8.2. Reducción de dimensiones

Smart City

LAB CTIC UNI

Dr. Manuel Castillo-Cara
Machine Learning con Python
Web: www.smartcityperu.org

Índice

- Introducción
- Principal Components Analysis.
- Varianza y selección de k.
- PCA como algoritmos de aprendizaje no supervisado

Smart City

LAB CTIC UNI

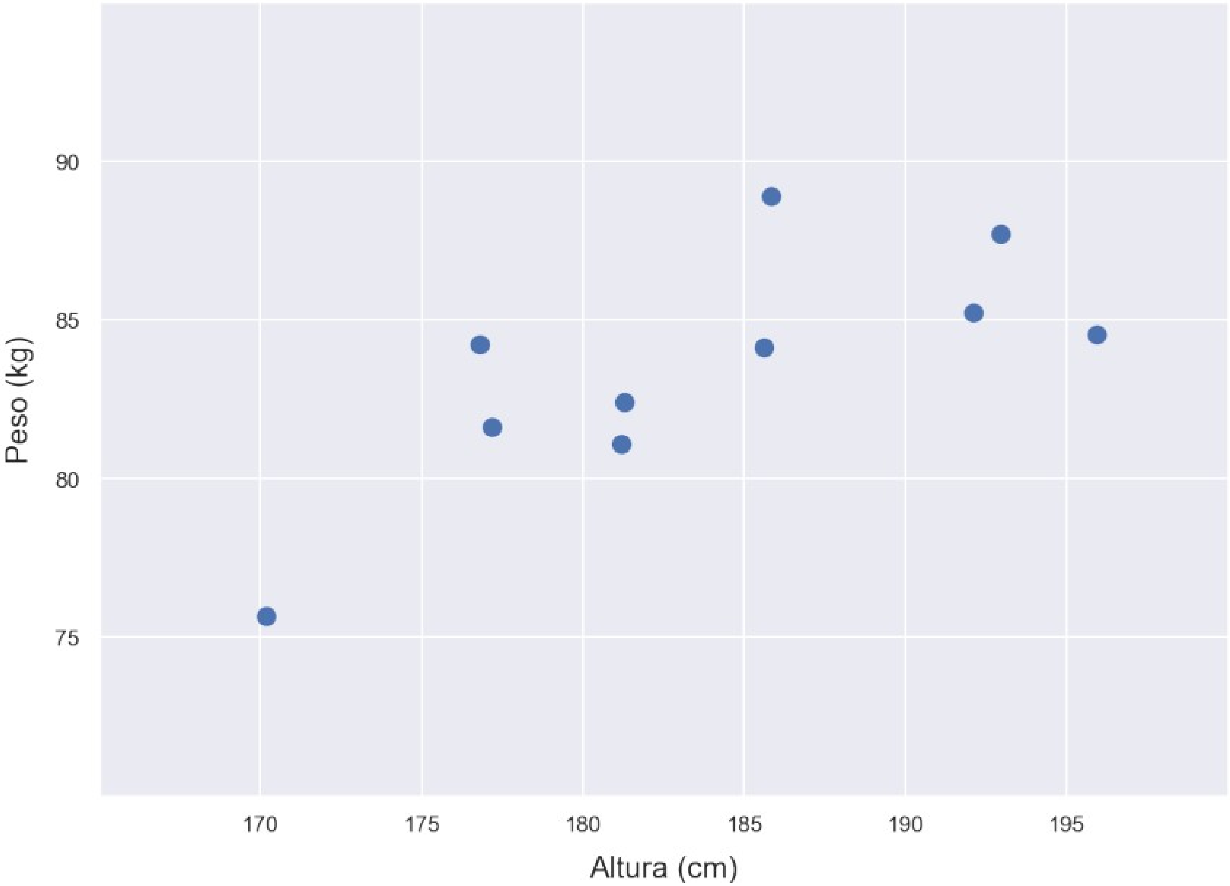
Reducción de dimensiones

1. Definición

- El análisis de componentes principales es una técnica de **aprendizaje no supervisado**.
- Consiste en crear un conjunto de características (componentes) que representen la información **relevante** que proporcionan las variables originales.
- Se persigue que el número de características sea **sustancialmente menor** que el conjunto de variables.
- Las características han de **maximizar** la proporción de **varianza** de los datos que **retienen** (explican).
  - Maximizan la información que nos proporcionan.

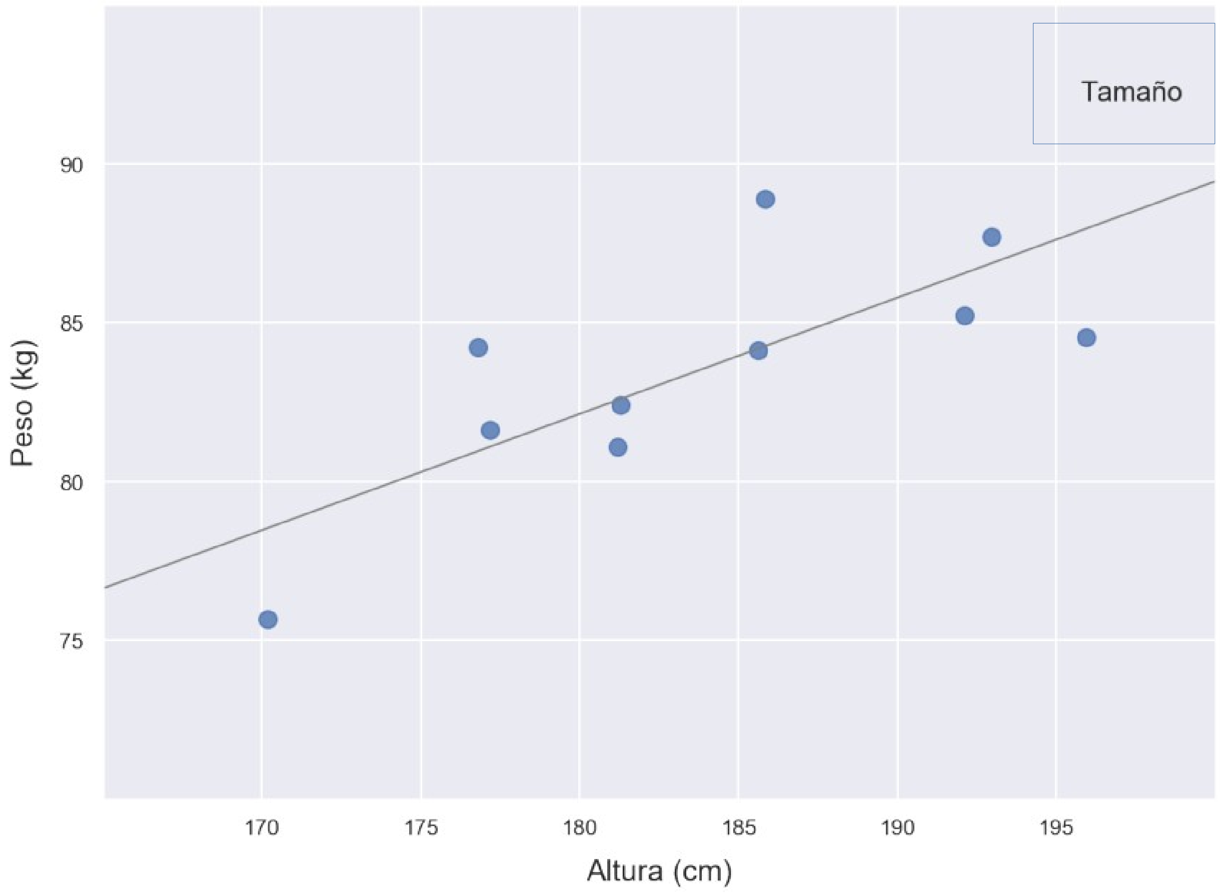
2. Datos fuertemente correlacionados
Ejemplo

- Parte de la información que reflejan, corresponde a **una** misma característica.



2. Datos fuertemente correlacionados
Ejemplo

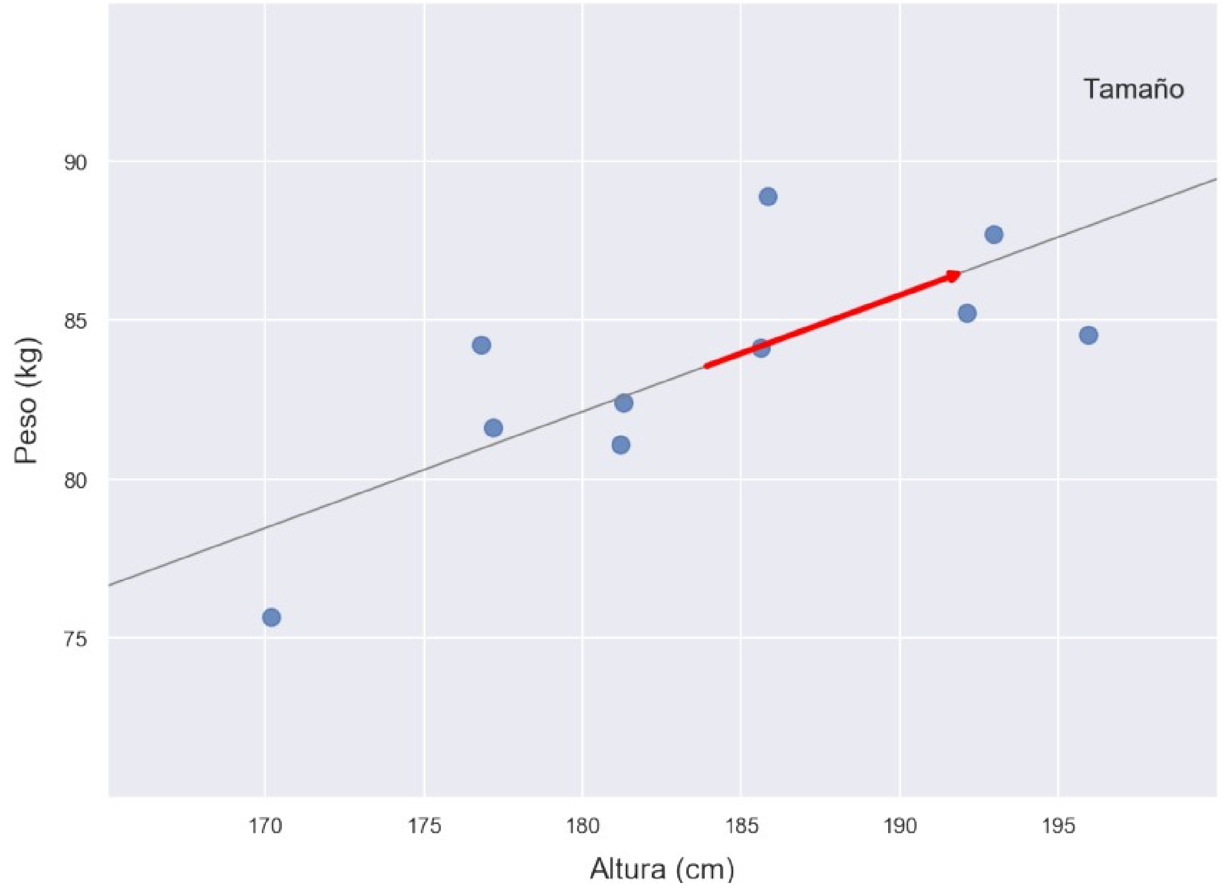
- Parte de la información que reflejan, corresponde a **una** misma característica.



Característica subyacente que representa a las dos características correlacionadas: tamaño y peso

2. Datos fuertemente correlacionados
Ejemplo

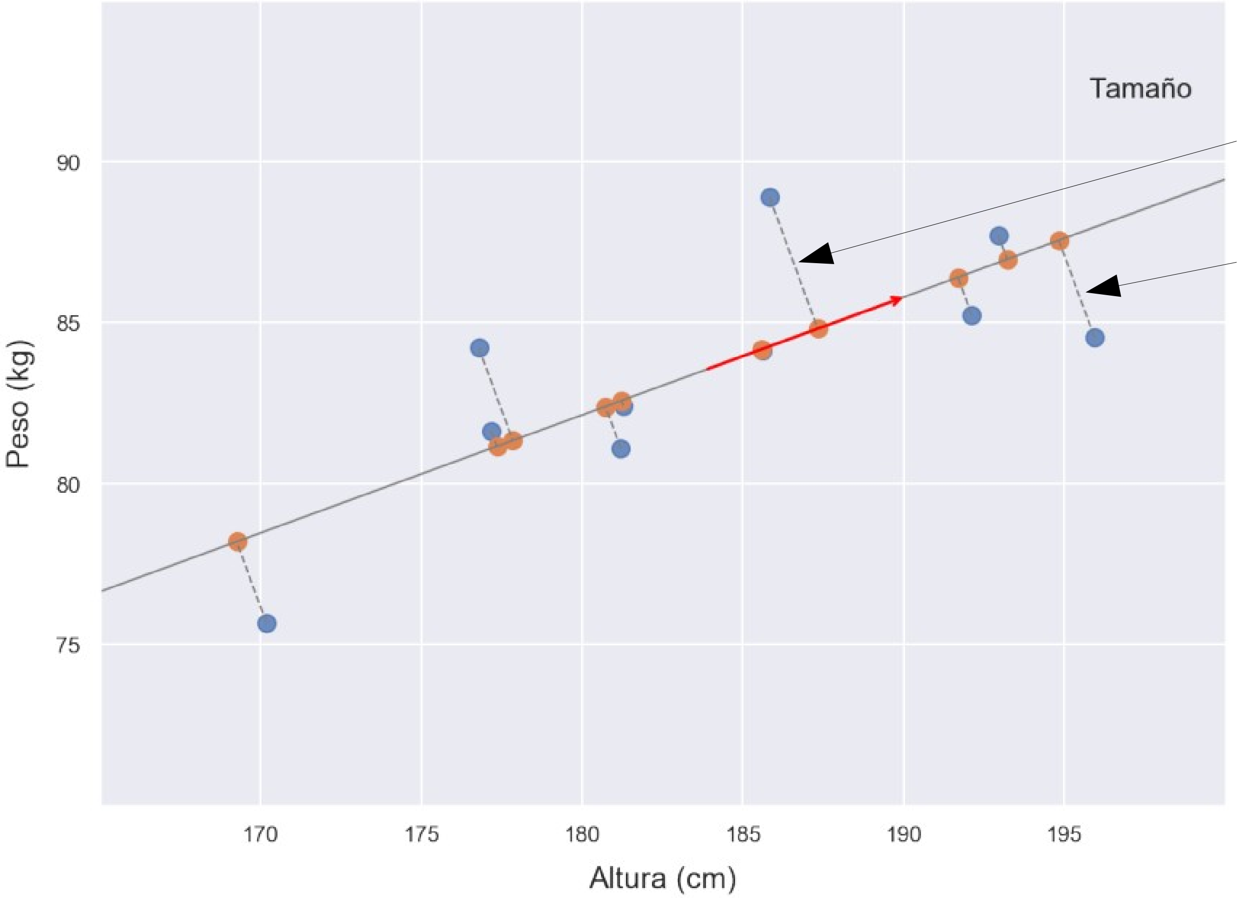
- Definida esa característica, es posible representar los datos en **una sola dimensión**.



La representación unidimensional (tamaño) se puede representar como un vector (rojo)

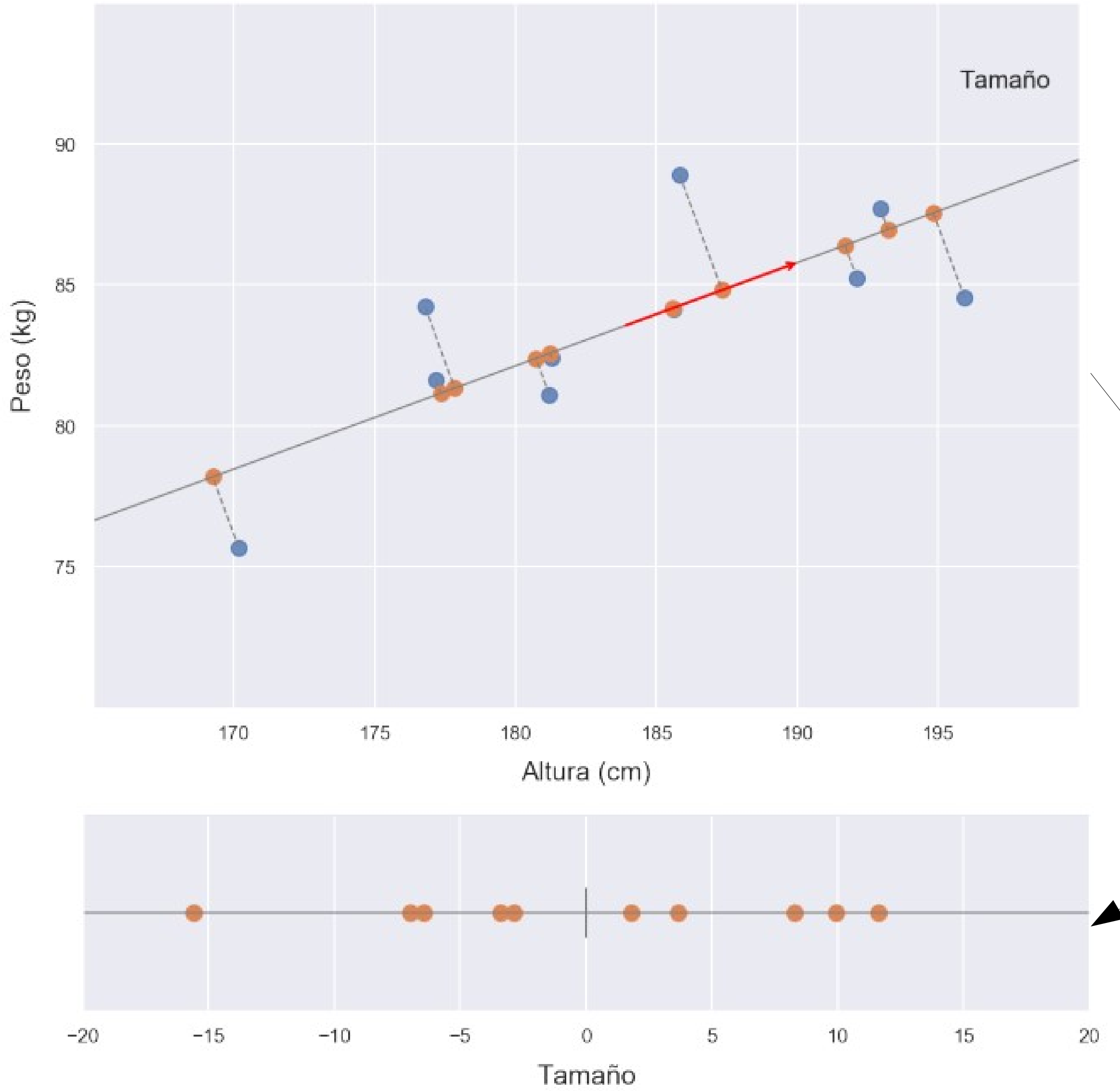
2. Datos fuertemente correlacionados
Ejemplo

- Definida esa característica, es posible representar los datos en **una sola dimensión**.



Pero ese vector con nuevas instancias tienen asociado un error que va desde el punto del vector hasta el punto original de la interesección de las dos características iniciales

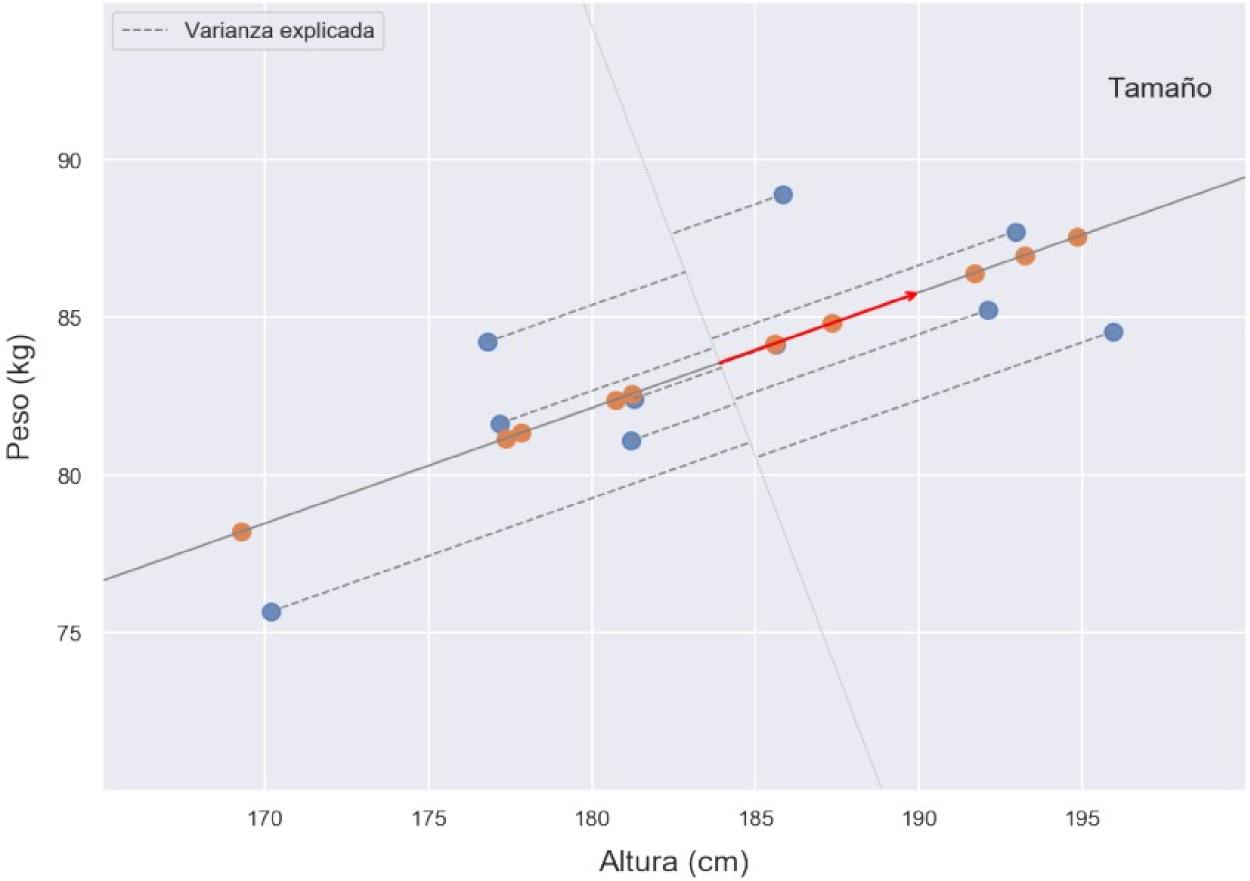
2. Datos fuertemente correlacionados
Ejemplo



Podemos ver la representación final en una dimesión de las característcas de dos dimensiones originales. Representa la misma información

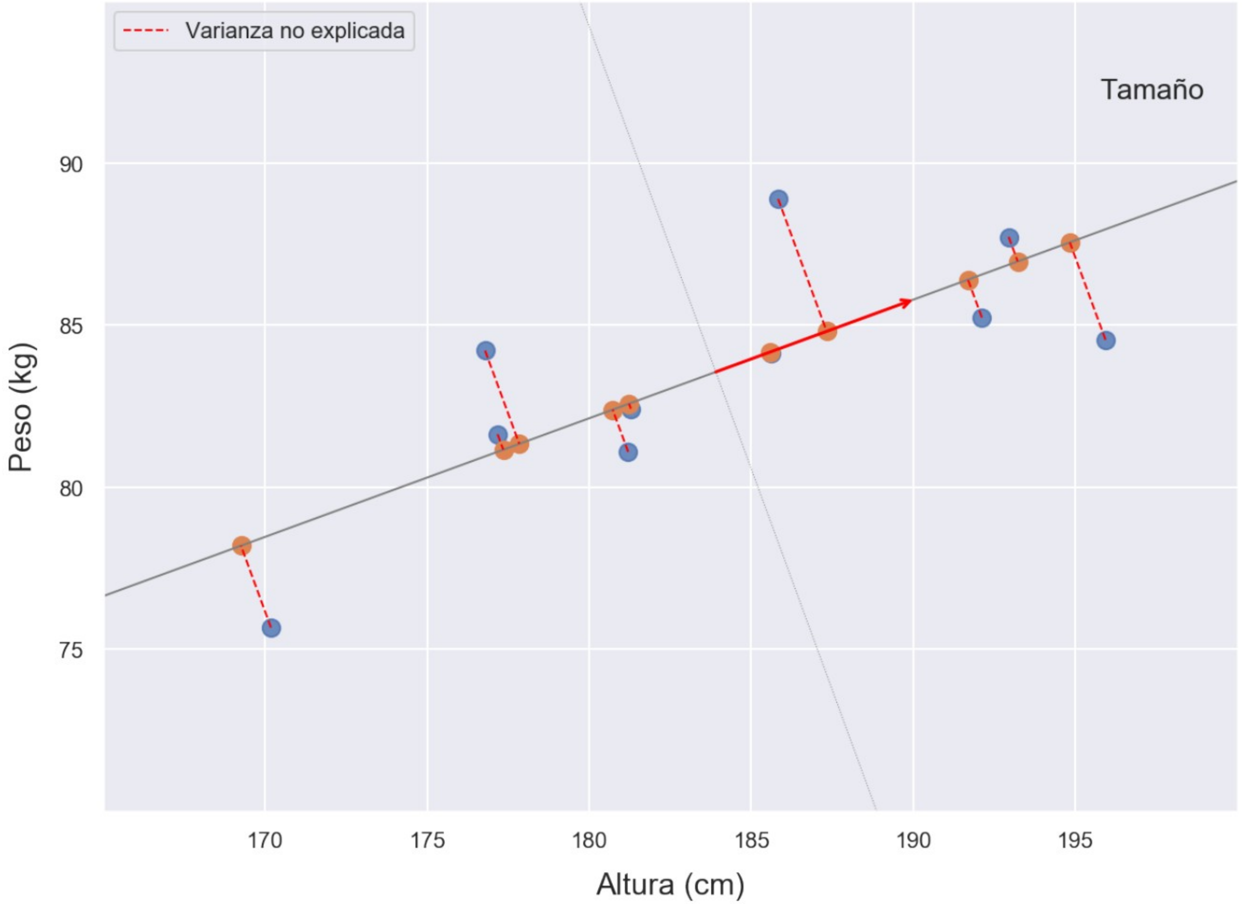
2. Datos fuertemente correlacionados
Ejemplo

- La nueva característica **retiene** gran parte de la **varianza** presente en los datos originales.



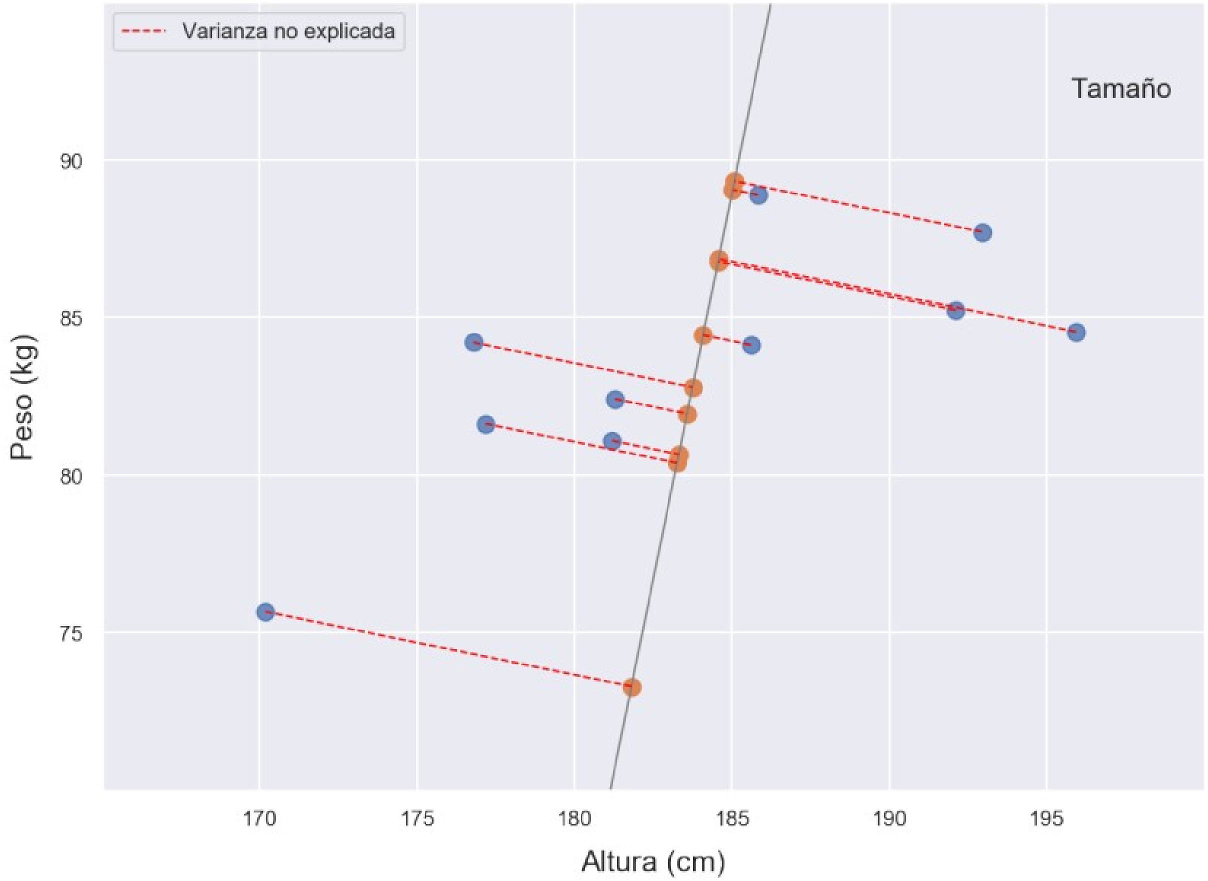
2. Datos fuertemente correlacionados
Ejemplo

- La nueva característica **no** retiene otra parte **varianza** presente en los datos originales.



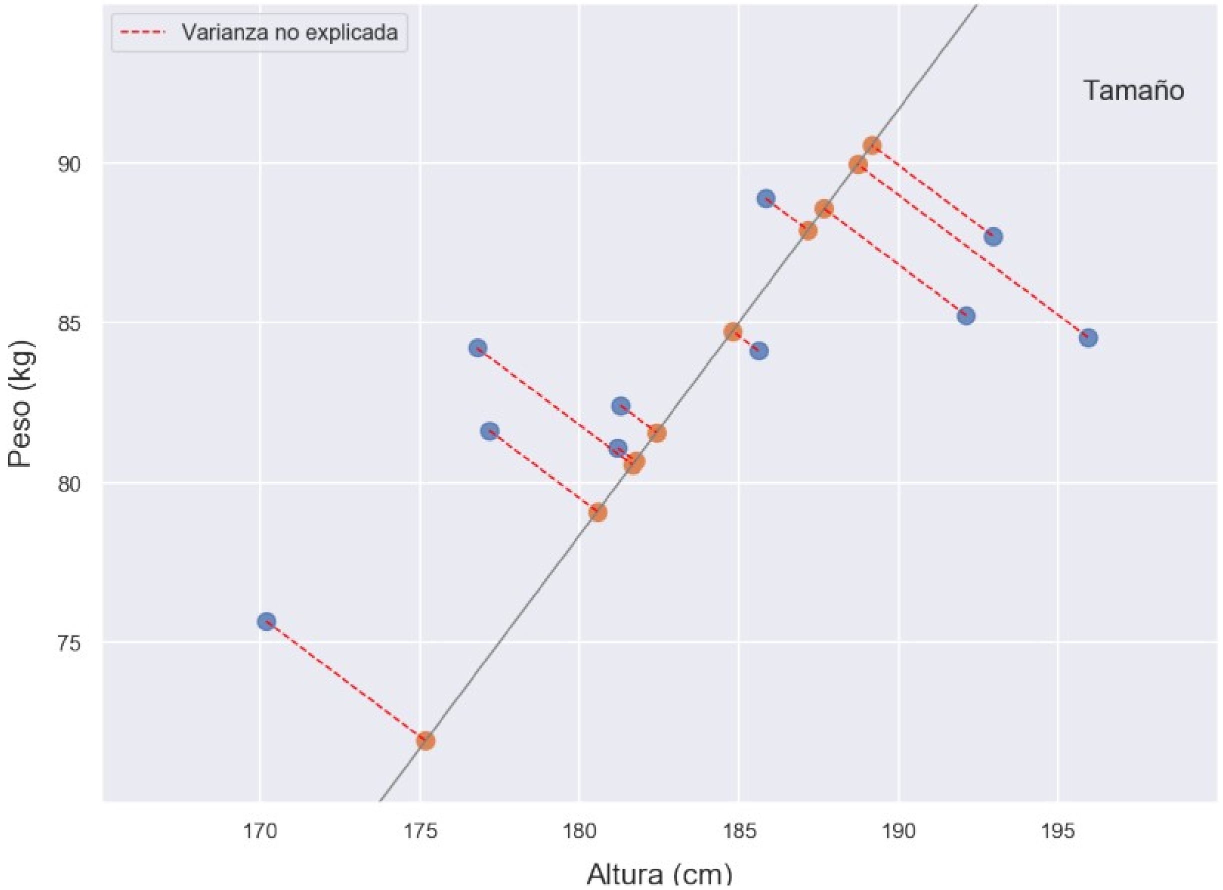
2. Datos fuertemente correlacionados
Ejemplo

- El objetivo es encontrar **la componente que maximice la varianza** retenida (**minimice** la **no** retenida).



2. Datos fuertemente correlacionados
Ejemplo

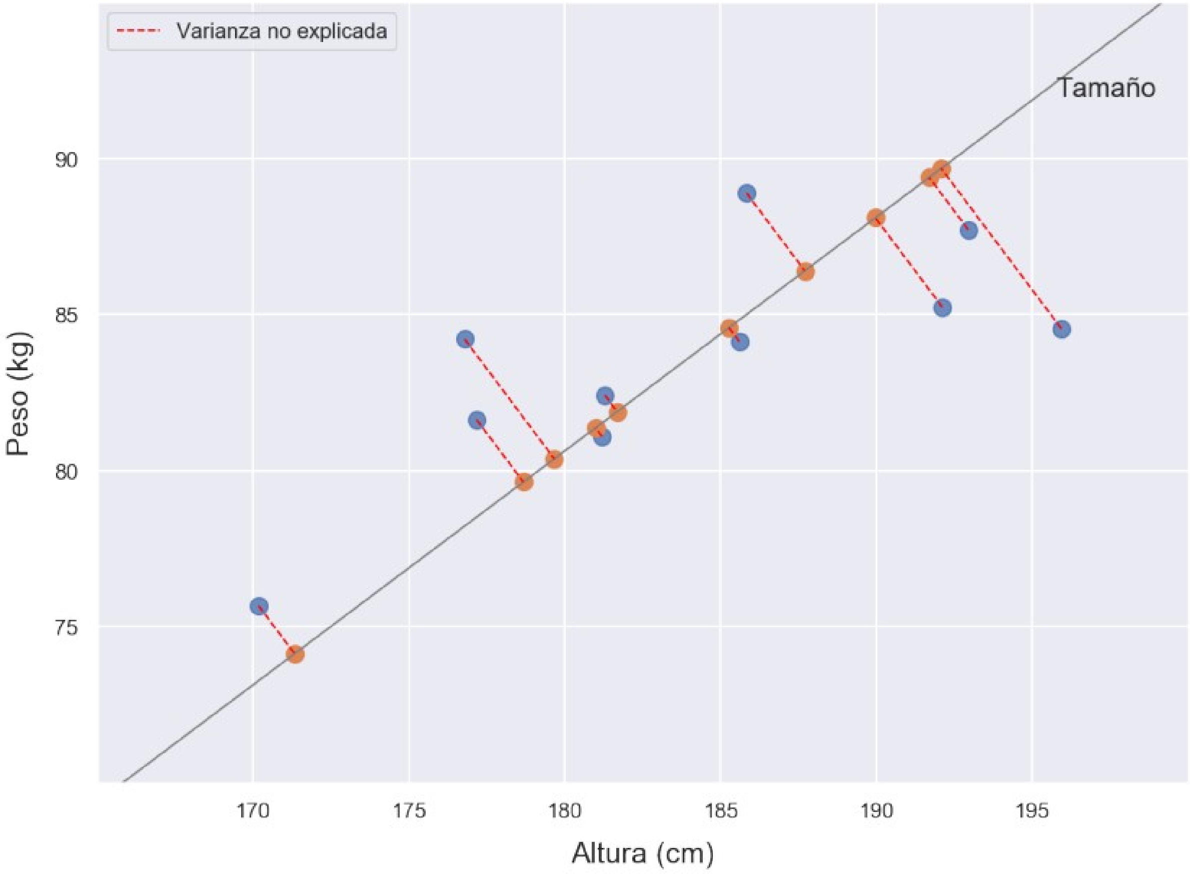
- El objetivo es encontrar **la componente que maximice la varianza** retenida (**minimice** la **no** retenida).



La varianza (líneas en rojo) van disminuyendo

2. Datos fuertemente correlacionados
Ejemplo

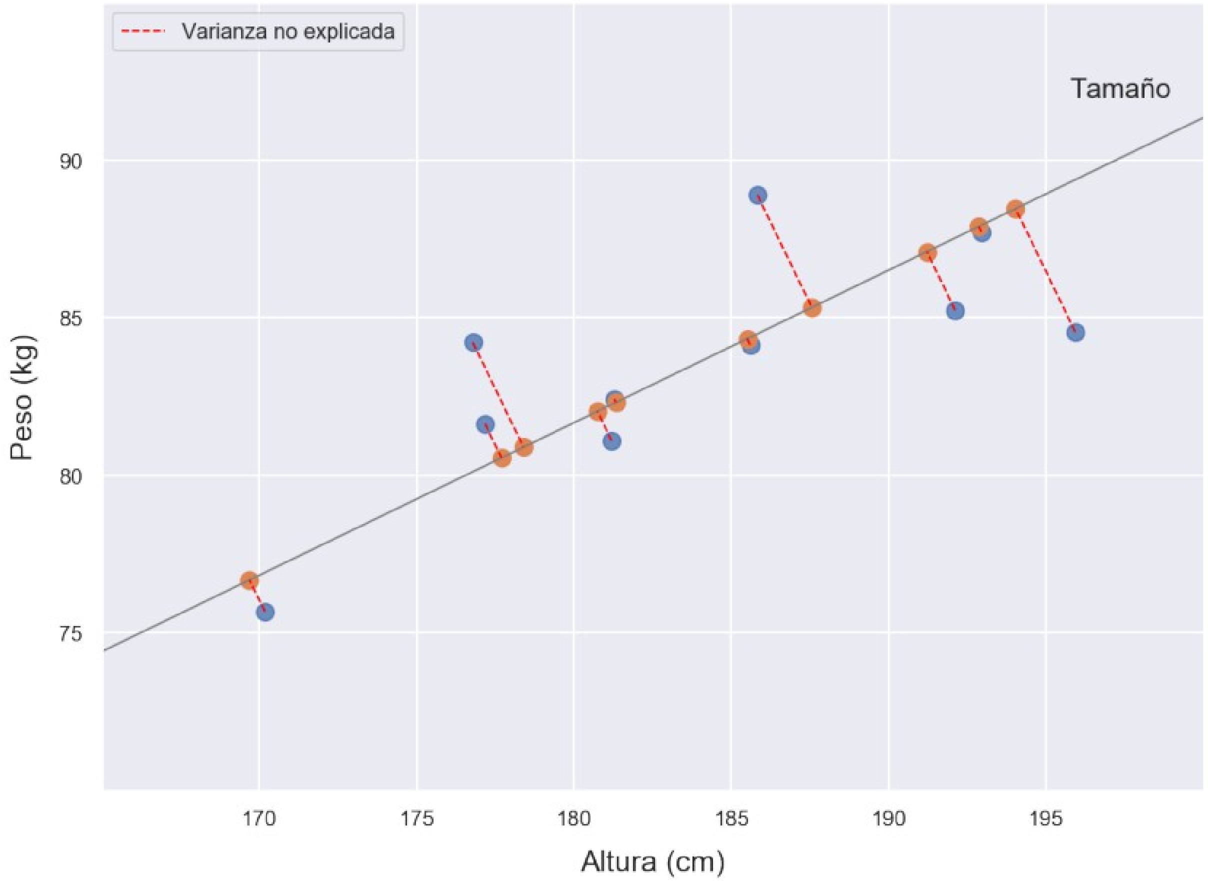
- El objetivo es encontrar **la componente que maximice la varianza** retenida (**minimice** la **no** retenida).



La varianza (líneas en rojo) van disminuyendo…

2. Datos fuertemente correlacionados
Ejemplo

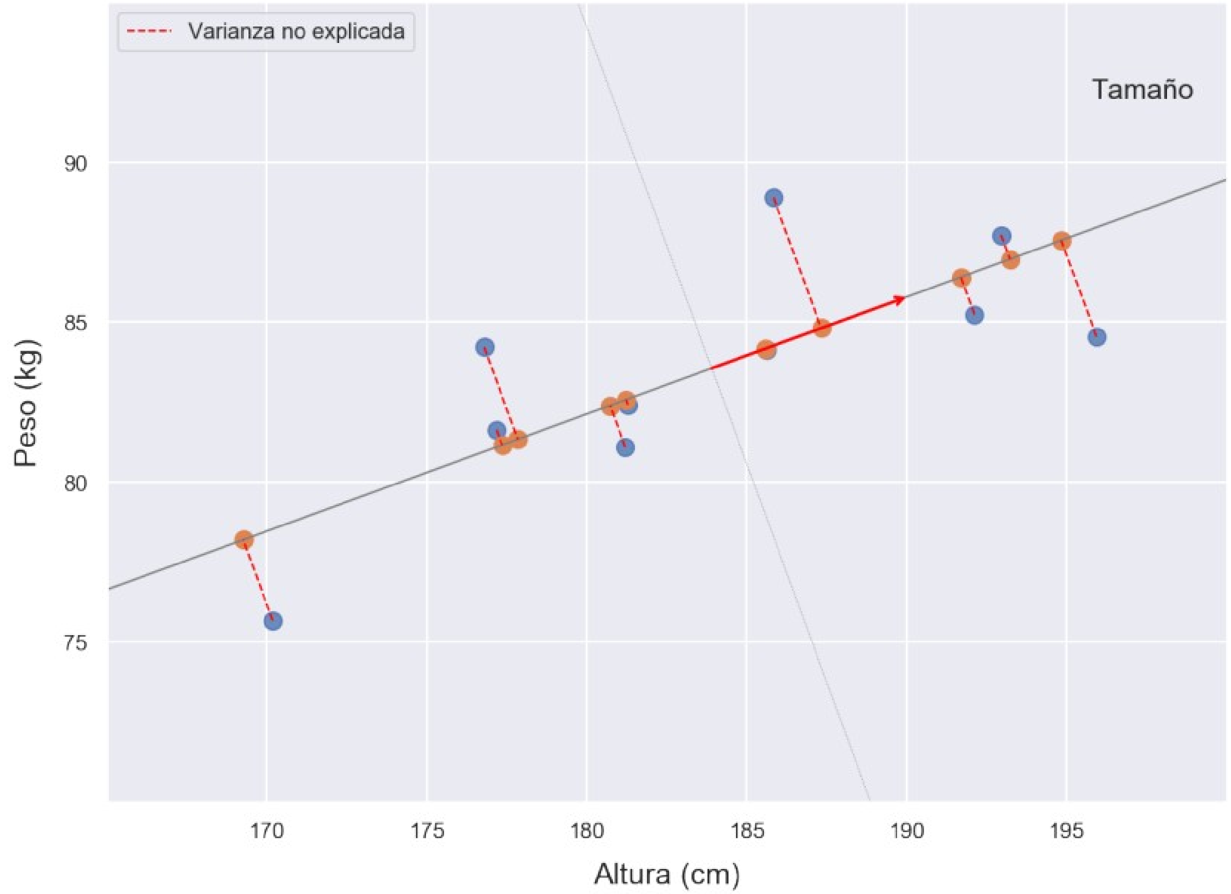
- El objetivo es encontrar **la componente que maximice la varianza** retenida (**minimice** la **no** retenida).



La varianza (líneas en rojo) van disminuyendo…

2. Datos fuertemente correlacionados
Ejemplo

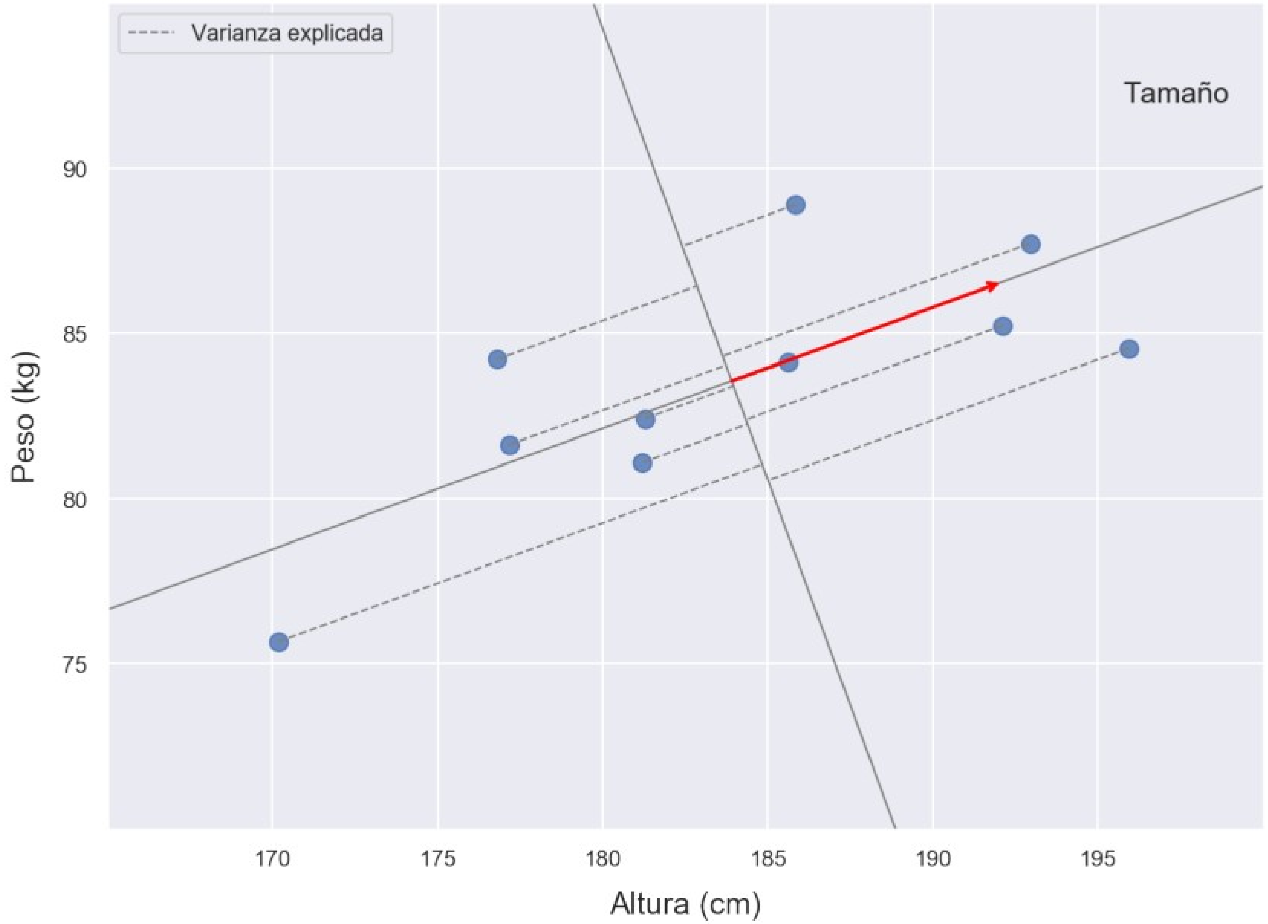
- El objetivo es encontrar **la componente que maximice la varianza** retenida (**minimice** la **no** retenida).



Hasta llegar a este.
Se ha elegido este vector porque maximiza la varianza retenida y miniza la no retenida.
Se le llama Componente Principal

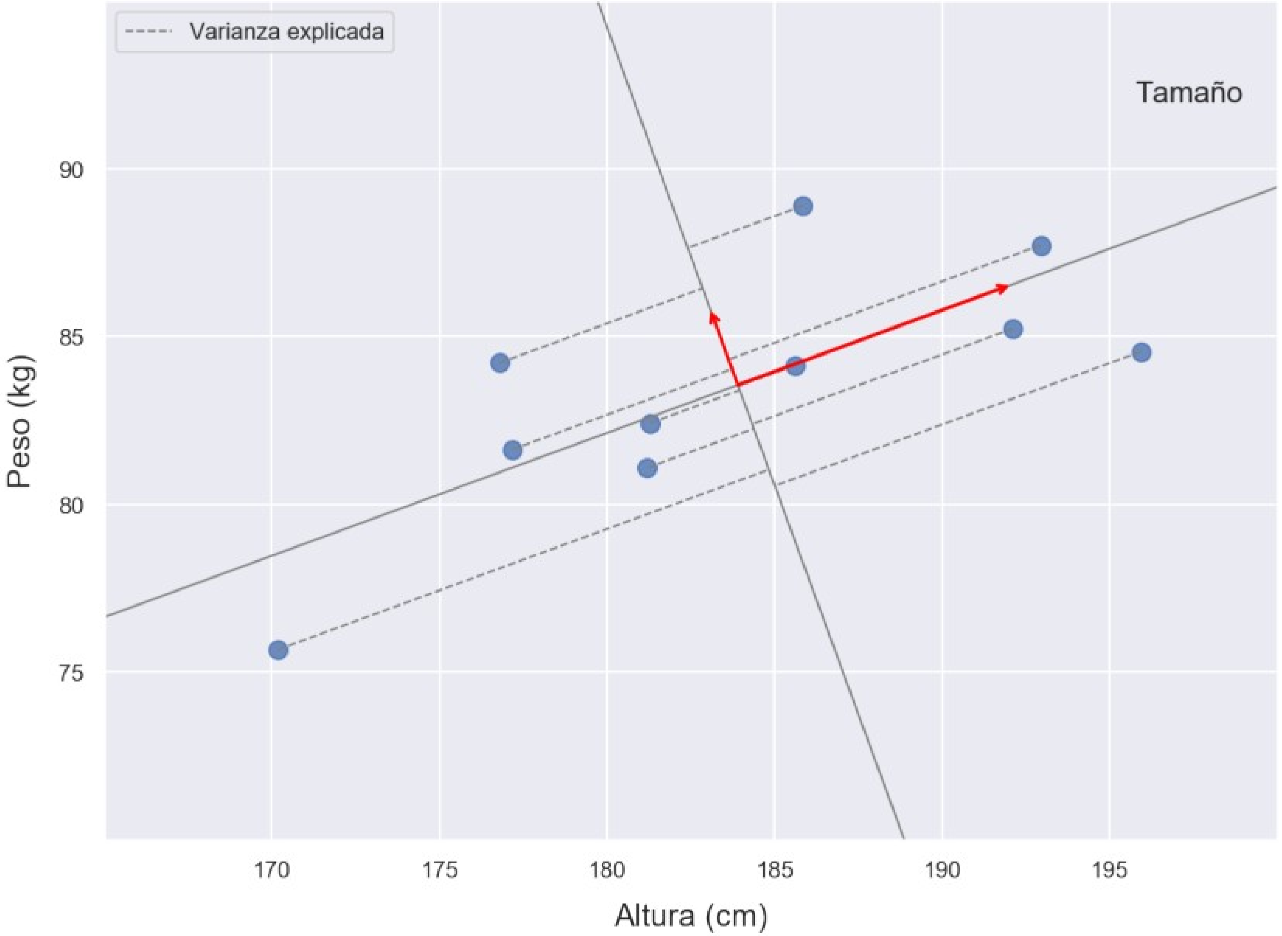
2. Datos fuertemente correlacionados
Ejemplo

- Se pueden definir componentes ortogonales a las existentes, tantas como características existen originalmente. Entre ellas, **retienen toda la varianza**.



2. Datos fuertemente correlacionados
Ejemplo

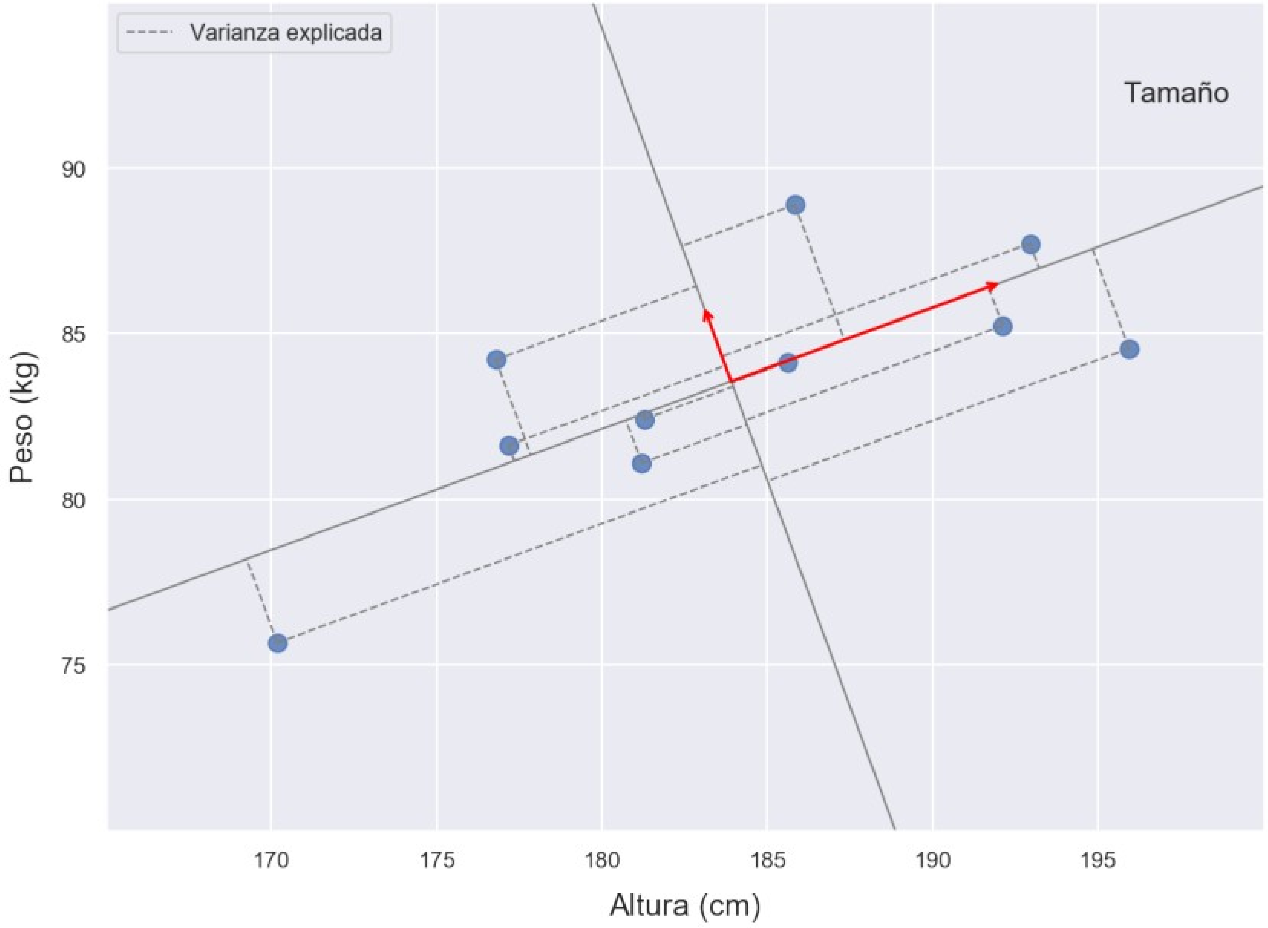
- Se pueden definir componentes ortogonales a las existentes, tantas como características existen originalmente. Entre ellas, **retienen toda la varianza**.



Este vector tira otro componente que uniformiza a los datos originales.

2. Datos fuertemente correlacionados
Ejemplo

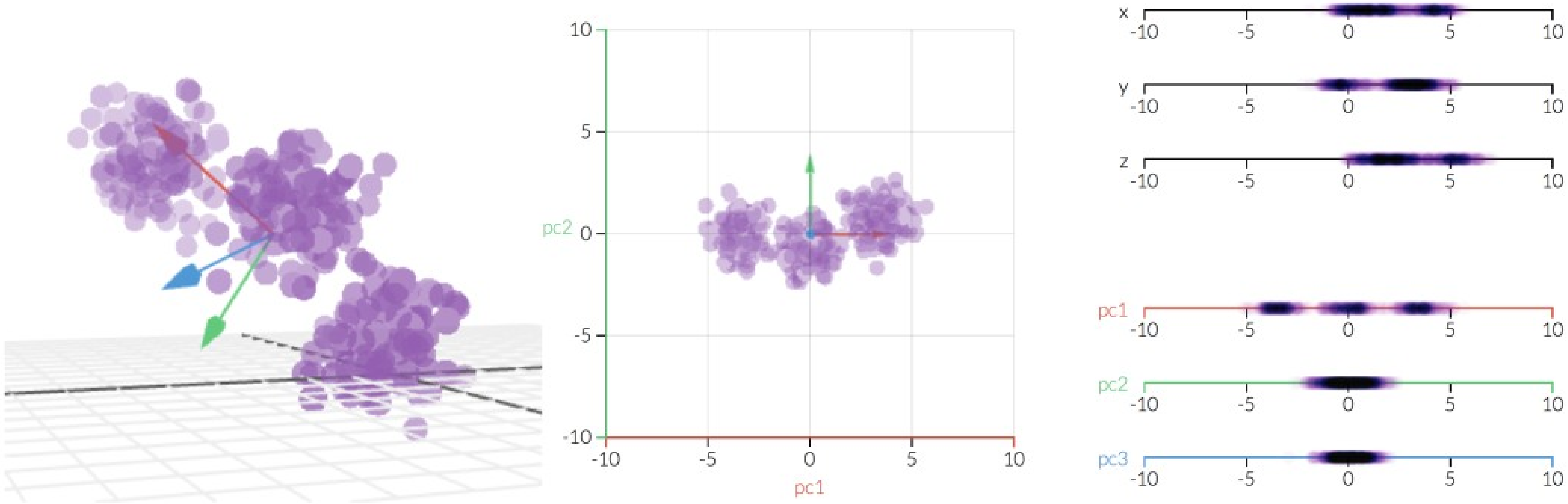
- Se pueden definir componentes ortogonales a las existentes, tantas como características existen originalmente. Entre ellas, **retienen toda la varianza**.



La idea es tener k < n, siendo k los componentes y n características

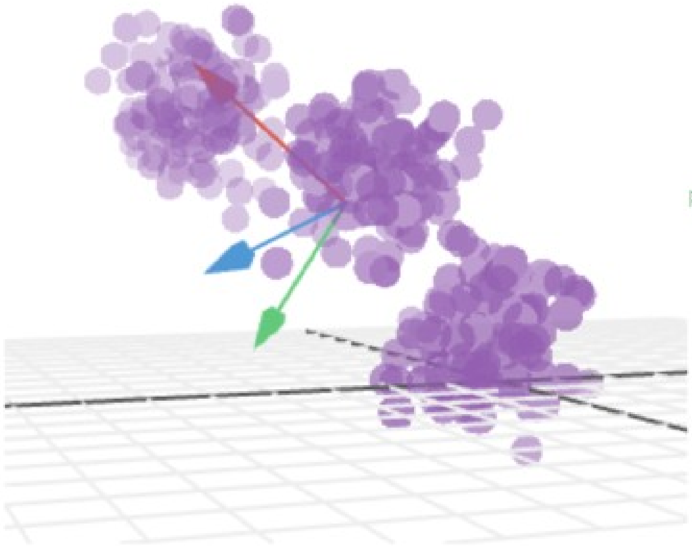
2. Datos tridimensionales
Ejemplo

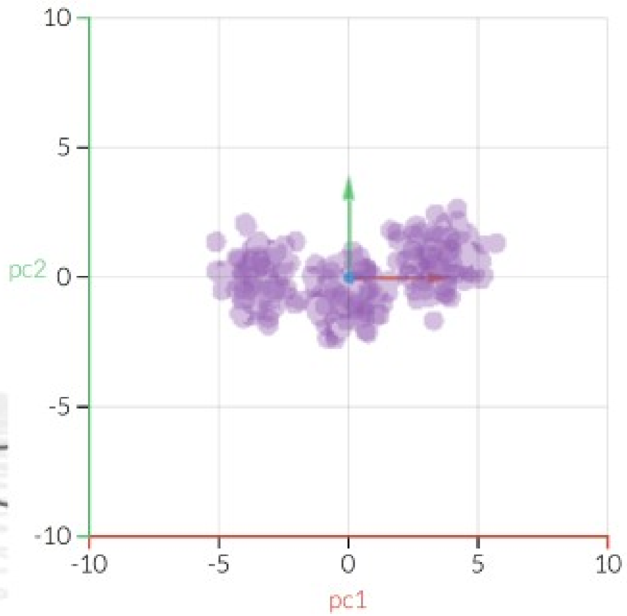
- Este ejemplo (animado e interactivo) se puede ver aquí.

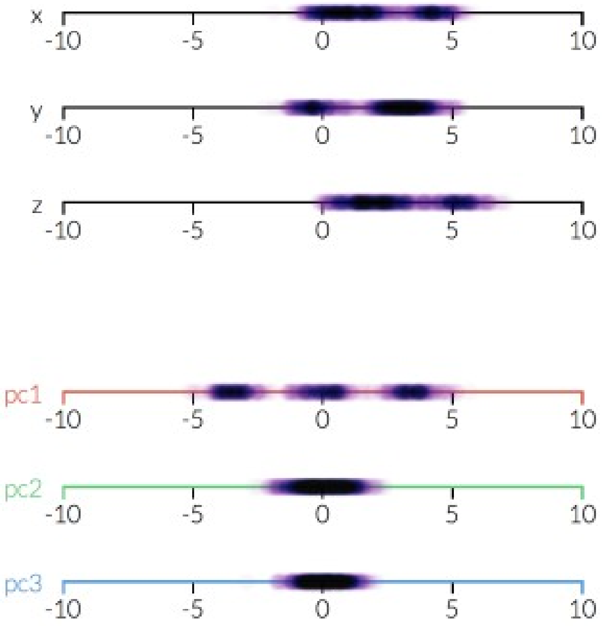


2. Datos tridimensionales
Ejemplo

- Este ejemplo (animado e interactivo) se puede ver aquí.
  - La componente roja es la que tiene más varianza.
  - La componente roja y verde representa a los datos en dos dimensiones.







Smart City

LAB CTIC UNI

Principal Components Analysis

1. PCA

- Dado un conjunto de datos $X \in R^{m \times n}$, análisis de componentes principales consiste en obtener un conjunto de k componentes $C \in R^{k \times n}$ que retengan la mayor parte de la varianza de los datos.
  - En el ejemplo anterior, $k = 1$ y $C \in R^{1 \times 2}$ (C es el vector que define la componente que representa el tamaño).
- PCA es útil cuando existen relaciones de **dependencia lineal** en los datos.
- Se utiliza para **reducir el número de características** original:
  - De cara a la exploración (características subyacentes)
  - Para almacenar los datos (con pérdidas) en menos espacio.
  - Para mejorar la eficiencia de algoritmos de aprendizaje supervisado.
  - Visualización (2 o 3 componentes).
  - etc.

2. PCA
scikit-learn

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$n = 4$

$X \in \mathbb{R}^{5\times 4}$

2. PCA
scikit-learn

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$n = 4$

$X \in \mathbb{R}^{5\times 4}$

- Si $k<n$, PCA devolvería las k primeras componentes (k primeras filas de C).
- **Importante**: Los algoritmos devuelven las componentes ordenadas de **mayor a menor varianza retenida**.

2. PCA
scikit-learn

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$n = 4$

$X \in \mathbb{R}^{5\times 4}$

- Si $k<n$, PCA devolvería las k primeras componentes (k primeras filas de C).
- **Importante**: Los algoritmos devuelven las componentes ordenadas de **mayor a menor varianza retenida**.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=4)
pca.fit(X)

C = pca.components_
>>> C = [[ 0.52 -0.49 -0.54  0.46]
         [ 0.53 -0.46  0.5  -0.5 ]
         [ 0.47  0.56 -0.47 -0.49]
         [-0.48 -0.48 -0.49 -0.55]]

pca.explained_variance_ratio_
>>> [0.9  0.08 0.02 0.  ]
```

3. Proyección de los datos
Dos Componentes

- Dadas la matriz de datos $X \varepsilon R^{m\times n}$, y las componentes $C \varepsilon R^{k\times k}$, la transformación consiste en obtener una nueva matriz de datos $X_{(pc)} \varepsilon R^{m\times k}$. Es decir, con k columnas.

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$$n = 4, k = 2$$

$$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}$$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

✎ La proyección se calcula como $X_{(pc)} = (X - \bar{\mathbf{x}})C^T$

```
X_pc = pca.transform(X)

>>> X_pc = [[ 7.67  2.01]
            [ 7.55 -2.06]
            ...
```

$$X_{(pc)} = (X - \bar{\mathbf{x}})C^T = \begin{bmatrix} 5.4 & -4.2 & -3.6 & 2 \\ 2.4 & -3.2 & -4.6 & 5 \\ -2.6 & 3.8 & 1.4 & -2 \\ -1.6 & 0.8 & 4.4 & -3 \\ -3.6 & 2.8 & 2.4 & -2 \end{bmatrix} \begin{bmatrix} 0.52 & 0.53 \\ -0.49 & -0.46 \\ -0.54 & 0.5 \\ 0.46 & -0.5 \end{bmatrix} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

3. Proyección de los datos
Dos Componentes

- Dadas la matriz de datos $X \varepsilon R^{m \times n}$, y las componentes $C \varepsilon R^{k \times k}$, la transformación consiste en obtener una nueva matriz de datos $X_{(pc)} \varepsilon R^{m \times k}$. Es decir, con k columnas.

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$$n = 4, k = 2$$

$$X \in \mathbb{R}^{5 \times 4}, C \in \mathbb{R}^{2 \times 4}$$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

```
X_pc = pca.transform(X)

>>> X_pc = [[ 7.67  2.01]
            [ 7.55 -2.06]
            ...
```

✎ La proyección se calcula como $X_{(pc)} = (X - \bar{\mathbf{x}})C^T$

Fijaros qeu tenemos dos grupos, por ejemplo a Alicia le gusta Falafel y Cereales y no le gusta kebab y shushi

$$X_{(pc)} = (X - \bar{\mathbf{x}})C^T = \begin{bmatrix} 5.4 & -4.2 & -3.6 & 2 \\ 2.4 & -3.2 & -4.6 & 5 \\ -2.6 & 3.8 & 1.4 & -2 \\ -1.6 & 0.8 & 4.4 & -3 \\ -3.6 & 2.8 & 2.4 & -2 \end{bmatrix} \begin{bmatrix} 0.52 & 0.53 \\ -0.49 & -0.46 \\ -0.54 & 0.5 \\ 0.46 & -0.5 \end{bmatrix} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

3. Proyección de los datos
Dos Componentes

- Dadas la matriz de datos $X \,\varepsilon\, R^{m\times n}$, y las componentes $C \,\varepsilon\, R^{k\times k}$, la transformación consiste en obtener una nueva matriz de datos $X_{(pc)} \,\varepsilon\, R^{m\times k}$. Es decir, con k columnas.

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$$n = 4, k = 2$$

$$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}$$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

C1 es característica 1 y 4 positivo y (comida vegetariana) y 2 y 3 negativos (no vegetariana). Representa el gusto por la comida vegetariana

✎ La proyección se calcula como $X_{(pc)} = (X - \bar{\mathbf{x}})C^T$

```
X_pc = pca.transform(X)

>>> X_pc = [[ 7.67  2.01]
           [ 7.55 -2.06]
           ...
```

$$X_{(pc)} = (X - \bar{\mathbf{x}})C^T = \begin{bmatrix} 5.4 & -4.2 & -3.6 & 2 \\ 2.4 & -3.2 & -4.6 & 5 \\ -2.6 & 3.8 & 1.4 & -2 \\ -1.6 & 0.8 & 4.4 & -3 \\ -3.6 & 2.8 & 2.4 & -2 \end{bmatrix} \begin{bmatrix} 0.52 & 0.53 \\ -0.49 & -0.46 \\ -0.54 & 0.5 \\ 0.46 & -0.5 \end{bmatrix} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

3. Proyección de los datos
Dos Componentes

- Dadas la matriz de datos $X \; \varepsilon \; R^{m \times n}$, y las componentes $C \; \varepsilon \; R^{k \times k}$, la transformación consiste en obtener una nueva matriz de datos $X_{(pc)} \; \varepsilon \; R^{m \times k}$. Es decir, con k columnas.

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$$n = 4, k = 2$$

$$X \in \mathbb{R}^{5 \times 4}, C \in \mathbb{R}^{2 \times 4}$$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

C2 es característica 1 y 3 positivo (comida semi-sana) y 2 y 4 negativos (no sana). Representa (en cierto modo) el gusto por la comida sana

✎ La proyección se calcula como $X_{(pc)} = (X - \bar{\mathbf{x}})C^T$

```
X_pc = pca.transform(X)

>>> X_pc = [[ 7.67  2.01]
            [ 7.55 -2.06]
            ...
```

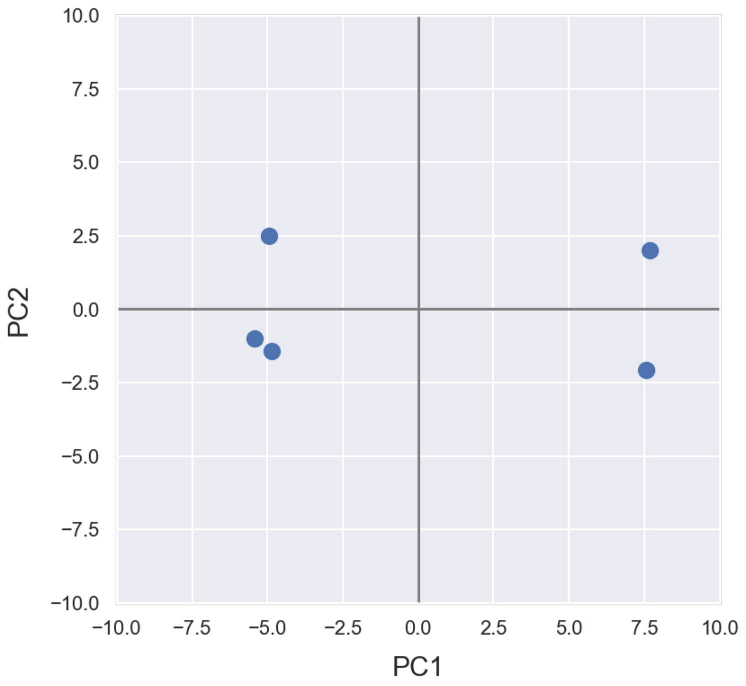
$$X_{(pc)} = (X - \bar{\mathbf{x}})C^T = \begin{bmatrix} 5.4 & -4.2 & -3.6 & 2 \\ 2.4 & -3.2 & -4.6 & 5 \\ -2.6 & 3.8 & 1.4 & -2 \\ -1.6 & 0.8 & 4.4 & -3 \\ -3.6 & 2.8 & 2.4 & -2 \end{bmatrix} \begin{bmatrix} 0.52 & 0.53 \\ -0.49 & -0.46 \\ -0.54 & 0.5 \\ 0.46 & -0.5 \end{bmatrix} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

4. Visualización

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

- La reducción a dos (o 3) componentes permite visualizar datos multidimensionales.
- Podemos ver en esta representación los gustos de cada instancia según los componentes reconstruidos en 2D.



5. Interpretación

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$X \in \mathbb{R}^{5\times4}, C \in \mathbb{R}^{2\times4}, X_{(pc)} \in \mathbb{R}^{5\times2}$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

- Los elementos (*loadings*) de una componente reflejan la relación con la variable correspondiente. En algunos casos, es útil la interpretación.
  - Ej: la primera componente refleja la preferencia por comida vegetariana (X_1 y X_4).
- Cada elemento $x^{(i)}$ se puede construir como una combinación lineal de las k componentes C. $x^{(i)}_{(pc)}$ representa el peso de cada componente, y $x^{(i)}_{(app)}$ su reconstrucción. Ejemplo:

$$x^{(1)}_{app} = \bar{\mathbf{x}} + 7.67 \cdot \begin{bmatrix} 0.52 \\ -0.49 \\ -0.54 \\ 0.46 \end{bmatrix} + 2.01 \cdot \begin{bmatrix} 0.53 \\ -0.46 \\ 0.5 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 9.62 \\ 0.54 \\ 2.48 \\ 7.5 \end{bmatrix}$$

5. Interpretación

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}, X_{(pc)} \in \mathbb{R}^{5\times 2}$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

X_{PC1} vegetal

X_{PC2} Sano

- Los elementos (*loadings*) de una componente reflejan la relación con la variable correspondiente. En algunos casos, es útil la interpretación.
  - Ej: la primera componente refleja la preferencia por comida vegetariana (X_1 y X_4).
- Cada elemento $x^{(i)}$ se puede construir como una combinación lineal de las k componentes C. $x^{(i)}_{(pc)}$ representa el peso de cada componente, y $x^{(i)}_{(app)}$ su reconstrucción. Ejemplo:

$$x^{(1)}_{app} = \bar{\mathbf{x}} + 7.67 \cdot \begin{bmatrix} 0.52 \\ -0.49 \\ -0.54 \\ 0.46 \end{bmatrix} + 2.01 \cdot \begin{bmatrix} 0.53 \\ -0.46 \\ 0.5 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 9.62 \\ 0.54 \\ 2.48 \\ 7.5 \end{bmatrix}$$

5. Interpretación

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}, X_{(pc)} \in \mathbb{R}^{5\times 2}$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

- Los elementos (*loadings*) de una componente reflejan la relación con la variable correspondiente. En algunos casos, es útil la interpretación.
  - Ej: la primera componente refleja la preferencia por comida vegetariana (X_1 y X_4).
- Cada elemento $x^{(i)}$ se puede construir como una combinación lineal de las k componentes C. $x^{(i)}_{(pc)}$ representa el peso de cada componente, y $x^{(i)}_{(app)}$ su reconstrucción. Ejemplo:

$$x^{(1)}_{app} = \bar{\mathbf{x}} + 7.67 \cdot \begin{bmatrix} 0.52 \\ -0.49 \\ -0.54 \\ 0.46 \end{bmatrix} + 2.01 \cdot \begin{bmatrix} 0.53 \\ -0.46 \\ 0.5 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 9.62 \\ 0.54 \\ 2.48 \\ 7.5 \end{bmatrix}$$

No salen los datos exactos porque se han utilizado dos componentes y queda una poca varianza sin explicar (pierde información) pero son muy aproximados

6. Reconstrucción a partir de las componentes

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}, X_{(pc)} \in \mathbb{R}^{5\times 2}$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

✎ La reconstrucción, $X_{(app)} \in \mathbb{R}^{5\times 4}$ se obtiene como:

$$X_{(app)} = \bar{\mathbf{x}} + X_{(pc)}C$$

```
X_app = pca.inverse_transform(X_pc)

>>> X_app = [[9.62 0.54 2.48 7.5 ]
             [7.4  2.49 0.51 9.49]
             ...
```

$$X_{(app)} = \bar{\mathbf{x}} + \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix} \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix} = \begin{bmatrix} 9.6 & 0.5 & 2.5 & 7.5 \\ 7.4 & 2.5 & 0.5 & 9.5 \\ 1.3 & 8.2 & 7.5 & 3.5 \\ 3.4 & 6.4 & 9.5 & 1.5 \\ 1.3 & 8.3 & 8.0 & 3.0 \end{bmatrix}$$

6. Reconstrucción a partir de las componentes

	Falafel	Kebab	Shushi	Cereales
Alicia	10	1	2	7
Bernardo	7	2	1	10
Carmen	2	9	7	3
Diego	3	6	10	2
Elena	1	8	8	3

$X \in \mathbb{R}^{5\times 4}, C \in \mathbb{R}^{2\times 4}, X_{(pc)} \in \mathbb{R}^{5\times 2}$

$$C = \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix}$$

$$X_{(pc)} = \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix}$$

✎ La reconstrucción, $X_{(app)} \in \mathbb{R}^{5\times 4}$ se obtiene como:

$$X_{(app)} = \bar{\mathbf{x}} + X_{(pc)} C$$

```
X_app = pca.inverse_transform(X_pc)

>>> X_app = [[9.62 0.54 2.48 7.5 ]
             [7.4  2.49 0.51 9.49]
             ...
```

$$X_{(app)} = \bar{\mathbf{x}} + \begin{bmatrix} 7.67 & 2.01 \\ 7.55 & -2.06 \\ -4.85 & -1.44 \\ -4.95 & 2.49 \\ -5.42 & -1. \end{bmatrix} \begin{bmatrix} 0.52 & -0.49 & -0.54 & 0.46 \\ 0.53 & -0.46 & 0.5 & -0.5 \end{bmatrix} = \begin{bmatrix} 9.6 & 0.5 & 2.5 & 7.5 \\ 7.4 & 2.5 & 0.5 & 9.5 \\ 1.3 & 8.2 & 7.5 & 3.5 \\ 3.4 & 6.4 & 9.5 & 1.5 \\ 1.3 & 8.3 & 8.0 & 3.0 \end{bmatrix}$$

Smart City

LAB CTIC UNI

Varianza y selección de k

1. Varianza

- La varianza de los datos se computa como la suma de las varianzas de cada variable:

$$var(X) = \sum_{j=1}^{n} var(X_j)$$

- La varianza **explicada** por las componentes se calcula como las varianzas las características correspondientes:
  - Las componentes principales corresponden con autovectores de la matriz de covarianza de los datos. Cada $X_{(pc),\,j}$ es el autovalor correspondiente al vector j.

$$var(X_{(pc)}) = \sum_{j=1}^{k} var(X_{(pc),j})$$

- Se cumple que, si $k = n \rightarrow var(X) = var(X_{(pc)})$
- Interesa la fracción de varianza que explica cada componente C_j, $\frac{var(X_{(pc),j})}{var(X_{(pc)})}$

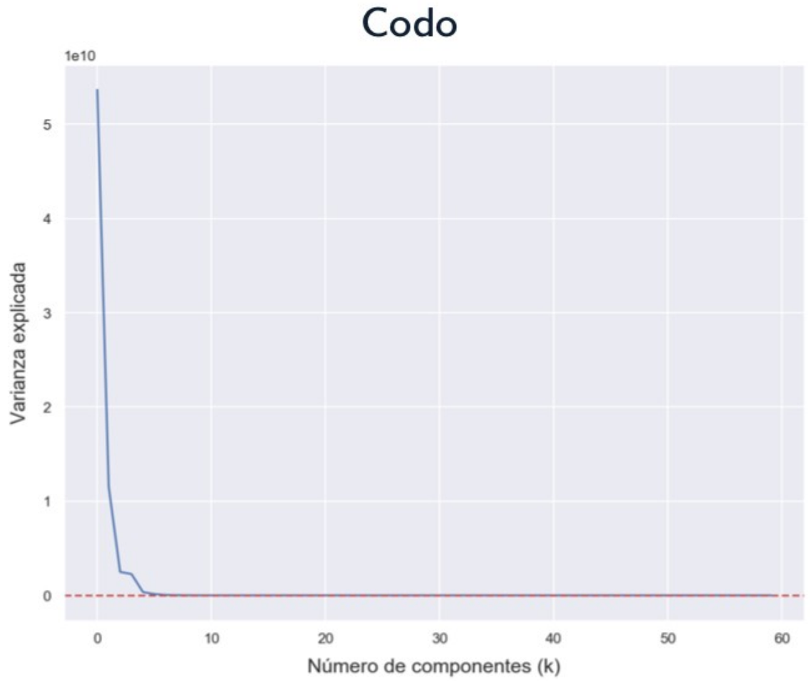
```
pca.explained_variance_

>>> [48.34709495  4.38266532]
```

```
pca.explained_variance_ratio_

>>> [0.89864489 0.08146218]
```

```
pca.explained_variance_ratio_

>>> [0.89864489 0.98010707]
```

2. Elección de k

- El número de componentes, k, determina la precisión con la que las componentes representan los datos originales (la varianza explicada).
- Existen dos métodos principales para seleccionar el valor de k:



- Un tercer método, denominado **Kaiser-Guttman** consiste en escoger las componentes que expliquen una varianza mayor que uno (con datos estandarizados).

Smart City

LAB CTIC UNI

PCA como algoritmos de aprendizaje no supervisado

1. Hipótesis y objetivos

☞ Nota

Es **necesario** que los datos están normalizados a media cero , es decir, $\bar{\mathbf{x}} = [0, 0, ...]$. Se recomienda también que las, varianzas sean uno.

- Dados el conjunto de datos X, y el conjunto de componentes C:

$$X_{(pc)} = XC^T \qquad X_{(app)} = X_{(pc)}C$$

- Por tanto, la reconstrucción de los datos puede expresarse como:

$$X_{(app)} = (XC^T)C$$

- Se define la función de hipótesis como:

$$h_\theta(x) = xC^T C, \quad x \in \mathbb{R}^n, \theta = \{C \in \mathbb{R}^{k \times n}\}$$

- El coste se expresa como:

$$J(\theta) = \|h_\theta(x) - x\|^2 = \| xC^T C - x\|^2$$

- Por tanto, el objetivo es:

$$\underset{C}{argmin} \sum_{i=1}^{m} \| xC^T C - x\|^2$$

Que según los componentes la pérdida de información (varianza) sea mínima

2. Algoritmos

- La mayoría de las implementaciones de PCA se basan en descomposición en valores singulares (SVD).
- Una matriz $X \, \varepsilon \, R^{m \times n}$ puede descomponerse como:

$$X = UsV^T, \quad U \in \mathbb{R}^{m \times n}, s \in \mathbb{R}^n, V \in \mathbb{R}^{n \times N}$$

- **Método 1**: dados X y k, se pueden obtener C y $X_{(pc)}$ como:

$$C = V^T_{:,1:k} \quad X_{(pc)} = U_{:,1:k} s_{:k}$$

- **Método 2**: dados X y k, se calcula la matriz de covarianzas de X como:

$$\Sigma = \frac{1}{m} X^T X$$

Mediante SVD se descompone $\Sigma = UsV^T$, de modo que puede obtenerse C como:

$$C = U_{:,1:k}$$

C está formada por los k autovectores (*eigenvectors*) de la matriz de covarianzas con mayor autovalor (*eigenvalue*).

¡GRACIAS!

Smart City

LAB CTIC UNI

Dr. Manuel Castillo-Cara
Machine Learning con Python
Web: www.smartcityperu.org