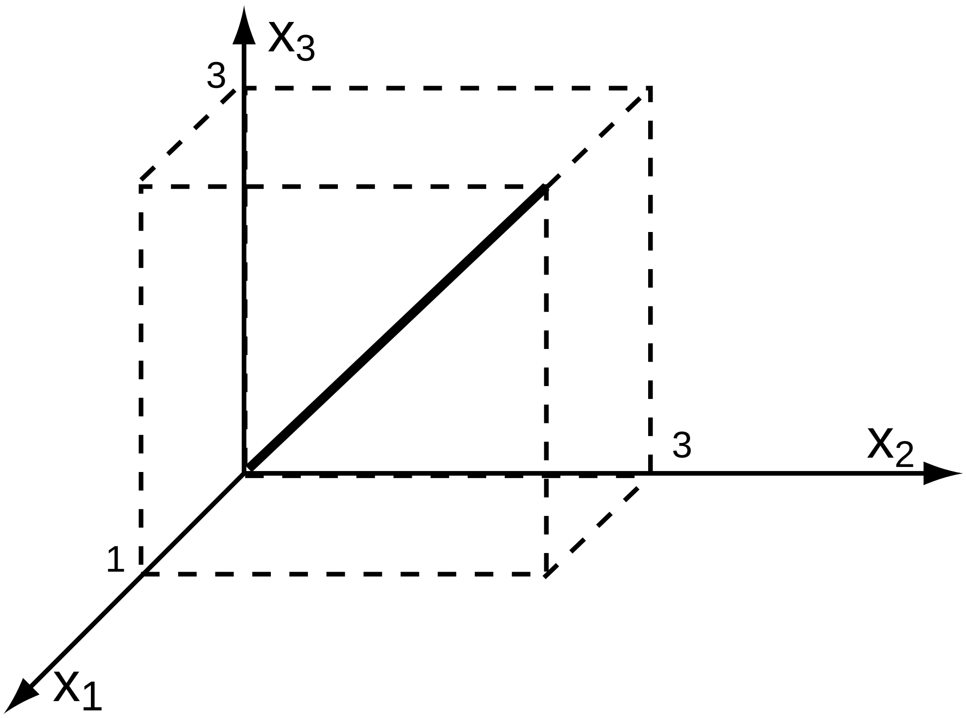
Statistics 5041
6. Geometry

Gary W. Oehlert
School of Statistics
313B Ford Hall
612-625-1557
gary@stat.umn.edu

We are going to be working with multivariate data, n cases with p variables each. For some purposes, it works best to think of n vectors of dimension p. For other purposes, we want p vectors of dimension n.

We'll need to study some geometry in multiple dimensions, ideas of length, distance, angle, and so on.

To begin, let's look at a generic vector, say $v = (1, 3, 3)'$ with dimension $k = 3$.



The *norm* or *length* of a vector v is

$$||v|| = \sqrt{< v, v >} = \sqrt{v'v} = \sqrt{\sum_{i=1}^{k} v_i^2}$$

$||v||$ is just the Euclidean distance from the point v to 0, via Pythagorean Theorem.

For scalar c, $||cv|| = |c| \ \ ||v||$.

```
Cmd> v <- vector(1,3,3)

Cmd> sqrt(sum(v^2))
(1)       4.3589

Cmd> sqrt(v'%*%v)
```

```
(1,1)        4.3589

Cmd> length(v)
(1)             3
```

`length()` is MacAnova is number of elements, not the norm of a vector.
A simple macro to automate computing the norm:

```
Cmd> norm <- macro("sqrt(sum($1^2))")
```

The first argument replaces `$1`.

```
Cmd> norm(v)
(1)       4.3589
```

Too large or too small values can lead to numerical overflow or underflow.

```
Cmd> vb <- 10^200 * v; norm(vb)
WARNING: result of arithmetic too large, set to MISSING; operation is ^ in macro norm
WARNING: MISSING values found by sum() in macro norm
(1)             0

Cmd> vc <- v / 10^200; norm(vc)
(1)             0
```

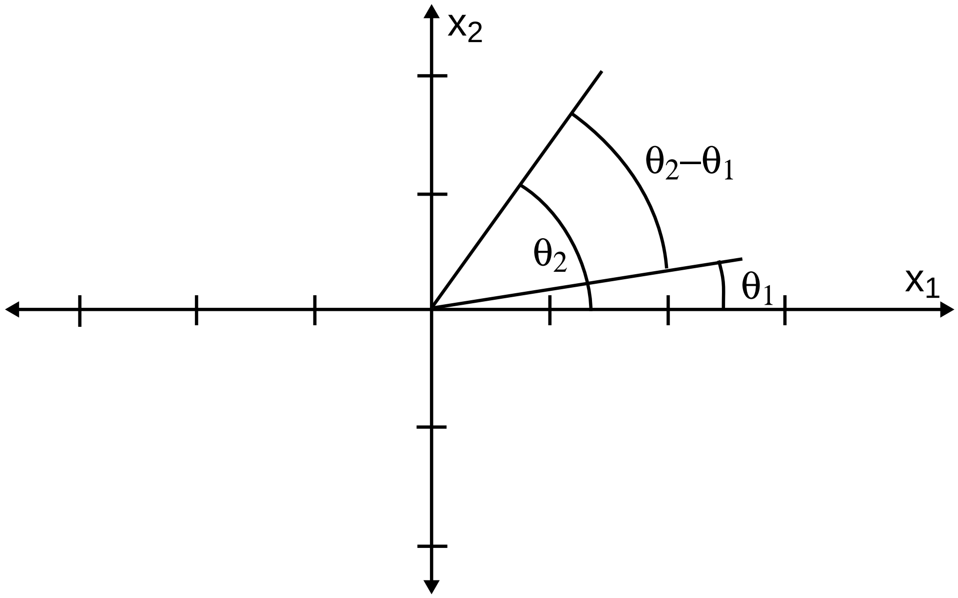
Try a more careful macro.

```
Cmd> norm <- macro("@v <- $1
@scale <- max(abs(@v))
@scale*sqrt(sum((@v/@scale)^2))",dollars:T)

Cmd> norm(vb)
(1)  4.3589e+200

Cmd> norm(vc)
(1)  4.3589e-200
```

Rescaling prevents under- or overflow when squaring.
There will be some angle between any two vectors (0 to 180 in degrees, or 0 to π in radians).



If θ is the angle between vectors v and w, then

$$\cos(\theta) = \frac{< v, w >}{||v||\ ||w||} = \frac{\sum_{i=1}^{k} v_i w_i}{\sqrt{\sum_{i=1}^{k} v_i^2\ \sum_{i=1}^{k} w_i^2}}$$

If v and w have unit length (norm equal to 1), then

$$\cos(\theta) = \ < v, w > \ = \sum_{i=1}^{k} v_i w_i$$

Vectors v and w are *orthogonal* or *perpendicular* if

$$0 = \ < v, w > \ = \sum_{i=1}^{k} v_i w_i$$

Then $\cos(\theta) = 0$ so that $\theta = \pm 90^\circ = \pm\pi/2$.

Vectors $x_1, x_2, \ldots, x_k$ are *mutually orthogonal* if and only if $< x_i, x_j >= 0$ for all i, j. In that case, $\mathbf{X} = [x_1, x_2, \ldots, x_k]$ has rank k, and $\mathbf{X'X}$ is a diagonal matrix with $< x_i, x_i >$ in the ith diagonal position.

Consider bivariate data $(x_i, y_i), i = 1, \ldots, n$, and let $\tilde{x}_i = x_i - \bar{x}$ and $\tilde{y}_i = y_i - \bar{y}$. Then the correlation between x and y is

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} x_i^2\ \sum_{i=1}^{n} y_i^2}} = \frac{< \tilde{x}, \tilde{y} >}{||\tilde{x}||\ ||\tilde{y}||} = \cos\theta$$

where θ is the angle between the deviations vectors $\tilde{x}$ and $\tilde{y}$.

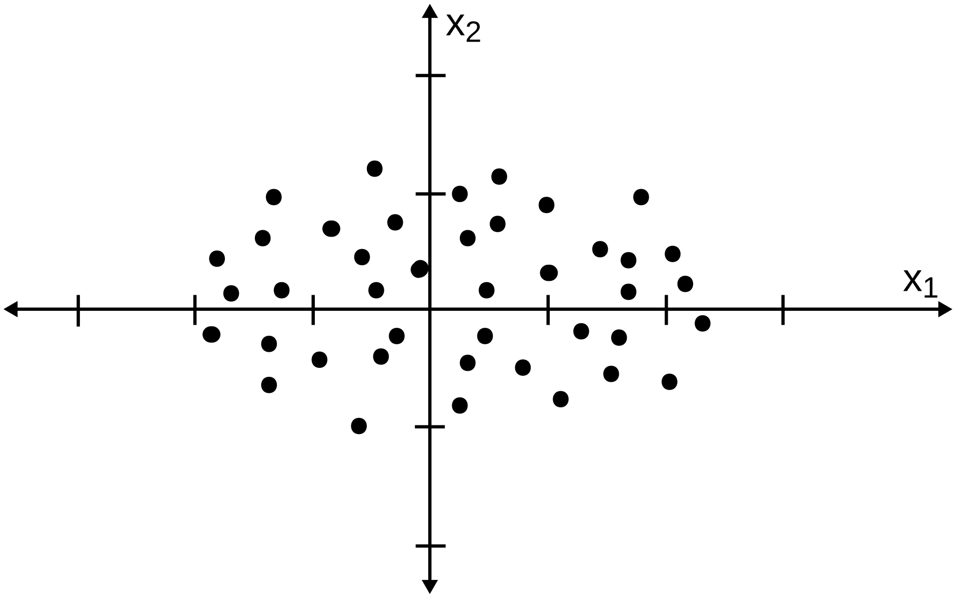
Correlation measures angle between mean-adjusted vectors.

Euclidean distance between vectors x, y, and 0 (the origin) is just Pythagorean Theorem (square root of sums of squares):

$$d(0, x) = \sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$$

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_p - y_p)^2}$$

OK for ordinary geometry, but may not be what we need for statistics.
Is Euclidean distance in the horizontal direction as "important" as Euclidean distance in the vertical direction in this point cloud?



A "statistical" distance takes into account the scale (or spread or variation) of the variables. In the preceding example, we would like to rescale to take the standard deviations into account. We would get

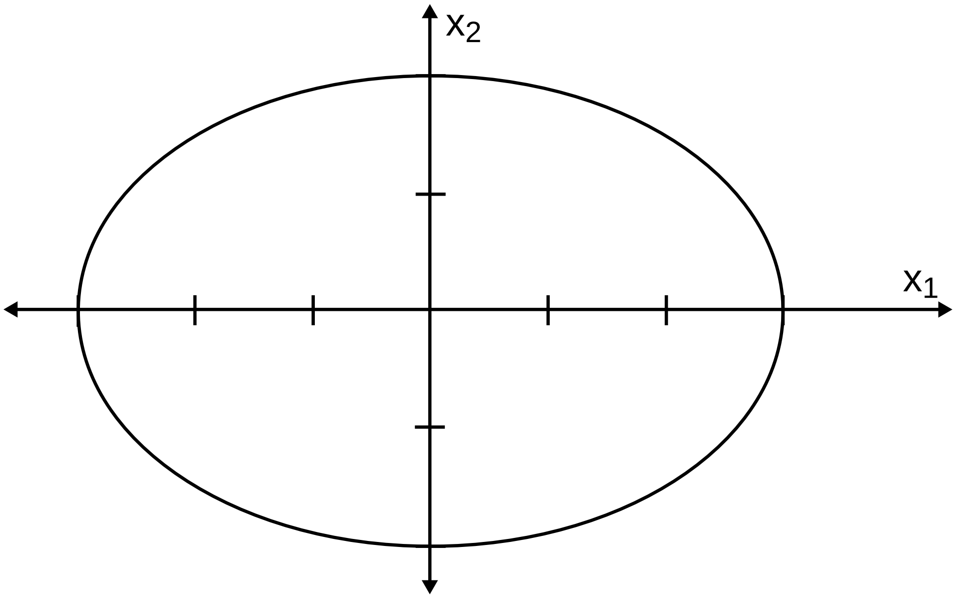
$$d(0,x) = \sqrt{\frac{x_1^2}{s_1^2} + \frac{x_2^2}{s_2^2} + \ldots + \frac{x_p^2}{s_p^2}}$$

Contours of constant distance are ellipses. The locus of points that satisfies

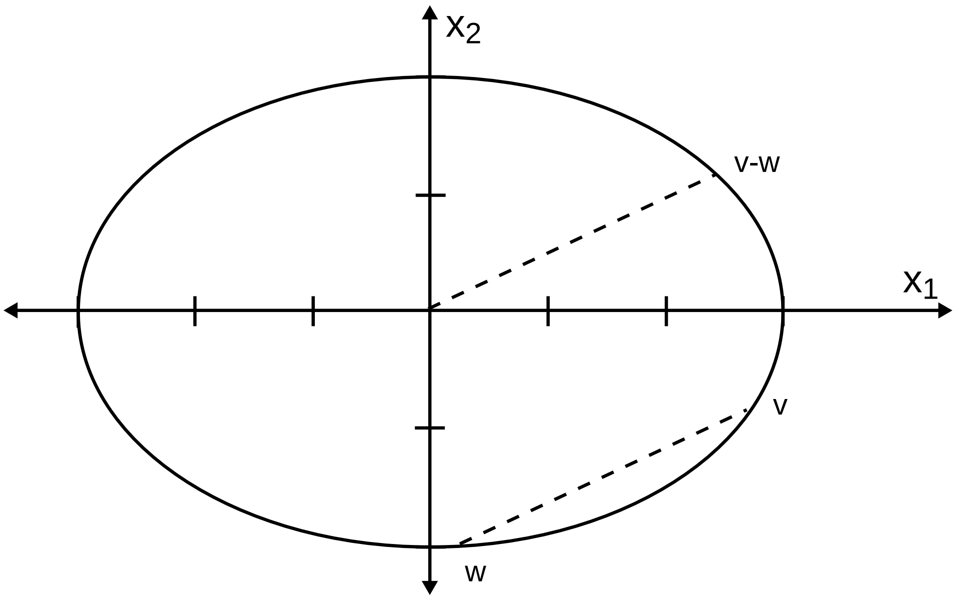
$$D = \sqrt{\frac{x_1^2}{s_1^2} + \frac{x_2^2}{s_2^2} + \ldots + \frac{x_p^2}{s_p^2}}$$

is an ellipse.
Consider

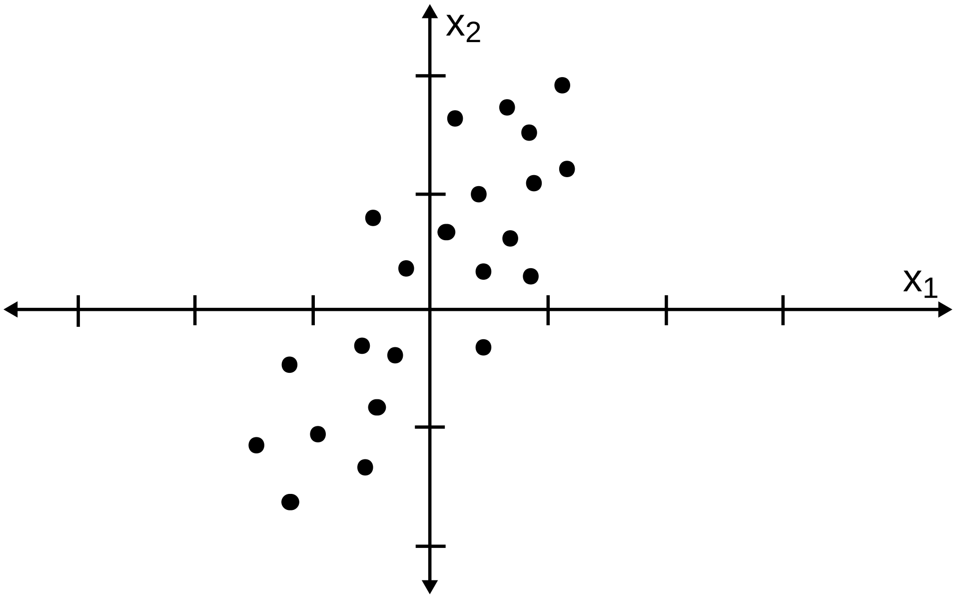
$$1 = \sqrt{\frac{x_1^2}{9} + \frac{x_2^2}{4}}$$



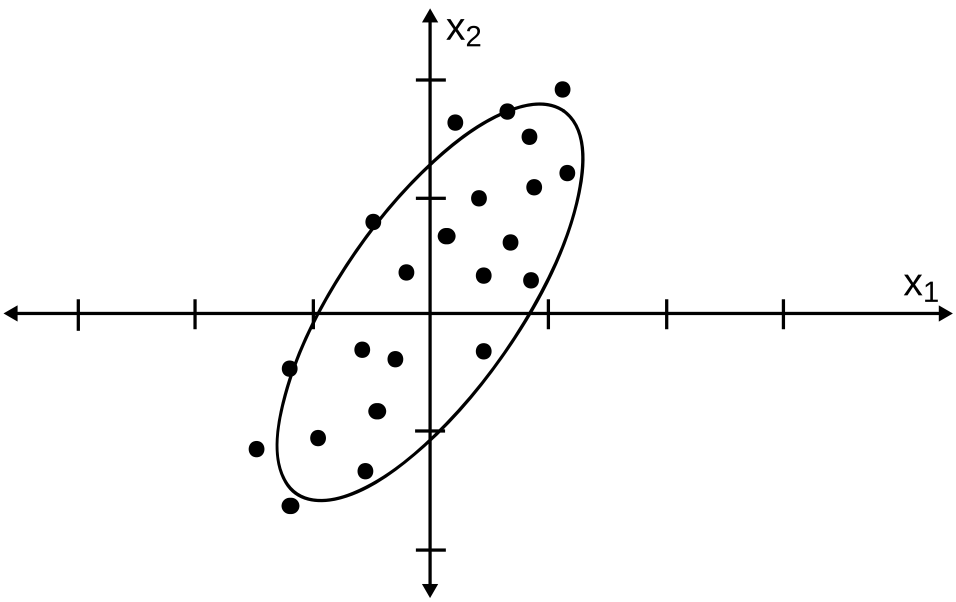
The distance from x to y is the distance of $x - y$ to 0 (or $y - x$ to 0).



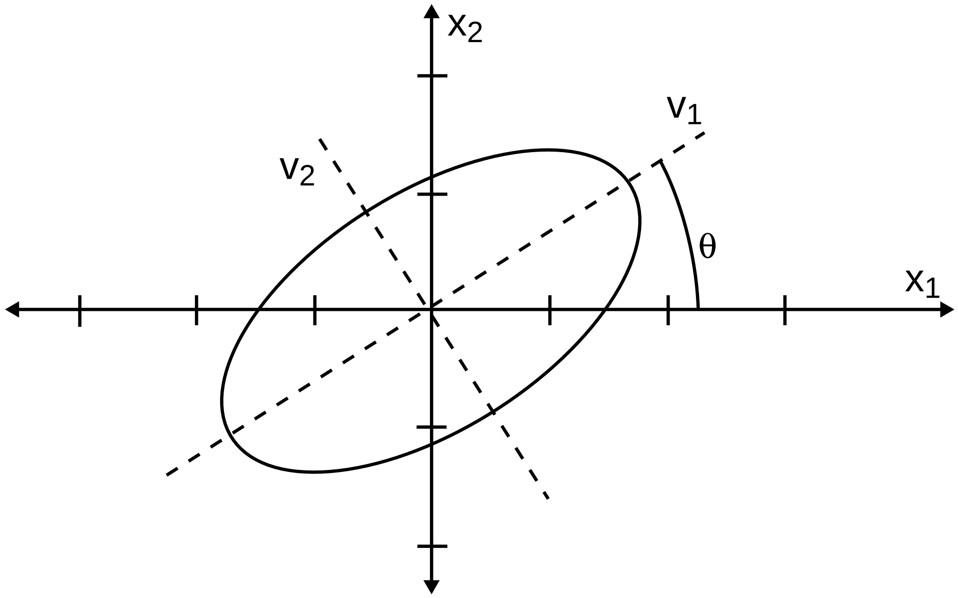
What if the cloud of points is not oriented along the axes?



The axes-oriented ellipse of constant distance is no longer appropriate, because the axes we want are tilted.



What we want are new axes v_1, v_2 rotated through an angle θ.



Then do elliptical distance on rotated axes.
For two dimensions, the rotation can be expressed:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

In p dimensions, let's just consider

$$v = \mathbf{A}x$$

Once we have v, compute squared distance by scaling the elements of v and adding the squares.

$$d(0, v) = \sqrt{\frac{v_1^2}{s_1^2} + \frac{v_2^2}{s_2^2} + \ldots + \frac{v_p^2}{s_p^2}}$$

In matrix form,

$$d(0, v)^2 = x'\mathbf{A}' \begin{bmatrix} \frac{1}{s_1^2} & 0 & \ldots & 0 \\ 0 & \frac{1}{s_2^2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \frac{1}{s_p^2} \end{bmatrix} \mathbf{A}x = x'\mathbf{B}x$$

where $\mathbf{B}$ is an $p \times p$ matrix that incorporates both the rotation and the scaling.
Let $\mathbf{X}$ by an $n \times p$ matrix of data, and let $\mathbf{S}$ be the $p \times p$ covariance matrix of the data: $\mathbf{S}_{ii} = \text{var}(\check{\mathbf{X}}_i)$ and $\mathbf{S}_{ij} = \text{covar}(\check{\mathbf{X}}_i, \check{\mathbf{X}}_j)$. Then the *Mahalanobis* distance from v to w is

$$d(v, w) = (v - w)'\mathbf{S}^{-1}(v - w)$$

That is, the magic matrix $\mathbf{B}$ is the inverse of the covariance matrix $\mathbf{S}$.
Rules for distance:

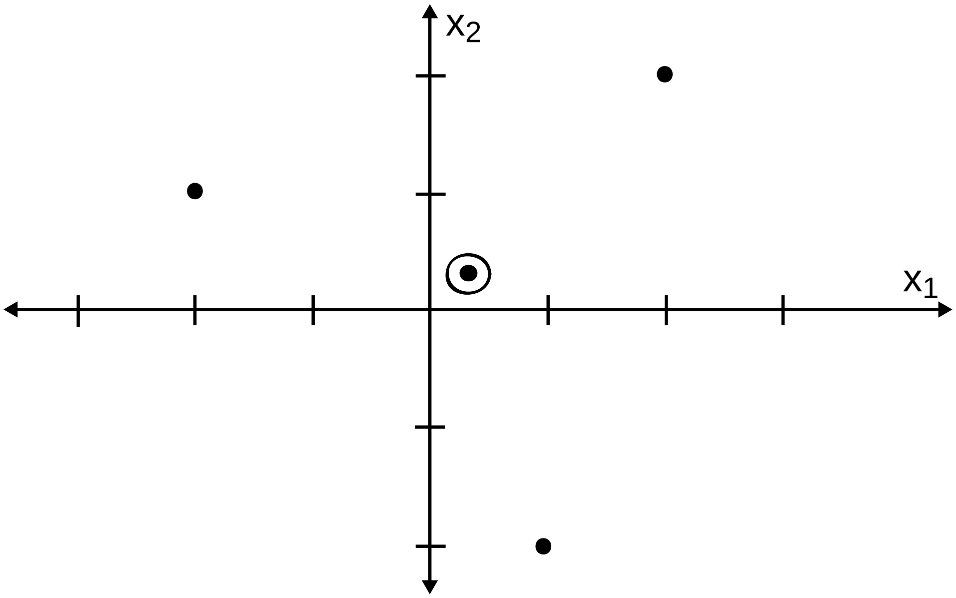
$$\begin{aligned} d(v, w) &= d(w, v) \\ d(v, w) &> 0 \text{ if } v \neq w \\ d(v, w) &= 0 \text{ if } v = w \\ d(v, w) &\leq d(v, z) + d(w, z) \end{aligned}$$

The last is called the triangle inequality.

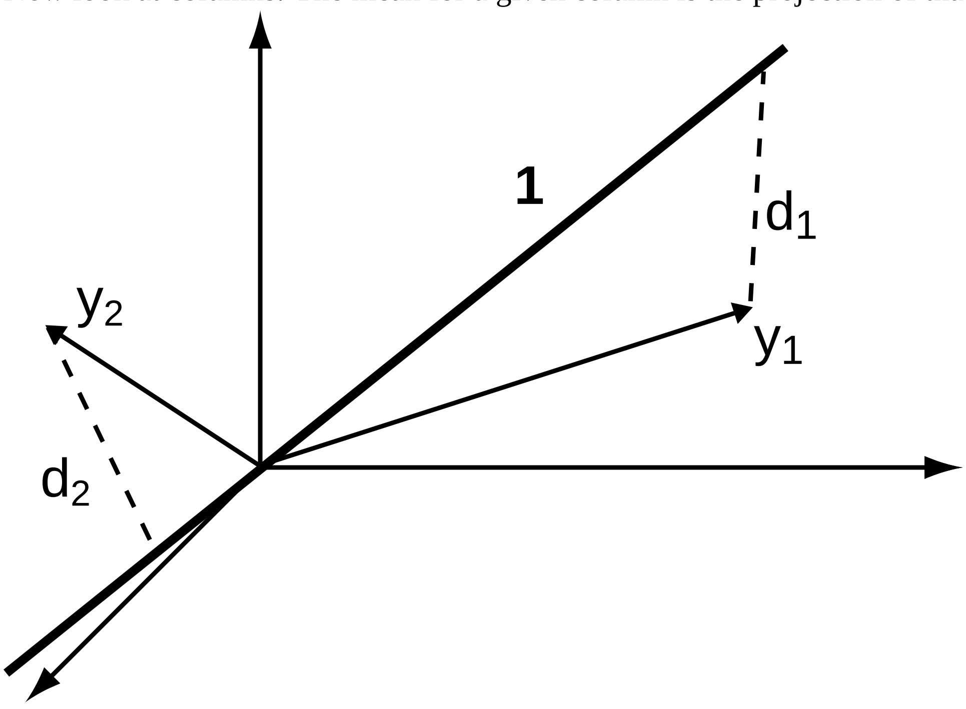
Let's go back to $\mathbf{X}$, an $n \times p$ matrix of data. We will illustrate with $n = 3$ and $p = 2$, bivariate data with three observations.

We can think of this as n vectors of dimension p (rows) or p vectors of dimension n (columns).

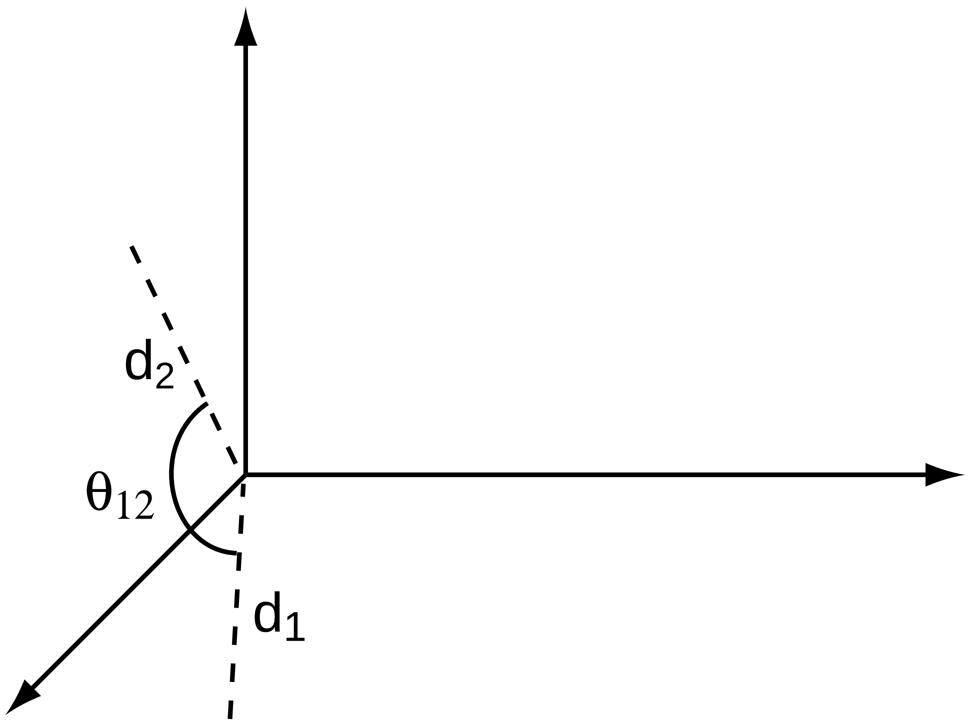
First look at rows. The p-dimensional point consisting of all the variable means is the center of mass of the data cloud.



Now look at columns. The mean for a given column is the projection of that vector onto the vector of all ones.



If we now move the deviations vectors to the origin, the cosine(s) of the angle(s) between these vectors are the correlations between the variables.


d2
θ12
d1