Try Hausholder reflectors to find $A = QR$ where A is not square. $A \in \mathbb{R}^{m \times n}$ where $m > n$.

m: # of equations

n: # of variables

Goal minimize $\|Ax - b\|$ using QR factorization.

since $m > n$ the equation $Ax = b$ is overdetermined.

Example $m = 5$ and $n = 3$

```
julia> A=rand(5,3)
5×3 Matrix{Float64}:
 0.320727  0.388933   0.681836
 0.79072   0.0768611  0.131238
 0.419896  0.593692   0.212764
 0.756835  0.666969   0.298797
 0.360863  0.272446   0.0304287

julia> using LinearAlgebra
```

```
julia> a1=A[:,1]
5-element Vector{Float64}:
 0.32072700349930194
 0.7907195643369205
 0.41989585565864607
 0.7568349909367742
 0.3608625456766106

julia> e1=[1,0,0,0,0]
5-element Vector{Int64}:
 1
 0
 0
 0
 0
```

```
julia> norm(a1-norm(a1)*e1)
1.549712159101821

julia> norm(a1+norm(a1)*e1)
2.0070224773625043

julia> c=-norm(a1)
-1.2678472897697846
```

bigger one in the denominator

```
julia> v=(a1-c*e1)/norm(a1-c*e1)
5-element Vector{Float64}:
 0.7915079732224451
 0.39397643686384204
 0.20921332989276997
 0.37709343042901866
 0.17979995229093407
```

note $H_1 \in \mathbb{R}^{5\times 5}$

```
julia> H1=I-2*v*v'
5×5 Matrix{Float64}:
 -0.25297   -0.623671  -0.331188   -0.596945  -0.284626
 -0.623671   0.689565  -0.16485    -0.297132  -0.141674
 -0.331188  -0.16485    0.91246    -0.157786  -0.0752331
 -0.596945  -0.297132  -0.157786    0.715601  -0.135603
 -0.284626  -0.141674  -0.0752331  -0.135603   0.935344
```

```
julia> H1*A
5×3 Matrix{Float64}:
 -1.26785     -0.818637    -0.511824
 -1.53666e-16 -0.524212    -0.462911
 -6.44245e-17  0.274504    -0.102747
 -9.56311e-17  0.0916533   -0.269891
 -9.57246e-18 -0.00186686  -0.240725
```

zero

work on this submatrix

```
julia> A2=(H1*A)[2:5,2:3]
4×2 Matrix{Float64}:
 -0.524212    -0.462911
  0.274504    -0.102747
  0.0916533   -0.269891
 -0.00186686  -0.240725
```

```
julia> e1=[1,0,0,0]
4-element Vector{Int64}:
 1
 0
 0
 0

julia> a1=A2[:,1]
4-element Vector{Float64}:
 -0.5242122774339462
  0.2745044436144049
  0.0916533453237448
 -0.0018668623370580023
```

```
julia> norm(a1-norm(a1)*e1)
1.159698689150701

julia> norm(a1+norm(a1)*e1)
0.29886291075403804
```

bigger

```
julia> c=norm(a1)
0.5987946411421512

julia> v2=(a1-c*e1)/norm(a1-c*e1)
4-element Vector{Float64}:
 -0.9683609450300623
  0.2367032455778981
  0.07903203321792718
 -0.0016097822257824653
```

Extend the vector so we can make the H_2 reflector.

```
julia> v2x=[0,v2...]
5-element Vector{Float64}:
  0.0
 -0.9683609450300623
  0.2367032455778981
  0.07903203321792718
 -0.0016097822257824653
```

```
julia> H2=I-2*v2x*v2x'
5×5 Matrix{Float64}:
  1.0   0.0         -0.0         -0.0          0.0
  0.0  -0.875446     0.458428     0.153063    -0.0031177
 -0.0   0.458428     0.887943    -0.0374143    0.000762081
 -0.0   0.153063    -0.0374143    0.987508     0.000254449
  0.0  -0.0031177    0.000762081  0.000254449  0.999995
```

this part is the 1x1 identity matrix

```
julia> H2*H1*A
5×3 Matrix{Float64}:
 -1.26785      -0.818637    -0.511824
  1.06978e-16   0.598795     0.317592
 -1.02164e-16  -4.55617e-17 -0.293531
 -2.55909e-17   1.19169e-16 -0.333591
 -1.4966e-17    6.02808e-17 -0.239427
```

zero

need this to be zero so upper triangular.

last submatrix

```
julia> A3=(H2*H1*A)[3:5,3]
3-element Vector{Float64}:
 -0.2935307516889512
 -0.3335912868234489
 -0.23942713546125546
```

One more reflector for the third column...

```
julia> a1=A3[:,1]
3-element Vector{Float64}:
 -0.2935307516889512
 -0.3335912868234489
 -0.23942713546125546

julia> e1=[1,0,0]
3-element Vector{Int64}:
 1
 0
 0

julia> norm(a1-norm(a1)*e1)
0.8976941077123972

julia> norm(a1+norm(a1)*e1)
0.46175805037442896
```

bigger so $c = \|a_1\|$

```
julia> c=norm(a1)
0.5047462749013485

julia> v3=(a1-c*e1)/norm(a1-c*e1)
3-element Vector{Float64}:
 -0.8892528309276273
 -0.3716090859430316
 -0.26671349784325754
```

extend the vector to make H_3

```
julia> v3x=[0,0,v3...]
5-element Vector{Float64}:
  0.0
  0.0
 -0.8892528309276273
 -0.3716090859430316
 -0.26671349784325754
```

```
julia> H3=I-2*v3x*v3x'
5×5 Matrix{Float64}:
  1.0  -0.0   0.0        0.0        0.0
 -0.0   1.0   0.0        0.0        0.0
  0.0   0.0  -0.581541  -0.660909  -0.474351
  0.0   0.0  -0.660909   0.723813  -0.198226
  0.0   0.0  -0.474351  -0.198226   0.857728
```

I is in upper Corner.

```
julia> R=H3*H2*H1*A
5×3 Matrix{Float64}:
 -1.26785      -0.818637     -0.511824
  1.06978e-16   0.598795      0.317592
  2.22239e-16   2.49068e-17   0.504746
  1.65662e-17  -8.13371e-18  -5.78385e-17
  4.61204e-17   4.82014e-17  -2.55356e-17
```

zero

$H_3 H_2 H_1 A = R$ $\qquad A = H_1^{-1} H_2^{-1} H_3^{-1} R = \underbrace{H_1 H_2}H_3 R$

Since $\underbrace{H_i^{-1} = H_i^T}_{\text{orthogonal}} \underbrace{= H_i}_{\text{symmetric}}$ $\qquad Q = H_1 H_2 H_3$

```
julia> Q=H1*H2*H3
5×5 Matrix{Float64}:
 -0.25297    0.303681    0.903252   -0.0620444   0.155149
 -0.623671  -0.724287    0.0833204  -0.222455   -0.173246
 -0.331188   0.538698   -0.25326    -0.630481   -0.372118
 -0.596945   0.297744   -0.200685    0.709889   -0.103868
 -0.284626   0.0658669  -0.269777   -0.212639    0.892555
```

should be A.

```
julia> Q*R
5×3 Matrix{Float64}:
 0.320727  0.388933   0.681836
 0.79072   0.0768611  0.131238
 0.419896  0.593692   0.212764
 0.756835  0.666969   0.298797
 0.360863  0.272446   0.0304287
```

$\|A-QR\|$

```
julia> opnorm(A-Q*R)
8.848100600596786e-16
```

rounding error is small.

In summary $A \in \mathbb{R}^{5\times3}$ and we used Householder reflectors to find $Q = H_1 H_2 H_3 \in \mathbb{R}^{5\times5}$ and $R \in \mathbb{R}^{5\times3}$ so

$$\underset{5\times3}{A} = \underset{5\times5}{Q}\ \underset{5\times3}{R} = \begin{bmatrix} \underset{5\times3}{\tilde{Q}} & \underset{5\times2}{\tilde{\tilde{Q}}} \end{bmatrix} \begin{bmatrix} \tilde{R} \\ 0_{2\times3} \end{bmatrix} = \tilde{Q}\tilde{R} + \tilde{\tilde{Q}}\,0 = \underset{5\times3}{\tilde{Q}}\ \underset{3\times3}{\tilde{R}}$$

reduced QR factorization

```
julia> R=H3*H2*H1*A
5×3 Matrix{Float64}:
 -1.26785       -0.818637     -0.511824
  1.06978e-16    0.598795      0.317592
  2.22239e-16    2.49068e-17   0.504746
  1.65662e-17   -8.13371e-18  -5.78385e-17
  4.61204e-17    4.82014e-17  -2.55356e-17
```

non-zero

all zero

$$R = \begin{bmatrix} \tilde{R} \\ 0_{2\times3} \end{bmatrix} \qquad \tilde{R} \in \mathbb{R}^{3\times3}$$

If the 3 columns of A are independent to begin with, then the 3 columns of $\tilde{R}$ have to be independent. That means the diagonal of $\tilde{R}$ is non-zero

Compare with Gram-Schmidt... This uses column operations to make the columns of A orthonormal...

$t_1 = a_1 \qquad q_1 = t_1 / \|t_1\|$

$t_2 = a_2 - (a_2 \cdot q_1)\, q_1 \qquad q_2 = t_2 / \|t_2\|$

$t_3 = a_3 - (a_3 \cdot q_1)\, q_1 - (a_3 \cdot q_2)\, q_2 \qquad q_3 = t_3 / \|t_3\|$

those are the entries in R

```
julia> t1=A[:,1]
5-element Vector{Float64}:
 0.32072700349930194
 0.7907195643369205
 0.41989585565864607
 0.7568349909367742
 0.3608625456766106

julia> q1=t1/norm(t1)
5-element Vector{Float64}:
 0.25296974334940564
 0.6236709820790004
 0.33118803742909025
 0.5969449136687432
 0.2846261916465791
```

```
julia> t2=A[:,2]-q1'A[:,2]*q1
5-element Vector{Float64}:
  0.18184275754447027
 -0.4336992767939237
  0.3225695675421843
  0.17828760746761502
  0.039440768460186015

julia> q2=t2/norm(t2)
5-element Vector{Float64}:
  0.30368133755776483
 -0.7242871712523651
  0.5386981535554652
  0.2977441600471677
  0.06586693625874142
```

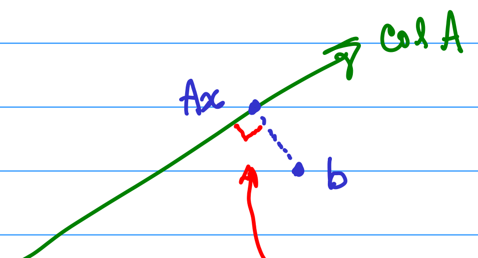
```
julia> t3=A[:,3]-q1'A[:,3]*q1-q2'A[:,3]*q2
5-element Vector{Float64}:
  0.4559128529500368
  0.042055684005038535
 -0.12783200125203328
 -0.10129485816890359
 -0.13616873043785052

julia> q3=t3/norm(t3)
5-element Vector{Float64}:
  0.9032515456189 23
  0.08332044454069171
 -0.25325992009949505
 -0.2006847067642083
 -0.26977659312981433
```

```
julia> Q=H1*H2*H3
5×5 Matrix{Float64}:
 -0.25297    0.303681   0.903252   -0.0620444   0.155149
 -0.623671  -0.724287   0.0833204  -0.222455   -0.173246
 -0.331188   0.538698  -0.25326    -0.630481   -0.372118
 -0.596945   0.297744  -0.200685    0.709889   -0.103868
 -0.284626   0.0658669 -0.269777   -0.212639    0.892555
```

These columns can be anything that makes the matrix orthogonal.

Idea minimize $\|Ax - b\|$ $\qquad A = QR = \tilde{Q}\tilde{R}$

suppose $b \notin \text{Col } A = \{Ax : x \in \mathbb{R}^n\}$.



closest point Ax in the column space is such that this is perpendicular.