

## 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.6 5 行目	実際に帰無仮説が採択される	実際に帰無仮説が棄却される	
p.6 6 行目	有意水準 $\alpha = 0.05$ 以下	有意水準 $\alpha = 0.05$ 未満	
p.7 コード 内のコメント	データが当初の倍になるまで増やし続ける	データが当初の 3 倍になるまで増やし続ける	
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.18 L7	B は A と同じく x を 3 列 2 行に	B は A と同じく x を 3 行 2 列に	row は行, col 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。	array 関数が配列を指定するものです。
p.26	R コード <code>Species = "setosa"</code>	<code>Species == "setosa"</code>	等価演算子は <code>==</code> 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.26 脚注 8	magritter パッケージ	magrittr パッケージ	
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、おかしな挙動になります。	両方 i で回すと、内側の i ループが外側の i ループ分繰り返されるという動きになります
p.46	R コード <code>ifelse(a &gt; 10, print("a &gt; 10"), print("a &lt;=10"))</code>	<code>print()</code> は不要です	
p.47 脚注 22	! や !!	本文中のコードと同様に、 <code>\!</code> や <code>\!\!</code>	
p.48 冒頭	a が 10 以下か、b が 10 以上	a が 10 以下、かつ、b が 10 以上	コード的には問題ないんですが、条件文の意味が違いました。
p.49	モジュロ ; odulo	モジュロ ; modulo	
p.55	?proc.time	?system.time	
p.57	iter <- 10000	iter <- 100000	
p.80	R コード <code>MC_demo()</code>	<code>mc_demo()</code>	

page	誤	正	解説
p.104	R コード最後の行内 df(line_x,df1 = nu_1, df2= nu_2)	df(line_x,df1 = 1, df2= nu)	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.124	図 4.2 の結果を導いた	図 4.1 の結果を導いた	
P.134	R コード var_p	var_p()	R のネイティブパイプは、関数 () の形に渡すことが必要です (magritter のパイプ演算子 %>% 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.152 コード最終行	df = n - 2	df = n - 1	p151 の数式と対応するので、-1 が正しいです
P.182	R コード cor.test(sample_r)\$conf1)ntdat_obs[, 2])\$conf.int[1:2]	cor.test(dat_obs[, 2])\$conf.int[1:2]	出力も [1] 0.3787639 0.8187475 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。
P.247	ただし、ここでは簡単のためにサンプルサイズが群ごとに等しいことを仮定します。	ただし、ここでは簡単のためにサンプルサイズが群ごとに等しいことを仮定します。	誤字