

w02c — Estimating a rotation: solution

With $(x', y') = R(\theta)(x, y)$:

$$x' = x \cos \theta - y \sin \theta, \quad y' = x \sin \theta + y \cos \theta.$$

Solve the pair for $\cos \theta$ and $\sin \theta$:

$$\cos \theta = \frac{xx' + yy'}{x^2 + y^2}, \quad \sin \theta = \frac{xy' - yx'}{x^2 + y^2},$$

and therefore

$$\theta = \text{atan2}(xy' - yx', \quad xx' + yy')$$

Why `atan2` and not `arctan`. Using `arctan(sin / cos)` collapses the denominator and loses the quadrant, so rotations differing by 180° become indistinguishable. `atan2` keeps both signs and returns a value in $(-180^\circ, 180^\circ]$. This is the point of the question — a student who uses plain `arctan` should lose a mark even if their example happens to work.

Worked example. $(x, y) = (1, 0)$, $(x', y') = (0, 1)$:

$$\cos \theta = \frac{1 \cdot 0 + 0 \cdot 1}{1} = 0, \quad \sin \theta = \frac{1 \cdot 1 - 0 \cdot 0}{1} = 1 \quad \Rightarrow \quad \theta = 90^\circ.$$

Implementation.

```
import numpy as np
def rotation_angle(x, y, xp, yp):
    d = x*x + y*y
    return np.degrees(np.arctan2((x*yp - y*xp)/d, (x*xp + y*yp)/d))
```

Note a single correspondence suffices only because the rotation is about the origin and there is no translation or scale. Worth asking a strong student what breaks if the transform also translates.