Supplementary Material document for
Bounded-memory adjusted scores estimation in generalized linear models with large data sets

Patrick Zietkiewicz*[1] and Ioannis Kosmidis†[1]

[1]Department of Statistics, University of Warwick
Gibbet Hill Road, Coventry, CV4 7AL, UK

May 30, 2024

## Supplementary material

The Supplementary Material provides the Supplementary Material document that is cross-referenced above and contains an exposition of Givens rotations (Section S1), pseudo-code for iteratively reweighted least squares in chunks (Algorithm S3), pseudo-code for the one- and two-pass implementations for solving the adjusted scores equations in chunks (Algorithm S4 and Algorithm S4, respectively), and all numerical results from the case study of Section 4. The Supplementary Material also provides R code to reproduce all numerical results in the main text and in the Supplementary Material document.

The R code is organized in the two directories `diverted-flights` and `biglm`. The former directory provides code for the case study of Section 4; the `README` file provides specific instructions to reproduce the results, along with the versions of the contributed R packages that have been used to produce the results in the main text. The `biglm` directory has a port of the `biglm` R package (Lumley, 2020), which implements the one- and two-pass IWLS variants for solving the bias-reducing adjusted score equations (Firth, 1993) and for maximum Jeffreys'-penalized likelihood estimation (Kosmidis and Firth, 2021). The Supplementary Material is available at `https://github.com/ikosmidis/bigbr-supplementary-material`.

## S1 Givens rotation matrix

Givens rotation matrices are fully described in Golub and Van Loan (2013, Section 5.1.8), but we also include their definition and properties here for reference. The Givens rotation matrix is

*patrick.zietkiewicz@warwick.ac.uk
†ioannis.kosmidis@warwick.ac.uk

defined as

$$
G(i,j,\theta) = \begin{array}{cc} & \begin{array}{ccccccc} \phantom{0} & \phantom{\cdots} & i & \phantom{\cdots} & j & \phantom{\cdots} & \phantom{0} \end{array} \\ \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \\ \end{array} & \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c_\theta & \cdots & s_\theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s_\theta & \cdots & c_\theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \end{array}
$$

where $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$. Then, $G(i,j,\theta)^\top r$ is a counter-clockwise rotation of $\theta$ radians of $r$ in the $(i,j)$ coordinate plane, $G(i,j,\theta)$ is orthogonal ($G(i,j,\theta)^\top = G(i,j,\theta)^{-1}$), and the product $G(i,j,\theta)^\top R$ affects just the $i$, $j$ rows of $R$. We can use Givens rotations to eliminate arbitrary elements of $R$. For example, we can eliminate the $(3,2)$ element of $R$ by taking the product $G(2,3,\theta)^\top R$ and choosing $s_\theta$ and $c_\theta$ as the solution to the system of two equations $s_\theta^2 + c_\theta^2 = 1$ and $s_\theta r_{22} + c_\theta r_{32} = 0$ with respect to $s_\theta$ and $c_\theta$; see, for example, Golub and Van Loan (2013, Algorithm 5.1.3) for an algorithm that solves that system. Note that there is no need to compute $\theta$.

## S2 Procedures `updateQR`, `incrLS`, `incrIWLS`, `incrIWLS1`, `incrIWLS2`

**Algorithm S1** Update the current value of $\bar{R}$ and $\bar{b}$ (`Rbar` and `bbar`, respectively) for the regression of $b$ on $A$ with observations weights $w$, given chunks of $A$, $b$, and $w$ (`Ac`, `bc`, and `wc`, respectively). `INCL` is a streamlined interface to the `INCLUD` implementation in Miller (1992, Algorithm AS274.1). Specifically, $\texttt{INCL}(\texttt{p}, \texttt{Ai}, \texttt{bi}, \texttt{wi}, \texttt{Rbar}, \texttt{bbar})$ returns the updated values of `Rbar` and `bbar` given a row `Ai` of `Ac`, an element `bi` of `bc`, and an element `wi` of `wc`. `INCL` internally calls $\texttt{INCLUD}(\texttt{p}, \texttt{p}(\texttt{p}-1)/2, \texttt{wi}, \texttt{Ai}, \texttt{bi}, \tilde{\texttt{D}}', \tilde{\texttt{R}}', \tilde{\texttt{b}}, 0, 0)$, which updates $\tilde{R}$ (an upper triangular matrix with ones or zeros in the diagonal), $\tilde{D}$ (a diagonal matrix), and $\tilde{b}$, where $\tilde{R}'$ is a vector with the upper-triangular elements of $\tilde{R}$ stored by row and excluding the diagonal elements, and $\tilde{D}'$ is the diagonal elements of $\tilde{D}$. In the notation of the current work, $\tilde{D}^{1/2}\tilde{R} = \bar{R}$, $\tilde{D}^{1/2}\tilde{b} = \bar{b}$.

**procedure** UPDATEQR(`c`, `p`, `Ac`, `bc`, `wc`, `Rbar`, `bbar`)
    **for** $\texttt{i} \in \{1, 2, \ldots, \texttt{c}\}$ **do**
        $\texttt{Ai} \leftarrow [\texttt{Ac}[\texttt{i}, 1], \ldots, \texttt{Ac}[\texttt{i}, \texttt{p}]]$
        $\texttt{bi} \leftarrow \texttt{bc}[\texttt{i}]$
        $\texttt{wi} \leftarrow \texttt{wc}[\texttt{i}]$
        $\{\texttt{Rbar}, \texttt{bbar}\} \leftarrow \texttt{INCL}(\texttt{p}, \texttt{Ai}, \texttt{bi}, \texttt{wi}, \texttt{Rbar}, \texttt{bbar})$
    **end for**
    **return** $\{\texttt{Rbar}, \texttt{bbar}\}$
**end procedure**

**Algorithm S2** `incrLS` takes as input (a storage location for) $A$, $b$, and $w$, and a chunk size $c$. Internally, `updateQR` is used to compute the least squares estimate $\hat{\psi}$ (`psihat`) of the regression of $b$ on $A$ with observation weights $w$, by applying it sequentially on all chunks of $c$ observations, with the last chunk containing $c$ or less observations.

```
procedure incrLS(A, b, c, w)
    n ← numberofrows(A)        ▷ must be equal to length(b) and length(w), and greater or equal to c
    p ← numberofcolumns(A)
    Rbar ← a p × p matrix of zeros
    bbar ← a vector of p zeros
    K ← ⌈n/c⌉                   ▷ ⌈a⌉ is the ceiling of a
    for k ∈ {1, 2, . . . , K} do
        i1 ← (k − 1) ∗ c + 1
        if k < K then
            i2 ← k ∗ c
        else
            c ← n − i1 + 1
            i2 ← n
        end if
        Ac ← A[i1 : i2, 1 : p]
        bc ← b[i1 : i2]
        wc ← w[i1 : i2]
        {Rbar, bbar} ← updateQR(c, p, Ac, bc, wc, Rbar, bbar)
    end for
    psihat ← inverse(Rbar) · bbar   ▷ using back substitution; · is matrix product
    return {psihat, Rbar}
end procedure
```

**Algorithm S3** `incrIWLS` is an implementation of incremental IWLS for generalized linear models. It takes as input (a storage location for) $X$, $y$, and the observation weights $m$, the chunk size $c$, an implementation of the inverse link function $F$ and its derivative $f$, an implementation of the variance function $V$, starting values for $\hat{\beta}$ (`betahat`), a tolerance $\epsilon$ (`epsilon`), and the maximum allowed number of IWLS iterations (`maxit`). `incrIWLS` sequentially goes through all chunks of $X$ and $y$, updates the corresponding chunk of the working variates $z$ at the current value of $\hat{\beta}$, and uses `updateQR` in Algorithm S1 to update the $\bar{R}$ and $\bar{b}$ from the chunks of $z$, $X$, and $w$. Once all chunks have been visited the weighted least squares estimate $\hat{\beta}$ is computed. The process is repeated until the norm of the difference of the current iterate of $\hat{\beta}$ from the previous one is smaller than $\epsilon$, or the maximum number of allowed iterations is exceeded.

```
 1: procedure incrIWLS(X, y, m, c, F, f, V, betahat, epsilon, maxit)
 2:     n ← numberofrows(X)      ▷ must be equal to length(y) and length(m), and greater or equal to c
 3:     p ← numberofcolumns(X)   ▷ must be equal to length(betahat) and greater or equal to c
 4:     df ← n − p
 5:     K ← ⌈n/c⌉                                             ▷ ⌈a⌉ is the ceiling of a
 6:     iter ← 1
 7:     Rbar ← a p × p matrix of zeros
 8:     bbar ← a vector of p zeros
 9:     SSR ← 0
10:     phihat ← 1                                 ▷ if GLM has an unknown dispersion parameter
11:     for k ∈ {1, 2, . . . , K} do
12:         i1 ← (k − 1) ∗ c + 1
13:         if k < K then
14:             i2 ← k ∗ c
15:         else
16:             c ← n − i1 + 1
17:             i2 ← n
18:         end if
19:         Xc ← X[i1 : i2, 1 : p]
20:         yc ← y[i1 : i2]
21:         mc ← m[i1 : i2]
22:         etac ← Xc · betahat                                   ▷ · is matrix product
23:         muc ← [F(etac[1]), . . . , F(etac[c])]
24:         dc ← [f(etac[1]), . . . , f(etac[c])]
25:         vc ← [V(muc[1]), . . . , V(muc[c])]
26:         wc ← mc ∗ dc ∗ dc/vc                                  ▷ elementwise ∗, /
27:         zc ← etac + (yc − etac)/dc                            ▷ elementwise +, −, /
28:         SSR ← SSR + sum(wc ∗ (zc − eta)^2)     ▷ if GLM has an unknown dispersion parameter
29:         {Rbar, bbar} ← updateQR(c, p, Xc, zc, wc, Rbar, bbar)
30:     end for
31:     beta0 ← betahat                            ▷ if GLM has an unknown dispersion parameter
32:     RbarInv ← inverse(Rbar)                                   ▷ using back substitution
33:     betahat ← RbarInv · bbar
34:     phihat ← SSR/df
35:     if ||betahat − beta0|| > epsilon then
36:         if iter < maxit then
37:             iter ← iter + 1
38:             Go to 7
39:         else
40:             print algorithm not converged
41:             return {betahat, Rbar}
42:         end if
43:     else
44:         return {betahat, phihat, Rbar}
45:     end if
46: end procedure
```

**Algorithm S4** `incrIWLS1` is an implementation of the one-pass incremental IWLS for the adjusted score estimation of generalized linear models. It takes as input (a storage location for) $X$, $y$, and the observation weights $m$, the chunk size $c$, an implementation of the inverse link function $F$ and its first and second derivative $f$ and $f'$ (`fdash`), respectively, an implementation of the variance function $V$ and its derivative $V'$ (`Vdash`), starting values for $\hat{\beta}$ (`betahat`), a tolerance $\epsilon$ (`epsilon`), and the maximum allowed number of IWLS iterations (`maxit`). The mJPL estimates are computed if `mjpl` is true. Otherwise, the mBR estimates are computed.

```
procedure incrIWLS1(X, y, m, c, F, f, fdash, V, Vdash, betahat, epsilon, maxit, mjpl)
    Lines 2-28 from Algorithm S3
    if iter > 1 then
        etac0 ← Xc · beta0                                        ▷ · is matrix product
        muc0 ← [F(etac0[1]), . . . , F(etac0[c])]
        dc0 ← [f(etac0[1]), . . . , f(etac0[c])]
        vc0 ← [V(muc0[1]), . . . , V(muc0[c])]
        wc0 ← mc ∗ dc0 ∗ dc0/vc0                                  ▷ elementwise ∗, /
        xR0 ← Xc · RbarInv
        hc0 ← (xR0 · t(xR0)) ∗ wc0       ▷ t(A) is the transpose of matrix A; elementwise ∗
        ddashc ← [fdash(etac[1]), . . . , fdash(etac[c])]
        hkappac ← hc0 ∗ ddashc/(2 ∗ dc ∗ wc)                      ▷ elementwise ∗, /
        zc ← zc + phihat ∗ hkappac                                ▷ elementwise +
        if mjpl = true then
            vdashc ← [Vdash(muc[1]), . . . , Vdash(muc[c])]
            zc ← zc + phihat ∗ (hkappac − hc0 ∗ vdashc/(2 ∗ mc ∗ dc))   ▷ elementwise +, −, ∗, /
        end if
    end if
    Lines 29-45 from Algorithm S3
end procedure
```

**Algorithm S5** `incrIWLS2` is an implementation of the two-pass incremental IWLS for the adjusted score estimation of generalized linear models. It takes as input (a storage location for) $X$, $y$, and the observation weights $m$, the chunk size $c$, an implementation of the inverse link function $F$ and its first and second derivative $f$ and $f'$ (`fdash`), respectively, an implementation of the variance function $V$ and its derivative $V'$ (`Vdash`), starting values for $\hat{\beta}$ (`betahat`), a tolerance $\epsilon$ (`epsilon`), and the maximum allowed number of IWLS iterations (`maxit`). The mJPL estimates are computed if `mjpl` is true. Otherwise, the mBR estimates are computed.

```
procedure incrIWLS2(X, y, m, c, F, f, fdash, V, Vdash, betahat, epsilon, maxit, mjpl)
    Lines 2-32 from Algorithm S3
    for k ∈ {1, 2, ..., K} do                                          ▷ Second pass
        i1 ← (k − 1) ∗ c + 1
        if k < K then
            i2 ← k ∗ c
        else
            c ← n − i1 + 1
            i2 ← n
        end if
        Xc ← X[i1 : i2, 1 : p]
        mc ← m[i1 : i2]
        etac ← Xc · betahat                                            ▷ · is matrix product
        muc ← [F(etac[1]), ..., F(etac[c])]
        dc ← [f(etac[1]), ..., f(etac[c])]
        vc ← [V(muc[1]), ..., V(muc[c])]
        wc ← mc ∗ dc ∗ dc/vc                                           ▷ elementwise ∗, /
        xR ← Xc · RbarInv
        hc ← (xR · t(xR)) ∗ wc            ▷ t(A) is the transpose of matrix A; elementwise ∗
        ddashc ← [fdash(etac[1]), ..., fdash(etac[c])]
        hkappac ← phihat ∗ hc ∗ ddashc/(2 ∗ dc ∗ wc)                   ▷ elementwise ∗, /
        if mjpl = true then
            vdashc ← [Vdash(muc[1]), ..., Vdash(muc[c])]
            hkappac ← 2 ∗ hkappac − phihat ∗ hc ∗ vdashc/(2 ∗ mc ∗ dc)  ▷ elementwise ∗, −, +
        end if
        {Rbar, bbarA} ← updateQR(c, p, Xc, hkappac, wc, Rbar, bbarA)
    end for
    betahat ← RbarInv · (bbar + bbarA)                                 ▷ elementwise +
    Lines 34-45 from Algorithm S3
end procedure
```

## S3 Estimates and estimated standard errors for the parameters of model (14)

Table S1: Estimates and estimated standard errors for the parameters of model (14) through ML, mBR and mJPL with one- and two-pass IWLS implementations. The table also reports the elapsed time, number of iterations, and average time per iteration for each method and implementation. The parameters corresponding to reference categories are set to 0. The first and second column under the ML heading show the ML estimates when allowing for 15 and 20 IWLS iterations.

| | | ML | | mBR | | mJPL | |
|---|---|---|---|---|---|---|---|
| | | | | 1-pass | 2-pass | 1-pass | 2-pass |
| $\alpha$ | | −5.52 | −6.38 | −4.10 | −4.10 | −4.12 | −4.12 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\beta_1$ | January | 0 | 0 | 0 | 0 | 0 | 0 |
| $\beta_2$ | February | −0.04 | −0.04 | −0.04 | −0.04 | −0.04 | −0.04 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_3$ | March | −0.10 | −0.10 | −0.10 | −0.10 | −0.10 | −0.10 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_4$ | April | −0.08 | −0.08 | −0.08 | −0.08 | −0.08 | −0.08 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_5$ | May | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_6$ | June | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_7$ | July | −0.01 | −0.01 | −0.01 | −0.01 | −0.01 | −0.01 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_8$ | August | −0.00 | −0.00 | −0.00 | −0.00 | −0.00 | −0.00 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_9$ | September | −0.09 | −0.09 | −0.09 | −0.09 | −0.09 | −0.09 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_{10}$ | October | −0.13 | −0.13 | −0.13 | −0.13 | −0.13 | −0.13 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_{11}$ | November | −0.15 | −0.15 | −0.15 | −0.15 | −0.15 | −0.15 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\beta_{12}$ | December | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_1$ | Monday | 0 | 0 | 0 | 0 | 0 | 0 |
| $\gamma_2$ | Tuesday | −0.01 | −0.01 | −0.01 | −0.01 | −0.01 | −0.01 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_3$ | Wednesday | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_4$ | Thursday | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_5$ | Friday | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_6$ | Saturday | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\gamma_7$ | Sunday | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\delta_1$ | AA | 2.49 | 3.36 | 1.08 | 1.08 | 1.09 | 1.09 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_2$ | AQ | 0 | 0 | 0 | 0 | 0 | 0 |
| $\delta_3$ | AS | 2.59 | 3.45 | 1.17 | 1.17 | 1.18 | 1.18 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_4$ | CO | 2.33 | 3.19 | 0.91 | 0.91 | 0.92 | 0.92 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_5$ | DL | 2.36 | 3.22 | 0.94 | 0.94 | 0.95 | 0.95 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_6$ | HP | 2.18 | 3.05 | 0.76 | 0.76 | 0.78 | 0.78 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_7$ | NW | 2.44 | 3.30 | 1.02 | 1.02 | 1.04 | 1.04 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_8$ | TW | 2.36 | 3.23 | 0.94 | 0.94 | 0.96 | 0.96 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_9$ | UA | 2.35 | 3.22 | 0.94 | 0.94 | 0.95 | 0.95 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_{10}$ | US | 2.54 | 3.41 | 1.12 | 1.12 | 1.14 | 1.14 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\delta_{11}$ | WN | 2.41 | 3.28 | 0.99 | 0.99 | 1.01 | 1.01 |
| | | (4.89) | (53.18) | (0.35) | (0.35) | (0.36) | (0.36) |
| $\zeta_{(d)}$ | Departure time | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) |
| $\zeta_{(a)}$ | Arrival Time | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) |
| $\rho$ | Distance | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) |
| $\psi_{(d),1}$ | Origin x | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\psi_{(d),1}$ | Origin y | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\psi_{(d),1}$ | Origin z | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 | −0.02 |
| | | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) |
| $\psi_{(a),1}$ | Destination x | −0.00 | −0.00 | −0.00 | −0.00 | −0.00 | −0.00 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\psi_{(a),2}$ | Destination y | −0.05 | −0.05 | −0.05 | −0.05 | −0.05 | −0.05 |
| | | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) |
| $\psi_{(a),3}$ | Destination z | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| | | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) | (0.00) |
| Time (sec) | | 215.55 | 273.09 | 219.23 | 379.66 | 217.04 | 378.72 |
| Iterations | | 15 | 20 | 12 | 12 | 12 | 12 |
| Time/Iteration (sec) | | 14.37 | 13.65 | 18.27 | 31.64 | 18.09 | 31.56 |

## References


Firth D (1993). "Bias reduction of maximum likelihood estimates." *Biometrika*, **80**(1), 27–38. doi:10.2307/2336755.

Golub GH, Van Loan CF (2013). *Matrix computations*. The Johns Hopkins University Press, Baltimore, 4th edition. ISBN 978-1-4214-0794-4.

Kosmidis I, Firth D (2021). "Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models." *Biometrika*, **108**, 71–82. doi:10.1093/biomet/asaa052.

Lumley T (2020). *biglm: Bounded memory linear and generalized linear models*. R package version 0.9-2.1, URL `https://CRAN.R-project.org/package=biglm`.

Miller AJ (1992). "Algorithm AS 274: Least squares routines to supplement those of Gentleman." *Applied Statistics*, **41**(2), 458. ISSN 00359254. doi:10.2307/2347583.