

プログラミング教育におけるエラー分析の一例 (第二報)

張 学健 平林 隆一 菅野 文友
(東京理科大学)

1. はじめに

高度情報社会に要求されるソフトウェア技術者養成のニーズおよび、パソコンの普及などの教育環境変化の背景に対して、大学における情報処理関連教育の進め方が、一つの重要な課題になっている。

本報告では、このようなニーズおよび背景を踏まえて、東京理科大学経営工学科2年次のプログラミング講義を受講した71人の学生を対象として、学生のレポート中のエラーデータおよびアンケート調査の回答を分析した。今回の分析では、特にパソコンを持っている学生と持っていない学生との間に、ソフトウェア・エラーの発生に対する影響の異同があるかないか、また血液型によるエラーの出方の異同について、注目した。

2. 1 学生に出題した問題

(1) 第1回：ベーシックを使ったプログラミング

自分の学籍番号(入学年度(2桁)+通し番号(3桁)の5桁)を前の2桁と後の3桁に分けて、簡単な四則演算を行ない、その結果に対して、有効数字を合わせるために四捨五入して答とする。

(2) 第2回：アセンブラを使ったプログラム

4つの整数を番地A, B, C, Dに入れ、2番目に大きい数字をP番地に入れる。

(3) 第3回：フォートランを使ったプログラム

X_i ; ($i = 1, 2, \dots, N$) で与えられる数列の最大値をBigに入れ、その数が数列の何番目であったかをNNに入れる。

(4) 第4回：ベーシックを使ったプログラム

ワイブル分布に従う乱数を発生する。

2. 2 データの種類と記号

本報告で取り扱うデータの種類と記号を、表2.1に示す。

表2.1 データの種類と記号

種類	記号	内容
第1回: ベーシックを使っ たプログラム	B	B1 構文エラー
		B2 ロジックエラー
		B3 桁による数字の分解エラー
		B4 四捨五入エラー
第2回: アセンブラを使っ たプログラム	A	A1 構文エラー
		A2 ロジックエラー
		A3 数と条件判定エラー
第3回: フォートランを使 ったプログラム	F	F1 構文エラー
		F2 ロジックエラー
		F3 フォートランとベーシックの混合エラー
第4回: ベーシックを使っ たプログラム	P	P1 構文エラー
		P2 m捨(m+1)入エラー
		P3 アルゴリズムエラー(問題の不理解)
		P4 ロジックエラー(基本的言語論理)
アンケート	Q	Q1 パソコンを持っているか。
		Q2 パソコンを使う機会があるか。
		Q3 ベーシックを知っていたか。
		Q4 フォートランを知っていたか。
血液型	K	(A型; 1)(B型; 2)(AB型; 3)(O型; 4)

3. エラーデータの解析

3.1 共通エラーの解析

出題した4つの問題に対して、構文エラーとロジックエラーは共通したエラーである。この2種類のエラーに対して、分析を行なう。

(1) 構文エラーに対する分析

まず、分割表を使って、分析を行なう。

表3.1 各回ごとの場合と連続した場合の
分類と構文エラーの2×5分割表(単位:%)

問題		B	AUF	A	F	P	計	
分類	各回	16.7	27.8	12.5	19.4	2.8	79.4	
	連続	16.7	8.6	4.2	4.2	0.0	33.7	
計		33.4	36.4	16.7	23.6	2.8	112.9	
χ^2		10.15*						

すなわち、各問題によって、各回ごとの場合と連続した場合の構文エラーの出現率に、差があるといえる。

次に、上記のデータを逆正弦変換して分析を行なう。

表3.2 構文エラーの出現率に対する逆正弦変換の結果(単位:deg.)

問題		B	AUF	A	F	P
逆正弦変換 $\text{Sin}^{-1} \frac{\text{連続}}{\text{連続+各回}}$	連続					
	連続+各回	30	13.9	14.5	10.4	0

表3.3 逆正弦変換の結果による分散分析表

要因	S	ϕ	V	F.
問題	464.9	4	116.2	10.02**
誤差	—	—	11.6	—
計	464.9	4	—	—

つまり、逆正弦変換による分析結果は、分割表による結果と同じである。

(2) ロジックエラーに対する分析

まず、分割表を使って、分析を行なう。

表3.4 各回ごとの場合と連続した場合の分類
とロジックエラーの2×5分割表(単位:%)

問題		B	AUF	A	F	P	計	
分類	各回	18.1	29.2	19.4	13.9	11.1	91.7	
	連続	18.1	4.2	2.8	2.8	0.0	27.9	
計		36.2	33.4	22.2	16.7	11.1	119.6	
χ^2		21.76*						

すなわち、各問題によって、各回ごとの場合と連続した場合のロジックエラーの出現率に、差があるといえる。

そして、逆正弦変換による分析を行なう。

表3.5 ロジックエラーによる出現率に対する逆正弦変換の結果 (単位: deg.)

問題		B	A U F	A	F	P
		逆正弦変換				
Sin	-1	連続				
	N	連続+各回	3.0	7.2	7.2	9.6

表3.6 逆正弦変換の結果による分散分析

要員	S	ϕ	V	F.
問題	5 1 2 . 6 4	4	1 2 8 . 1 6	1 1 . 0 5 **
誤差	—	—	1 1 . 6	—
計	5 1 2 . 6 4	4	—	—

つまり、逆正弦変換による結果は、分割表による結果と同じである。

次に、エラーの出現の差の内容や時系列的な傾向を調べるために、図3.1と図3.2の棒グラフで比較する。

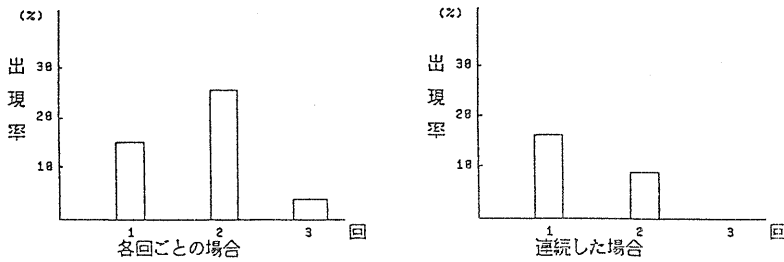


図3.1 構文エラーの出現率

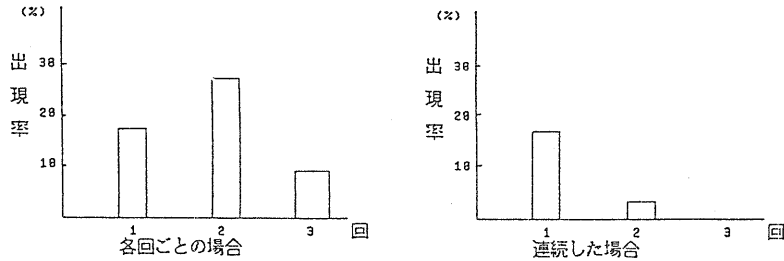


図3.2 ロジックエラーの出現率

図3.1および図3.2から、構文エラーとロジックエラーの両方とも連続した場合の出現率は、確実に減少していることがわかる。

3.2 アンケートおよびエラーデータの解析

(1) 分割表を利用した解析

表3.7 Q1の回答とベーシックエラーの2×4分割表 (単位: %)

アンケート		エラー				計
		B 1	B 2	B 3	B 4	
Q 1	Yes	1 3 . 0 4	4 . 5 5	1 8 . 1 8	0 . 0 0	3 6 . 3 7
	No	1 6 . 6 7	3 0 . 0 0	3 3 . 3 3	2 3 . 3 3	1 0 3 . 3 3
計		3 0 . 3 1	3 4 . 5 5	5 1 . 5 1	2 3 . 3 3	1 3 9 . 7 0
χ^2		1 9 . 2 9 *				

すなわち、パソコンを持っているか否かが、ベーシックエラーの出現率に影響を与えているといえる。

次に、逆正弦変換による分析を行なう。

表3.8 Q1の回答とベーシックエラーに対する逆正弦変換の結果(単位: deg.)

逆正弦変換		エラー			
		B 1	B 2	B 3	B 4
Sin ⁻¹	$\frac{\text{Yes}}{\text{Yes}+\text{No}}$	25.8	7.6	20.7	0

表3.9 逆正弦変換による分散分析表

要員	S	ϕ	V	F.
エラー	420.19	3	140.06	12.07**
誤差	—	—	11.6	—
計	420.19	3	—	—

つまり、逆正弦変換による結果は、分割表による結果と同じである。

同様に、Q1~Q4の解答とB、A、Fの3種類の言語に対して、分割表によって、分析を行なう。その結果をまとめて表3.10に示す。

表3.10 分割表による検定の結果

アンケート		エラー		
		B	A	F
Q 1		*		*
Q 2		*		*
Q 3		*		*
Q 4		*	*	*

(*:水準5%で有意)

すなわち、アセンブラ言語に対して、ベーシック言語の習熟は関連していないといえる。

(2) グラフによる比較

次にこの12の組合せについて、グラフ(図3.3~図3.14)から、エラーの発生傾向を分析する。

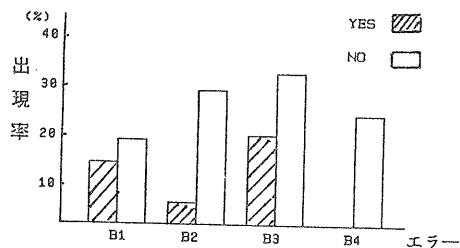


図3.3 Q1の解答別ベーシックエラーの出現率

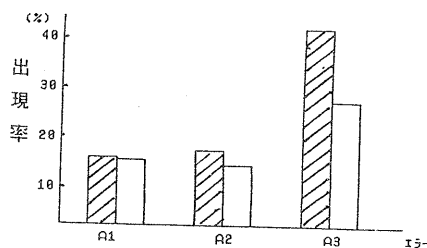


図3.4 Q1の解答別アセンブラエラーの出現率

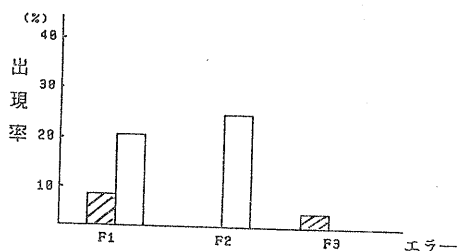


図3.5 Q1の解答別フォートランエラーの出現率

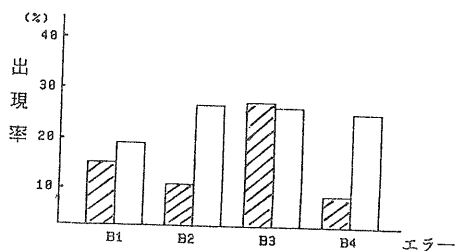


図3.6 Q2の解答別ベーシックエラーの出現率

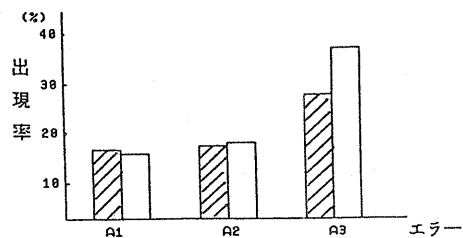


図 3. 7 Q 2 の解答別アセンブラエラーの出現率

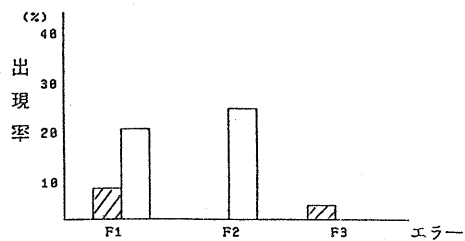


図 3. 8 Q 2 の解答別フォートランエラーの出現率

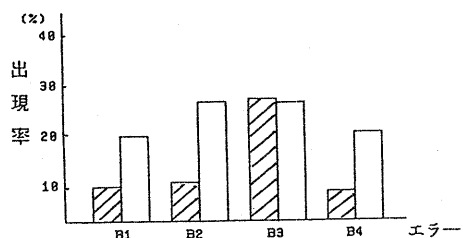


図 3. 9 Q 3 の解答別ベーシックエラーの出現率

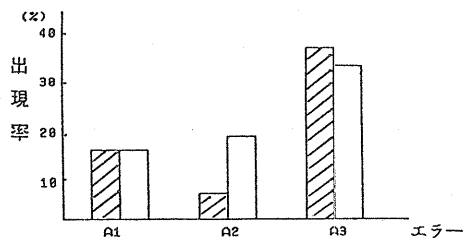


図 3. 10 Q 3 の解答別アセンブラエラーの出現率

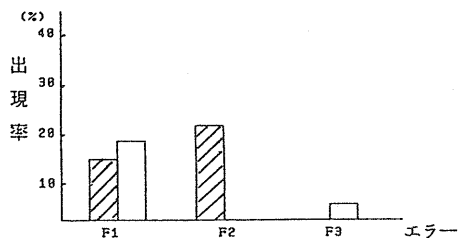


図 3. 11 Q 3 の解答別フォートランエラーの出現率

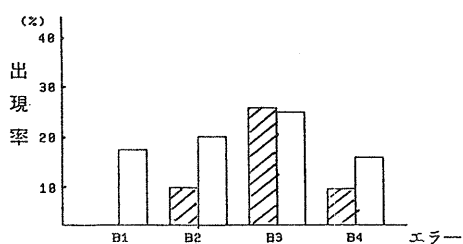


図 3. 12 Q 4 の解答別ベーシックエラーの出現率

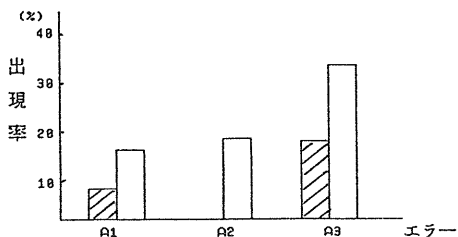


図 3. 13 Q 4 の解答別アセンブラエラーの出現率

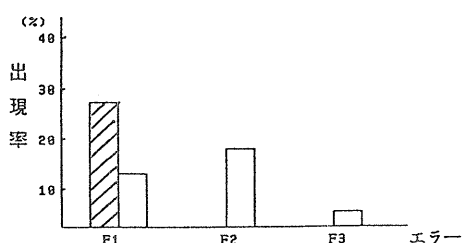


図 3. 14 Q 4 の解答別フォートランエラーの出現率

まず、ベーシックエラーの出現率のグラフ（図3.3、図3.6、図3.9、図3.12）から、アンケートに“Yes”と答えた学生のエラー出現率は、全般的に低い。したがって、ベーシックエラーの出現率は、表3.4からもアンケートの回答によって影響を受けることがわかる。

次に、表3.4から、アンケートの回答によって、フォートランエラーの発生傾向の差があると思われる。しかし、フォートランエラーの出現率のグラフ（図3.5、図3.8、3.14）を見ると、意味のある差や傾向はないことがわかる。したがって、アンケートの回答とフォートランエラーの発生は、無関係であるとした方が良いと思われる。

(3) 相関分析を利用した解析

ベーシックあるいはフォートランを知っている学生と、知らない学生に分けて、4つのプログラムエラーの合計数について、相関分析を行なう。検定の結果が有意となり、寄与率も高かったのは、フォートランを知っている学生に関して、アセンブラとフォートランエラーの間に、正の相関関係があっただけである。

表3.11 Q4に“Yes”と答えた学生のエラーの相関分析表

問題	A	B	F	P
A		-0.404	-0.232	0.311
B	0.163		0.667*	-0.279
F	0.054	0.445		-0.454
P	0.097	0.078	0.206	

(右上方：相関係数、左下方：寄与率、*：5%有意)

表3.11から、アセンブラとフォートランのエラーの合計数とは正の相関関係があることがわかる。これは、アセンブラエラーの発生に対して、フォートランを知っていることが影響を与えているといえる。

(4) 数量化理論第I類を用いた解析

説明変量は、アンケートQ1～Q4の4つで、それぞれのカテゴリ変量は、全て2つである。

表3.12の外的基準について、解析を行なう。

検定の結果、有意であったのは、項番1であった。この内容を表3.13に示す。

表3.12 外的基準

項番	外的基準
1	Bのエラー合計数
2	Aのエラー合計数
3	Fのエラー合計数
4	Pのエラー合計数
5	構文エラー合計数
6	ロジックエラー合計数

表3.13 外的基準がBのエラー合計数の場合

要因	カテゴリウエイト		レンジ	重相関係数	寄与率	F.
	YES	NO				
Q1	0.218	1.090	0.872	0.452	0.203	3.017*
Q2	0.615	0.000	0.615			
Q3	-0.345	0.000	0.345			
Q4	-0.349	0.000	0.349			

(*：5%有意)

表3.13で、Q1の“NO”のカテゴリウエイトが高いことから、パソコンを持っていないことが、エラーの発生を増加させていることがわかる。Q2の“YES”のカテゴリウエイトが高いのは、パソコンを使う機会があっても、ベーシック言語の習熟しているわけではないことを示している。Q3とQ4の“YES”のカテゴリウエイトの値が、負なので、ベーシック、あるいはフォートランを知っていることが、ベーシックエラーの発生を抑制する傾向があると考えられる。

3.3 血液型、アンケート、エラーデータの解析

(1) 分割表を利用した解析

血液型と各種類のエラーの出現率について、分割表を利用して分析した。その結果を表3.14に示す。

表 3. 1 4 分割表による検定の結果

分類	血液型
各回ごとの場合のエラーの出現率	*
構文エラーの出現率	*
ロジックエラーの出現率	*
アンケートの回答	*

表 3. 1 4 から、血液型の違いが、プログラミング技術に、それぞれ異なった影響を与えていることがわかる。そこで、実際にエラーの発生状態やアンケートの回答に、どのような差があるのかを調るために、グラフを用いて比較する。

(2) グラフを用いた分析

血液型別の各回ごとの場合の出現率のグラフを図 3. 1 5 ~ 図 3. 1 8 に示す。

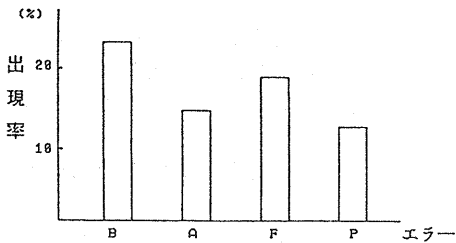


図 3. 1 5 A型

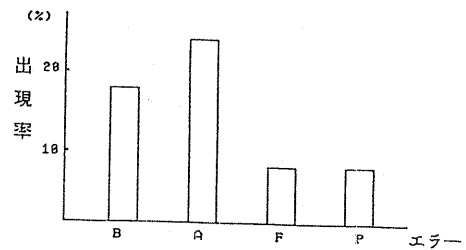


図 3. 1 6 B型

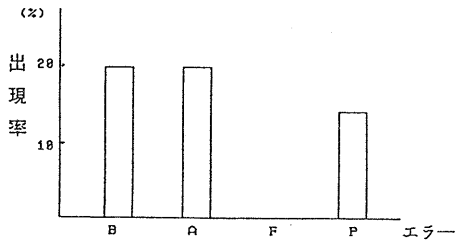


図 3. 1 7 AB型

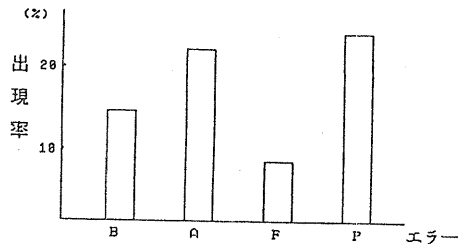


図 3. 1 8 O型

図 3. 1 5 ~ 図 3. 1 8 から見ると、A型の学生は、フォートランを知っている割合が高いにもかかわらず、フォートランエラーの出現率は、他に較べて極めて高いことを示している。

また、ベーシックエラーの出現率に注目すると、A型、B型、AB型の学生のエラー出現率は、下降傾向を示している。しかし、O型の学生には、下降傾向は、見られなかった。特に、各回ごとの場合のエラー出現率のグラフでは、上昇傾向さえ示している。

(3) 分散分析による解析

4つの問題、2種類の共通エラーおよび血液型について、3元配置分散分析を行なう。但し、血液型については、AB型のデータが少ないので、省略した。

表 3. 1 5 基準化したデータ (単位: %)

問題 血液型	B		A		F		P	
	構文	ロジック	構文	ロジック	構文	ロジック	構文	ロジック
A	16.67	22.22	11.11	11.11	33.33	27.78	0	11.11
B	18.18	18.18	18.18	18.18	9.09	9.09	0	0
O	11.11	16.67	11.11	22.22	11.11	16.67	11.11	16.67

表 3. 16 分散分析表 (プーリングした結果)

要因	S	ϕ	V	F	ρ %
血液型	114.25	2	57.13	4.57*	5.9
問題	497.64	3	165.88	13.27*	30.9
エラー	63.05	1	63.05	5.04*	3.4
血液型×問題	678.27	6	113.05	9.04*	40.5
誤差	137.55	11	12.50		19.3
計	1490.76	23			100

分散分析の結果から、血液型、問題、エラーおよび血液型×問題の4つの要因が有意であることがわかる。

4. 考察

- (1) 一般に、プログラミングにおいて、連続して同種のエラーを発生することは少ない。
- (2) フォートランをすでに知っているも、フォートランエラーの発生頻度とは無関係と見られる。これは、知っているというレベルの問題と、フォートラン言語の習熟しにくさの両方を示している。
- (3) フォートランをすでに知っていることが、アセンブラエラーの発生を抑制する傾向に大きな影響を与えている。
- (4) ベーシックに習熟していても、また、していなくても、フォートランとアセンブラのエラーの発生頻度には、ほとんど無関係と見られる。
- (5) パソコンを持っていることおよびパソコンを使う機会があることと、ベーシックを知っていることとは、ほぼ同等の意味であると見てよい。
- (6) パソコンを使用していることが、ベーシックのプログラミング技術を習得することに役立つ。このことから、急速に普及してきたパソコンが、ソフトウェア教育に有効であることがわかる。
- (7) 血液型によるエラーの発生状態についても、かなり特徴的を差異が見られる。特に、A型の学生は、フォートランのプログラミングには向いていないところがあった。また、A型、B型およびAB型の学生は、ベーシックのプログラミング学習において、習熟効果が顕著であるのに、O型の学生には習熟効果がなく、プログラミングには不適である面が見られた。

5. おわりに

今回の報告で使用したデータは、71人のエラーデータとアンケート回答で、かなり少なかった。そのため、解析の精度が充分ではなかったと思われる。また、今回は、血液型による分類等で解析を行なったが、今後の課題として、性別による分類、理科系と文科系の学生による分類等で、エラーの発生状態を調査する必要があるだろう。さらに、コボル、パスカル等の他のプログラミング言語についても、今後、調査を進める必要がある。

今後、このような調査を、根気よく繰り返して行ない、データを収集し縦と横の繋がりを広げて、広範な視野から解析を加えることが重要であろう。

6. 参考文献

- (1) 菅野文友：「ヒューマン・エラーのメカニズム」、日科技連出版社 (1980)。
- (2) 菅野文友：「ソフトウェアの信頼性」、日科技連出版社 (1983)。
- (3) 古林隆：「統計解析」、培風館 (1981)。
- (4) 平林隆一、菅野文友、猪股邦臣：“プログラミング教育におけるエラー分析の一例 (第1報)”，情報処理学会、第28回ソフトウェア工学研究会 (1983)。