統計研究彙報　第 69 号　2012 年 3 月　（23～52）

多変量外れ値の検出
～繰返し加重最小二乗（IRLS）法による欠測値の補定方法～

和田かず美†

Detection of Multivariate Outliers
– Regression Imputation by the Iteratively Reweighted Least Squares –

WADA, Kazumi

統計調査の欠測値を回帰補定する場合、補定値の推計には最小二乗法（OLS: Ordinary Least Squares）を用いるのが一般的であるが、回帰パラメータ推計時にデータに影響の大きな外れ値が存在する場合、除外など影響を制限するような処理をしなければパラメータの妥当性に問題が生じ、結果として回帰パラメータや補定値が大きく変動することになる。

本稿では、企業財務データを用いて、売上高を従業者数で説明する単回帰モデルを作成し、通常の最小二乗法に代えてロバスト回帰の一種で統計調査の集計実務に適用可能な簡便なアルゴリズムの繰返し加重最小二乗法（IRLS: Iteratively Reweighted Least Squares）を用いることにより、自動的に外れ値の影響を抑えて安定した補定値を得られることを示す。

キーワード：　繰返し加重最小二乗法（IRLS: Iteratively Reweighted Least Squares）、M 推定量、回帰補定、外れ値

In the tabulation process of statistical surveys, missing values are imputed in case they may cause a bias in the final statistical tables. Among various imputation methods, this paper focuses on the regression imputation. The Ordinary Least Squares (OLS) method is widely used for this purpose; however, any extreme outliers can easily influence the OLS Estimator (OLSE). Furthermore, they may distort the validity of the imputation.

Since outliers are often unavoidable, the Iteratively Reweighted Least Squares (IRLS) method is examined in this paper to improve the regression imputation. Enterprise financial statements data are used to impute total sales by number of employees.

IRLS is an algorithm to compute an M-estimator. It is easy to compute, it eliminates or restricts the influence of outliers, and it provides more stable estimation than OLS.

Keywords: Iteratively Reweighted Least Squares (IRLS), M-estimator, Regression imputation, Outlier

原稿受理日　平成 24 年 1 月 10 日

†　独立行政法人統計センター情報技術部研究主幹室

はじめに

政府統計調査において調査票の未回収や未記入により欠測値が存在し、その欠測値が欠落のまま集計すると集計結果が偏る恐れがある場合、欠測値を何らかの方法で補う補定(imputation)という作業を行うことがある。補定方法には様々なものがあるが、ここでは特に回帰モデルにより補定値を推計する回帰補定について取り上げる。

例えば企業や事業所の売上高は、一般に従業者数などの企業規模を示す変数と関係性が高いので、売上高が欠測したときに従業者数による回帰補定を行うことが考えられるが、金額データは外れ値（データの大部分と傾向が異なる異常値）が発生しやすい。このような場合、OLS では推計前に何らかの方法で外れ値を除外しなければならないが、外れ値があってもその影響を制限あるいは緩和することのできるロバスト回帰を利用するという方法もある。

本稿で取り上げる IRLS は、ロバスト回帰法の一種である M 推定量を算出するためのアルゴリズムで、破綻点(break down point)は OLS と同じ 1/n で説明変数の外れ値にも弱いが、計算負荷が低く複雑な統計量の計算を必要とせず、重回帰にも対応できるため、既に広く実用化されている。

政府の統計調査は一般にデータ量が膨大で集計や製表業務を行う担当者は必ずしも統計の専門家とは限らないために、大量のデータ処理には向かず専門知識を必要とする統計ソフトは利用しにくく、上述のような IRLS の長所は実用化の観点から非常に重要である。

本稿の第 I 章では、OLS の問題点と繰返し加重最小二乗法（IRLS: Iteratively Reweighted Least Squares）について取り上げ、第 II 章で使用データと回帰モデルの選択法について解説する。第 III 章では企業財務データを用いて産業別に行った補定の試算とその推計値の安定性を検討し、第 IV 章で結果と考察を述べる。なお、本稿の試算は統計環境 R のバージョン 2.11.1 上で行い、作成した R2.14.0 上で動作確認済みの IRLS 関数のコードとその使用例を別紙として添付する。

Ⅰ．方法論

統計調査データは図 1 に示すような流れで処理されており、最初にデータのチェックが行われる。外れ値の検出もこの段階で行われ、検出されたデータに何らかの誤りがある場合は数値が修正される。問題のデータが誤りではなく単に他の大部分のデータと傾向が異なる場合はそのまま集計されるが、このような外れ値をそのまま OLS を用いた補定値の推計にも使用すると、推計結果に大きな影響を与える可能性がある。一方で、補定時の推計からこのような外れ値を除外しようとすると、何らかの方法で数多い補定ドメイン毎に正常値の範囲を設定しなければならない。

このような問題に対する一つの対処法として、政府統計部局のデータチェック作業を改善しコスト削減に資することを目的に国連が 1997 年に刊行した Statistical Data Editing, Volume No.2 の中で、Bienias らがロバスト回帰の一種である繰返し加重最小二乗法(IRLS)を紹介している。

図１．統計調査データの集計の流れ

１． 最小二乗法（OLS）と外れ値

欠測を補定したい目的変数を y_i、説明変数を x_i として(1)式のような線形回帰モデルを考える。誤差項 ε_i が期待値 0 で分散一定、自己相関がなく正規分布という仮定を満たしていれば、回帰パラメータ $\boldsymbol{\beta}$ の最小二乗推定量(OLSE)は最良線形不偏推定量(BLUE: Best Linear Unbiased Estimator)である。誤差の正規性の前提が崩れたとしても OLS 推定量は一致性をもつため、回帰パラメータ推定には OLS が広く使用されている。ただし、誤差項が正規分布よりも裾が長いときなど OLS による回帰分析に大きな影響を与える外れ値が存在する場合、OLSE は一致推定量であっても有効推定量（最小の分散をもつ不偏推定量）ではなくなり、結果の妥当性に問題が生じる可能性がある。

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i \qquad (1)$$

実例として、データ数 100、平均が 100、分散 1、相関 0.7 の 2 変量正規乱数データを作成し、少し外れたところに 2 つの外れ値を人工的に付与し単回帰分析を行った結果を図 2 に示す。黒点が正規乱数データ、赤点が人工外れ値を示している。OLS で正規乱数の 100 データだけを使用して回帰直線を求めると橙の実線になるが、外れ値を含めた 102 データ全てを使用した回帰線は赤の点線になり、二つの外れ値が推計に与える影響の大きさがわかる。

一方、IRLS を用いて 102 データ全てを使用して求めた回帰線は青の点線となる。表 1 には上述の三通りの試算による回帰パラメータの推計値 $\hat{\beta}_0$（切片）及び $\hat{\beta}_1$（傾き）の値を示した。IRLS の結果は正常値のみの OLS の結果と完全に一致はしないが、かなり近い値が得られることがわかる。

図２．**OLS** の問題点と **IRLS** の効果

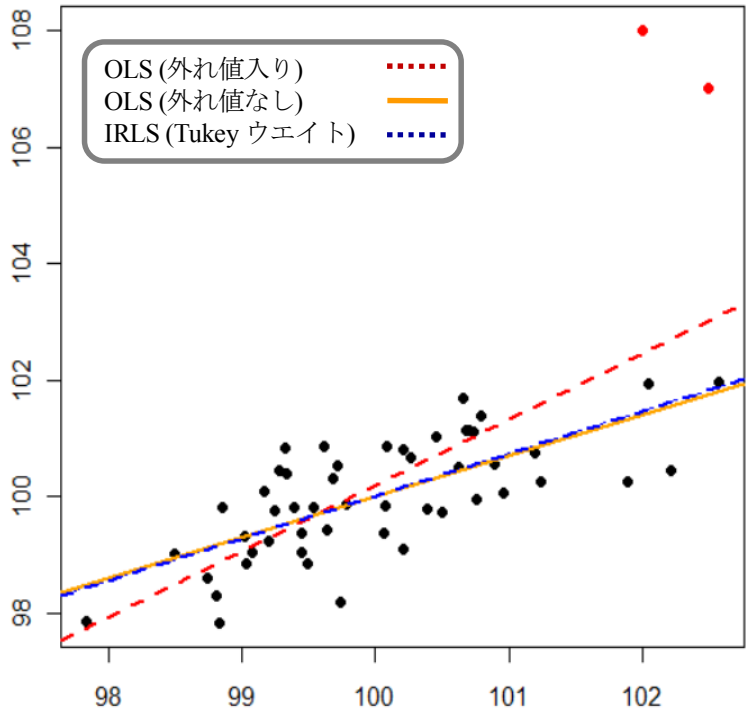


表１．回帰パラメータの推定値

	切片($\hat{\beta}_0$)	傾き($\hat{\beta}_1$)
OLS (外れ値入り)	4.74	0.95
OLS (外れ値なし)	30.00	0.70
IRLS (Tukey, c=8)	29.94	0.70
IRLS (Tukey, c=4)	29.10	0.71

２． M推定量について

Huber(1964)は、位置パラメータ（平均値ベクトル）のロバスト推計について体系的に論じ、以下のようなM推定量を提案、その一致性と漸近正規性を証明した。

標本$x_i, \cdots, x_n$は、互いに独立で同一の分布に従うものとする。標本平均Tは、$T = \sum_i x_i / n$ により得られるが、これは$\sum_i (x_i - T)^2$を最小化するTを求めることによっても得ることができる。このとき、x_iに外れ値があれば、標本平均Tはその外れ値が他の大部分のデータから乖離すればするほど無制限に大きな影響を受けることになるので、標本平均はロバストではない。ここにデータの影響を制御するための関数ρを導入し、$\sum_i \rho(x_i - T)$ を最小化するTを求めるというのがM推定量の考え方である。ρは損失関数と呼ばれる。

Huber(1973)は、位置パラメータについての M 推定量の考え方を線形回帰モデルに拡張し、Holland & Welsch(1977)は、このM推定量とその計算アルゴリズムであるIRLSについて、以下のように整理している。

(1)式の線形回帰モデルを、行列の形で以下のように表現する。

$$y = \mathbf{X}\boldsymbol{\beta} + \varepsilon \qquad (2)$$

ここで、nはデータ数、pは説明変数の数で

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} 1 & x_{11} \cdots x_{1p} \\ \vdots & \vdots \ddots \vdots \\ 1 & x_{n1} \ldots x_{np} \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

とすると、(2)式の回帰パラメータ$\boldsymbol{\beta}$の M 推定量は、x_iを行列$\mathbf{X}$の i 行目として、下式の最小化により得られる。σはスケールパラメータと呼ばれて誤差分散の尺度を示し、下式の$y_i - x_i\boldsymbol{\beta}/\sigma$は$\varepsilon_i / \sigma$であり、つまり誤差を$\sigma$で標準化している。

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - x_i\boldsymbol{\beta}}{\sigma} \right)$$

損失関数ρが微分可能で、0 のまわりで対称な凸関数のとき、$\boldsymbol{\beta}$の推定値である$\hat{\boldsymbol{\beta}}$は下式を解

いて得られる。損失関数ρを微分して得られるψは影響関数と呼ばれる。

$$\sum_{i=1}^{n} x_{ij}\psi\left(\frac{y_i - x_i\hat{\boldsymbol{\beta}}}{\sigma}\right) = 0 \quad for\ all\ j \qquad (3)$$

(3)式を解くためには繰返し計算が必要となる。いくつかの計算方法があるが、Holland & Welsch(1977)や Bienias ら(1997)は、理論的に望ましいが計算が難しいニュートン法や収束の遅い Huber 法ではなく、Beaton & Tukey(1974)が提案した計算しやすく既存の加重最小二乗法(WLS)を利用できる IRLS を採用した。これは、ウエイト関数wを$w(e) = \psi(e)/e$、< >で囲んだ部分はウエイトを対角成分とした n×n の正方行列として、適当な初期値$\hat{\boldsymbol{\beta}}^{(0)}$を用いてより良い次の推定値$\hat{\boldsymbol{\beta}}^{(1)}$を算出し、収束するまで繰り返して推定値を改善する。

$$\hat{\boldsymbol{\beta}}^{(j)} = \hat{\boldsymbol{\beta}}^{(j-1)} + \left(\mathbf{X}^T\left\langle w\left(\frac{y_i - x_i\hat{\boldsymbol{\beta}}^{(j-1)}}{\sigma}\right)\right\rangle\mathbf{X}\right)^{-1}\mathbf{X}^T\left\langle w\left(\frac{y_i - x_i\hat{\boldsymbol{\beta}}^{(j-1)}}{\sigma}\right)\right\rangle(y - \mathbf{X}\hat{\boldsymbol{\beta}}^{(j-1)})$$

３．繰返し加重最小二乗法(IRLS)のアルゴリズム

本稿では、Bienias ら(1997)に準拠し、以下のアルゴリズムの IRLS 関数を使用する。スケールパラメータσには残差eの平均絶対偏差s、ウエイト関数wは Tukey の biweight 及び Huber ウエイトの二種類を使用している。

1)　初期値算出

OLS により回帰係数を算出し、それを初期値$\hat{\boldsymbol{\beta}}^{(0)}$とする。このときの残差$e_i^{(0)}$の平均絶対偏差$s^{(0)}$と、0 から 1 までの値をとる IRLS ウエイト$w_i^{(0)}$を算出する。w_iの算出法については第 3 節で取り上げるが、各データポイントが回帰線からどの程度乖離しているかにより値が決まり、回帰線から遠いほど小さい値をとるので外れ値の影響に制約を加えることができる。

2)　繰返し 1 回目

$w_i^{(0)}$を用いた WLS により$\hat{\boldsymbol{\beta}}^{(1)}$を算出。併せて、残差$e_i^{(1)}$の平均絶対偏差$s^{(1)}$及び新たな IRLS ウエイト$w_i^{(1)}$を算出する。

3)　繰返し j 回目

j 回目の繰返しの際，j－1 回目の残差$e_i^{(j-1)}$とその平均絶対偏差$s^{(j-1)}$により算出した IRLS ウエイト$w_i^{(j-1)}$を用いて下式により回帰係数$\hat{\boldsymbol{\beta}}^{(j)}$を求める．ここで、$\mathbf{W}^{(j-1)} = diag\{w_i^{j-1}\}$とする。$\mathbf{W}^{(j-1)}$は対角成分が$w_i^{(j-1)}$、それ以外は 0 の n×n 行列である。

$$\hat{\boldsymbol{\beta}}^{(j)} = \left[\mathbf{X}^T\mathbf{W}^{(j-1)}\mathbf{X}\right]^{-1}\mathbf{X}^T\mathbf{G}\mathbf{W}^{(j-1)}\mathbf{y} \qquad (4)$$

4)　収束条件

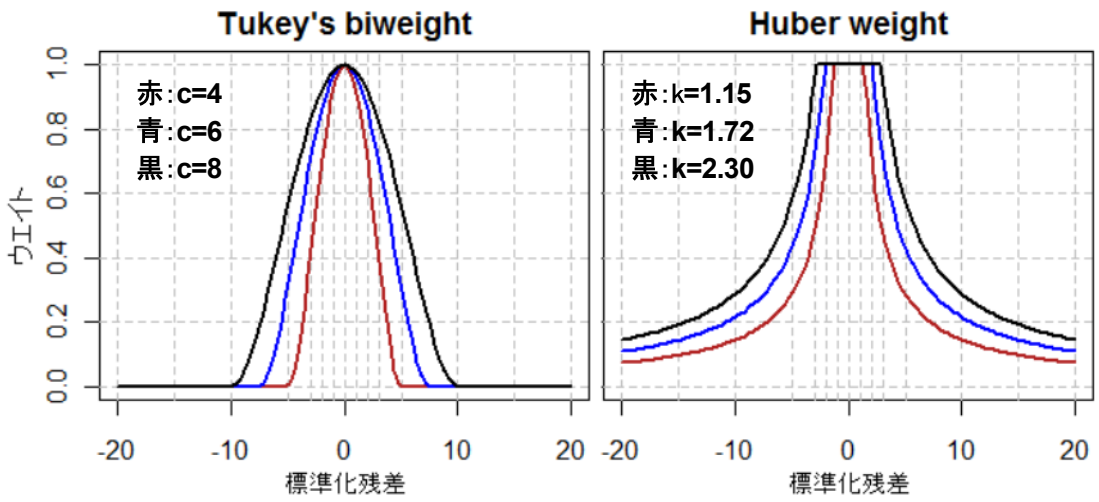
$s^{(j)}/s^{(j-1)}$ が収束するまで 3)を繰り返す。ここでは収束条件を Bienias ら(1997)と同様に 0.01 未満とした。

３． ウエイト関数

IRLS のウエイト関数についてはさまざまなものが提案されているが、本稿ではその中で代表的な Tukey の biweight 関数と、Huber のウエイト関数の二つを使用した。関数形は表２に示す。

Tukey の biweight は、Beaton & Tukey(1974)が提案し、Bienias ら(1997)が採用している。ロバスト性を制御するためにユーザーが任意で設定する調整定数 c の値は大きいほどロバスト性が緩やかになり、Bienias ら(1997)は経験則から 4 から 8 の間で設定することを推奨している。biweight 関数を使用した M 推定量は、データが正規分布に従いスケールパラメータ σ が標準偏差のとき、c=4.685 で漸近有効性が 95%になる。この Tukey ウエイトは、初期値により局所解に陥ったり、その構造上収束せずに無限ループを起こす可能性もあるが、図３に示すとおり、ある程度以上回帰線から垂直距離が遠いデータポイントのウエイトに 0 が付与されるので、極端な外れ値はその影響を完全に排除できるという性質を持つ。

図３．ウエイト関数の形状



一方、Huber ウエイトは Huber(1964)により考案され、Huber(1973)で回帰モデルの IRLS に適用されている。初期値によらず大局解に必ず収束する。調整定数 k は、データが正規分布に従い、スケールパラメータ σ が標準偏差のとき、k=1.345 で漸近有効性が 95%になる。Tukey ウエイトの場合は回帰線から少しでも離れるとウエイトが 1 から削られていくが、Huber ウエイトの場合は回帰線の近くでウエイトが 1 の領域があり、さらに分布の裾部分が非常に長く、相当回帰線から離れたデータポイントでも、ウエイトは小さくはなるが 0 にはならないという性質を持つ。

ここではスケールパラメータ σ に平均絶対偏差 s を使用するが、標準偏差を σ_{SD} とすると両者には(5)式のような関係があるので、s について漸近有効性が 95％となる c と k は、$c = 4.685/0.8 \approx 5.856$ 及び $k = 1.345/0.8 \approx 1.681$ ということになる。これらの値から、c=4～8 の場合に対応する k の値を算出し、試算時に使用する調整定数の値を表 3 のように設定した。

$$\frac{s}{\sigma_{SD}} = \frac{E|e|}{\sqrt{E(e^2)}} = \sqrt{\frac{2}{\pi}} \approx 0.80 \qquad (5)$$

表２．損失関数・影響関数及びウエイト関数

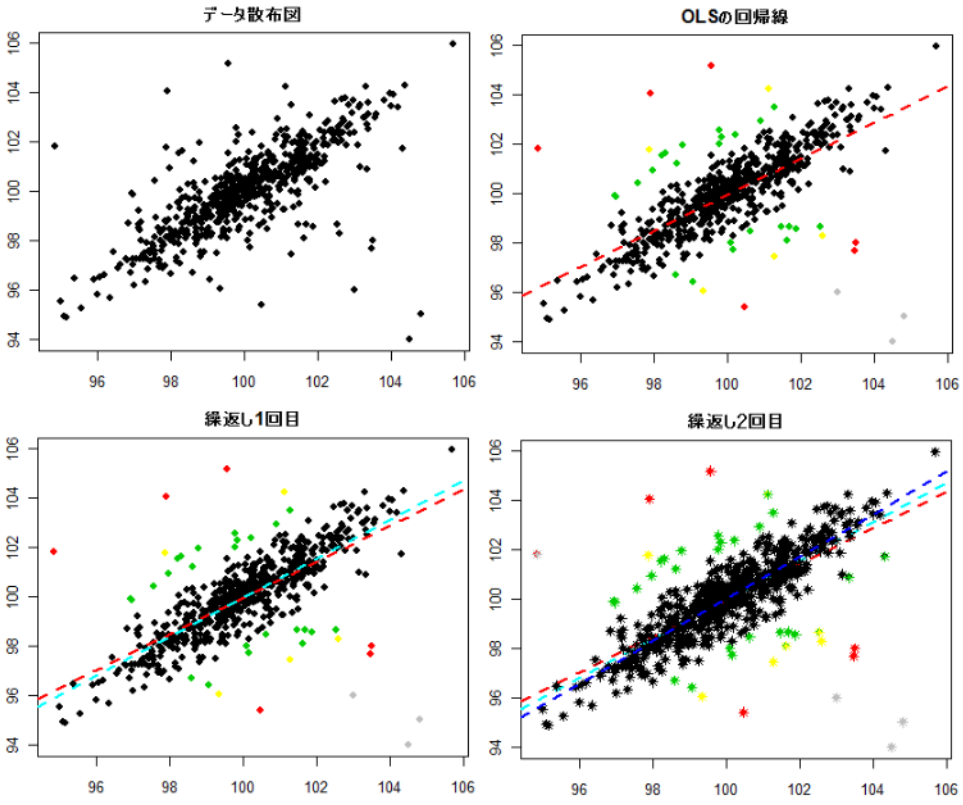
	Tukey の biweight	Huber ウエイト
損失関数 ρ	$\rho(e) = \begin{cases} \frac{c^2}{6}(1-[1-\left(\frac{e}{c}\right)^2]^3) & \lvert e\rvert \le c \\ \frac{c^2}{6} & \lvert e\rvert > c \end{cases}$	$\rho(e) = \begin{cases} \frac{e^2}{2} & \lvert e\rvert \le k \\ k\lvert e\rvert - \frac{k^2}{2} & \lvert e\rvert > k \end{cases}$
影響関数 ψ	$\psi(e) = \begin{cases} e\left[1-\left(\frac{e}{c}\right)^2\right]^2 & \lvert e\rvert \le c \\ 0 & \lvert e\rvert > c \end{cases}$	$\psi(e) = \begin{cases} e & \lvert e\rvert \le k \\ k & \lvert e\rvert > k \\ -k & \lvert e\rvert < -k \end{cases}$
ウエイト関数 w	$w(e) = \begin{cases} \left[1-\left(\frac{e}{c}\right)^2\right]^2 & \lvert e\rvert \le c \\ 0 & \lvert e\rvert > c \end{cases}$	$w(e) = \begin{cases} 1 & \lvert e\rvert \le k \\ \frac{k}{\lvert e\rvert} & \lvert e\rvert > k \end{cases}$

表３．調整定数の基準値と試算条件

	漸近有効性 95%の値		試算条件 ［平均絶対偏差 s ］		
	標準偏差 σ_{SD}	平均絶対偏差 s			
Tukey の c	4.685	5.856	4	6	8
Huber の k	1.345	1.681	1.15	1.72	2.30

図 4 に、単回帰モデルに従い誤差項に裾の長い t 分布を使用した擬似乱数データについて IRLS をステップごとに適用した結果を図示している。左上のような分布のデータについて、OLS 推定量を算出した結果が右上の図の赤い点線である。これを初期値として IRLS ウエイトを Tukey の biweight により算出し、WLS により得た回帰線が左下の図の水色の点線になる。各データポイントに付与された IRLS ウエイトも色分けで示した。回帰線が動けば各データポイントの残差の値も変わるので、再び新たな残差から IRLS ウエイトを算出して新たに右下の図の青い点線のような回帰線を得た。このときの IRLS ウエイトはアスタリスク(*)印への色付けで示している。この時点で平均絶対偏差の変化率が 0.01 よりも小さくなり、このデータについては初期値計算を含めて繰返し三回で収束した。

図４．IRLS の仕組み



Ⅱ．使用データとモデルの選択

１．使用データ

企業売上高を従業者数により推計する回帰補定を想定し、(株)東京商工リサーチの企業財務データを用いて産業別にモデル選択を行い、OLS と IRLS について補定値を算出し、その安定性を比較した。使用した企業財務データは 6775 社分、うち 5524 社分は従業者数 500 人以上の国内企業全数、1251 社はおおむね 50 人以上の中小企業で、産業・従業者階級による層別にランダム抽出をしており、2002 年 3 月期決算データを最新として、直近三期分の売上高と最新決算期の従業者数データ及び産業分類（1993 年第 10 回改定版）の情報を持つ。分析を行ったデータの産業区分及び区分毎のデータ数は第 3 節の表 5 に示した。

２．回帰補定モデルの候補とその推計式

補定を行う場合、使用するモデルは単純なものが望ましい。企業売上高は、企業規模に関する情報で大部分が説明できることが多く、残差と説明変数に相関も残らなかったため、説明変数は従業者数のみとした。企業売上高は企業規模が大きくなるほど誤差分散が大きくなる傾向があるため、等分散を前提とした単純な単回帰モデルではなく、誤差項が説明変数に応じて大きくなる比推定モデルや、データ変換により誤差項を等分散化し偏りのある分布を対象にするようなモデルを候補とした。候補となる四種類のモデルとその推定値の推計式を以下に示す。適合モデルの選択は第 3 節の表 5 に示す産業区分別に行った。p は回帰パラメータの数で、比推定モデルの B では 1、線形モデルの C 及び D は 2 となる。

A. 変換のない比推定モデル

モデル式　$y/x = \beta_0 + \varepsilon$

補定値の推計式　$\hat{y}_i = \hat{\beta} \cdot x_i$

B. 平方根変換＋比推定モデル

モデル式　$\sqrt{y}/\sqrt{x} = \beta_0 + \varepsilon$

補定値の推計式　$\hat{y}_i = (\beta_0^{\,2} + \dfrac{\sum_{i=1}^{n} e_i^{\,2}}{n-p})x_i$　(6)

C. 平方根変換＋線形モデル

モデル式　$\sqrt{y} = \beta_0 + \beta_1\sqrt{x} + \varepsilon$

補定値の推計式　$\hat{y}_i = \hat{\beta}_0^2 + \hat{\beta}_1^2 x_i + 2\hat{\beta}_0\hat{\beta}_1\sqrt{x_i} + \dfrac{\sum_{i=1}^{n} e_i^{\,2}}{n-p}$　(7)

D. 対数変換＋線形モデル

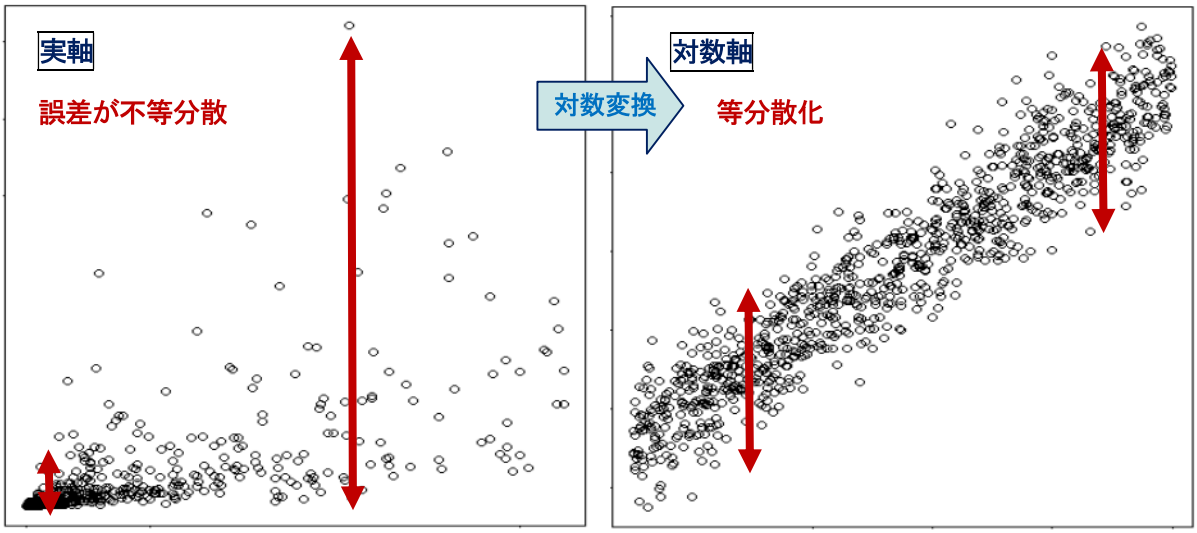
モデル式　$\log y = \hat{\beta}_0 + \hat{\beta}_1 \cdot \log x + \varepsilon$

補定値の推計式

$$\hat{y}_i = \exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot \log x_i) \cdot \exp(\frac{1}{2} \cdot \frac{\sum_{i=1}^{n} e_i^{\,2}}{(n-p)}) \quad (8)$$

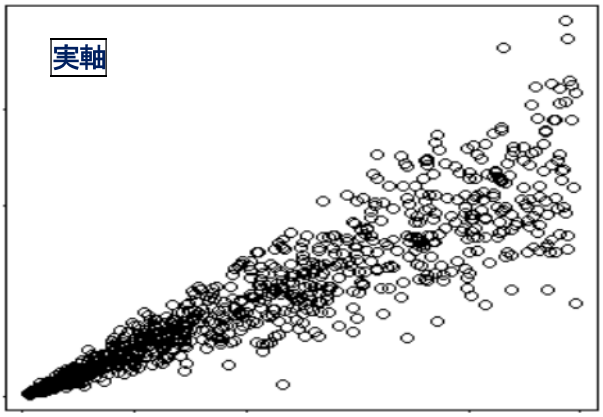
図 5 は、擬似乱数により作成した、対数変換後に線形回帰に従うデータの散布図である。左図のように X 軸のデータが大きくなるにつれて誤差分散が大きくなり、データの形が扇形に広がってしまうような場合に、適切なデータ変換により右図のようにデータが線形で等幅の帯状になれば、その変換データは線形回帰モデルが適合する。

図５．対数変換すると線形回帰モデルに従うデータの散布図



一方で、比推定モデルは変換しない場合でも説明変数と比例して大きくなる誤差項を持つ。図 6 は、乱数を無変換で比推定モデルに従うデータの散布図である。

図６．比推定モデルに従う乱数データ



３．モデル選択の方法

ここで示すモデル選択法は、総務省の岡本政人氏にご教授いただいた。ただし、以下の記述に何らかの不備や誤りがあれば、その責任は筆者にある。

（1）データ変換の選択

全ての候補モデルについて説明変数は 1 つで、データ変換は無変換、平方根変換及び対数変換の三種類である。このため y_i/x_i、$\sqrt{y_i}/\sqrt{x_i}$、$\log(y_i)/\log(x_i)$ について正規性検定（あるいは正規分布への適合度検定）を行い、最も正規性が高く p 値が大きくなるデータ変換法を選択する。

単変量データの正規性の検定方法には多くの種類があるが、Geary 検定や D'Agostino-Pearson 検定などのモーメントを使用するタイプのものは、正規乱数に 1 つ外れ値を加えるだけで p 値が極端に小さくなってしまうため、外れ値を含むことを前提としたデータの検定には適していない。一方で、Kolmogorov-Smirnov 検定に代表されるような経験分布関数を使用する検定法は、順序統計量を使用するノンパラメトリックな方法で、若干の外れ値でモーメント検定のように極端な p 値の変化を起こさないので、正規性があまり高くないデータについても比較を行うことができる。

本稿では、表 4 に示す R の既存の関数を用いて比較を行い、主に Kolmogorov-Smirnov 検定を改良した Lilliefors 検定と、参照用に標本分散と順序統計量の両方を使う Shapiro-Wilk 検定の結果を参照してデータ変換方法の選択を行った。

一般的に、経験分布関数を使用する検定法の中では Lilliefors 検定よりも Anderson-Darling 検定が良いとされるが、以下に示す R の関数を用いた場合、p 値が小さい場合に Anderson-Darling 検定の関数の戻り値が非数(NaN)になり、正規性の低いデータについて比較ができないことがある。

産業別に検定を行ったところ、同じデータでも p 値の大きさは検定の種類によりかなり異なるが、データ変換の種類についての p 値の大小関係については検定の種類にかかわらず同じ結果が得られた。

表４．正規性の検定に使用した R の関数

検定の種類	関数	収録パッケージ	備考
Lilliefors	lillie.test	nortest	データ数 5 以上
Anderson-Darling	ad.test	nortest	データ数 8 以上。非数(NaN)の戻り値がある。
Shapiro-Wilk	shapiro.test	stats	データ数 4～5000。

（2）モデル選択

適合するデータ変換を行った上で、比推定及び単回帰モデルを適用し、Q-Q プロットや残差プロット、実軸及び変換軸での回帰線入りの散布図を作成し、それらをもとに適合モデルを総合的に判断した。

残差プロットは、残差分散比推定モデル［A 及び B］に適合する場合、残差を変換済みの説明変数で割ったものを Y 軸、変換済み説明変数を X 軸にプロットした散布図が、X 軸を中心として等幅の帯状に近くなる。一方、線形回帰モデル［C 及び D］の場合は、残差と変換した説明変数

の散布図が同じように等幅の帯状に近くなる。また、残差と目的変数をプロットして何らかの関係性が残っている場合は説明変数の不足を示唆するが、今回のデータでそのような産業はみられなかった。

IRLS の場合、データに回帰線からの垂直距離に応じてウエイトを付与しているので、データに適合しないモデルを使用した場合、モデルに合わない部分のデータのウエイトを削って強引にモデルにデータを合わせてしまうことになる。このため、合わないモデルを使用する弊害は OLS の場合よりも大きい。また同じ理由で IRLS の場合は誤差分散や決定係数などの数値から判断することはできず、上述のような各種プロット図による総合判断が必要となる。

産業別のデータ変換及びモデル選択の結果は、表 5 のとおり。

表５．適合モデルと期別データ数一覧

産業	適合モデル	データ数		
		当期	1 期前	2 期前
[1] 全産業	対数変換＋線形	6775	6627	6456
[2] D 鉱業	対数変換＋線形	56	54	54
[3] E 建設業	平方根変換＋比推定	415	414	411
[4] F 製造業	対数変換＋線形	1902	1882	1860
[5] G 電気ガス熱水道	対数変換＋線形	72	70	67
[6] H1 運輸業	対数変換＋線形	577	568	552
[7] H2 通信業	対数変換＋線形	95	90	82
[8] I1 卸売業	対数変換＋線形	517	509	500
[9] I2 小売業	平方根変換＋比推定	600	591	574
[10] I3 飲食業	平方根変換＋比推定	163	159	156
[11] J 金融・保険業	対数変換＋線形	488	472	458
[12] K 不動産業	対数変換＋線形	209	205	201
[13] L サービス業	対数変換＋線形	1681	1613	1541
[14] L01 宿泊業	平方根変換＋比推定	53	51	50
[15] L02 娯楽業	平方根変換＋線形	65	65	64
[16] L03 情報サービス・調査業	対数変換＋線形	302	296	283
[17] L04 専門サービス業	対数変換＋線形	131	127	118
[18] L05 協同組合	対数変換＋線形	128	117	108
[19] L06 その他の事業サービス業	対数変換＋線形	386	382	371
[20] L07 医療業	平方根変換＋比推定	264	246	238
[21] L08 教育業	対数変換＋線形	105	99	88

Ⅲ．IRLS による回帰補定

本章では、第 II 章で行ったモデル選択の結果に基づき、産業別に適合モデルにより IRLS 及び OLS により補定のための回帰パラメータを算出し、補定値の安定性を確認する。

１．IRLS の適用例

IRLS の効用は、外れ値の影響コントロールである。以下に、産業別の結果の中から、外れ値のパターン別に IRLS 適用の効果を整理し、代表的な散布図を実軸及び変換軸で例示した。データの分散が小さく外れ値が存在しないときは、OLS も IRLS もあまり変わらない結果が得られる。一方、たとえ梃子比が小さくても、外れ値は補定値に影響を与えることが確認できた。

散布図作成にあたり、使用した企業売上高は全て当期で、IRLS は極端な外れ値の影響を完全排除できるために Huber ウエイトよりも OLS との乖離が大きくなる Tukey の biweight 関数による結果を示した。

(１)　外れ値がないとき

データ分散が小さくモデルの当てはまりも良く、IRLS を適用しても大きくウエイトが削られる外れ値が存在しないようなデータの場合は、OLS で補定値を算出して何の問題もない。代表例として、図 7 に L03 情報サービス・調査業の散布図を示す。変換軸でのプロットも実軸でも、OLS と IRLS の回帰線の乖離はかなり小さい。

(２)　梃子比の大きな外れ値があるとき

このような外れ値があるとき、推計値への影響は最も大きくなる。

図 8 は L04 専門サービス業の散布図で、上の変換軸プロット図で右下の方に赤丸で囲んだグレーのデータポイントが三つ存在している。これらは、Tukey のウエイト関数では調整定数 c の値にかかわらずウエイトに 0 が付与されている極端な外れ値で、梃子比も大きい位置にあるため、変換軸上でも回帰線の傾きに大きく影響を与えていることがわかる。

(３)　梃子比が小さい外れ値がある例

J 金融・保険業の例を図 9 に示す。外れ値の梃子比が小さく、変換軸上で OLS と IRLS の回帰直線にさほど乖離がないように見えても、外れ値があれば誤差分散が大きくなる。データ変換を要するモデルを適用している場合、推計値の算出時に式(6) 、(7)及び(8)に示すように残差分散による推計値の補正が必要なので、誤差分散が大きい場合に補定のための推計値は上方に偏ってしまう。一方で、IRLS の場合は外れ値の残差分散もウエイト付けにより抑制されるために補定値は過大になりにくいという特徴がある。

図７．外れ値がない例：L03 情報サービス・調査業

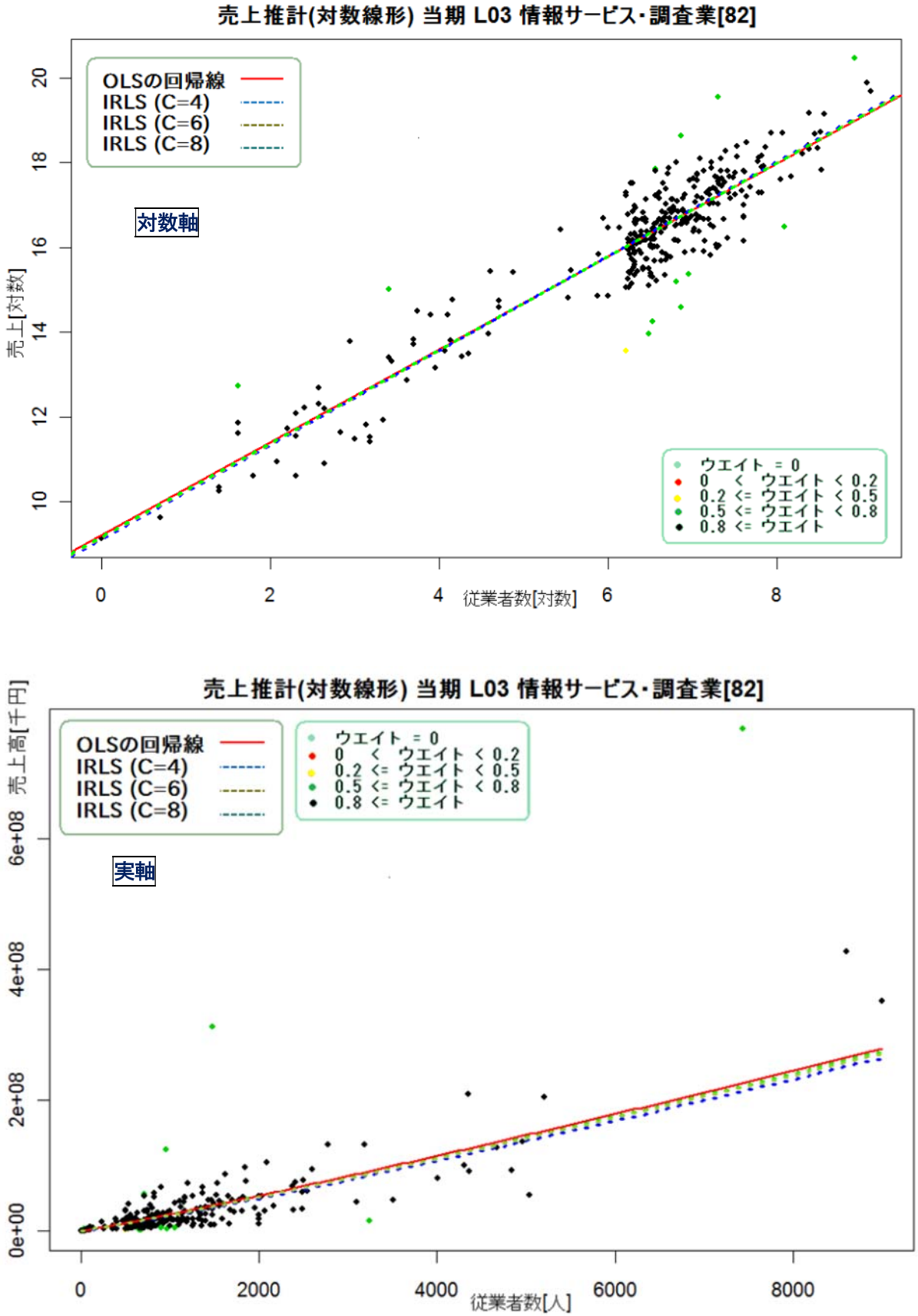


図８．梃子比の大きな外れ値がある例：L04 専門サービス業

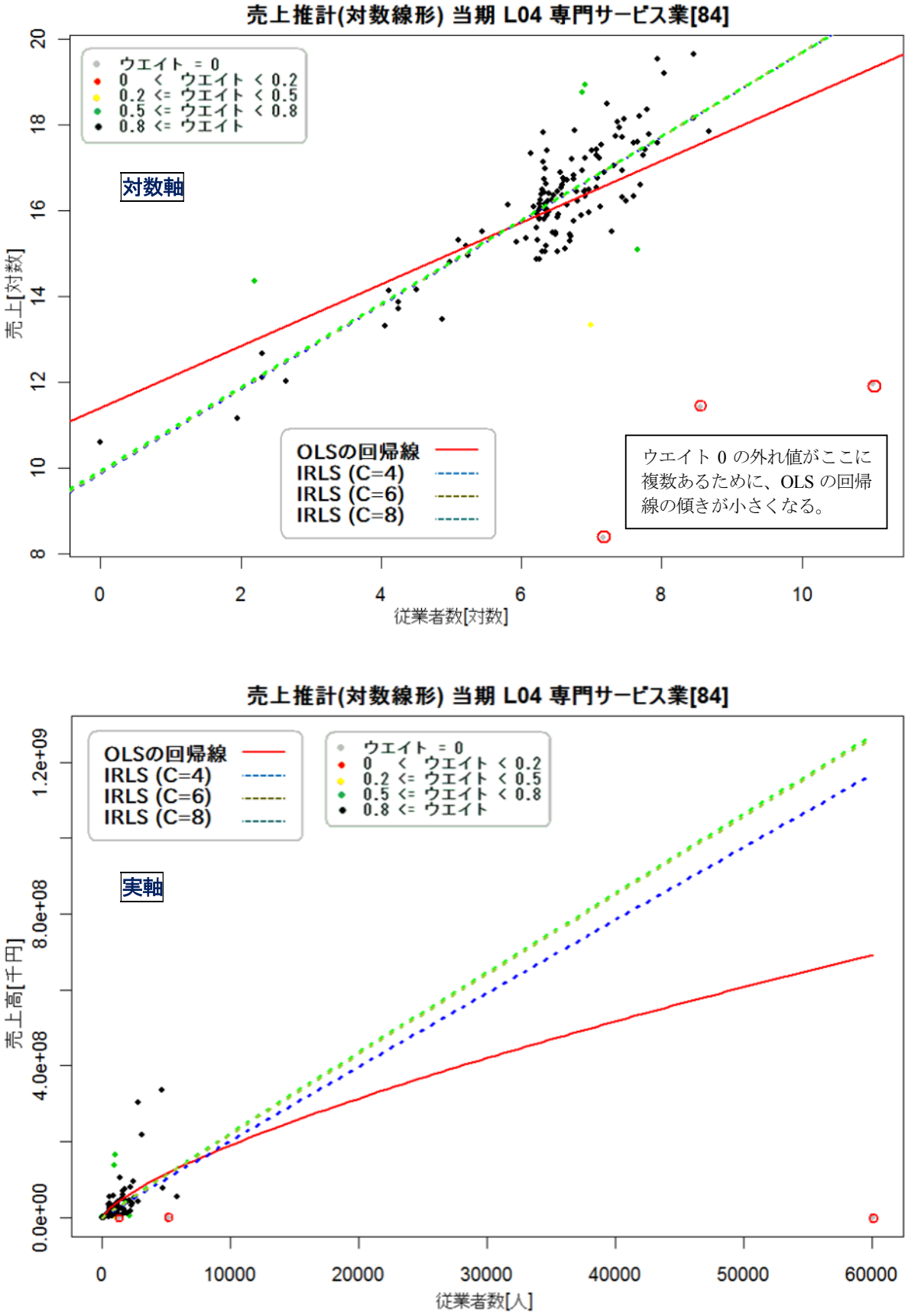
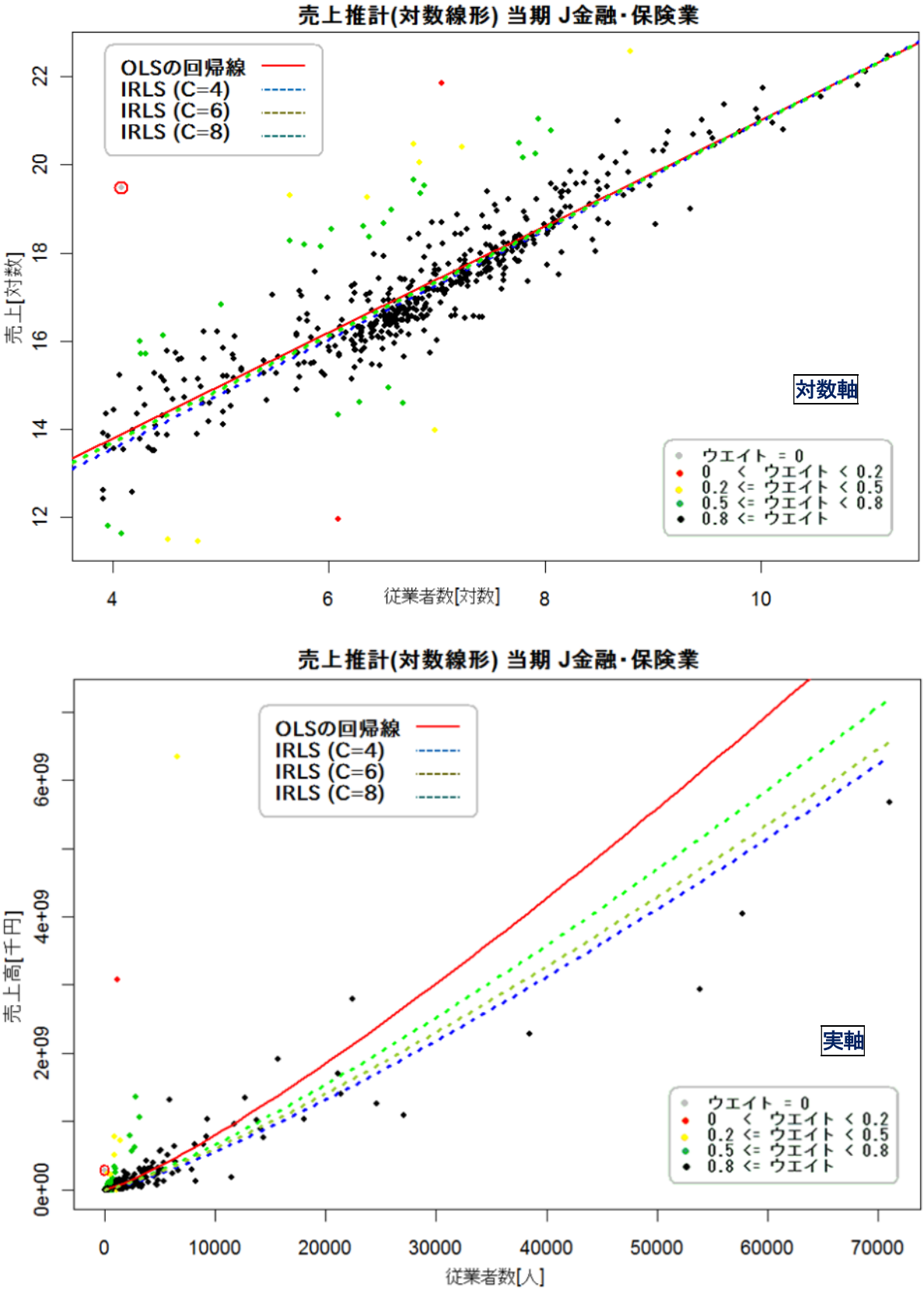


図９．梃子比が小さい外れ値がある例：J 金融・保険業



２．推計値の安定性

OLS、IRLS(Tukey ウエイト, c=8)及び IRLS(Huber ウエイト, k=2.30)についてそれぞれ産業別・期別に適合モデルの回帰パラメータを推計し、全ての従業者数x_iについて補定値となる売上高の推計値$\hat{y}_i$を算出し、t 期の推計値平均M^tを算出する。次に当期を t=0、1 期前を t=-1 、2 期前を t=-2 とすると、M^0、M^{-1}、M^{-2}の 3 データで産業別の標準偏差$SD(M)$を求め、その数値により推計値の安定性を比較した。もともとデータは期毎に変動するものだが、外れ値が推計に影響を与えれば、OLS による推計値の変動が過大になる可能性がある。

産業別標準偏差$SD(M)$の値と、OLS の数値と比較した IRLS の数値の割合を表６に示す。

表６．推計値の安定性

産業分類	OLS	IRLS(Tukey, c=8)		IRLS(Huber, k=2.30)	
	$SD(M_{OLS})$	$SD(M_{TK8})$	$\frac{SD(M_{TK8})}{SD(M_{OLS})}$	$SD(M_{HB8})$	$\frac{SD(M_{HB8})}{SD(M_{OLS})}$
全産業	3027060	2757976	91.1%	2637967	87.1%
D 鉱業	447173	372625	83.3%	351178	78.5%
E 建設業	1953614	1991757	102.0%	2001045	102.4%
F 製造業	6122484	5404287	88.3%	5023427	82.0%
G 電気ガス熱水道	41802857	18955342	45.3%	24262074	58.0%
H1 運輸業	860311	1298195	150.9%	1061530	123.4%
H2 通信業	254148840	32575217	12.8%	67034575	26.4%
I1 卸売業	3567877	4497210	126.0%	4578660	128.3%
I2 小売業	2795270	2498792	89.4%	2497575	89.4%
I3 飲食業	561462	632154	112.6%	609382	108.5%
J 金融・保険業	11982947	8460241	70.6%	9080495	75.8%
K 不動産業	2420972	908766	37.5%	1250169	51.6%
L サービス業	1749592	524310	30.0%	1333100	76.2%
L01 宿泊業	390623	317858	81.4%	285112	73.0%
L02 娯楽業	4768370	8093258	169.7%	8363391	175.4%
L03 情報サービス・調査業	2670112	2518226	94.3%	2541856	95.2%
L04 専門サービス業	5105057	2164155	42.4%	2425667	47.5%
L05 協同組合	4860718	3836977	78.9%	4290818	88.3%
L06 その他の事業サービス業	1387513	1130346	81.5%	797309	57.5%
L07 医療業	540899	522910	96.7%	530070	98.0%
L08 教育業	252098	290183	115.1%	335834	133.2%
総平均(D 鉱業～L サービス業)	74472996	11383439	15.3%	20868917	28.0%

全産業を含め 21 の産業区分のうち、想定どおり IRLS の値が OLS よりも小さくなったものが 15 産業あり、大部分の産業で IRLS の利用により補定のための推計値が安定するといえる。

IRLS が OLS よりも変動が大きくなった 6 産業については、その原因を確認したところ、以下

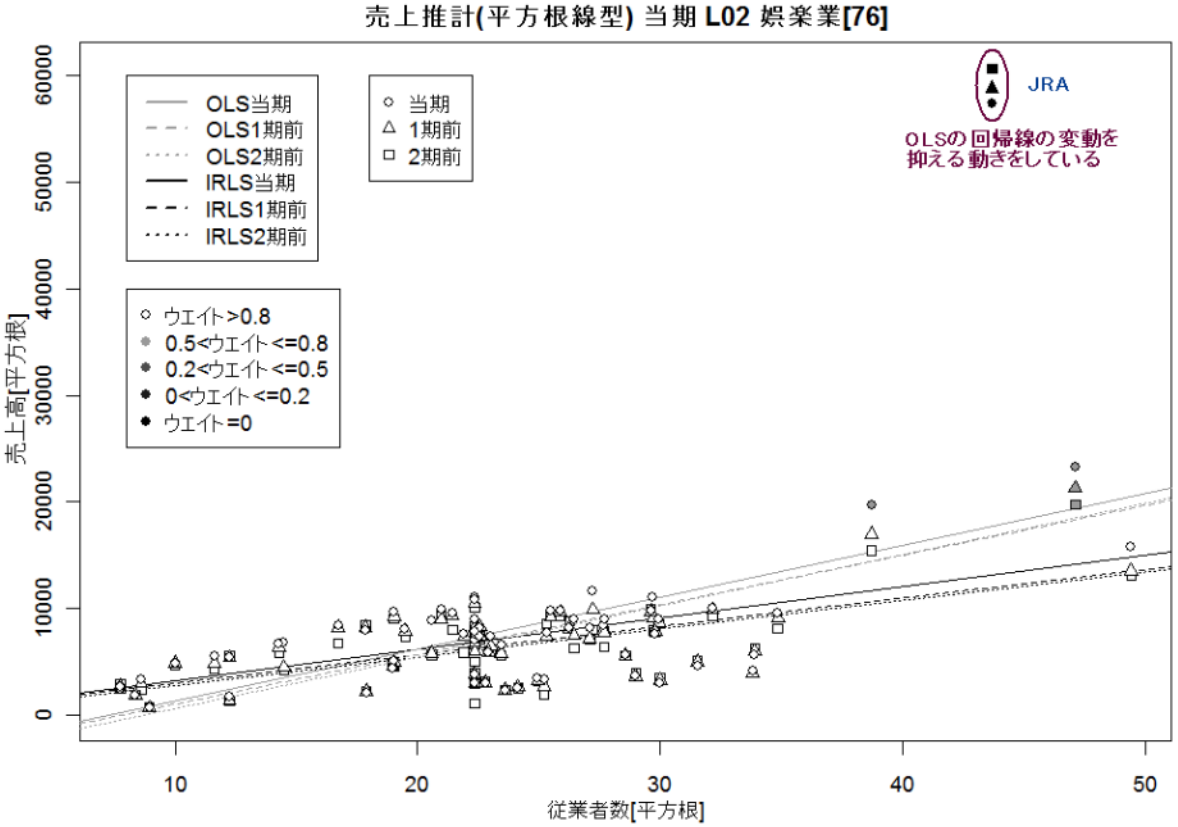
の二つの場合に分類することができた。

(1)　大部分のデータの動きと逆の傾向を持つ少数の極端な外れ値が存在する場合

L02 娯楽業がこれに該当する。三期分のデータを回帰線とともに変換軸にプロットした散布図を図 10 に示す。散布図内の IRLS は、Tukey の biweight 関数を c=8 で用いた。

散布図の右上に、ウエイトが 0 になる極端な外れ値があり、これは日本中央競馬会(JRA)である。他の大部分が、2 期前よりも 1 期前、1 期前よりも当期の売上高が大きいという傾向を持つが、この極端な外れ値はそれとは逆の動きをしており、期毎に売上高が小さくなっている。このため、OLS の場合、大部分のデータが期毎の上方に変動する動きを、逆の動きをする影響の大きい外れ値一つで打消してしまい、推計値の変動が JRA 以外のデータの変動よりも過小になることが原因である。

図１０．IRLS による推計値の変動が大きい原因(1)



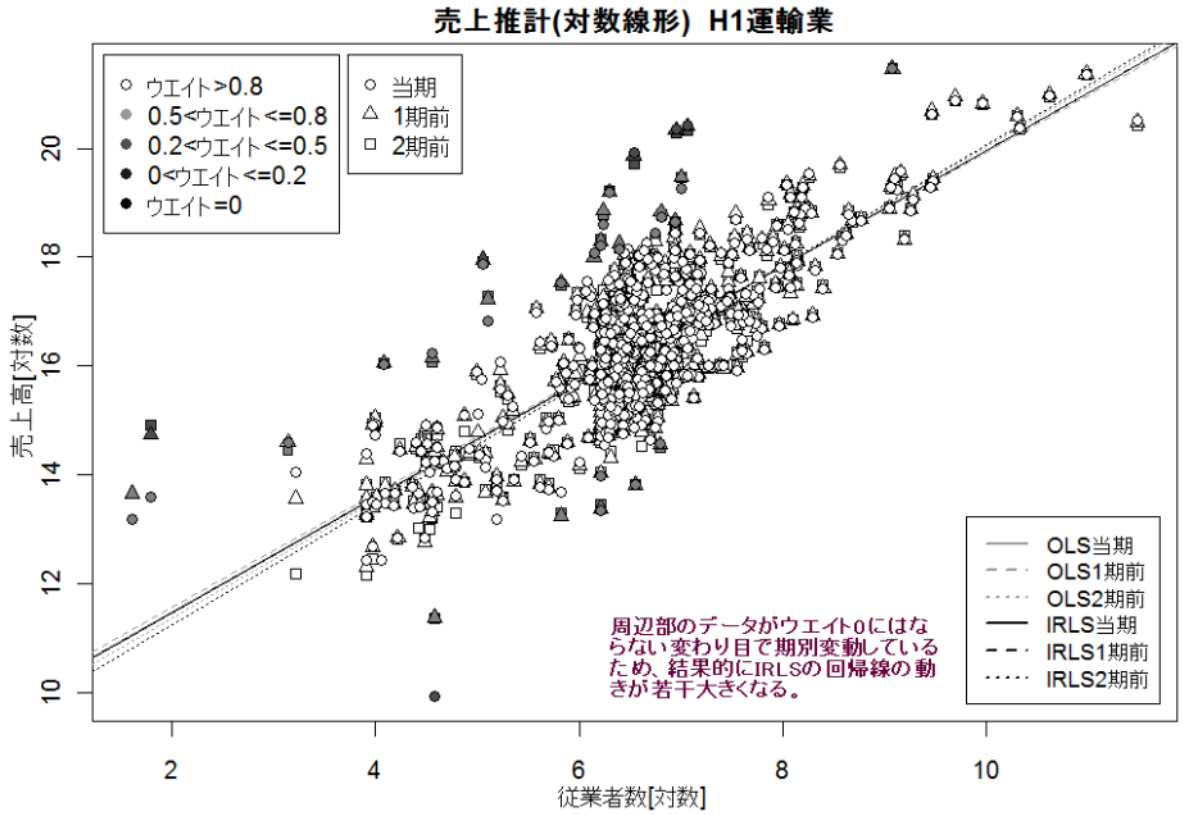
(2)　データ分布の周辺部で変動する外れ値が多く存在する場合

L02 娯楽業以外の 5 産業全てがこれに該当する。一例として、H1 運輸業の散布図を図 11 に示した。これも散布図の IRLS は Tukey の biweight 関数を c=8 で用いている。

この場合、ウエイトが 0 になるような極端な外れ値は存在しないが、データ分布の周辺部でウエイトが 0 に近くデータポイントの色が濃く表示されている部分に多くのデータが位置し、そのいくつかは期毎にウエイトが大きく変わるほど変動が大きい。このような場合、比較的影響のある周辺部のデータが、期毎に異なるウエイトを付与される結果、影響力が変動し、結果としてわずかに推計値の変動が OLS よりも大きくなる。

極端な外れ値がなく比較的データがきれいな場合に、IRLS は回帰線からの垂直距離に応じてデータにウエイト付けするその仕組み上、OLS よりも若干効率が落ちて推計値の分散が大きくなるという性質を持つ。ただし、一つでも影響の大きな外れ値が存在する場合の OLS の推計の妥当性の悪化に比較して、比較的汚れの少ないデータでの IRLS の効率の悪化の影響は相対的にかなり小さい印象がある。実際に５つの産業全てについて、散布図上で OLS と IRLS の回帰線はほとんど乖離していない。

図１１．IRLS による推計値の変動が大きい原因(2)



Ⅳ．結果と考察

１．IRLS の回帰補定への適用について

IRLS は、計算が簡単で実用に向いている。今回の試算はすべて統計ソフト R（Ver. 2.11.1）を用いて行っているが、モデル選択の作業を除いた補定部分については、特別な外部ライブラリなどを必要としないため、ウエイト付きで回帰パラメータを算出できれば、通常の集計プログラムの開発環境での適用が可能である。今回使用したデータでの最大繰返し回数は 5 回であったが、ほとんどのデータは 2～3 回で収束しており、計算負荷も高くない。

データが正規分布であれば、補定には OLSE を使用するのが最も良い方法であるが、外れ値が存在する場合は IRLS の適用により、外れ値の影響によりおこる回帰パラメータの変動を抑えることができるので、推計値が安定する。外れ値がない場合、IRLS は OLS よりも若干効率は落ちるが、推計結果の妥当性には大きな問題はないと思われる。

２．ウエイト関数について

本稿では、Tukey の biweight 関数に加えて、Huber のウエイト関数でも試算を行った。Huber ウエイトを使用すると、どんな外れ値でも必ず 0 にはならないウエイトが付与されるその性質上、ループは起こさず大局解に必ず収束し、収束スピードも Tukey よりもわずかに速いが、極端な外れ値でもその影響を完全排除はしないために、推計値は Tukey ウエイトを使用した場合よりも OLS に近くなる。

影響が大きな外れ値がデータに含まれる可能性があり、極端なものについては完全に推計から排除したい場合は Tukey の biweight の使用が望ましく、一方、データに外れ値がなく全てのデータを考慮した推計をしたい場合は Huber のウエイト関数が適している。

計算の繰返しは最大のもので 5 回だが、大部分のドメインが 2～3 回で収束しており、Tukey よりも Huber ウエイトの収束がわずかに早いが、計算負荷が低いので速度はどちらも問題がない。

調整定数の設定については、今回モデル選択時の正規性の検定でほとんどのドメインにおいて分布の裾が長いことから、本稿では Tukey は c=8、Huber は k=2.30 という設定での試算結果を採用した。一方で、和田・ 椿 (2011)において擬似乱数データを用いて IRLS のシミュレーションを Tukey も Huber も本稿と同じ調整定数の設定で行ったところ、データの誤差項の分布が変わっても、調整定数の値が最も大きい場合が総合的に良いという結果を得ている。

Tukey の biweight は、理論上局所解や無限ループの恐れはあるが、多峰性などの複雑な構造を持たず説明変数も少ない今回のようなデータについてそのような問題は起きなかった。和田・ 椿 (2011)のように擬似乱数を用いた多くのデータセットを用いてシミュレーションを行う場合には、データポイントの位置関係によって無限ループがまれに起きるが、その場合は調整定数の値を若干変えるだけで計算は収束する。

３．乗率の考慮について

補定ドメインは、可能であればドメイン内のデータの標本ウエイトが同じものだけになるよう設定することが望ましい。データ数の制約などで同じ補定ドメインに標本ウエイトが異なるデー

タが混在する場合でも、非常にきれいな正規分布データを対象とするのでない限りは、補定のための OLS による回帰推定に乗率を考慮することは、外れ値の悪影響が乗率分だけ増幅されるために推奨はできない。

IRLS を用いれば外れ値の影響を制御できるので、Wada & Abe(2011)においては、本稿と同じデータを用いて乗率を考慮した推計を行っている。本稿末に掲載した IRLS 関数のコードは、Wada & Abe(2011)や和田・ 椿 (2011)でも使用している乗率対応版である。IRLS 関数のパラメータに乗率 g_i を受け渡すと、第Ⅰ章３節に示した IRLS のアルゴリズムの(4)式の代わりに以下の(4)'式で計算され、IRLS ウエイトと集計乗率の両方を考慮した $\hat{\boldsymbol{\beta}}$ を得ることができる。ここでは、$\mathbf{G} = diag\{g_i\}$ とする。

$$\hat{\boldsymbol{\beta}}^{(j)} = \left[\mathbf{X}'\mathbf{G}\mathbf{W}^{(j-1)}\mathbf{X}\right]^{-1}\mathbf{X}'\mathbf{G}\mathbf{W}^{(j-1)}\mathbf{y} \qquad (4)'$$

４．回帰補定について

今回行った企業売上高データの従業者数による補定の場合、適合モデルは全て同じではないが、全ての産業について従業者数の多い大企業ほど売上高のデータの分散が非常に大きくなるようなモデルが選択されたという共通点がある。

このような場合に回帰補定を用いると、従業者数が多い企業ほど真の値と推計した補定値の乖離が大きい可能性がある。このため、大企業ほど極力個別の問い合わせや上場企業の場合は公開されている企業財務データの活用などを行い、他企業データからの推計による補定を避けることが望ましい。

別紙１- １　IRLS 関数の例: 単回帰モデル［Tukey の biweight］

```
#######################################################################
#  Tirls : Tukey の繰返し再加重最小二乗法(IRLS)  ウエイトは Tukey の biweight #
#---------------------------------------------------------------------#
#      Ver. 0    2010/06/14  オリジナル                                #
#      Ver. 1    2010/12/01  乗率対応版                                #
#      Ver. 1.1  2011/03/07  無限ループ回避                            #
#######################################################################
#  関数 Tirls パラメータ
#   y1      単回帰の目的変数（必須）
#   x1      単回帰の説明変数（必須）
#   rt      乗率。指定がない場合デフォルトは 1
#   c1      biweight 関数用調整定数。デフォルトは 8
#   c=4     とてもロバスト
#   c=8     ちょっとロバスト
#   dat     x1, y1 が含まれるデータフレーム。指定がなければ使わない
#   rp.max  ループ回数の上限。特に指定しなければ 50 回  # 2011.03.07
#######################################################################
#  関数 Tirls 戻り値
#   TK      最終結果
#   wt      ウエイト
#   rp      ループ回数
#   s1      平均絶対残差(MAD)の変遷データ
#######################################################################

Tirls <- function(y1, x1, rt=rep(1, length(y1)), c1=8, dat="", rp.max=50) {

  if (dat!="") attach(get(dat))          # データフレーム指定があるときのみ
  p.mad <- rep(0, rp.max)                # 2011.03.07 MAD の変遷を保存
  R0 <- lm(y1~x1, weights=rt)            # 初期値は普通の OLS

  Tk2 <- R0
  rp1 <- 1                               # ループ回数
  s0 <- 0                                # 最低１回はループするような値をセット
  s1 <- p.mad[rp1] <- mean(abs(Tk2$residuals))          # 平均絶対残差(MAD)

  #### ウエイト算出
        u1 <-Tk2$residuals/(c1*s1)
        w1 <- (1-u1**2)**2
        w1[which(u1>=1)] <- 0

  #### 繰返し計算
  for (i in 2:rp.max) {                                 # 2011.03.07
     # while (abs(1-s1/s0) >= 0.01) {
     if (abs(1-s1/s0) < 0.01) break                     # 2011.03.07
       Tk1 <-Tk2
       s0 <- s1
       Tk2 <- lm(y1~x1, weights=w1*rt)
       rp1 <- rp1 + 1                                   # ループカウンタ
       s1 <- p.mad[rp1] <- mean(abs(Tk2$residuals))     # 平均絶対残差
       u1 <-Tk2$residuals/(c1*s1)
       w1 <- (1-u1**2)**2
       w1[which(abs(u1)>=1)] <- 0
    }
        return(list(TK=Tk2, wt=w1, rp=rp1, s1=p.mad))
  }
###################### End of Tirls ###################################
```

別紙１－２　IRLS 関数の例：単回帰モデル［Huber ウエイト］

```
#########################################################################
#  Hirls : Tukey の繰返し再加重最小二乗法(IRLS)  Huber ウエイト版            #
#-----------------------------------------------------------------------#
#      Ver. 0    2010/08/05  オリジナル                                   #
#      Ver. 1    2010/12/01  乗率対応版                                   #
#      Ver. 1.1  2011/03/07  最大ループ回数設定                           #
#########################################################################
#  関数 Hirls パラメータ
#   y1      単回帰の目的変数（必須）
#   x1      単回帰の説明変数（必須）
#   rt      乗率。指定がない場合デフォルトは 1
#   c1      ウエイト関数用調整定数。デフォルトは 2.30。1.073 が正規分布想定。
#       1.15 で Tukey の biweight C=4 相当、 2.30 で Tukey の c=8 相当。
#   dat     x1, y1 が含まれるデータフレーム。指定がなければ使わない
#   rp.max  ループ回数の上限。特に指定しなければ 50 回  # 2011.03.07
#########################################################################
#  関数 Hirls 戻り値
#   HB      最終結果
#   wt      ウエイト
#   rp      ループ回数
#   s1      平均絶対残差(MAD)の変遷データ
#########################################################################

Hirls <- function(y1, x1, rt=rep(1, length(y1)), c1=2.30, dat="", rp.max=50) {

  if (dat!="") attach(get(dat))          # データフレーム指定があるときのみ
  p.mad <- rep(0, rp.max)                # 2011.03.07 MAD の変遷を保存
  R0 <- lm(y1~x1, weights=rt)            # 初期値はウエイトをつけない OLS

  Hb2 <- R0
  rp1 <- 1                               # ループ回数
  s0 <- 0                                # 最低 1 回はループするような値をセット
  s1 <- p.mad[rp1] <- mean(abs(Hb2$residuals))           # 平均絶対残差

  #### ウエイト算出
    w1 <- s1*c1 / abs(Hb2$residuals)
    w1[which(abs(Hb2$residuals) <= s1*c1)] <- 1

  #### 繰返し計算
  for (i in 2:rp.max) {                                  # 2011.03.07
    #  while (abs(1-s1/s0) >= 0.01) {
      if (abs(1-s1/s0) < 0.01) break                     # 2011.03.07
      Hb1 <-Hb2
      s0 <- s1
      Hb2 <- lm(y1~x1, weights=w1*rt)
      rp1 <- rp1 + 1                                     # ループカウンタ
      s1 <- p.mad[rp1] <- mean(abs(Hb2$residuals))       # 平均絶対残差
      w1 <- s1*c1 / abs(Hb2$residuals)
      w1[which(abs(Hb2$residuals) <= s1*c1)] <- 1
    }
        return(list(HB=Hb2, wt=w1, rp=rp1, s1=p.mad))
  }
###################### End of Hirls ####################################
```

別紙１-３　IRLS 関数の例: 比推定モデル［Tukey の biweight］

```
#########################################################################
#  RTirls : Tukey の繰返し再加重最小二乗法(IRLS)  ウエイトは Tukey の biweight    #
#          比推定用                                                       #
#------------------------------------------------------------------------ #
#      Ver. 0    2010/06/14  オリジナル                                    #
#      Ver. 1    2010/12/01  乗率対応版                                    #
#      Ver. 1.1  2011/03/07  MAD が収束しないときの無限ループ回避          #
#########################################################################
#  関数 RTirls パラメータ
#   y1    単回帰の目的変数（必須）
#   x1    単回帰の説明変数（必須）
#   rt    乗率。指定がない場合デフォルトは 1
#   c1    biweight 関数用調整定数。デフォルトは 8
#   c=4   とてもロバスト
#   c=8   ちょっとロバスト
#   dat   x1, y1 が含まれるデータフレーム。指定がなければ使わない
#   rp.max  ループ回数の上限。特に指定しなければ 50 回  # 2011.03.07
#########################################################################
#  関数 RTirls 戻り値
#   TK    最終結果
#   wt    ウエイト
#   rp    ループ回数
#   s1    平均絶対残差(MAD)の変遷データ
#########################################################################

RTirls <- function(y1, rt=rep(1, length(y1)), c1=8, dat="", rp.max=50) {

  if (dat!="") attach(get(dat))          # データフレーム指定があるときのみ
  p.mad <- rep(0, rp.max)                # 2011.03.07 MAD の変遷を保存
  R0  <- lm(y1~1, weights=rt)            # 初期値はウエイトをつけない OLS
  Tk2 <- R0
  rp1 <- 1                               # ループ回数
  s0  <- 0                               # 最低１回はループするような値をセット
  s1  <- p.mad[rp1] <- mean(abs(Tk2$residuals))          # 平均絶対残差

  #### ウエイト算出
  u1 <-Tk2$residuals/(c1*s1)
  w1 <- (1-u1**2)**2
  w1[which(u1>=1)] <- 0

  #### 繰返し計算
  for (i in 2:rp.max) {                                  # 2011.03.07
    # while (abs(1-s1/s0) >= 0.01) {
      if (abs(1-s1/s0) < 0.01) break                     # 2011.03.07
      Tk1 <-Tk2
      s0  <- s1
      Tk2 <- lm(y1~1, weights=w1*rt)
      rp1 <- rp1 + 1                                     # ループカウンタ
      s1 <- p.mad[rp1] <- mean(abs(Tk2$residuals))       # 平均絶対残差
      u1 <-Tk2$residuals/(c1*s1)
      w1 <- (1-u1**2)**2
      w1[which(abs(u1)>=1)] <- 0
    }
        return(list(TK=Tk2, wt=w1, rp=rp1, s1=p.mad))
  }
###################### End of RTirls ####################################
```

別紙１-４　IRLS 関数の例: 比推定モデル［Huber ウエイト］

```
##############################################################################
#  RHirls : Tukey の繰返し再加重最小二乗法(IRLS)  Huber ウエイト版 比推定用        #
#---------------------------------------------------------------------------- #
#      Ver. 0    2010/08/05  オリジナル                                        #
#      Ver. 1    2010/12/01  乗率対応版                                        #
#      Ver. 1.1  2011/03/07  最大ループ回数設定                                #
##############################################################################
#  関数 RHirls パラメータ
#   y1      目的変数／説明変数（必須）
#   rt      乗率。指定がない場合デフォルトは 1
#   c1      ウエイト関数用調整定数。デフォルトは 2.30
#         1.073 が正規分布想定。1.15 で Tukey の biweight C=4 相当、 2.30 で Tukey の c=8 相当。
#   dat     x1, y1 が含まれるデータフレーム。指定がなければ使わない
#   rp.max  ループ回数の上限。特に指定しなければ 20 回  # 2011.03.07
##############################################################################
#  関数 RHirls 戻り値
#   HB      最終結果
#   wt      ウエイト
#   rp      ループ回数
#   s1      平均絶対残差 (MAD) の変遷データ
##############################################################################

RHirls <- function(y1, rt=rep(1, length(y1)), c1=2.30, dat="", rp.max=50) {

if (dat!="") attach(get(dat))            # データフレーム指定があるときのみ
  p.mad <- rep(0, rp.max)                # 2011.03.07 MAD の変遷を保存
  R0 <- lm(y1~1, weights=rt)             # 初期値はウエイトをつけない OLS

  Hb2 <- R0
  rp1 <- 1                               # ループ回数
  s0  <- 0                               # 最低１回はループするような値をセット
  s1  <- p.mad[rp1] <- mean(abs(Hb2$residuals))          # 平均絶対残差

  #### ウエイト算出
    w1 <- s1*c1 / abs(Hb2$residuals)
    w1[which(abs(Hb2$residuals) <= s1*c1)] <- 1

  #### 繰返し計算
  for (i in 2:rp.max) {                                  # 2011.03.07
    # while (abs(1-s1/s0) >= 0.01) {
      if (abs(1-s1/s0) < 0.01) break                     # 2011.03.07
        Hb1 <-Hb2
        s0 <- s1
        Hb2 <- lm(y1~1, weights=w1*rt)
        rp1 <- rp1 + 1                                   # ループカウンタ
        s1 <- p.mad[rp1] <- mean(abs(Hb2$residuals))     # 平均絶対残差
        w1 <- s1*c1 / abs(Hb2$residuals)
        w1[which(abs(Hb2$residuals) <= s1*c1)] <- 1
    }

          return(list(HB=Hb2, wt=w1, rp=rp1, s1=p.mad))
  }
###################### End of RHirls ####################################
```

別紙１-５　IRLS 関数の使用例

```
# 以下のコードは、R の組込データを用いてモデル選択を行い、IRLS による回帰線入りの散布図を作成している。
# データの出典は、Verbeek (2004)。
############################
rm(list=ls(all=TRUE))     # 作業領域のクリア
setwd("d:/test")          # 関数ファイルを置いた任意の作業ディレクトリを指定する
source("Tirls.r")         #  線形モデル用   Tukey ウエイトの IRLS
source("RTirls.r")        #  比推定モデル用 Tukey ウエイトの IRLS
source("Hirls.r")         #  線形モデル用   Huber ウエイトの IRLS
source("RHirls.r")        #  比推定モデル用 Huber ウエイトの IRLS

data(Clothing, package="Ecdat") # Ecdat に含まれる Clothing データセットの呼び出し
  # この組込データはオランダの男性洋品店の 1990 年の年間売上高についてのもの
uriage1 <- Clothing$tsales      # 総売上高 uriage1  単位:ギルダー
emp1    <- Clothing$nfull + Clothing$npart + Clothing$naux
         # フルタイムとパートと非常勤を足して従業者数 emp1 とする

############################
#     正規性の検定
############################
require(nortest)    # for Lilliefors (Kolmogorov-Smirnov) test
  lillie.test(uriage1/emp1)$p.value        #[1] 0.002479864
  lillie.test(sqrt(uriage1/emp1))$p.value   #[1] 0.2780684
  lillie.test(log(uriage1/emp1))$p.value    #[1] 1.897207e-09

# 二番目の p 値が最も大きいので、平方根変換を採用する。以下のヒストグラムも平方根変換を示唆している。

##### ヒストグラム作成 #####
require(MASS)       #  for truehist
png(filename="Clothing_ヒストグラム.png", width=1024, height=768, pointsize = 20)
  par(mfrow=c(1,3))
  truehist(uriage1/emp1, main="変換なし")
  truehist(sqrt(uriage1/emp1), main="平方根変換")
  truehist(log(uriage1/emp1), main="対数変換")
dev.off()

############################
#     モデル選択
############################
# 変換は確定したので、回帰か比推定モデルかを決める。

n1 <- length(uriage1)     # データ数
# データの色分け用フラグ
f.TRR1 <- f.TLR1 <- f.TLR2 <- f.HRR1 <- f.HLR1 <- f.HLR2 <- rep(1, n1)

# 平方根変換 + 線形モデル ###########################################################
##### OLS #####
  OLS1 <- lm(sqrt(uriage1)~sqrt(emp1))
    s.OLS1 <- mean(OLS1$residuals^2) / (n1-2)*n1  # 推計用の補正係数算出

##### IRLS(Tukey, c=8) ######
  TLR1 <- Tirls(sqrt(uriage1), sqrt(emp1), c1=8)   # c1 は調整定数
    s.TLR1 <- mean(TLR1$TK$residuals^2) / (n1-2)*n1  # 推計用の補正係数算出
    # IRLS の最終ウエイトでデータを色分け
      f.TLR1[which(TLR1$wt < 0.8)]        <- 3
      f.TLR1[which(TLR1$wt < 0.5)]        <- 7
      f.TLR1[which(TLR1$wt < 0.2)]        <- 2
```

```
    f.TLR1[which(TLR1$wt == 0)]         <- 8

##### IRLS(Huber, k=2.3) ######
  HLR1 <- Hirls(sqrt(uriage1), sqrt(emp1), c1=2.30)   # c1 は調整定数
    s.HLR1 <- mean(HLR1$HB$residuals^2) / (n1-2)*n1  # 推計用の補正係数算出
    # IRLS の最終ウエイトでデータを色分け
      f.HLR1[which(HLR1$wt < 0.8)]        <- 3
      f.HLR1[which(HLR1$wt < 0.5)]        <- 7
      f.HLR1[which(HLR1$wt < 0.2)]        <- 2
      f.HLR1[which(HLR1$wt == 0)]         <- 8

# 平方根変換 + 比推定モデル ###################################################
##### OLS #####
  ORS1 <- lm(sqrt(uriage1)/sqrt(emp1)~1)
    s.ORS1 <- mean(ORS1$residuals^2) / (n1-1)*n1

##### IRLS(Tukey, c=8) #####
  TRR1 <- RTirls(sqrt(uriage1/emp1), c1=8)
    # 推計用の補正係数 s.RR1 算出
    s.TRR1 <- sum(TRR1$wt * TRR1$TK$residuals^2) / sum(TRR1$wt) /(n1-1)*n1
    # IRLS の最終ウエイトでデータを色分け
      f.TRR1[which(TRR1$wt < 0.8)]        <- 3
      f.TRR1[which(TRR1$wt < 0.5)]        <- 7
      f.TRR1[which(TRR1$wt < 0.2)]        <- 2
      f.TRR1[which(TRR1$wt == 0)]         <- 8

##### IRLS(Huber, k=2.3) #####
  HRR1 <- RHirls(sqrt(uriage1/emp1), c1=2.3)
    # 推計用の補正係数 s.RR1 算出
    s.HRR1 <- sum(HRR1$wt * HRR1$TK$residuals^2) / sum(HRR1$wt) /(n1-1)*n1
    # IRLS の最終ウエイトでデータを色分け
      f.HRR1[which(HRR1$wt < 0.8)]        <- 3
      f.HRR1[which(HRR1$wt < 0.5)]        <- 7
      f.HRR1[which(HRR1$wt < 0.2)]        <- 2
      f.HRR1[which(HRR1$wt == 0)]         <- 8

# 参照用 対数線形 ###########################################################
  OLS2 <- lm(log(uriage1)~log(emp1))
    s.OLS2 <- exp(sum(OLS2$residuals^2) / (n1-2) /2)

  TLR2 <- Tirls(log(uriage1), log(emp1), c1=8)
    s.TLR2 <- exp(sum(TLR2$TK$residuals^2) / (n1-2) /2)
    # IRLS の最終ウエイトでデータを色分け
      f.TLR2[which(TLR2$wt < 0.8)]        <- 3
      f.TLR2[which(TLR2$wt < 0.5)]        <- 7
      f.TLR2[which(TLR2$wt < 0.2)]        <- 2
      f.TLR2[which(TLR2$wt == 0)]         <- 8

  HLR2 <- Hirls(log(uriage1), log(emp1), c1=2.30)   # c1 は調整定数
    s.HLR2 <- mean(HLR2$HB$residuals^2) / (n1-2)*n1  # 推計用の補正係数算出
      f.HLR2[which(HLR2$wt < 0.8)]        <- 3
      f.HLR2[which(HLR2$wt < 0.5)]        <- 7
      f.HLR2[which(HLR2$wt < 0.2)]        <- 2
      f.HLR2[which(HLR2$wt == 0)]         <- 8

##### プロット ################################################################
# 白黒用カラー設定
  col1  <- c("white", "gray20", "gray60", "","","", "gray40", "black")  # for Tukey
```

```
  col2  <- c("white", "gray20", "gray60", "gray80","","", "gray40", "black")  # for Huber

# 凡例
  ltxt    <- c("OLS 線形", "Tukey8 線形", "Huber8 線形",
               "OLS 比推定", "Tukey8 比推定", "Huber8 比推定")
  ltxt_TK <- c("TK ウエイト>0.8", "0.5<ウエイト<=0.8", "0.2<ウエイト<=0.5",
                "0<ウエイト<=0.2", "ウエイト=0")
  ltxt_HB <- c("HB ウエイト= 1","1 > ウエイト>0.8", "0.5<ウエイト<=0.8",
                "0.2<ウエイト<=0.5", "0<ウエイト<=0.2", "ウエイト=0")

png(filename="Clothing_回帰線入り散布図.png", width=2339, height=3307, pointsize = 40)
  par(mfrow=c(2,1))

  ### 平方根軸散布図 ###
  plot(sqrt(emp1), sqrt(uriage1), bg=col1[f.TLR1], pch=21,
              main="Clothing: 平方根軸散布図", xlab="従業者数[平方根]", ylab="売上高[平方根]")
    abline(a=OLS1$coeff[1],    b=OLS1$coeff[2],     col="red",   lty=1, lwd=1)
          abline(a=TLR1$TK$coeff[1], b=TLR1$TK$coeff[2],  col="blue",  lty=1, lwd=3)
    abline(a=HLR1$HB$coeff[1], b=HLR1$HB$coeff[2],  col="green", lty=1, lwd=1)

    abline(a=0, b=ORS1$coeff,             col="red",       lty=2, lwd=1)
          abline(a=0, b=TRR1$TK$coeff,    col="blue",      lty=3, lwd=3)
    abline(a=0, b=HRR1$HB$coeff,          col="green",     lty=3, lwd=1)

    legend(3.7, 800, ltxt, col=rep(c("red", "blue", "green"), 2), lty=c(1,1,1,3,3,3), lwd=2)
    legend(1.7, 2200, ltxt_TK, pch = c(21, 16, 16, 16, 16),
         col=c("black", "gray60", "gray40", "gray20", "black"))
#   legend(1.65, 2300, ltxt_HB, pch = c(21, 16, 16, 16, 16, 16),
#        col=c("black", "gray80", "gray60", "gray40", "gray20", "black"))

    ### 実軸散布図 ###plot(emp1, uriage1, bg=col1[f.TLR1], pch=21,
              main="Clothing: 実軸散布図", xlab="従業者数[人]", ylab="売上[ギルダー]")
    curve((OLS1$coeff[1]    + OLS1$coeff[2]    * sqrt(x))^2 + s.OLS1,
                col="red",   lty=1, lwd=1, add=T)
    curve((TLR1$TK$coeff[1] + TLR1$TK$coeff[2] * sqrt(x))^2 + s.TLR1,
                col="blue",  lty=1, lwd=3, add=T)
    curve((HLR1$HB$coeff[1] + HLR1$HB$coeff[2] * sqrt(x))^2 + s.HLR1,
                col="green", lty=1, lwd=1, add=T)

    curve((ORS1$coeff^2    + s.ORS1) * x,  col="red",     lty=2, lwd=2, add=T)
    curve((TRR1$TK$coeff^2 + s.TRR1) * x,  col="blue",    lty=3, lwd=3, add=T)
    curve((HRR1$HB$coeff^2 + s.HRR1) * x,  col="green",   lty=2, lwd=2, add=T)

    legend(15, 1.6e+6, ltxt, col=rep(c("red", "blue", "green"), 2),
          lty=c(1,1,1,3,3,3), lwd=2)
    legend(3, 5e+6, ltxt_TK, pch = c(21, 16, 16, 16, 16),
          col=c("black", "gray60", "gray40", "gray20", "black"))
#   legend(11, 5e+6, ltxt_HB, pch = c(21, 16, 16, 16, 16, 16),
#         col=c("black", "gray80", "gray60", "gray40", "gray20", "black"))

dev.off()

#-------------------------------------------------------------------------------#
png(filename="Clothing_残差分析_OLS.png", width=768, height=1024, pointsize = 20)
  par(mfrow=c(3,2))
  plot(sqrt(emp1), scale(OLS1$residuals),           pch=19, main="sqrt(x) - e")
       abline(h=0, col = 'green')
    qqnorm(OLS1$residuals, pch=19, main="平方根線形 OLS Q-Q plot")
```

```
      qqline(OLS1$residuals, col="green")

  plot(sqrt(emp1), scale(ORS1$residuals/sqrt(emp1)),  pch=19, main="sqrt(x) - e/sqrt(x)")
      abline(h=0, col = 'green')
    qqnorm(ORS1$residuals, pch=19, main="平方根比推定 OLS Q-Q plot")
      qqline(ORS1$residuals, col="green")

  plot(log(emp1), scale(OLS2$residuals),            pch=19, main="log(x) - e")
      abline(h=0, col = 'green')
    qqnorm(OLS2$residuals, pch=19, main="対数線型 OLS Q-Q plot")
      qqline(OLS2$residuals, col="green")
dev.off()


png(filename="Clothing_残差分析_Tukey8.png", width=768, height=1024, pointsize = 20)
  par(mfrow=c(3,2))
  plot(sqrt(emp1), scale(TLR1$TK$residuals),        pch=19, col=f.TLR1,
      main="sqrt(x) - e", xlab="sqrt(x)", ylab="residuals")
      abline(h=0, col = 'green')
    qqnorm(TLR1$TK$residuals, pch=19, col=f.TLR1, main="平方根線形 Tukey8 Q-Q plot")
      qqline(TLR1$TK$residuals, col="green")

  plot(sqrt(emp1), scale(TRR1$TK$residuals/sqrt(emp1)),  pch=19, col=f.TRR1,
      main="sqrt(x) - e/sqrt(x)", xlab="sqrt(x)", ylab="residuals/sqrt(x)")
      abline(h=0, col = 'green')
    qqnorm(TRR1$TK$residuals, pch=19, col=f.TRR1, main="平方根比推定 Tukey8 Q-Q plot")
      qqline(TRR1$TK$residuals, col="green")

  plot(log(emp1), scale(TLR2$TK$residuals),          pch=19, col=f.TLR2,
      main="log(x) - e", xlab="log(x)", ylab="resudals")
      abline(h=0, col = 'green')
    qqnorm(TLR2$TK$residuals, pch=19, col=f.TLR2, main="対数線型 Tukey8 Q-Q plot")
        qqline(TLR2$TK$residuals, col="green")
dev.off()

png(filename="Clothing_残差分析_Huber8.png", width=768, height=1024, pointsize = 20)
  par(mfrow=c(3,2))
  plot(sqrt(emp1), scale(HLR1$HB$residuals),        pch=19, col=f.HLR1,
      main="sqrt(x) - e", xlab="sqrt(x)", ylab="residuals")
      abline(h=0, col = 'green')
    qqnorm(HLR1$HB$residuals, pch=19, col=f.HLR1, main="平方根線形 Huber8 Q-Q plot")
      qqline(HLR1$HB$residuals, col="green")

  plot(sqrt(emp1), scale(HRR1$HB$residuals/sqrt(emp1)),  pch=19, col=f.HRR1,
      main="sqrt(x) - e/sqrt(x)", xlab="sqrt(x)", ylab="residuals/sqrt(x)")
      abline(h=0, col = 'green')
    qqnorm(HRR1$HB$residuals, pch=19, col=f.HRR1, main="平方根比推定 Huber8 Q-Q plot")
      qqline(HRR1$HB$residuals, col="green")

  plot(log(emp1), scale(HLR2$HB$residuals),          pch=19, col=f.HLR2,
      main="log(x) - e", xlab="log(x)", ylab="resudals")
      abline(h=0, col = 'green')
    qqnorm(HLR2$HB$residuals, pch=19, col=f.HLR2, main="対数線型 Huber8 Q-Q plot")
        qqline(HLR2$HB$residuals, col="green")
dev.off()
#-------------------------------------------------------------------------------#
#  平方根変換が良いのは明らか。どちらかといえば線形モデルよりも比推定モデルがよいと思われる。
```

参考文献

[1] Antoch, J. and Ekblom, H. (1995) Recursive robust regression computational aspects and comparison, *Computational Statistics & Data Analysis* 19, 115-128

[2] Beaton, A. E. and Tukey, J. W. (1974) The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data, *Technometrics* 16, 147-185

[3] Bienias, J. L., Lassman, D. M. Scheleur, S. A. and Hogan H. (1997) Improving Outlier Detection in Two Establishment Surveys. *Statistical Data Editing 2 - Methods and Techniques*. (UNSC and UNECE eds.), 76-83.

[4] Holland, P. W. and Welsch, R. E. (1977), Robust Regression Using Iteratively Reweighted Least-Squares, *Communications in Statistics – Theory and Methods* A6(9), 813-827

[5] Huber, P. J. (1964) Robust Estimation of a Location Parameter, *Annals of Mathematical Statistics* 35(1), 73-101

[6] Huber, P. J. (1973) Robust Regression: Asymptotics, Conjectures and Monte Carlo, *Annals of Statistics* 1(5), 799-821.

[7] Lenth, R. V. and P. J. Green (1987) Consistency of Deviance-Based M-Estimators, *Journal of the Royal Statistical Society, Series B(Methodology)* 49(3), 326-330

[8] Mosteller, F. and J.W. Tukey (1977) *Data Analysis and Regression*, Addison Wesley, Reading, MA.

[9] Rousseeuw & Leroy(1987), Robust Regression and Outlier Detection, John Wiley & Sons, Inc.

[10] Verbeek, Marno (2004) *A guide to modern econometrics*, John Wiley and Sons, chapter 3

[11] Wada, K. and Abe, Yutaka (2011), Multivariate Outlier Detection for Regression – Imputation and Aggregation Weight Calibration by IRLS –, *58th Session of the ISI World Statistics Congress Proceedings*, Dublin

[12] 岡本政人（2004），多変量外れ値検出法の研究動向，*製表技術研究レポート* 1，（独）統計センター研究センター，pp.1-34

[13] 岩崎学（2002）*不完全データの統計解析*, エコノミスト社

[14] 蓑谷千凰彦（1992）*計量経済学における頑健推定*, 多賀出版

[15] 和田かず美, 椿広計 (2011), ロバスト回帰を用いた外れ値に対する乗率の補正, *応用統計学会 2011 年度年会講演予稿集*, 51-56