3. Experiments

3.1 The Test Collections

Experiments were done on 3 test collections; NPL, Evans and Evans Titles. Their sizes are summarised in the following table.

	DOCUMENTS				QUERIES				RELEVANCE DOCUMENTS		
	No.	MAX	MIN	AV.	No.	MAX	MIN	AV.	MAX	MIN	AV.
NPL	11429	105	1	19.96	93	10	2	7.1	84	1	22.4
EVANS	2542	57	3	16.12	33	36	12	24.5	53	3	24.18
EVANS TITLES	2542	18	0	6.51	33	35	12	23.58	53	3	24.18

Notes

The NPL collection is almost identical to the collection used by Robertson, van Rijsbergen and Porter (1981). The only change made to the collection was the removal of terms (a total of 4) from the queries to make the large queries slightly shorter. More information about the collection can be found in Vaswani and Cameron (1970).

The "Evans" and "Evans Titles" collections are based on the collections used by Sparck-Jones and Webster (1980), but the collections used by us have been modified in the following ways :-

(i) A fairly large stop-list (about 300 words) was used to remove common and non-useful words (like please) from documents and queries.

(ii) Documents and queries terms were stemmed using Martin Porter's stemming algorithm (Porter 1980)

(iii) The six largest queries were discarded. This was necessary because our program is limited to a maximum query size of 36 terms. This limit was imposed so that we could represent interactions by bit-maps.

The original documents consisted of a title followed by a number of manually chosen keywords and phrases. For the "Evans" collection the title and keywords were merged together and used as the document description. For the "Evans Titles" collection only the title was used. The small difference between the queries of the two collections arises because terms that didn't occur in any document were removed from the queries. It should also be noted that our two collections differ from those used by Sparck Jones and Bates in that we used the "need statement" form of the request in both cases, but in one case used the manual indexing as well as titles. Further information albout this collection can be found in Evans (1975 a,b).

3.2 The Experiments

We did two different types of feedback experiments : the usual half-collection experiments and continued-searching experiments in which an initial search is done to obtain feedback information which is used to complete the search.

3.2.1 Half-collection experiments

These were done in the usual way. First the document collection was divided into two parts, odd and even numbered documents. In some experiments an initial search is done on one half of the collection and a small number (usually 20) documents are retrieved. The relevant documents in those 20 are then used to recalculate the weights

which are evaluated by doing a full search on the other half of the collection. In the rest of the experiments the relevant documents in the whole of one half of the collection are used to calculate weights which are then evaluated on the other half of the collection.

3.2.2 Continued searching experiments

These are intended to be an imitation of what would happen if relevance feedback was used in an actual retrieval search.

First a few documents are retrieved (5,10,15 or 20) using the best weights available without relevance information. Any relevant documents found are then used to recalculate the weights and the search is continued for the rest of the collection. The two searches that are then compared are those in which

(i) the first few documents are retrieved without relevance informatin and then the remainder are retrieved using information obtained in the first part of the search.

(ii) the first few documents are retrieved as above but then the search is continued without changing the weights.

Incidently, there must be an optimum size for the initial search. If it consists of zero documents there will be no relevance information and if it consists of the whole collection there will be no continued search. In both cases the two searches (i) and (ii) will be identical.

3.2.3 Evaluation

Our principal method of evaluation was to produce recall-precision graphs. These were obtained using the standard recall cutoff method (van Rijsbergen 1979) and average values of precision were calculated for recall at 10% intervals from 10% to 90%.

The only siginficance test with any theoretical justification which is applicable to these experiments is the sign test. This is based on the fact that, given the null hypothesis that one sort of query-weight is no better than another, we would expect the number of queries in which the first type of weight does better to have a binomial distribution with probability 0.5.

To do this test we need some overall measure of retrieval performance that can be used , query by query,to compare two sets of weights. There are plenty of possibilities, for example we could use precision at some fixed level of recall, but all our tests were done with two measures : normalised recall and normalised precision. Both measures are based on the whole recall-precision graph but normalised precision is biased to the low recall, high precision end of the graph (van Rijsbergen, 1979).

3.3 Logistic Weights

For the simple linear logistic model, without interactions but with a normal prior, the logistic weights are those which maximise $g(\underline{w})$ defined in equation (4) section 2.3. That is

$$g(\underline{w}) = \sum_{t \varepsilon Q} w_t r_t - \sum_{\substack{T \\ T \subseteq Q}} n_T \log(1 + \exp \sum_{t \varepsilon T} w_t) - \frac{1}{2\sigma^2} \sum_{t \varepsilon Q} (w_t - \mu_t)^2$$

where

- w_t is the weight associated with term t
- r_t is the number of known relevent documents containing term
- Q is the set of terms being considered, in this case the set of terms in the query
- T is an arbitary subset of Q.
- n_T is the number of documents which contain all the the terms in and none of the terms in $Q \setminus T$
- μ_t, σ^2 are the means and variance for the assumed normal prior (see 3.3.2)

Newton's method was used to maximize $g(\underline{w})$. This is an iterative algorithm which takes the following form.

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \alpha_k \, (\underline{H}(\underline{w}^{(k)}))^{-1} \, \underline{b} \, (\underline{w}^{(k)})$$

where $\underline{b}\,(\underline{w})$ is the vector of first partial derivatives of g, and $\underline{H}(\underline{w})$ is the Hessian, or matrix of second partial derivatives of g.

α_k is a step length, and is chosen to make

$$g(\underline{w}^{(k+1)}) > g(\underline{w}^{(k)})$$

As $\underline{H}(\underline{w})$ is negative definite we can always find an $\alpha_k > 0$. There are a number of algorithms available for choosing α_k but we used the following rather crude one.

(1) Set $\alpha_k = 1$
(2) If $g(\underline{w}^{(k+1)}) > g(\underline{w}^{(k)})$ then use that value of α_k
(3) Otherwise set $\alpha_k = 0.8 \times \alpha_k$ and go to (2)

This is certain to find an adequate α_k but it is unlikely to find the best one. In fact α_k will equal 1 for all but the first few iterations.

The iteration was continued until $\left\| \underline{w}^{(k+1)} - \underline{w}^{(k)} \right\|_2 < \varepsilon$ where ε is a fixed small number (we used $\varepsilon = 1 \times 10^{-7}$). The number of iteration steps needed depends mostly on how far $\underline{w}^{(o)}$ is from the final value. As we used $\underline{w}^{(o)} = \underline{\mu}$ this is strongly dependent on the size of σ^2.

The amount of work that has to be done for each iteration depends mostly on the size of Q, that is on the number of terms in the query. Because of the way our queries were stored we had to limit our query size to a maximum of 36 terms and for queries of that size convergence took typically 3-4 seconds of CPU time on

the Honeywell 6000. For large queries most of the time is probably spent inverting the Hessian and the algorithm could probably be speeded up by using the same Hessian for several iterations.

3.3.1 The evaluation of $\underline{b}$ and $\underline{\underline{H}}$

If the formula for $g(\underline{w})$ above is differentiated we get the following forumulae for $\underline{b}\,(\underline{w})$ and $\underline{\underline{H}}\,(\underline{w})$

$$b_t(\underline{w}) = r_t - \sum_{\substack{T \\ t \epsilon T}} n_T \frac{\exp(\sum_{u \epsilon T} w_u)}{(1 + \exp(\sum_{u \epsilon T} w_u))} - \frac{1}{\sigma^2}(w_t - \mu_t)$$

$$H_{s,t}(\underline{w}) = - \sum_{\substack{T \\ t \epsilon T \\ \text{and } s \epsilon T}} n_T \frac{\exp(\sum_{u \epsilon T} w_u)}{(1 + \exp(\sum_{u \epsilon T} w_u))^2} - \frac{\delta_{st}}{\sigma^2}$$

where

$$\delta_{st} = \begin{cases} 1 & s = t \\ 0 & s \neq t \end{cases}$$

A table of r_t and of non-zero values of n_T needs to be evaluated and stored just once for each query.

3.3.2 The prior distribution

In the present experiments, we have generally assumed that the Sparck Jones collection frequency weights provide appropriate prior information about the relevance weights. In other words, the prior mean μ_t is taken to be the collection frequency weight according to expression (2) of section 1.2.2. It will be clear that the logistic

model is neutral in this respect : any particular formula could be built into the prior.

The variance of the prior is interpreted as measuring the relative reliance that is to be placed on the prior mean, relative to direct evidence from relevance feedback (a low variance implies high reliance on the prior mean). Experiments with different variances are described.

3.3.3 The inclusion of interactions

It is quite easy to include specific term interactions into a query. For example the inclusion of a two-term interaction will have exactly the same effect on the model as the addition of an extra term which is contained in all the documents which contain both the terms. Hence we can treat interactions as "pseudo-terms" in much the same way that we include a term t_o which is assumed to be in every document.

3.4 Results of the experiments

The weighting systems used in the experiments are as follows :

word : unit weights are used (equivalent to level of coordination).

frequency : collection frequency weights are used, according to expression (2) of section 1.2.2.

RSJ : weights are calculated according to the Robertson/Sparck Jones model, "0.5" formula (expression (1) of section 1.2.1), Complement method for non-relevant documents.

Logistic : $v = x$ weights are calculated according to the logistic model, normal prior with means equal to the frequency weights, variance x , maximum posterior estimates, complement method for non-relevant documents.

3.4.1 Continued searching experiments

In all of these the initial search (retrieving 5,10, 15 or 20 documents) is done using the frequency weights. Then the search is continued using weights obtained from feedback information obtained in the initial search.

For each experiment we list, in order,

(i) the length of the initial search

(ii) the precision values for recall values of 10%, 20% ....,90%

(iii) significance levels for the sign test of the null hypothesis that feedback produces no improvement. Two values are given, using normalised recall and normalised precision respectively to compare performance. Both values are given as a percentage. Hence if both (or either) value is less than 1 say we would reject the null hypothesis at the 1% level.

The NPL Collection

Search without feedback (frequency weights).

54 45 37 31 24 18 15 11 6

RSJ Weights

5	56	47	40	32	26	20	15	11	7	0.2	0.2
10	56	47	40	32	26	19	15	11	7	3	3
15	55	47	40	33	26	20	16	12	7	0.2	0.2
20	55	47	39	32	26	20	16	12	8	0.6	0.6

Logistic weights $v = 1$

5	55	45	38	31	25	18	15	11	6	86	27
10	55	46	39	32	25	19	15	11	7	24	5
15	55	46	38	32	26	19	15	11	7	30	2
20	55	45	38	32	26	19	15	11	6	50	7

The EVANS Collection

	62	48	39	33	27	22	13	9	5	

RSJ weights

1	62	51	41	35	29	22	14	9	6	5	3
5	65	52	45	38	32	26	15	11	8	0.2	0.2
10	63	51	44	38	33	26	16	12	7	0.3	0.2
15	63	50	43	38	33	26	16	12	7	0.02	0.1
20	63	50	42	37	32	26	17	13	8	0.01	0.01

Logistic weights $v = 1$

5	62	50	40	33	28	22	14	9	5	50	100
10	63	49	41	35	29	23	15	10	6	3	12
15	63	49	41	36	29	23	15	10	6	0.6	6.2
20	63	49	42	35	30	23	15	10	6	0.03	1.6

The EVANS TITLES Collection

	48	35	23	15	11	7	5	4	2

RSJ weights

5	50	37	27	18	12	8	6	4	2	50	14
10	49	37	27	19	12	8	6	4	2	34	36
15	48	37	26	18	12	8	6	4	2	34	50
20	49	36	27	18	13	8	6	4	2	50	14

Logistic weights $v = 1$

5	48	36	23	15	11	7	5	4	2	75	75
10	48	36	24	16	11	7	5	4	4	75	75
15	48	35	25	16	11	7	5	4	2	50	50
20	48	35	25	16	11	7	5	4	2	50	75

3.4.2 Half Collection experiments

We have grouped these according to the half-collection used to evaluate the weights. For each experiment we give the source of the weights and then the average precision values for recall at 10% intervals from 10% to 90%.

When giving the source of the weights the following abreviations are used.

even, odd : All the known relevant documents in the half collections (either even or odd numbered documents) are used to calculate the term weights.

even/20 :
odd/20 : A preliminary search is done on the specified half collection using unit weights and the relevant documents found among the first 20 retrieved documents are used to calculate the weights.

The NPL Collection, even numbered documents

word	51	43	36	28	24	20	14	11	7
frequency	57	48	39	31	26	23	17	14	10
RSJ, odd	62	56	49	39	34	29	21	17	12
logistic v = 4, odd	66	57	48	40	34	29	22	18	12
RSJ, odd/20	60	52	43	35	30	25	17	14	10
logistic v = 4, odd/20	61	52	42	34	29	25	19	14	10

NPL, odd numbered documents

word	49	39	29	24	22	16	12	8	5
frequency	57	48	39	33	28	21	17	12	8
RSJ, even	65	56	46	38	34	26	21	15	10
logisitic v = 4, even	66	58	47	41	37	29	22	16	9
RSJ, even/20	59	50	40	34	29	22	18	13	8
logistic v = 4, even/20	61	51	42	36	31	24	18	12	7

EVANS, even numbered documents

word	55	46	35	32	27	16	14	11	8
frequency	58	53	39	35	32	21	17	13	9
RSJ, odd	74	69	63	54	46	36	26	21	14
logistic v = 1, odd	64	57	47	45	39	28	22	15	12
logistic v = 4, odd	67	64	54	49	44	30	23	16	13
logistic v = 10, odd	70	68	58	50	44	31	23	18	14
logistic v = 20, odd	73	69	60	51	44	30	22	16	12
RSJ, odd/20	69	64	56	48	38	27	19	15	11
logistic v = 1, odd/20	61	54	43	38	33	21	17	13	10
logistic v = 4, odd/20	62	57	46	40	35	23	19	14	11

EVANS, odd numbered documents

word	63	51	40	35	30	21	14	10	6
frequency	70	60	47	42	35	24	15	12	8
RSJ, even	72	66	57	53	44	37	26	20	10
logistic v = 1, even	69	60	52	49	42	30	19	15	8
logistic v = 4, even	69	61	52	48	41	32	21	16	8
logistic v = 9, even	68	59	52	46	40	32	21	17	9
logistic v = 25, even	69	58	51	46	39	31	22	17	9
RSJ, even/20	68	61	51	45	35	29	20	17	10
logistic v = 1, even/20	69	61	48	44	37	27	17	14	9
logistic v = 4, even/20	66	60	47	43	36	28	16	14	7

EVANS-TITLES, even numbered documents

Word	42	34	27	21	16	11	9	7	6
frequency	50	35	27	22	20	13	10	7	6
RSJ, odd	58	49	38	27	24	15	12	8	7
logistic $v = 1$, odd	54	39	31	24	22	14	11	8	6
logistic $v = 4$, odd	57	45	32	26	23	16	12	8	7
RSJ, odd/20	54	45	33	26	23	15	12	8	7
logistic $v = 1$, odd/20	54	39	30	24	22	14	11	8	7
logistic $v = 4$, odd/20	55	44	32	24	22	15	12	8	7

EVANS-TITLES, odd numbered documents

Word	50	35	22	18	12	8	5	3	2
frequency	54	43	27	20	14	8	4	3	2
RSJ, even	65	58	43	30	22	14	7	5	3
logistic $v = 1$, even	56	50	34	23	15	11	6	5	3
logistic $v = 4$, even	58	48	36	26	17	12	6	4	3
RSJ, even/20	59	52	33	25	19	12	5	5	3
logistic $v = 1$, even/20	54	43	27	20	13	9	5	5	3
logistic $v = 4$, even/20	54	44	29	20	14	9	5	4	3

3.5 Survey of the results

3.5.1 Continued searching experiments

This form of evaluation, it should be pointed out, is a new one, proposed as an alternative to the traditional "residual ranking" experiments. In the present experiments, the documents used for feedback are retained in the evaluation set. We suggest that this procedure is more realistic (i.e. simulates a real-life search better) than residual ranking.

As might be expected, the continued searching experiments show smaller performance differences between feedback and no feedback than residual ranking. Feedback always improves performance a little, but often not significantly. RSJ weights, always perform better than logistic weights with the same feedback set, though again often not significantly. A detailed examination of the figures suggests that (a) the larger the feedback set, the better the performance at the tail (high recall) end of the curve; (b) a small feedback set seems likely to produce a substantial performance improvement in just a few queries; a larger feedback set gives slighter improvements spread over a larger number of queries (particularly evident in Evans). Some of these points are illustrated in Fig. 1, which gives some results from the Evans collection.

3.5.2 Half-collection experiments

It has proved very difficult to generalize from these results. Out of the six sets of experiments, one can find examples where RSJ outperforms logistic and examples of the opposite; examples where higher-variance priors for the logistic improve performance and examples where they depress it. However, in general the results show no strong evidence for logistic weights.

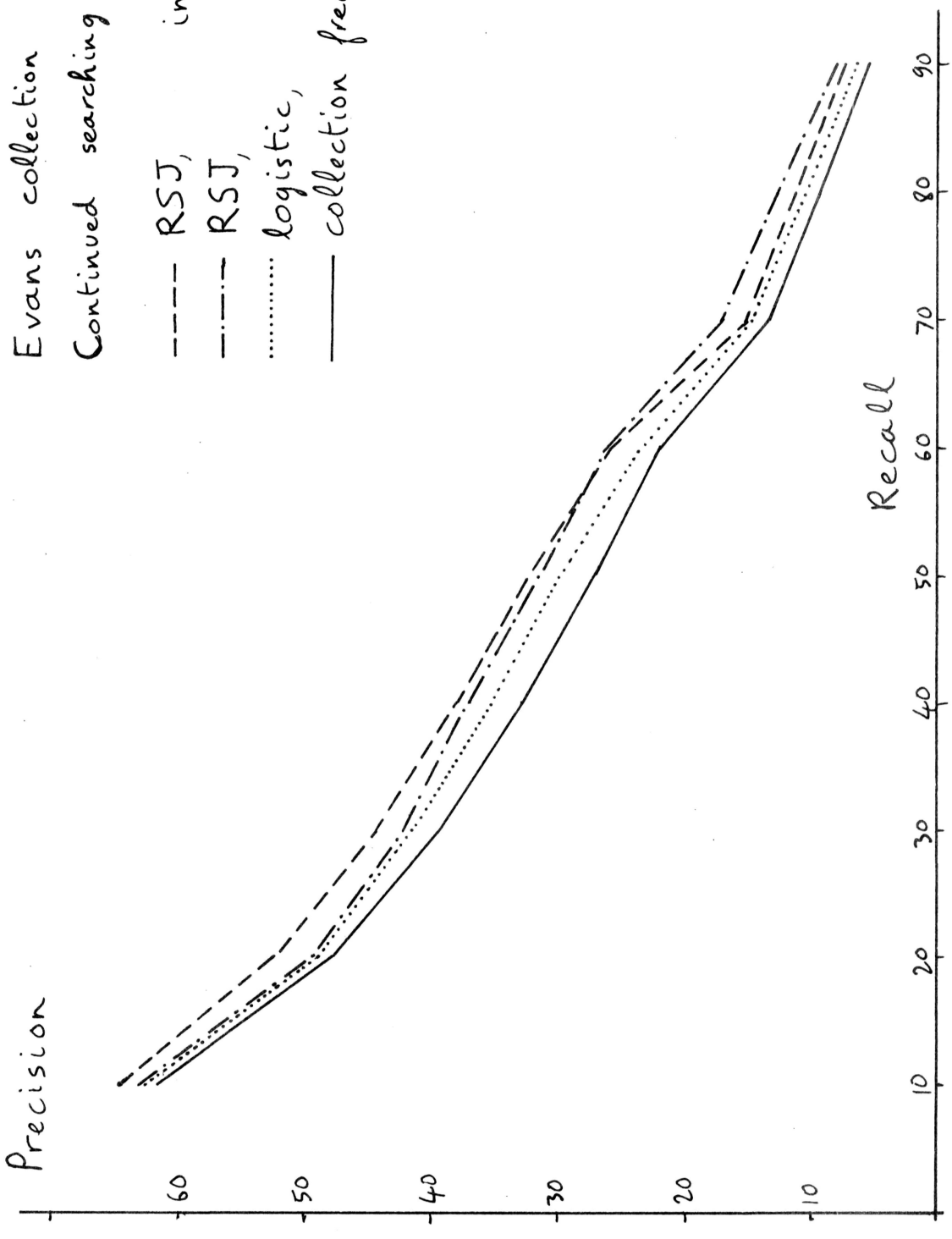
Evans collection
Continued searching
---- RSJ, initial search 5
-·-·- RSJ, " 20
..... logistic, " 20
—— collection frequency
Precision
60
50
40
30
20
10
10
20
30
40
50
60
70
80
90
Recall

Figure 1

3.5.3 Other experiments

Some further experiments were performed early in the project, as a basis for further work. The results of these experiments are not presented in detail here; however, it is worth indicating some of the main findings.

Some experiments were done on the inclusion of interactions : selected pairs of query terms were included as indicated in section 3.3.3. The pairs were selected by inspection, as likely to have meaning as a phrase which is not apparent in the separate words (e.g. high, frequency). All these experiments were negative in the sense that including interactions depressed performance; the more interactions were included, the more performance was depressed. This somewhat strange result lead us to attempt a theoretical answer to the question : When should an interaction be included ? The resulting theoretical development is presented and discussed in section 4.

Some experiments were done with the prior for the logistic model having zero mean (rather than collection-frequency based) and large variance - i.e. as neutral a prior distribution as possible. These performed worse than the frequency-based mean. No experiments, however, have been done using other forms of prior, such as constant (non-zero) mean or a peaked function such as Salton's.

We attempted to get some direct evidence on the possible ideal form of frequency-based prior. We plotted, for individual query terms, the neutral-prior retrospective logistic weight against the collection-frequency weight (Fig.2). This scatter diagram is notable only for showing virtually no pattern - there is a very slight positive correlation, but certainly no possibility of distinguishing between alternative relationships. We considered the possibility that this lack of pattern was due to variations between queries : that is, that collection-frequency weights might be a good predictor of the relative values of different terms in a

given query, but not their values in relation to terms from other queries. So we normalized the weights for each query. The result is given in fig.3; the correlation is a little highter, but still not really good enough to draw any useful conclusions. This area is discussed further in section 5.

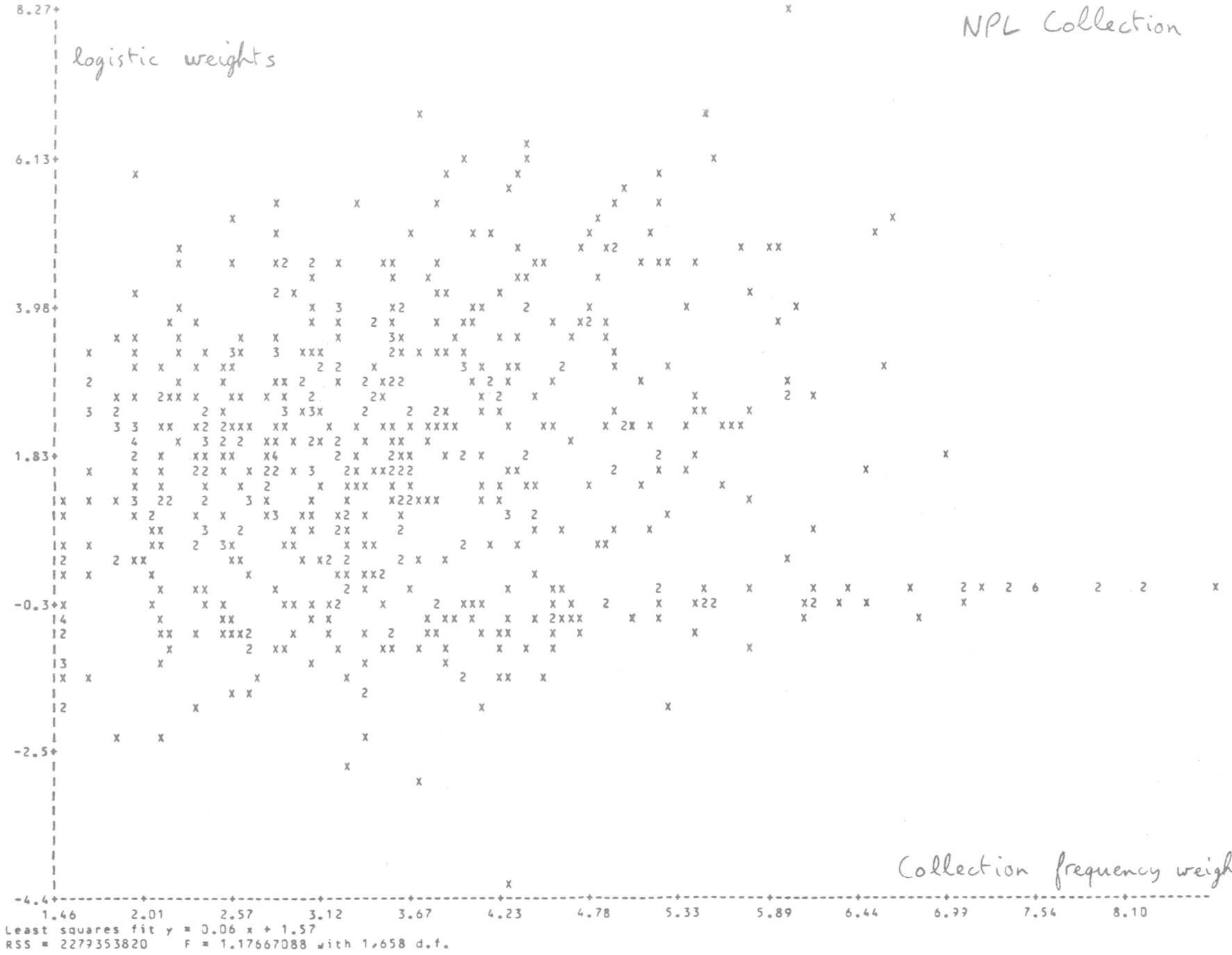
NPL Collection
logistic weights
8.27+
6.13+
3.98+
1.83+
-0.3+
-2.5+
-4.4+
Collection frequency weights
1.46
2.01
2.57
3.12
3.67
4.23
4.78
5.33
5.89
6.44
6.99
7.54
8.10
Least squares fit y = 0.06 x + 1.57
RSS = 2279353820 F = 1.17667088 with 1,658 d.f.

Figure 2

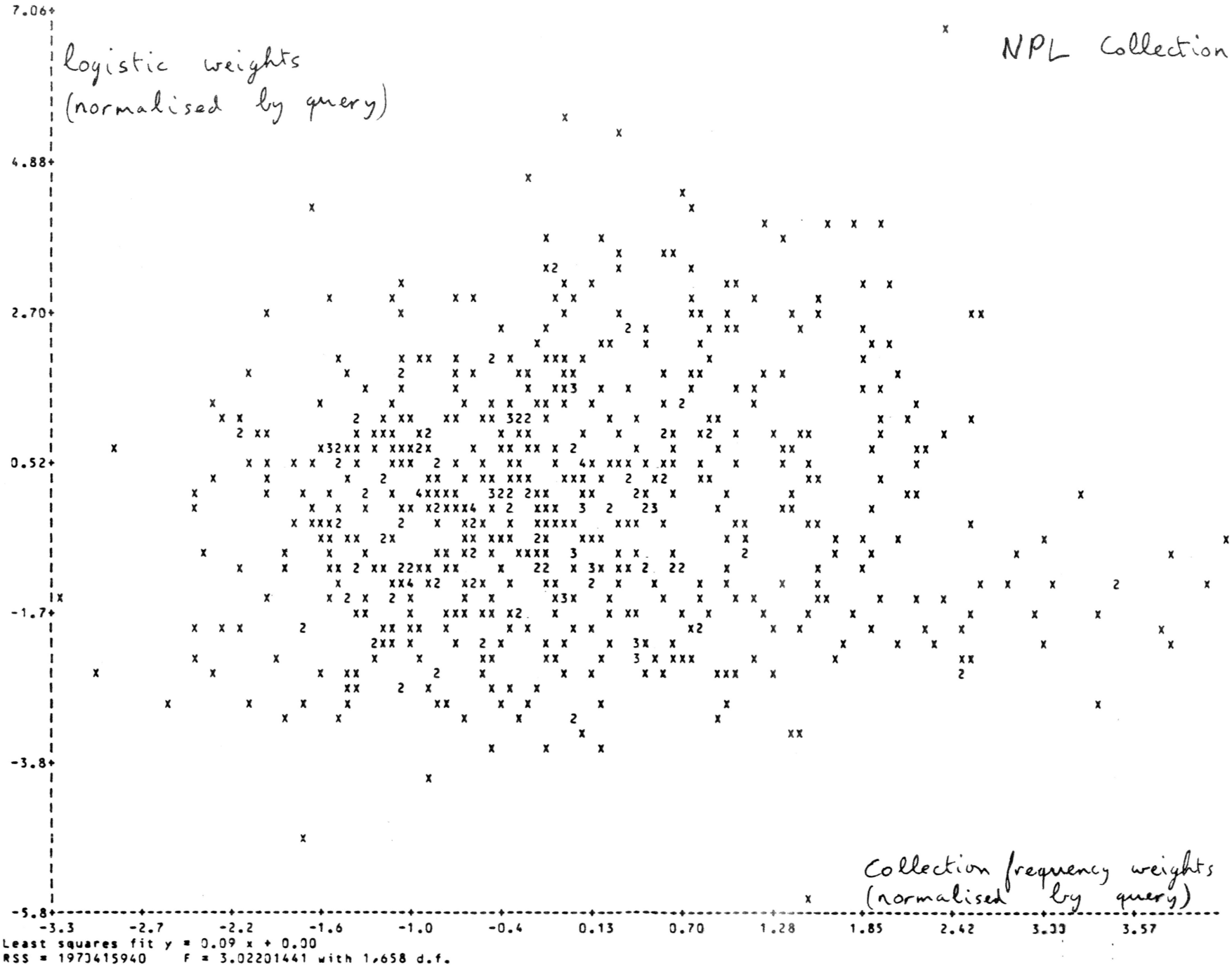
NPL Collection
logistic weights
(normalised by query)
7.06+
4.88+
2.70+
0.52+
-1.7+
-3.8+
-5.8+
-3.3
-2.7
-2.2
-1.6
-1.0
-0.4
0.13
0.70
1.28
1.85
2.42
3.00
3.57
Collection frequency weights
(normalised by query)
Least squares fit y = 0.09 x + 0.00
RSS = 1973415940 F = 3.02201441 with 1,658 d.f.

Figure 3