\mathbf{B}$ is of size $s \times p$.

The biplot is a plot of the rows of $\mathbf{A}$ and the columns of $\mathbf{B}$.

Typically, we plot the rows as points and the columns as vectors.

# A PCA biplot

The PCA approximation to a matrix $\mathbf{X}$ is given by

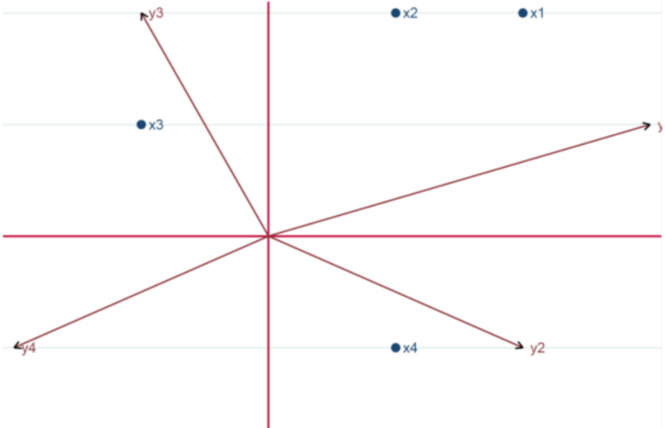
$$\mathbf{Y}\mathbf{J}_k\mathbf{P}'.$$

Typically, we take the first two columns of $\mathbf{Y}$ and the first two columns of $\mathbf{P}$ to form the biplot.

The rows of $\mathbf{Y}$ are the scores (representing the observations) and the rows of $\mathbf{P}'$ are the loadings (representing the variables).

# Interpreting a biplot

- An observation has (approximately) an above-average value of a variable if it is close to the tip of the vector representing that variable.
- Two variables are (approximately) correlated if their vectors are close together.
  - They are (approximately) uncorrelated if they are orthogonal.


y3
x2
x1
x3
y4
x4
y2