# A Appendix: The QR decomposition and least squares problems

Here we give a brief review of the QR decomposition, and the application of this decomposition to least squares problems. Chapter 5 of Golub & Van Loan (1996) is an excellent reference.

## A.1 The QR decomposition of a full column rank matrix

Let $A \in \mathbb{R}^{m\times n}$ with $\text{rank}(A) = n$ (this implies that $m \geq n$). Then there exists matrices $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$ such that $A = QR$, where $Q$ is orthogonal (its first $n$ columns form a basis for the column space of $A$), and $R$ is of the form

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

$R_1 \in \mathbb{R}^{n\times n}$ being upper triangular. This is (not surprisingly) called the *QR decomposition*, and it can be computed in $O(mn^2)$ operations (Golub & Van Loan 1996).

The decomposition $A = QR$ is used primarily for solving least squares problems. For example, given $b \in \mathbb{R}^m$, suppose that are interested in finding $x \in \mathbb{R}^n$ to minimize

$$\|b - Ax\|_2^2. \tag{23}$$

Since $\text{rank}(A) = n$, the minimizer $x$—also referred to as the solution—is unique. Let $Q_1 \in \mathbb{R}^{m\times n}$ denote the first $n$ columns of $Q$, and let $Q_2 \in \mathbb{R}^{m\times(m-n)}$ denote the last $m - n$ columns. Then

$$\|b - Ax\|_2^2 = \|Q^T(b - Ax)\|_2^2 = \|Q_1^T b - R_1 x\|_2^2 + \|Q_2^T b\|_2^2,$$

and so minimizing the left-hand side is equivalent to minimizing $\|Q_1^T b - R_1 x\|_2^2$. This can be done quickly, by solving the equation $R_1 x = c$ where $c = Q_1^T b$. Recalling the triangular structure of $R_1$, this looks like:

$$\begin{bmatrix} \square & \square & \square & \square & \square \\ & \square & \square & \square & \square \\ & & \square & \square & \square \\ & & & \square & \square \\ & & & & \square \end{bmatrix} \cdot x = c,$$

where the boxes denote nonzero entries, and blank spaces indicate zero entries. We first solve the equation given by last row (an equation in one variable), then we substitute and solve the second to last row, etc. This *back-solve* procedure takes $O(n^2)$ operations. Hence, finding the least squares solution of (23) requires $O(mn) + O(n^2) = O(mn)$ operations in total (the first term counts the multiplication by $Q_1^T$ to form $c = Q_1^T b$). Note that this does not count the $O(mn^2)$ operations required to compute the QR decomposition of $A$ in the first place; and importantly, if we want to minimize multiple criterions of the form (23) for different vectors $b$, then we only compute the QR decomposition of $A$ once, and use this decomposition to find each solution quickly in $O(mn)$ operations.

## A.2 The QR decomposition of a column rank deficient matrix

Let $A \in \mathbb{R}^{m\times n}$ with $\text{rank}(A) = k \leq n$. Then there exists $P \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{m\times m}$, and $R \in \mathbb{R}^{m\times n}$ such that $AP = QR$, where $P$ is a permutation matrix, $Q$ is orthogonal (its first $k$ columns span the column space of

$A$), and $R$ decomposes as

$$R = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix},$$

where $R_1 \in \mathbb{R}^{k \times k}$ is upper triangular, and $R_2 \in \mathbb{R}^{k \times (n-k)}$ is dense. Visually, $R$ looks like this (when the order of rank deficiency is $n - k = 2$):

$$\begin{bmatrix} \square\square\square\square\square \\ \square\square\square\square \\ \square\square\square \\ \\ \end{bmatrix}.$$

Note that $AP$ just permutes the columns of $A$. This decomposition takes $O(mnk)$ operations (Golub & Van Loan 1996).

The least squares criterion in (23) can now admit many solutions $x$ (in fact, infinitely many) if $\text{rank}(A) < n$. If we simply want any solution $x$—Golub & Van Loan (1996) refer to this as a *basic solution*—then we can use the QR decomposition $AP = QR$. We write

$$\begin{aligned} \|b - Ax\|_2^2 &= \|b - APP^T x\|_2^2 \\ &= \|Q^T(b - APP^T x)\|_2^2 \\ &= \|Q_1^T b - \begin{bmatrix} R_1 & R_2 \end{bmatrix} P^T x\|_2^2 + \|Q_2^T b\|_2^2, \end{aligned}$$

where $Q_1 \in \mathbb{R}^{m \times k}$ contains the first $k$ columns of $Q$, and $Q_2 \in \mathbb{R}^{m \times (m-k)}$ contains the last $m - k$ columns. We can now consider $z = P^T x$ as the optimization variable, and solve

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = Q_1^T b, \tag{24}$$

where we have decomposed $z = (z_1, z_2)$ with $z_1 \in \mathbb{R}^k$ and $z_2 \in \mathbb{R}^{n-k}$. Note that to solve (24), we can take $z_2 = 0$, and then back-solve to compute $z_1$ in $O(k^2)$ operations. Letting $x = Pz$, we have hence computed a basic least squares solution in $O(mk) + O(k^2) + O(n) = O(mn)$ operations.

## A.3 The minimum $\ell_2$ norm least squares solution

Suppose again that $A \in \mathbb{R}^{m \times n}$ and $\text{rank}(A) = k \leq n$. If we want to compute the unique solution[1] $x^*$ that has the minimum $\ell_2$ norm across all least squares solutions $x$ in (23), then the strategy given in the last section does not necessarily work (in fact, it does not produce $x^*$ unless $R_2 = 0$). However, we can modify the QR decomposition $AP = QR$ from Section A.2 in order to compute $x^*$. For this, we need to apply Givens rotations to $R$. These are covered in the next section, but for now, the key message is that there exists an orthogonal transformation $G \in \mathbb{R}^{n \times n}$ such that

$$RG = \widetilde{R} = \begin{bmatrix} 0 & \widetilde{R}_1 \\ 0 & 0 \end{bmatrix}, \tag{25}$$

where $\widetilde{R}_1 \in \mathbb{R}^{k \times k}$ is upper triangular. Applying a single Givens rotation to (the columns of) $R$ takes $O(k)$ operations, and $G$ is composed of $k(n - k)$ of them, so forming $RG$ takes $O(k^2(n - k))$ operations. Hence

[1] Uniqueness follows from the fact that the set of least squares solutions forms a convex set. Note that this is given by $x^* = A^+ b$, where $A^+$ is the Moore-Penrose pseudoinverse of $A$.

the decomposition $APG = Q\widetilde{R}$ requires the same order of complexity, $O(mnk) + O(k^2(n-k)) = O(mnk)$ operations in total.

Now we write

$$\|b - Ax\|_2^2 = \|b - APGz\|_2^2,$$

where $z = G^T P^T x$. Since $P, G$ are orthogonal, we have $\|x\|_2 = \|z\|_2$, and therefore our problem is equivalent to finding the minimum $\ell_2$ norm minimizer $z^*$ of the right-hand side above. As before, we now utilize the QR decomposition, writing

$$\|b - APGz\|_2^2 = \|Q^T(b - APGz)\|_2^2 = \|Q_1^T b - \begin{bmatrix} 0 & \widetilde{R}_1 \end{bmatrix} z\|_2^2 + \|Q_2^T b\|_2^2,$$

where $Q_1 \in \mathbb{R}^{m\times k}$ and $Q_2 \in \mathbb{R}^{m\times(m-k)}$ give, respectively, the first $k$ and the last $m-k$ columns of $Q$. Hence we seek the minimum $\ell_2$ norm solution of

$$\begin{bmatrix} 0 & \widetilde{R}_1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = Q_1^T b,$$

where $z = (z_1, z_2)$ with $z_1 \in \mathbb{R}^{(n-k)}$ and $z_2 \in \mathbb{R}^k$. For $z^*$ to have minimum $\ell_2$ norm, we must have $z_1^* = 0$. Then $z_2^*$ is given by back-solving, which takes $O(k^2)$ operations. Finally, we let $x^* = PGz^*$, and count $O(mk) + O(k^2) + O(n^2) + O(n) = O(n \cdot \max\{m, n\})$ operations in total to compute the minimum $\ell_2$ norm least squares solution.

## A.4 The minimum $\ell_2$ norm least squares solution and the transposed QR

Given $A \in \mathbb{R}^{m\times n}$ with $\text{rank}(A) = k \le n$, it can be advantageous in some problems to use a QR decomposition of $A^T$ instead of $A$. (For example, this is the case when we want to update the QR decomposition after $A$ has changed by one column; see Section D.2.) By what we just showed, we can compute a decomposition $A^T PG = Q\widetilde{R}$, where $P \in \mathbb{R}^{m\times m}$ is a permutation matrix, $G \in \mathbb{R}^{m\times m}$ is an orthogonal matrix of Givens rotations, $Q \in \mathbb{R}^{n\times n}$ is orthogonal, and $\widetilde{R} \in \mathbb{R}^{n\times m}$ is of the special form (25) with $\widetilde{R}_1 \in \mathbb{R}^{k\times k}$ upper triangular, in $O(mnk)$ operations.

To find the minimum $\ell_2$ norm minimizer $x^*$ of (23), we can employ a similar strategy to that of Section A.3. Using the orthogonality of $P, G$, and the computed decomposition $G^T P^T A = \widetilde{R}^T Q^T$, we have

$$\|b - Ax\|_2^2 = \|G^T P^T (b - Ax)\|_2^2 = \|c_1 - \begin{bmatrix} \widetilde{R}_1^T & 0 \end{bmatrix} z\|_2^2 + \|c_2\|_2^2,$$

where $z = Q^T x$, and $c_1, c_2$ denote the first $k$, respectively last $m-k$ coordinates of $c = G^T P^T b$. As $Q$ is orthogonal, we have $\|x\|_2 = \|z\|_2$, and hence it suffices to find the minimum $\ell_2$ norm minimizer $z^*$ of the right-hand side above. This is the same as finding the minimum $\ell_2$ norm solution of the linear equation

$$\begin{bmatrix} \widetilde{R}_1^T & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = c_2,$$

where $z = (z_1, z_2)$ with $z_1 \in \mathbb{R}^k$ and $z_2 \in \mathbb{R}^{(n-k)}$. Therefore $z_2^* = 0$, and $z_1^*$ can be computed by foward-solving (the same concept as back-solving, except we start with the first row), requiring $O(k^2)$ operations. We finally take $x^* = Qz^*$. The total number of operations is $O(n) + O(m^2) + O(k^2) + O(nk) = O(m \cdot \max\{m, n\})$.

# B Appendix: Givens rotations

We describe Givens rotations, orthogonal transformations that help maintain (or create) maintain upper triangular structure. Givens rotations provide a way to efficiently update the QR decomposition of a given matrix after a row or column has been added or deleted. (They also provide a way to compute the QR decomposition in the first place.) Our explanation and notation here are based largely on Chapter 5 of Golub & Van Loan (1996).

## B.1 Simple Givens rotations in two dimensions

The main idea behind a Givens rotation can be expressed by considering the $2 \times 2$ rotation matrix

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $c = \cos\theta$ and $s = \sin\theta$, for some $\theta \in [0, 2\pi]$. Multiplication by $G^T$ amounts to a counterclockwise rotation through an angle $\theta$; since it is a rotation matrix, $G$ is clearly orthogonal. Furthermore, given any vector $(a, b) \in \mathbb{R}^2$, we can choose $c, s$ (choose $\theta$) such that

$$G^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix},$$

for some $d \in \mathbb{R}$. This is simply rotating $(a, b)$ onto the first coordinate axis, and by inspection we see that we must take $c = a/\sqrt{a^2 + b^2}$ and $s = -b/\sqrt{a^2 + b^2}$. Note that, from the point of view of computational efficiency, we never have to compute $\theta$ (which would require inverse trigonometric functions).

## B.2 Givens rotations in higher dimensions

The same idea extends naturally to higher dimensions. Consider the $n \times n$ Givens rotation matrix

$$G = \begin{bmatrix} 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix};$$

in other words, $G$ is the $n \times n$ identity matrix, except with four elements $G_{ii}, G_{ij}, G_{ji}, G_{jj}$ replaced with the corresponding elements of the $2 \times 2$ Givens rotation matrix. We will write $G = G(i, j)$ to emphasize the dependence on $i, j$. It is straightforward to check that $G$ is orthogonal. Applying $G^T$ to a vector $x \in \mathbb{R}^n$

only affects components $x_i$ and $x_j$, and leaves all other components untouched: with $z = G^T x$, we have

$$z_k = \begin{cases} cx_i - sx_j & \text{if } k = i \\ sx_i + cx_j & \text{if } k = j \\ x_k & \text{otherwise} \end{cases}.$$

Because $G^T$ only acts on two components, we can compute $z = G^T x$ in $O(1)$ operations. And as in the $2 \times 2$ case, we can make $z_j = 0$ by taking

$$c = x_i / \sqrt{x_i^2 + x_j^2} \quad \text{and} \quad s = -x_j / \sqrt{x_i^2 + x_j^2}.$$

Now we consider Givens rotations applied to matrices. If $A \in \mathbb{R}^{m \times n}$ and $G = G(i,j) \in \mathbb{R}^{m \times m}$, then pre-multiplying $A$ by $G^T$ (as in $G^T A$) only affects rows $i$ and $j$, and hence computing $G^T A$ takes $O(n)$ operations. Moreover, with the appropriate choice of $c, s$, we can selectively zero out an element in the $j$th row of $G^T A$. A common application of $G^T$ looks like the following:

$$G^T \cdot \begin{bmatrix} \square & \square & \square & \square \\ & \square & \square & \square \\ & & \square & \square \\ & & \square & \square \\ & & & \square \end{bmatrix} = \begin{bmatrix} \square & \square & \square & \square \\ & \square & \square & \square \\ & & \square & \square \\ & & & \square \\ & & & \square \end{bmatrix},$$

where in this example $G = G(3,4)$, and $c, s$ have been chosen so that the element in the 4th row and 3rd column of the output is zero. Importantly, the first 2 columns of rows 3 and 4 were all zeros to begin with, and zeros after pre-multiplication, so that this zero pattern has not been disturbed (think of the $2 \times 2$ case: rotating $(0,0)$ still gives $(0,0)$). Applying a second Givens rotation to the output gives an upper triangular structure:

$$G_2^T \cdot \begin{bmatrix} \square & \square & \square & \square \\ & \square & \square & \square \\ & & \square & \square \\ & & & \square \\ & & & \square \end{bmatrix} = \begin{bmatrix} \square & \square & \square & \square \\ & \square & \square & \square \\ & & \square & \square \\ & & & \square \\ & & & \end{bmatrix},$$

where $G_2 = G_2(4,5)$ is another Givens rotation matrix.

On the other hand, post-multiplying $A \in \mathbb{R}^{m \times n}$ by a Givens rotation matrix $G = G(i,j) \in \mathbb{R}^{n \times n}$ (as in $AG$) only affects columns $i$ and $j$. Therefore computing $AG$ requires $O(m)$ operations. The logic is very similar to the pre-multiplication case, and by choosing $c, s$ appropriately, we can zero out a particular element in the $j$th column of $AG$. A common application looks like:

$$\begin{bmatrix} \square & \square & \square & \square & \square \\ & \square & \square & \square & \square \\ & & \square & \square & \square \\ & & & & \square \end{bmatrix} \cdot G = \begin{bmatrix} \square & \square & \square & \square & \square \\ & \square & \square & \square & \square \\ & & & \square & \square \\ & & & & \square \end{bmatrix},$$

where $G = G(3,4)$ and $c, s$ were chosen to zero out the element in the 3rd row and 3rd column. Now applying

two more Givens rotations yields an upper triangular structure:

$$\begin{bmatrix} \square\square\square\square\square \\ \square\square\square\square \\ \square\square \\ \square \end{bmatrix} \cdot G_2 G_3 = \begin{bmatrix} \square\square\square\square \\ \square\square\square \\ \square\square \\ \square \end{bmatrix}.$$

# C Appendix: Updating the QR decomposition in the full rank case

In this section we cover techniques based on Givens rotations for updating the QR decomposition of a matrix $A \in \mathbb{R}^{m \times n}$, after a row or column has been either added or removed to $A$. We assume here that $\text{rank}(A) = n$; the next section covers the rank deficient case, which is more delicate. For the full rank update problem, a good reference is Section 12.5 of Golub & Van Loan (1996). Hence suppose that we have computed a decomposition $A = QR$, with $Q \in \mathbb{R}^{m \times m}$ and $R \in \mathbb{R}^{m \times n}$, as described in Section A.1, and we subsequently want to compute a QR decomposition of $\widetilde{A}$, where $\widetilde{A}$ differs from $A$ by either one row or one column. As motivation, we may have already solved the least squares problem

$$\|b - Ax\|_2^2,$$

and now want to solve the new least squares problem

$$\|c - \widetilde{A}x\|_2^2.$$

As we will see, computing a QR decomposition of $\widetilde{A}$ by updating that of $A$ saves an order of magnitude in computational time when compared to the naive route (computing the QR decomposition "from scratch"). We treat the row and column update problems separately.

## C.1 Adding or removing a row

Suppose that $\widetilde{A} \in \mathbb{R}^{(m+1) \times n}$ is formed by adding a row to $A$, following its $i$th row, so

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} A_1 \\ w^T \\ A_2 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{i \times n}$, $A_2 \in \mathbb{R}^{(m-i) \times n}$, and $w \in \mathbb{R}^n$ is the row to be added. Let $Q_1 \in \mathbb{R}^{i \times m}$ denote the first $i$ rows of $Q$ and $Q_2 \in \mathbb{R}^{(m-i) \times m}$ denote its last $m - i$ rows. By rearranging both the rows of $A$ the rows of $Q$ in the same way, the product $Q^T A$ remains the same:

$$\begin{bmatrix} Q_2^T & Q_1^T \end{bmatrix} \begin{bmatrix} A_2 \\ A_1 \end{bmatrix} = Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where $R_1 \in \mathbb{R}^{n\times n}$ is upper triangular. Therefore

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & Q_2^T & Q_1^T \end{bmatrix} \begin{bmatrix} w^T \\ A_2 \\ A_1 \end{bmatrix} = \begin{bmatrix} w^T \\ R_1 \\ 0 \end{bmatrix}.$$

We can now apply Givens rotations $G_1, \dots G_n$ so that

$$\widetilde{R} = G_n^T \dots G_1^T \begin{bmatrix} w^T \\ R_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{R}_1 \\ 0 \end{bmatrix},$$

where $\widetilde{R}_1 \in \mathbb{R}^{n\times n}$ is upper triangular. Hence defining

$$\widetilde{Q} = \begin{bmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{bmatrix} G_1 \dots G_n,$$

and noting that $\widetilde{Q} \in \mathbb{R}^{(m+1)\times(m+1)}$ is still orthogonal, we have $\widetilde{A} = \widetilde{Q}\widetilde{R}$, the desired QR decomposition. This update procedure uses $n$ Givens rotations, and therefore it requires a total of $O(mn)$ operations. Compare this to the usual cost $O(mn^2)$ of computing a QR decomposition (without updating).

On the other hand, suppose that $\widetilde{A} \in \mathbb{R}^{(m-1)\times n}$ is formed by removing the $i$th row of $A$. Hence

$$A = \begin{bmatrix} A_1 \\ w^T \\ A_2 \end{bmatrix} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{(i-1)\times n}$, $A_2 \in \mathbb{R}^{(m-i)\times n}$, and $w \in \mathbb{R}^n$ is the row to be deleted. (We assume without a loss of generality that $m > n$, so that removing a row does not change the rank; updates in the rank deficient case are covered in the next section.) Let $q \in \mathbb{R}^m$ denote the $i$th row of $Q$, and note that we can compute Givens rotations $G_1, \dots G_{m-1}$ such that

$$G_{m-1}^T \dots G_1^T q = se_1,$$

with $e_1 = (1, 0, \dots 0) \in \mathbb{R}^m$ the first standard basis vector, and $s = \pm 1$. Let $\widetilde{Q} = QG_1 \dots G_{m-1}$; then, as $\widetilde{Q}$ is still orthogonal, it has the form

$$\widetilde{Q} = \begin{bmatrix} 0 & \widetilde{Q}_1 \\ s & 0 \\ 0 & \widetilde{Q}_2 \end{bmatrix},$$

where $\widetilde{Q}_1 \in \mathbb{R}^{(i-1)\times(m-1)}$ and $\widetilde{Q}_2 \in \mathbb{R}^{(m-i)\times(m-1)}$. Furthermore, defining $\widetilde{R}G_{m-1}^T \dots G_1^T R$, we can see that

$$\widetilde{R} = \begin{bmatrix} v^T \\ \widetilde{R}_1 \\ 0 \end{bmatrix},$$

where $\widetilde{R}_1 \in \mathbb{R}^{n\times n}$ is upper triangular and $v \in \mathbb{R}^n$. By construction $A = QR = \widetilde{Q}\widetilde{R}$, and defining

$$\widetilde{Q}_0 = \begin{bmatrix} \widetilde{Q}_1 \\ \widetilde{Q}_2 \end{bmatrix} \quad \text{and} \quad \widetilde{R}_0 = \begin{bmatrix} \widetilde{R}_1 \\ 0 \end{bmatrix},$$

we have $\widetilde{A} = \widetilde{Q}_0\widetilde{R}_0$, as desired. We performed $m-1$ Givens rotations, and hence $O(m^2)$ operations.

## C.2 Adding or removing a column

Suppose that $\widetilde{A} \in \mathbb{R}^{m\times(n+1)}$ is formed by adding a column to $A$, say, after its $j$th column. Then

$$Q^T\widetilde{A} = \begin{bmatrix} R_1 & \vdots & R_2 \\ 0 & w & R_3 \\ 0 & \vdots & 0 \end{bmatrix},$$

where $R_1 \in \mathbb{R}^{j\times j}$ and $R_3 \in \mathbb{R}^{(n-j)\times(n-j)}$ are upper triangular, $R_2 \in \mathbb{R}^{j\times(n-j)}$ is dense, and $w \in \mathbb{R}^m$. (We are assuming here, without a loss of generality, that the added column does not lie in the span of the existing ones; updates in the rank deficient case are covered in the next section.) We can apply Givens rotations $G_1, \dots G_{n-j}$ to the rows of $Q^T\widetilde{A}$ so that

$$G_{n-j}^T \dots G_1^T Q^T \widetilde{A} = \begin{bmatrix} \widetilde{R}_1 \\ 0 \end{bmatrix} = \widetilde{R},$$

where $\widetilde{R}_1 \in \mathbb{R}^{(n+1)\times(n+1)}$ is upper triangular. Therefore with $\widetilde{Q} = QG_1 \dots G_{n-j}$, we have $\widetilde{A} = \widetilde{Q}\widetilde{R}$. We applied $O(n)$ Givens rotations, so this update procedure requires $O(mn)$ operations.

If instead $\widetilde{A} \in \mathbb{R}^{m\times(n-1)}$ is formed by removing the $j$th column of $A$, then

$$Q^T\widetilde{A} = \begin{bmatrix} R_1 & R_2 \\ 0 & w^T \\ 0 & R_3 \\ 0 & 0 \end{bmatrix},$$

where $R_1 \in \mathbb{R}^{(j-1)\times(j-1)}$ and $R_3 \in \mathbb{R}^{(n-j)\times(n-j)}$ are upper triangular, $R_2 \in \mathbb{R}^{(j-1)\times(n-j)}$ is dense, and $w \in \mathbb{R}^{n-j}$. Note that we can apply Givens rotations $G_1, \dots G_{n-j}$ to the rows of $Q^T\widetilde{A}$ to produce

$$G_{n-j}^T \dots G_1^T Q^T \widetilde{A} = \begin{bmatrix} \widetilde{R}_1 \\ 0 \end{bmatrix} = \widetilde{R},$$

where $\widetilde{R}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ is upper triangular. Hence with $\widetilde{Q} = QG_1 \dots G_{n-j}$, we see that $\widetilde{A} = \widetilde{Q}\widetilde{R}$. Again we used $O(n)$ Givens rotations, and $O(mn)$ operations.

# D Appendix: Updating the QR decomposition in the rank deficient case

Here we again consider techniques for updating a QR decomposition, but study the more difficult case in which $A \in \mathbb{R}^{m\times n}$ with $\text{rank}(A) = k \leq n$. In particular, we are interested in computing the minimum $\ell_2$ norm minimizer of

$$\|b - Ax\|_2^2, \tag{26}$$

and subsequently, computing the minimum $\ell_2$ norm minimizer of

$$\|c - \widetilde{A}x\|_2^2. \tag{27}$$

where $\widetilde{A}$ has either one more or one less row that $A$, or else one more of one less column than $A$. Depending on whether whether our goal is to update the rows or columns, we actually need to use a different QR decomposition for the the initial least squares problem (26).

## D.1 Adding or removing a row

We compute the minimum $\ell_2$ norm minimizer of the initial least squares criterion (26) using the QR decomposition $APG = QR$ described in Section A.3, where $P \in \mathbb{R}^{n\times n}$ is a permutation matrix, $G \in \mathbb{R}^{n\times n}$ is a product of Givens rotations matrices, $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$ is of the special form

$$R = \begin{bmatrix} 0 & R_1 \\ 0 & 0 \end{bmatrix},$$

with $R_1 \in \mathbb{R}^{k\times k}$ upper triangular (we note, in order to avoid confusion, that $R, R_1$ were written as $\widetilde{R}, \widetilde{R}_1$ in Section A.3).

First suppose that $\widetilde{A} \in \mathbb{R}^{(m+1)\times n}$ is formed by adding a row to $A$, after its $i$th row. Write

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} A_1 \\ w^T \\ A_2 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{i\times n}$, $A_2 \in \mathbb{R}^{(m-i)\times n}$, and $w \in \mathbb{R}^n$ is the row to be added. Also let $Q_1 \in \mathbb{R}^{i\times m}$ and $Q_2 \in \mathbb{R}^{(m-i)\times m}$ denote the first $i$ and last $m-i$ rows of $Q$, respectively. The logic at this step is similar to that in the full rank case: we can rearrange both the rows of $A$ and the rows of $Q$ so that the product $Q^TA$ will not change, hence

$$\begin{bmatrix} Q_2^T & Q_1^T \end{bmatrix} \begin{bmatrix} A_2 \\ A_1 \end{bmatrix} PG = Q^TAPG = R = \begin{bmatrix} 0 & R_1 \\ 0 & 0 \end{bmatrix}.$$

Therefore

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & Q_2^T & Q_1^T \end{bmatrix} \begin{bmatrix} w^T \\ A_2 \\ A_1 \end{bmatrix} PG = \begin{bmatrix} w^TPG \\ Q^TAPG \end{bmatrix} = \begin{bmatrix} d_1^T & d_2^T \\ 0 & R_1 \\ 0 & 0 \end{bmatrix}, \tag{28}$$

where $d_1 \in \mathbb{R}^{n-k}$ and $d_2 \in \mathbb{R}^k$ are the first $n-k$ and last $k$ components, respectively, of $d = G^TP^Tw$. Now we must consider two cases. First, assume that $\text{rank}(\widetilde{A}) = \text{rank}(A)$, so adding the new row to $A$ did not

change its rank. This implies that $d_1 = 0$, and we can apply Givens rotations $G_1, \dots G_k$ to the right-hand side of (28) so that

$$\widetilde{R} = G_k^T \dots G_1^T \begin{bmatrix} 0 & d_2^T \\ 0 & R_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \widetilde{R}_1 \\ 0 & 0 \end{bmatrix},$$

where $\widetilde{R}_1 \in \mathbb{R}^{k \times k}$ is upper triangular. Letting

$$\widetilde{Q} = \begin{bmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{bmatrix} G_1 \dots G_k,$$

we complete the desired decomposition $\widetilde{A}PG = \widetilde{Q}\widetilde{R}$. Note that this QR decomposition is of the appropriate form to compute the minimum $\ell_2$ norm solution of the least squares problem (27).

The second case to consider is $\operatorname{rank}(\widetilde{A}) > \operatorname{rank}(A)$, which means that adding the new row to $A$ increased the rank. Then at least one component of $d_1$ is nonzero, and we can apply Givens rotations $G_1, \dots G_{n-k}$ to the right-hand side of (28) so that

$$\widetilde{R} = \begin{bmatrix} d_1^T & d_2^T \\ 0 & R_1 \\ 0 & 0 \end{bmatrix} G_1 \dots G_{n-k} = \begin{bmatrix} 0 & \widetilde{R}_1 \\ 0 & 0 \end{bmatrix},$$

where $\widetilde{R}_1 \in \mathbb{R}^{(k+1)\times(k+1)}$ is upper triangular. We let

$$\widetilde{Q} = \begin{bmatrix} 0 & Q_1 \\ 1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad \text{and} \quad \widetilde{G} = GG_1 \dots G_{n-k},$$

and observe that $\widetilde{A}P\widetilde{G} = \widetilde{Q}\widetilde{R}$ is a QR decomposition of the desired form, so that we may compute the minimum $\ell_2$ norm minimizer of (27). Finally, in either case (an increase in rank or not), we used $O(n)$ Givens rotations, and so this update procedure requires $O(n \cdot \max\{m, n\})$ operations.

Alternatively, suppose that $\widetilde{A} \in \mathbb{R}^{(m-1)\times n}$ is formed by removing the $i$th row of $A$, so

$$A = \begin{bmatrix} A_1 \\ w^T \\ A_2 \end{bmatrix} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

where $A_1 \in \mathbb{R}^{(i-1)\times n}$, $A_2 \in \mathbb{R}^{(m-i)\times n}$, and $w \in \mathbb{R}^n$ is the row to be deleted. We follow the same arguments as in the full rank case: we let $q \in \mathbb{R}^m$ denote the $i$th row of $Q$, and compute Givens rotations $G_1, \dots G_{m-1}$ such that

$$G_{m-1}^T \dots G_1^T q = se_1,$$

with $e_1 = (1, 0, \dots 0) \in \mathbb{R}^m$ and $s = \pm 1$. Defining $\widetilde{Q} = QG_1 \dots G_{m-1}$, we see that

$$\widetilde{Q} = \begin{bmatrix} 0 & \widetilde{Q}_1 \\ s & 0 \\ 0 & \widetilde{Q}_2 \end{bmatrix},$$

for $\widetilde{Q}_1 \in \mathbb{R}^{(i-1)\times(m-1)}$ and $\widetilde{Q}_2 \in \mathbb{R}^{(m-i)\times(m-1)}$, and defining $\widetilde{R} = G_{m-1}^T \dots G_1^T R$, we have

$$\widetilde{R} = \begin{bmatrix} v_1^T & v_2^T \\ 0 & \widetilde{R}_1 \\ 0 & 0 \end{bmatrix},$$

where $\widetilde{R}_1 \in \mathbb{R}^{k\times k}$ has zeros below its diagonal, and $v_1 \in \mathbb{R}^{n-k}$, $v_2 \in \mathbb{R}^k$. As $APG = QR = \widetilde{Q}\widetilde{R}$, we let

$$\widetilde{Q}_0 = \begin{bmatrix} \widetilde{Q}_1 \\ \widetilde{Q}_2 \end{bmatrix} \quad \text{and} \quad \widetilde{R}_0 = \begin{bmatrix} 0 & \widetilde{R}_1 \\ 0 & 0 \end{bmatrix},$$

and conclude that $\widetilde{A}PG = \widetilde{Q}_0\widetilde{R}_0$, which is almost the desired QR decomposition. We say almost because, if $\operatorname{rank}(\widetilde{A}) < \operatorname{rank}(A)$ (removing the $i$th row decreased the rank), then the diagonal of $\widetilde{R}_1$ will have a zero element and hence it will not be upper triangular. If the $q$th diagonal element is zero, then we can perform Givens rotations $J_1, \dots J_{q-1}$ and $J_{q+1}, \dots J_k$ resulting in

$$\bar{R}_0 = J_{q-1}^T \dots J_1^T \widetilde{R}_0 J_{q+1} \dots J_k = \begin{bmatrix} 0 & \bar{R}_1 \\ 0 & 0 \end{bmatrix},$$

where $\bar{R}_1 \in \mathbb{R}^{(k-1)\times(k-1)}$ is upper triangular. It helps to see a picture: with its 2nd diagonal element zero, $\widetilde{R}_0$ may look like

$$\begin{bmatrix} \square\square\square\square \\ \quad\square\square \\ \quad\square\square \\ \quad\quad\square \\ \quad\quad\quad \end{bmatrix},$$

so applying 2 Givens rotations to the rows,

$$J_2^T J_1^T \cdot \begin{bmatrix} \square\square\square\square \\ \quad\square\square \\ \quad\square\square \\ \quad\quad\square \\ \quad\quad\quad \end{bmatrix} = \begin{bmatrix} \square\square\square\square \\ \quad\square\square \\ \quad\quad\square \\ \quad\quad\quad \\ \quad\quad\quad \end{bmatrix}.$$

and a single Givens rotation to the columns,

$$\begin{bmatrix} \square\square\square\square \\ \quad\square\square \\ \quad\quad\square \\ \quad\quad\quad \\ \quad\quad\quad \end{bmatrix} \cdot J_4 = \begin{bmatrix} \quad\square\square\square \\ \quad\square\square \\ \quad\quad\square \\ \quad\quad\quad \\ \quad\quad\quad \end{bmatrix},$$

which has desired form. Letting $\bar{Q}_0 = \widetilde{Q}_0 J_1 \dots J_{q-1}$ and $\widetilde{G} = G J_{q+1} \dots J_k$, we have constructed the proper QR decomposition $\widetilde{A}P\widetilde{G} = \bar{Q}_0\bar{R}_0$. We used $O(m)$ Givens rotations, and $O(m \cdot \max\{m, n\})$ operations in total.

## D.2 Adding or removing a column

If $\widetilde{A}$ has one more or less column than $A$, then one cannot obviously update the QR decomposition $APG = QR$ of $A$ to obtain such a decomposition for $\widetilde{A}$. Note that, if $\widetilde{A}$ differs from $A$ by one row, then $\widetilde{A}PG$

also differs from $APG$ by one row; but if $\widetilde{A}$ differs from $A$ by one column, then $\widetilde{A}$ does not even have the appropriate dimensions for post-multiplication by $PG$.

However, because adding or a removing a column to $A$ is the same as adding or removing a row to $A^T$, we can compute a QR decomposition $A^T PG = QR$, where now $P \in \mathbb{R}^{m\times m}$, $G \in \mathbb{R}^{m\times m}$, $Q \in \mathbb{R}^{n\times n}$, and $R \in \mathbb{R}^{n\times m}$, and update it using the strategies disussed in the previous section. The update procedure for addition requires $O(m \cdot \max\{m, n\})$ operations, and that for removal requires $O(n \cdot \max\{m, n\})$ operations. Hence, to be clear, we first compute the decomposition $A^T PG = QR$ in order to solve the initial least squares problem (26), as described in Section A.4, and then update it to form $\widetilde{A}^T \widetilde{P}\widetilde{G} = \widetilde{Q}\widetilde{R}$ (for some $\widetilde{P}, \widetilde{G}, \widetilde{Q}, \widetilde{R}$), which we use to solve (27).

# E Appendix: More plots from the Chicago crime data example

The figures below display 5 more solutions along the fused lasso path fit to the Chicago crime data example, corresponding to 2, 5, 10, 15, and 25 degrees of freedom.

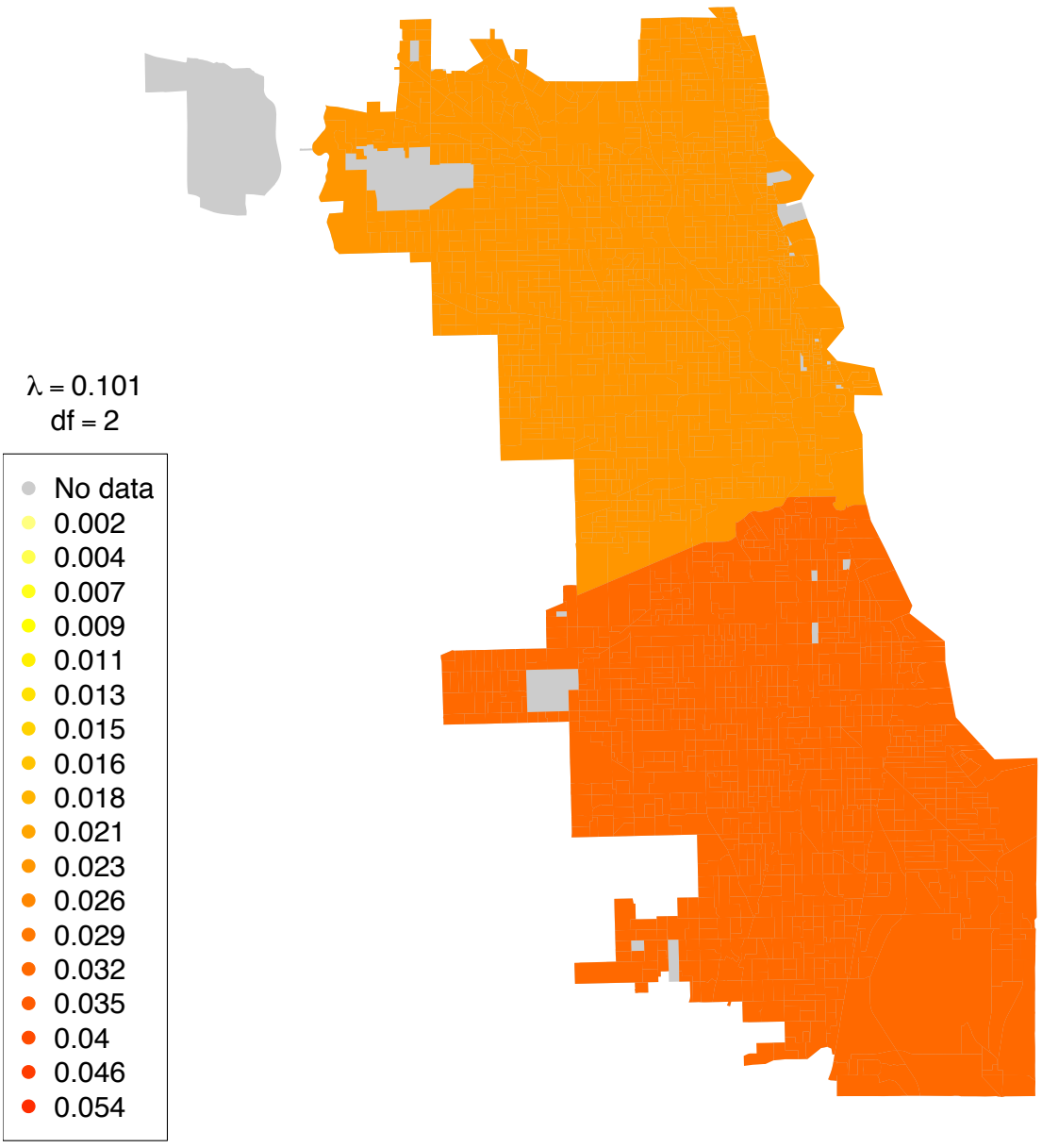


Figure 4: *Fused lasso solution, from the Chicago crime data example, with* $\lambda = 0.101$.

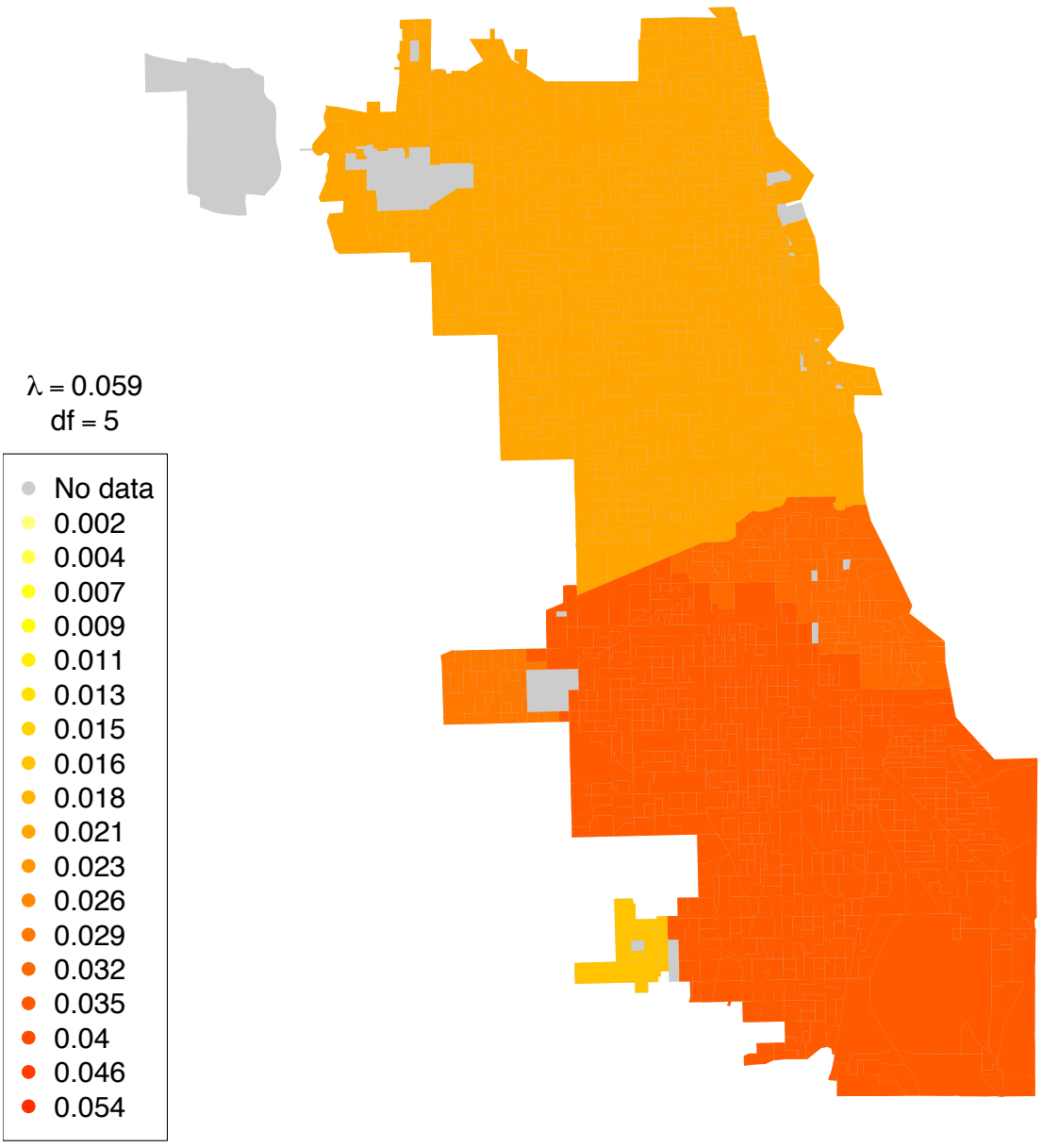


Figure 5: *Fused lasso solution, from the Chicago crime data example, with* $\lambda = 0.059$.

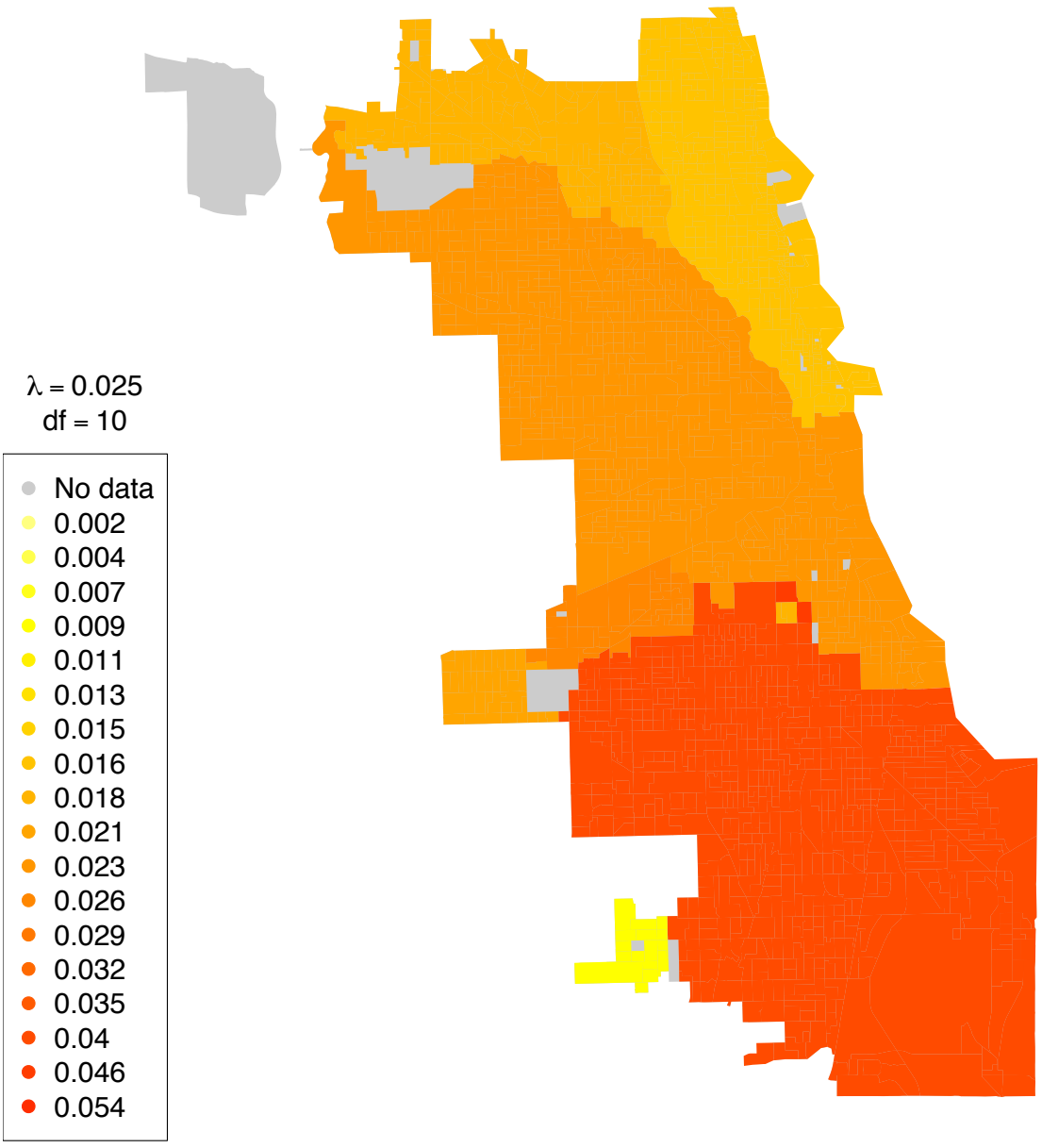


Figure 6: *Fused lasso solution, from the Chicago crime data example, with* $\lambda = 0.025$.

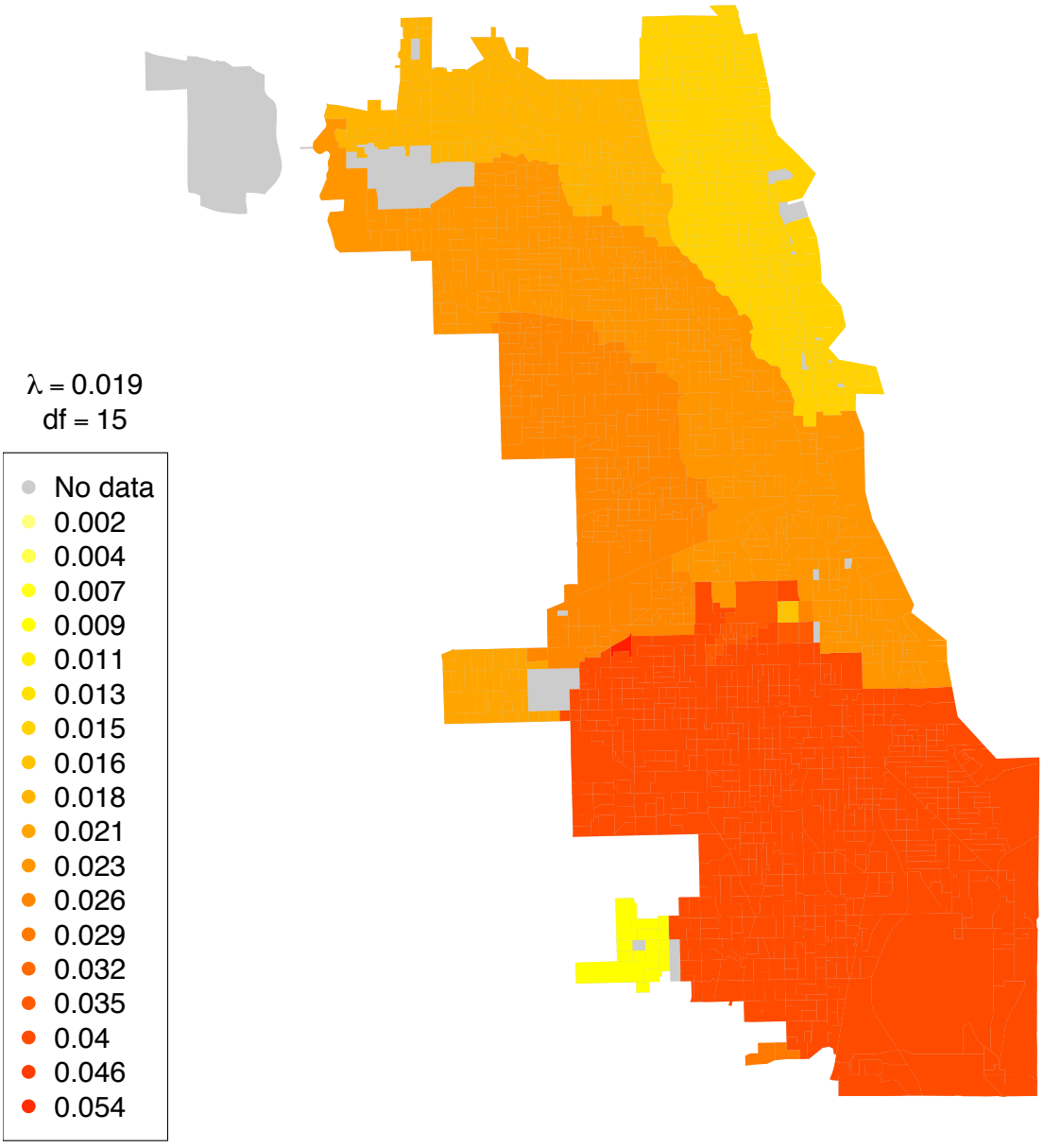


Figure 7: *Fused lasso solution, from the Chicago crime data example, with* $\lambda = 0.019$.

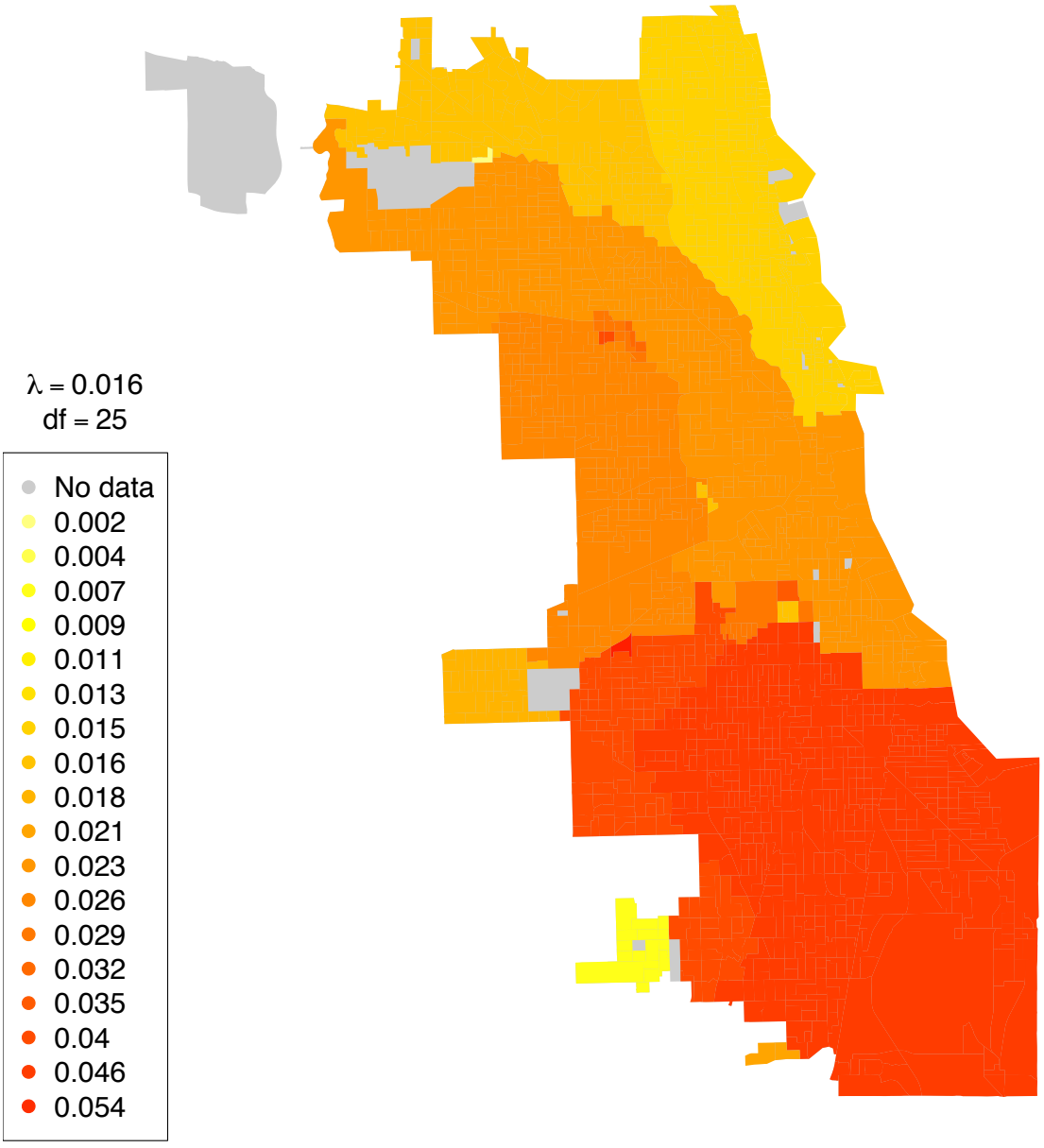


Figure 8: *Fused lasso solution, from the Chicago crime data example, with* $\lambda = 0.016$.