

TOPIC 1

Least squares: Cholesky, QR and Householder

4 QR decomposition and Householder transformations

Course: [Math 535 \(http://www.math.wisc.edu/~roch/mmidis/\)](http://www.math.wisc.edu/~roch/mmidis/) - Mathematical Methods in Data Science (MMiDS)

Author: [Sebastien Roch \(http://www.math.wisc.edu/~roch/\)](http://www.math.wisc.edu/~roch/), Department of Mathematics, University of Wisconsin-Madison

Updated: Sep 21, 2020

Copyright: © 2020 Sebastien Roch

4.1 Revisiting Gram-Schmidt

Let $\mathbf{a}_1, \dots, \mathbf{a}_m$ be linearly independent. In a previous notebook, we discussed the Gram-Schmidt algorithm to obtain an orthonormal basis of $\text{span}(\mathbf{a}_1, \dots, \mathbf{a}_m)$.

We revisit it in matrix form. Let

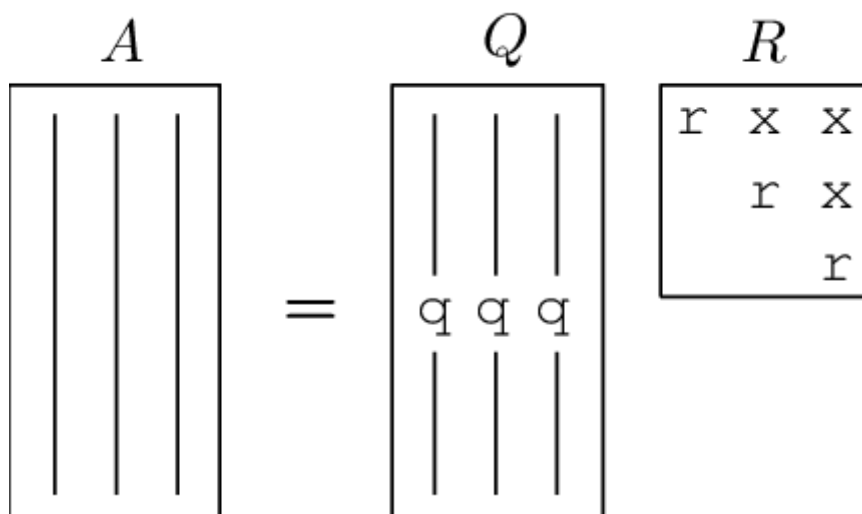
$$A = \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \dots & \mathbf{a}_m \\ | & & | \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} | & & | \\ \mathbf{q}_1 & \dots & \mathbf{q}_m \\ | & & | \end{pmatrix}.$$

The output of the Gram-Schmidt algorithm can be written in the following compact form, known as a [QR decomposition](https://en.wikipedia.org/wiki/QR_decomposition) (https://en.wikipedia.org/wiki/QR_decomposition),

$$A = QR$$

where column i of the $m \times m$ matrix R contains the coefficients of the linear combination of the \mathbf{q}_j 's that produce \mathbf{a}_i .

By the proof of Gram-Schmidt, $\mathbf{a}_i \in \text{span}(\mathbf{q}_1, \dots, \mathbf{q}_i)$. So column i of R has only zeros below the diagonal. Hence R has a special structure we have previously encountered: it is upper triangular.



(Source (https://www.researchgate.net/figure/QR-factorization-of-a-quasimatrix_fig8_270659665))

4.2 Least squares via QR

Let $A \in \mathbb{R}^{n \times m}$ be an $n \times m$ matrix with linearly independent columns and let $\mathbf{b} \in \mathbb{R}^n$ be a vector. Recall that a solution \mathbf{x}^* to the least-squares problem

$$\min_{\mathbf{x} \in \mathbb{R}^m} \|A\mathbf{x} - \mathbf{b}\|$$

satisfies the normal equations

$$A^T A \mathbf{x}^* = A^T \mathbf{b}.$$

As we will see later in the course, the normal equations approach via Cholesky decomposition has numerical issues. We present an alternative approach based on the QR decomposition. The QR decomposition approach to least squares proceeds as follows:

1. Construct an orthonormal basis of $\text{col}(A)$ through a QR decomposition

$$A = QR.$$

1. Form the orthogonal projection matrix

$$P = QQ^T.$$

1. Apply the projection to \mathbf{b} and observe that, by the proof of the normal equations, \mathbf{x}^* satisfies

$$A\mathbf{x}^* = QQ^T\mathbf{b}.$$

1. Plug in the QR decomposition for A to get

$$QR\mathbf{x}^* = QQ^T\mathbf{b}.$$

1. Multiply both sides by Q^T and use $Q^TQ = I_{m \times m}$

$$R\mathbf{x}^* = Q^T\mathbf{b}.$$

1. Solving this system for \mathbf{x}^* is straightforward because R is upper triangular via back substitution.

Theorem (Least Squares via QR): Let $A \in \mathbb{R}^{n \times m}$ be an $n \times m$ matrix with linearly independent columns, let $\mathbf{b} \in \mathbb{R}^n$ be a vector, and let $A = QR$ be a QR decomposition of A . The solution to the least-squares problem

$$\min_{\mathbf{x} \in \mathbb{R}^m} \|A\mathbf{x} - \mathbf{b}\|.$$

satisfies

$$R\mathbf{x}^* = Q^T\mathbf{b}.$$

NUMERICAL CORNER We implement the QR approach to least squares and return to our previous linear regression example.

```

In [1]: # Julia version: 1.5.1
using Plots, LinearAlgebra

function mmids_gramschmidt(A)
    n,m = size(A)
    Q = zeros(Float64,n,m)
    R = zeros(Float64,m,m)
    for j = 1:m
        v = A[:,j]
        for i = 1:j-1
            R[i,j] = dot(Q[:,i],A[:,j])
            v -= R[i,j]*Q[:,i]
        end
        R[j,j] = norm(v)
        Q[:,j] = v/R[j,j]
    end
    return Q,R
end

function mmids_backsubs(U,b)
    m = length(b)
    x = zeros(Float64,m)
    for i=m:-1:1
        x[i] = (b[i] - dot(U[i,i+1:m],x[i+1:m]))/U[i,i]
    end
    return x
end

```

Out[1]: mmids_backsubs (generic function with 1 method)

```

In [2]: function ls_by_qr(A, b)
        Q,R = mmids_gramschmidt(A)
        return mmids_backsubs(R,Q'*b)
    end

```

Out[2]: ls_by_qr (generic function with 1 method)

```

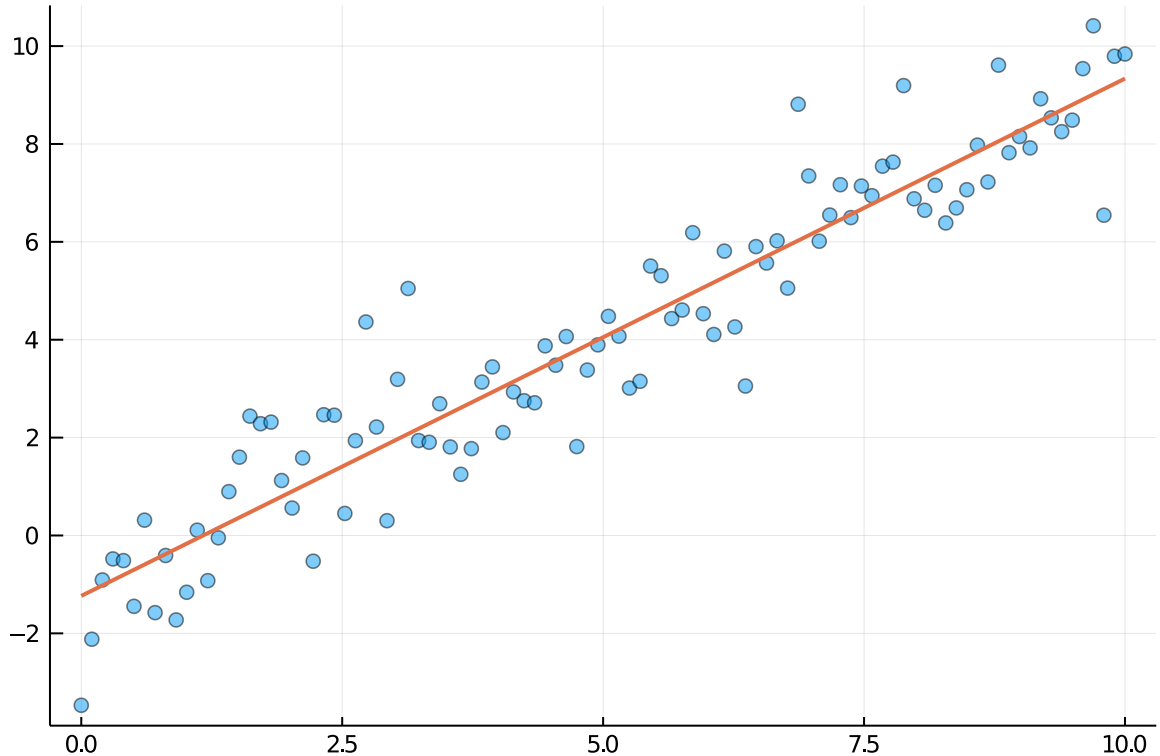
In [3]: n, b0, b1 = 100, -1, 1
        x = LinRange(0,10,n)
        y = b0 .+ b1*x .+ randn(n)
        A = hcat(ones(n),x)
        coeff = ls_by_qr(A,y)

```

Out[3]: 2-element Array{Float64,1}:
 -1.2341467593021789
 1.0569731809683849

```
In [4]: scatter(x,y,legend=False,alpha=0.5)
        plot!(x,coeff[1].+coeff[2]*x,lw=2)
```

Out[4]:



4.3 Householder transformations

While Gram-Schmidt gives a natural way to compute a QR decomposition, there are many other numerical algorithms for this purpose. Some have better numerical behavior, specifically in terms of how they handle roundoff error. Quoting [Wikipedia \(https://en.wikipedia.org/wiki/Round-off_error\)](https://en.wikipedia.org/wiki/Round-off_error):

A roundoff error, also called rounding error, is the difference between the result produced by a given algorithm using exact arithmetic and the result produced by the same algorithm using finite-precision, rounded arithmetic. Rounding errors are due to inexactness in the representation of real numbers and the arithmetic operations done with them. [...] When a sequence of calculations with an input involving roundoff error are made, errors may accumulate, sometimes dominating the calculation.

We will not prove this here, but the following method based on Householder reflectors is numerically more [stable \(https://en.wikipedia.org/wiki/Numerical_stability#Stability_in_numerical_linear_algebra\)](https://en.wikipedia.org/wiki/Numerical_stability#Stability_in_numerical_linear_algebra).

First some preliminary definitions.

Definition (Orthogonal Matrix): A square matrix $Q \in \mathbb{R}^{m \times m}$ is orthogonal if $Q^T Q = Q Q^T = I_{m \times m}$. ◁

In words, the matrix inverse of Q is its transpose. As we have seen before, this is equivalent to the columns of Q forming an orthonormal basis of \mathbb{R}^m .

Exercise: Show that the product of two orthogonal matrices Q_1 and Q_2 is also orthogonal. ◁

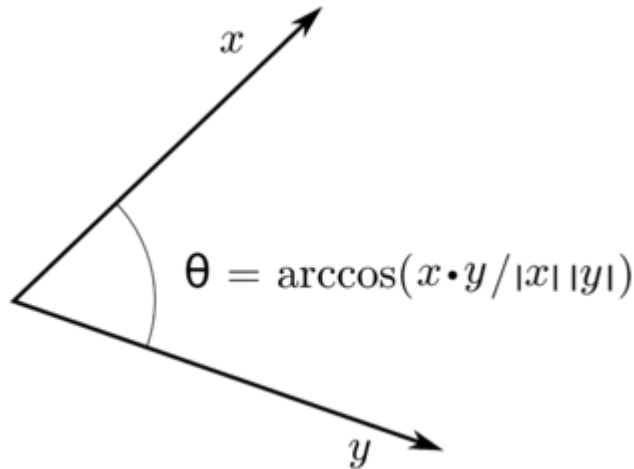
An important property of orthogonal matrices is that they preserve inner products: if $Q \in \mathbb{R}^{m \times m}$ is orthogonal, then for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$

$$\langle Q\mathbf{x}, Q\mathbf{y} \rangle = (Q\mathbf{x})^T Q\mathbf{y} = \mathbf{x}^T Q^T Q\mathbf{y} = \mathbf{x}^T \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle.$$

In particular, orthogonal matrices preserve norms and angles.

Recall that the angle θ between \mathbf{x} and \mathbf{y} satisfies

$$\cos \theta = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}.$$



(Source (<https://commons.wikimedia.org/wiki/File:Inner-product-angle.png>))

4.3.1 Reflections

One such family of transformations are reflections.

Definition (Hyperplane): A hyperplane W is a linear subspace of \mathbb{R}^m of dimension $m - 1$. ◁

Exercise: Let $W \subseteq \mathbb{R}^m$ be a hyperplane. Show that there exists a unit vector $\mathbf{z} \in \mathbb{R}^m$ such that

$$\mathbf{w} \in W \iff \langle \mathbf{z}, \mathbf{w} \rangle = 0.$$

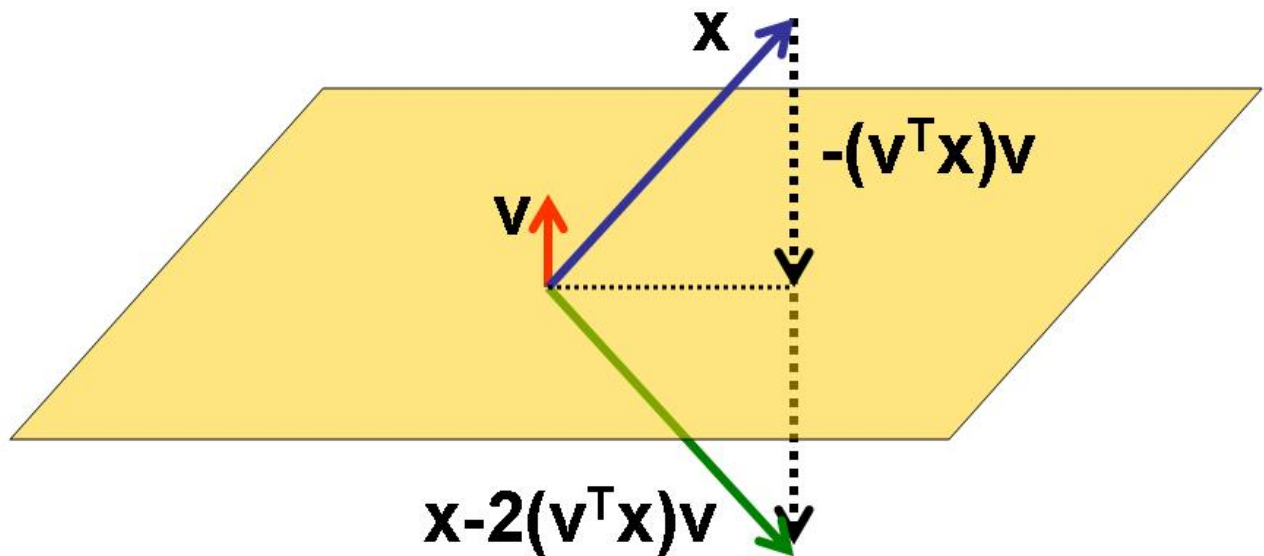
◁

Definition (Householder Reflection): Let $\mathbf{z} \in \mathbb{R}^m$ be a unit vector and let \mathcal{W} be the hyperplane orthogonal to it. The reflection across \mathcal{W} is given by

$$H = I_{m \times m} - 2\mathbf{z}\mathbf{z}^T.$$

This is referred to as a Householder reflection. \triangleleft

In words, we subtract twice the projection onto \mathbf{z} , as depicted below.



(Source (<https://ccjou.wordpress.com/householder-matrix/>))

Lemma: Let $H = I_{m \times m} - 2\mathbf{z}\mathbf{z}^T$ be a Householder reflections. Then H is an orthogonal matrix.

Proof: We check the definition:

$$\begin{aligned} H^T H &= (I_{m \times m}^T - 2(\mathbf{z}\mathbf{z}^T)^T)(I_{m \times m} - 2\mathbf{z}\mathbf{z}^T) \\ &= (I_{m \times m} - 2\mathbf{z}\mathbf{z}^T)(I_{m \times m} - 2\mathbf{z}\mathbf{z}^T) \\ &= I_{m \times m} - 2\mathbf{z}\mathbf{z}^T - 2\mathbf{z}\mathbf{z}^T + 4\mathbf{z}\mathbf{z}^T \end{aligned}$$

which is equal to $I_{m \times m}$. The calculation for HH^T is the same. \square

4.3.2 QR decomposition by introducing zeros

We return to QR decompositions. One way to construct a QR decomposition of a matrix $A \in \mathbb{R}^{n \times m}$ is to find a sequence of orthogonal matrices Q_1, \dots, Q_k that triangularize A :

$$Q_k \cdots Q_2 Q_1 A = R$$

for an upper-triangular matrix R . Indeed, by the properties of orthogonal matrices, we then have

$$A = Q_1^T Q_2^T \cdots Q_k^T R$$

where $Q = Q_1^T Q_2^T \cdots Q_k^T$ is itself orthogonal. So to proceed we need to identify orthogonal matrices that have the effect of introducing zeros below the diagonal, as illustrated below.

$$H_2 H_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \\ 0 & 0 & \boxtimes & \times & \times \end{bmatrix}$$

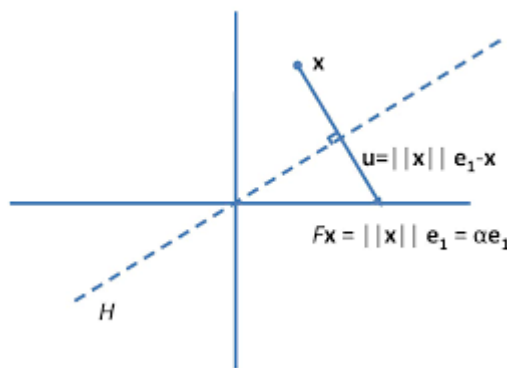
(Source) (https://hua-zhou.github.io/teaching/biostatm280-2017spring/slides/10-gr/householder_transform_diagram.png)

It turns out that a well-chosen Householder reflection does the trick. Let \mathbf{a}_1 be the first column of A and take

$$\mathbf{z}_1 = \frac{\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1}{\|\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1\|} \quad \text{and} \quad H_1 = I_{n \times n} - 2\mathbf{z}_1\mathbf{z}_1^T$$

where recall that \mathbf{e}_1 is the first element of the standard basis of \mathbb{R}^n . As depicted below, this choice sends \mathbf{a}_1 to

$$\|\mathbf{a}_1\| \mathbf{e}_1 = \begin{pmatrix} \|\mathbf{a}_1\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$



(Source) https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQPtlGW70euRloKwOXwwNErCtqpV1Z_04oZy0rd2yk5ksVdVV1T

Lemma: Let \mathbf{a}_1 , \mathbf{z}_1 and H_1 be as above. Then

$$H_1 \mathbf{a}_1 = \|\mathbf{a}_1\| \mathbf{e}_1.$$

Proof idea: The proof by picture is in the figure above.

Proof: Note that

$$\begin{aligned} \|\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1\|^2 &= (\|\mathbf{a}_1\| - a_{1,1})^2 + \sum_{j=2}^n a_{1,j}^2 \\ &= \|\mathbf{a}_1\|^2 - 2\|\mathbf{a}_1\| a_{1,1} + a_{1,1}^2 + \sum_{j=2}^n a_{1,j}^2 \\ &= 2(\|\mathbf{a}_1\|^2 - \|\mathbf{a}_1\| a_{1,1}) \end{aligned}$$

and

$$\begin{aligned}
 2\mathbf{z}_1 \mathbf{z}_1^T \mathbf{a}_1 &= 2\mathbf{z}_1 \frac{\|\mathbf{a}_1\| \mathbf{e}_1^T \mathbf{a}_1 - \mathbf{a}_1^T \mathbf{a}_1}{\|\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1\|} \\
 &= 2 \frac{\|\mathbf{a}_1\| a_{1,1} - \|\mathbf{a}_1\|^2}{\|\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1\|^2} (\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1) \\
 &= -(\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1)
 \end{aligned}$$

where we used the previous equation. Hence

$$\mathbf{H}_1 \mathbf{a}_1 = (I_{n \times n} - 2\mathbf{z}_1 \mathbf{z}_1^T) \mathbf{a}_1 = \mathbf{a}_1 + (\|\mathbf{a}_1\| \mathbf{e}_1 - \mathbf{a}_1) = \|\mathbf{a}_1\| \mathbf{e}_1.$$

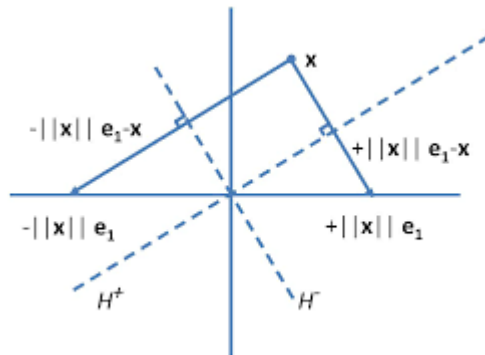
That establishes the claim. \square

There is another valid choice, as the next exercise establishes.

Exercise: Define

$$\tilde{\mathbf{z}}_1 = \frac{\|\mathbf{a}_1\| \mathbf{e}_1 + \mathbf{a}_1}{\|\|\mathbf{a}_1\| \mathbf{e}_1 + \mathbf{a}_1\|} \quad \text{and} \quad \tilde{\mathbf{H}}_1 = I_{n \times n} - 2\tilde{\mathbf{z}}_1 \tilde{\mathbf{z}}_1^T.$$

Show that $\tilde{\mathbf{H}}_1 \mathbf{a}_1 = -\|\mathbf{a}_1\| \mathbf{e}_1$.



(Source) <https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQ0lCpiodSnps555DzXtc5XSo2zvJWXfOjUZMuL6w4skmbpyEum>

4.3.3 Putting everything together

We have shown how to introduce zeros below the diagonal in the first column of a matrix $A \in \mathbb{R}^{n \times m}$. To introduce zeros in the second column below the diagonal we use a block matrix

$$H_2 = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & F_2 \end{pmatrix}$$

where F_2 is the following Householder reflection. Write the second column of $H_1 A$ as $(y^{(2)}, \mathbf{y}_2)^T$. That is, y_2 are the entries 2, ..., n of that column.

$$F_2 = I_{(n-1) \times (n-1)} - 2\mathbf{z}_2\mathbf{z}_2^T \quad \text{with} \quad \mathbf{z}_2 = \frac{\|\mathbf{y}_2\| \mathbf{e}_1 - \mathbf{y}_2}{\|\|\mathbf{y}_2\| \mathbf{e}_1 - \mathbf{y}_2\|}$$

where now $\mathbf{e}_1 \in \mathbb{R}^{n-1}$. Applying H_2 to $H_1 A$ preserves the first row and column, and introduces zeros under the diagonal on the second column.

And so on. At Step k , we split the k -th column of $H_{k-1} \cdots H_1 A$ into its first $k-1$ and last $n-k+1$ entries $(\mathbf{y}^{(k)}, \mathbf{y}_k)$ and form the matrix

$$H_k = \begin{pmatrix} I_{(k-1) \times (k-1)} & \mathbf{0} \\ \mathbf{0} & F_k \end{pmatrix}$$

where

$$F_k = I_{(n-k+1) \times (n-k+1)} - 2\mathbf{z}_k\mathbf{z}_k^T \quad \text{with} \quad \mathbf{z}_k = \frac{\|\mathbf{y}_k\| \mathbf{e}_1 - \mathbf{y}_k}{\|\|\mathbf{y}_k\| \mathbf{e}_1 - \mathbf{y}_k\|}$$

This time the first $k-1$ rows and columns are preserved, while zeros are introduced under the diagonal of the k -th column.

NUMERICAL CORNER We implement the procedure above in Julia. We will need the following function. For $\alpha \in \mathbb{R}$, let the sign of α be

$$\text{sign}(\alpha) = \begin{cases} 1 & \text{if } \alpha > 0 \\ 0 & \text{if } \alpha = 0 \\ -1 & \text{if } \alpha < 0 \end{cases}$$

For example, in Julia, using the function `sign` :

```
In [5]: sign(-10), sign(20), sign(0)
```

```
Out[5]: (-1, 1, 0)
```

The following function constructs the upper triangular matrix R by iteratively modifying the relevant block of A . Computing Q actually requires extra computational work that is often not needed. We saw that, in the context of the least-squares problem, we really only need to compute $Q^T \mathbf{b}$ for some input vector \mathbf{b} . This can be done as R is constructed, as follows.

See [here \(https://docs.julialang.org/en/v1/base/base/#\)](https://docs.julialang.org/en/v1/base/base/#) for an explanation of `copy` .

```
In [6]: function mmids_householder(A)
    n, m = size(A)
    Qt = Matrix{Float64}(I, n, n)
    R = copy(A)
    for k = 1:m
        # computing z
        y = R[k:n, k]
        e1 = zeros(n-k+1)
        e1[1] = 1
        z = sign(y[1])*norm(y)*e1 .+ y
        z = z/norm(z)

        # updating Qt
        Qt[k:n, 1:n] = Qt[k:n, 1:n] .- 2*z*z'*Qt[k:n, 1:n]

        # updating R
        R[k:n, k:m] = R[k:n, k:m] .- 2*z*z'*R[k:n, k:m]
    end
    return Qt[1:m, 1:n]', R[1:m, 1:m]
end
```

Out[6]: mmids_householder (generic function with 1 method)

In `mmids_householder`, we use both reflections defined above. We will not prove this here, but the particular choice made has good numerical properties. Quoting [TB, Lecture 10]:

Mathematically, either choice of sign is satisfactory. However, this is a case where numerical stability -- insensitivity to rounding errors -- dictates that one choice should be taken rather than the other. For numerical stability, it is desirable to reflect \mathbf{x} to the vector $z\|\mathbf{x}\|\mathbf{e}_1$ that is not too close to \mathbf{x} itself. [...] Suppose that [in the figure above] the angle between H^+ and the \mathbf{e}_1 axis is very small. Then the vector $\mathbf{v} = \|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$ is much smaller than \mathbf{x} or $\|\mathbf{x}\|\mathbf{e}_1$. Thus the calculation of \mathbf{v} represents a subtraction of nearby quantities and will tend to suffer from cancellation errors.

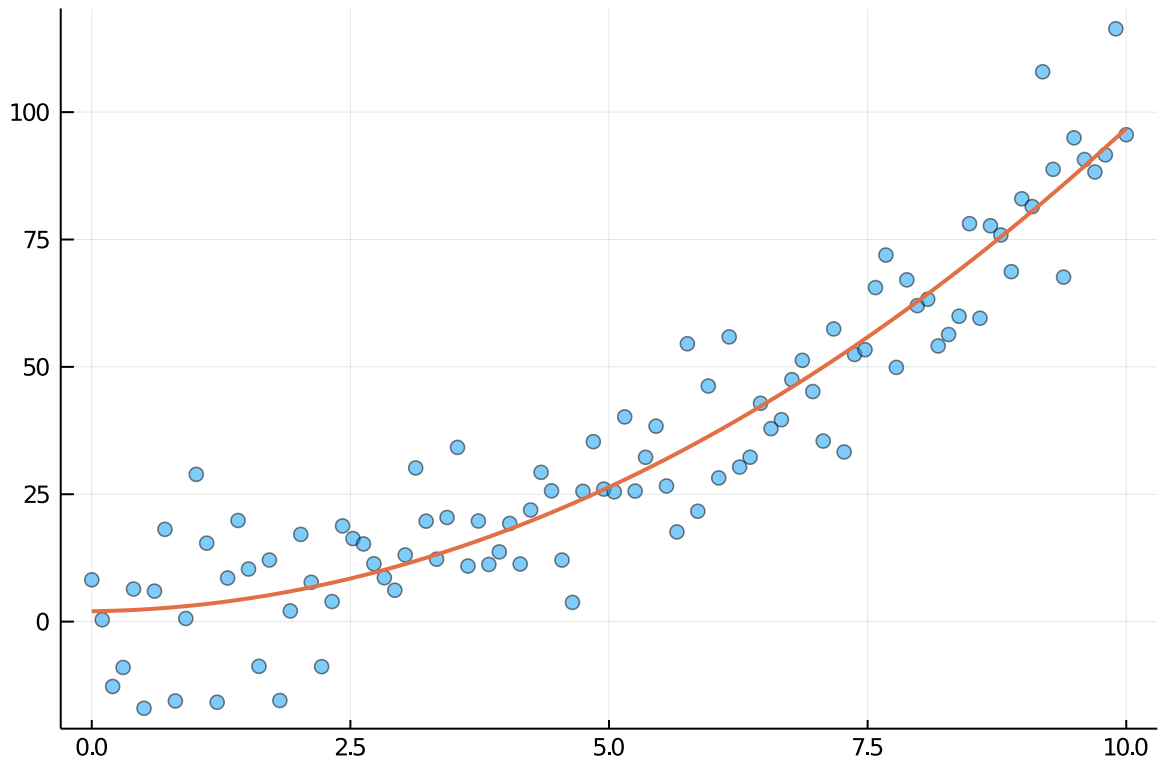
We return to our regression example.

```
In [7]: n, b0, b1, b2 = 100, 0, 0, 1
    x = LinRange(0, 10, n)
    y = b0 .+ b1*x .+ b2*x.^2 .+ 10*randn(n)
    A = reduce(hcat, [ones(n), x, x.^2])
    Q, R = mmids_householder(A)
    coeff = mmids_backsubs(R, Q'*y)
```

Out[7]: 3-element Array{Float64, 1}:
 2.0282989895406263
 0.2719746343547142
 0.9204381318085032

```
In [8]: scatter(x,y,legend=false,alpha=0.5)
plot!(x,coeff[1].+coeff[2]*x.+coeff[3]*x.^2,lw=2)
```

Out[8]:



One advantage of the Householder approach is that it produces a matrix Q with very good orthogonality, i.e., $Q^T Q \approx I$. We give a quick example below comparing Gram-Schmidt and Householder. (The choice of matrix A will become clearer when we discuss the singular value decomposition later in the course.)

```
In [9]: n = 50
U, W = qr(randn(n,n))
V, W = qr(randn(n,n))
S = Diagonal((1/2).^(1:n))
A = U*S*V'
```

```
In [10]: Qgs, Rgs = mmids_gramschmidt(A)
norm(A-Qgs*Rgs)
```

Out[10]: 1.3619835091208436e-16

```
In [11]: norm(Qgs'*Qgs - Matrix{Float64}(I,n,n))
```

Out[11]: 23.988875573848322

As you can see above, the Q and R factors produced by Gram-Schmidt do have the property that $QR \approx A$. However, Q is far from orthogonal.

On the other hand, Householder reflections perform much better in that respect as we show next.

```
In [12]: Qhh, Rhh = mmids_householder(A)
norm(A-Qhh*Rhh)
```

```
Out[12]: 3.327807615972261e-16
```

```
In [13]: norm(Qhh'*Qhh - Matrix{Float64}(I,n,n))
```

```
Out[13]: 6.643898557699989e-15
```