

## 正誤表

万全を期して作成したつもりですが、初版で既にいくつか間違いのご指摘をいただいております。ご指摘に御礼申し上げ、また、ここにお詫びして修正をご報告いたします。

page	誤	正	解説
p.6	R コード <code>n &lt;- 2500</code>	<code>n &lt;- 25</code>	テキストでは $n = 25$ の例として示していましたが、コードは続く $n = 2500$ の例を実行するものになっていました。
p.10（下から 5 行目）	わかりやすく書いる書籍	わかりやすく書いている書籍	
p.18L7	B は A と同じく <code>x</code> を 3 列 2 行に	B は A と同じく <code>x</code> を 3 行 2 列に	<code>row</code> は行, <code>col</code> は列です。失礼しました。
p.18	R の出力の要素が全て 0 になっている	1 から 24 までの数字が順に入ります。	<code>array</code> 関数が配列を指定するものです。
p.26-27	出力	コード	右肩に「出力」と書かれているブロックは、「コード」が正しいです。
p.35-36	出力	コード	右肩に「出力」と書かれているブロックは、「コード」が正しいです。
p.40	決して実行しないでくださいのコード	変更なし	R ではカウンタ変数は別途割り当てられるので、永久ループにはならないそうです。しかしプログラミング言語として、一般的に避けるべき作法です。
p.41	R が永遠の計算ループから抜け出せなくなります。	抜け出せなくなることはありませんが、おかしい挙動になります。	両方 <code>i</code> で回すと、内側の <code>i</code> ループが外側の <code>i</code> ループ分繰り返されるという動きになります。
p.104	R コード最後の行内 <code>df(line_x, df1 = nu_1, df2= nu_2)</code>	<code>df(line_x, df1 = 1, df2= nu)</code>	これに伴い、図 3.29 の曲線もわずかに変化します（ヒストグラムに変化はありません）。
P.134	R コード <code>var_p</code>	<code>var_p()</code>	R のネイティブパイプは、関数 <code>()</code> の形に渡すことが必要です ( <code>magrittr</code> のパイプ演算子 <code>%&gt;%</code> であれば問題ありません)
P.142 本文下から 3 行目	サンプルサイズ $n$ が 4、10、100 と大きくなるにつれて	サンプルサイズ $n$ が 4、20、100 と大きくなるにつれて	
P.143 図 4.14			誤った画像ファイルが挿入されていました。コードを実行して出力される図が正しいです
P.182	R コード <code>cor.test(dat_obs[, 1], t.test(sample_r)\$conf.int[1:2])\$conf.int[1:2]</code>	<code>cor.test(dat_obs[, 1], t.test(sample_r)\$conf.int[1:2])\$conf.int[1:2]</code>	出力も <code>[1] 0.3787639 0.8187475</code> となります。
P.181	パーセンタイル信頼区間の方が広がっています。	今回はパーセンタイル信頼区間の方が狭くなっています。	ここは一般的に狭くなるわけではないので。
P.182	Fisher の Z 変換の上限 (0.4973) と下限 (0.5011)	上限は 0.5897387, 下限は 0.4217412	R のコード変更に伴って修正させていただきます。