Part 1. Objectives

- **Long-term goal:** How do I fit a model to data?
- How does QR factorization work? What special cases are there?
- How do I solve (in a least-squares sense) an overdetermined linear system using QR?
- Why should I use QR and not the normal equations $A^T A x = A^T b$ for least-squares problems?

Part 2. Uniqueness of the least-squares solution

Consider the least-squares problem $A x \cong b$ where $A$ is tall-and-skinny (i.e. $m \times n$ with $m > n$) and has a nontrivial nullspace (i.e. there is a vector $n \neq 0$ so that $A n = 0$).

If you have a solution $x$ to $A x \cong b$, can you find another solution $x'$ (so that $\| A x - b \|_2 = \| A x' - b \|_2$)?

- (A) No
- (B) Yes

Part 3. Norms and Matrices

Given

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 3 \\ 4 \end{bmatrix},$$

what is $\| Q^T b \|_2^2$?

Part 4. Least-Squares Residual

Given a QR factorization $A = QR$ with $Q = I$, what is the square of the 2-norm of the residual of solving the least-squares problem $A x \cong b$ going to be if

$$b = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$
Part 5. Solving least-squares problems

You are given a matrix $A$ and a right-hand side vector $b$.
Use a QR factorization of $A$ (from `scipy.linalg.qr`) to solve the least-squares problem $Ax \cong b$.

Hints:

- `scipy.linalg.qr` returns the full QR factorization by default.
- Use `scipy.linalg.solve_triangular`

INPUTS: $A$ and $b$
OUTPUT: $x$

```python
import scipy.linalg as la

m, n = A.shape
```