The Distributional Characteristics of Lead Mine Yields

by

James R Warren BSc MSc PhD PGCE

The Logarithm of the Total Production of Lead Concentrate for each mine of Southern Wales was established[1] and plotted against Rank. A similar exercise was undertaken for the Logarithm of Total Production of Lead Metal in respect of the British prefectures of Shropshire and Montgomeryshire[2].

Graphs show that the mines group into Statistical Provinces or CLASSES whose plotted points describe polynomial curves definitive of each segregation.

The graph for Southern Wales shows two clear cubic curves for two productive classes plus a third degenerate point plotting the output of a tiny trial. The Shropshire graph also develops three classes: The most prolific follows a cubic curve, the middling class a quadratic and the small mines a linear relation. The Montgomeryshire graph is much more complex. Six or Nine separate classes can be resolved by inspection and the curves displayed are usually cubic or quadratic. It is only in the Montgomeryshire case that the full assemblage approximates the Zipfian paradigm of a linear relationship between the logarithm of size and the distributional rank.

The Co-ordination of Data

In order to facilitate analysis and comparison it is desirable to perform an elementary transformation of the data to express it in relative terms. This has the twin advantages of making Log Production, T, and Rank, R, both non-dimensional and also normalises each variable to the interval zero to unity.

To maintain a linkage of this distributional data to the original we may record T_{max}, the Log Production of the Most Prolific Mine, and m, the Number of Mines in the region.

The following equations define the required transformations:-

$$p_i = \frac{T_{\max} - T_i}{T_{\max}} \qquad \textbf{Eqn.1}$$

so long as $T_i \geq 0$ and:-

$$r_i = \frac{m - R_i}{m - 1} \qquad \textbf{Eqn.2}$$

It can be seen that under this scheme the most productive mine has (p_i=0:r_i=1) and the least ($p_i \rightarrow 1$:r_i=0).

A Descriptive Model

Experiments and trials were made with various model components applied to raw data (T,R). Both linear and polynomial regressions were essayed to eliminate trend from regional distributions. The former only eliminated 51% of variation but the latter some 97%. In both cases residuals tended to describe a damped oscillation which several heuristics based upon the approximate analytic function failed satisfactorily to approximate. It then occurred to me that the worksheet plot of the oscillating residual was disturbingly like Fig 10.1b of "Computational Mathematics"[3], an expression of the error remaining after fitting a polynomial to data.

It occurred to me that I was chasing chimeras and that I required to return to first principles and *contemplate* my discovery of the polynomial provinces rather than *ignore* it.

Taking this view I might summarise the functional predictor of production as:-

$$T_i = X_{\alpha,\omega} \cdot P_i \qquad \textbf{Eqn.3}$$

where Switching Function, X, has these characteristics:-

$$X(r) = \begin{cases} 1 & \text{if } \alpha \le r \le \omega \\ 0 & \text{elsewhere on the interval} \end{cases} \qquad \textbf{Eqn.4}$$

Modeling the Switching Function

A FOURIER SERIES approximates an arbitrary periodic function subject to the DIRICHLET CONDITIONS[4]:-

1. Function discontinuities (if any) are finite in any period

2. The function must contain a finite number of extrema in any period

3. The function must be absolutely integrable in any period

Of especial interest to us is the guarantee and limitation:-

> **"Suppose that f(x) is a periodic function of period 2π and is defined and bounded for $0 \le x < 2\pi$, and suppose that the interval (0,2π) can be split into finitely many subintervals in each of which the function is continuous and monotonic. Then the Fourier series of f(x) converges at each point of continuity x_0 to $f(x_0)$, and at a point a jump discontinuity x to the mean value .... of its left and right limiting values".**[5]

For Period = 2π/ω the FOURIER SERIES is defined by the equations[4]:-

$$P(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n \text{Cos} n\omega t + b_n \text{Sin} n\omega t) \qquad \textbf{Eqn.5}$$

whilst for n = 1,2,3,...:-

$$a_n = \frac{\omega}{\pi}\int_{-\pi/\omega}^{\pi/\omega} P(t)\text{Cos} n\omega t.dt \qquad \textbf{Eqn.6}$$

$$b_n = \frac{\omega}{\pi}\int_{-\pi/\omega}^{\pi/\omega} P(t)\text{Sin} n\omega t.dt \qquad \textbf{Eqn.7}$$

In practical computation the Equation Five summation must be taken to a finite number of series components (v say) depending upon the precision of the harmonic synthesis required. The coefficient integrals may be analytic but may in principle be numerical if necessary or convenient.

It is of course the case that we need a shiftable square-wave pulse of sufficient stability within the interval (0,2π).

This pulse is controlled by the Width (w) and Offset(Ω) Parameters illustrated below[6]:-

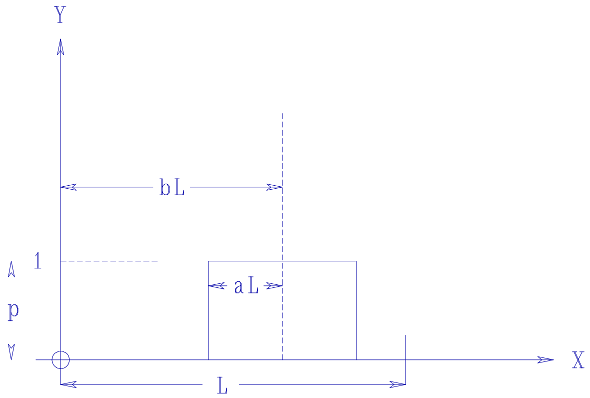


Figure 1

The relevant Fourier Coefficients are:-

$$\frac{1}{2}a_0 = 2pa \qquad \textbf{Eqn.8}$$

$$a_n = \frac{4p \ \mathrm{Sin} n\pi a \ \ \mathrm{Cos} n\pi b}{n\pi} \qquad \textbf{Eqn.9}$$

$$b_n = 0 \qquad \textbf{Eqn.10}$$

To express the system of Equations Five, Eight, Nine and Ten in computational terms we require to remember that mine output rank is assimilated to the interval (0,1) by Equation Two and that the Width and Offset Metrics a and b need to be expressed co-ordinately.

Accordingly, for $1 \leq i \leq m$ and $1 \leq j \leq v$:-

$$\frac{1}{2}a_0 = \frac{R_\omega - R_\alpha + 1}{m}$$

Eqn 11

$$a_i = \frac{4\mathrm{Sin}\left(j\pi.\frac{R_\omega - R\alpha + 1}{2m}\right).\mathrm{Cos}\left(j\pi.\left[\frac{R_\omega - R\alpha + 1}{2m} + \frac{R\alpha - 1}{m}\right]\right)}{j\pi}$$

Eqn 12

$$b_i = 0$$

Eqn 13

If for efficiency and convenience we define the Relative Range

Constant, z, using:-

$$z = \frac{R_\omega - R_\alpha + 1}{2m} \qquad \textbf{Eqn.14}$$

we may declare the Shifted Square-Wave Fourier Series as:-

$$X = 2z + \sum_{i=1}^{v} \frac{4\ \mathrm{Sin} j\pi z\ .\ \mathrm{Cos} j\pi\left(z + \frac{R_\alpha - 1}{m}\right).\ \mathrm{Cos} j\pi[1 - r_i]}{j\pi} \qquad \textbf{Eqn.15}$$

Numerical trials of Equation Fifteen using Program FOURSERI.BAS for a variety of m and v established that for acceptable accuracy v should compare with or exceed m. Special conditions involving few mines in a region (low m) could give surprisingly accurate X but in general the penalty of low v was considerable rounding of the range-pulse flanks and averaging ($p \approx 0.5$) would often persist for r_ω at the rank of the classe's least productive mine.

Modelling the Polynomial Function

The highest degree delineation so far detected is a Cubic, represented by the general form:-

$$P = a_0 r^0 + a_1 r^1 + a_2 r^2 + a_3 r^3 \qquad \textbf{Eqn.16}$$

or:-

$$P = \prod_{i=0}^{3} a_i r^i \qquad \textbf{Eqn.17}$$

The Cubic Polynomial is general enough to define the Quadratic and Linear Degrees by zeroisation of a_3 and a_3,a_2 respectively.

Measures of Fitment Quality

To assess the extent to which the segmentalised polynomial fitments agreed with the original data a series of Ranked Log Productions were synthesized from the Class Polynomial Regression Coefficients, each set of coefficients applied only to the Class Ranks concerned. (This synthetic series is headed "Logically Switched Model" in the Model Reports *.MOD).

This series was then compared with the Original Data computed directly from mine histories.

The formula employed was:-

$$C_d = \frac{\Sigma(T_{syn} - \mu_T)^2}{\Sigma(T_i - \mu_T)^2} \qquad \textbf{Eqn.18}$$

When it came to comparing Fourier Switched Models against this Logically Switched Series the Determination Coefficient Formula of Equation Eighteen proved unsatisfactory.

Clarity required the use of some formula which would measure the residual error in the approximation.

The Fourier Switched Models also embody polynomial syntheses for each Class by utilising the Switching Function of Equation Fifteen to bound the treated Classes.

The chosen Residual Error Formula was:-

$$\varepsilon = \frac{\Sigma(T_i - T_v)^2}{\Sigma T_i^2} \qquad \textbf{Eqn.19}$$

The Residual Error was also computed for the Logically-Switched Series against the Original Data in order to provide a reference metric for adjudging convergence.

The Organisation of Data

As aforementioned the Lead Mine Populations of three British provinces were analysed:-

(A) SOUTHERN WALES

An area of some 5720 km^2 comprising the old Welsh counties of Pembrokeshire, Carmarthenshire, Radnorshire, Breconshire, and the southern fringes of Cardiganshire. In 1974 the Welsh counties were abolished and the larger part of this area is now in Dyfed.

Local hills provide sheep turf, but the preponderant landscape is a low, well-watered and very verdant cow pasture dotted with large villages and small towns.

The typical country rock is Lower Palaeozoic argillite.

(B) SHROPSHIRE

This English county of 3487.581 km^2 contained almost all its lead mines on the Shelve Plateau, a lonely tract of scrub and pasture near the Montgomeryshire border.

The country rock is Paleozoic with a Precambrian quartzite inlier (The Stiperstones). None of the former facies are Dinantian blue limestone.

(C) MONTGOMERYSHIRE

This was a large (2064.342 km^2) county of Central Wales which, in 1974, became an important component of the new Powys prefecture.

Paleozoic argillites predominate in a landscape of grassy hills and silent, sequestered valleys.

I have organised the input and output data into sets of five reports for each province. Each set forms the respective Appendices One, Two and Three.

Province Report Components

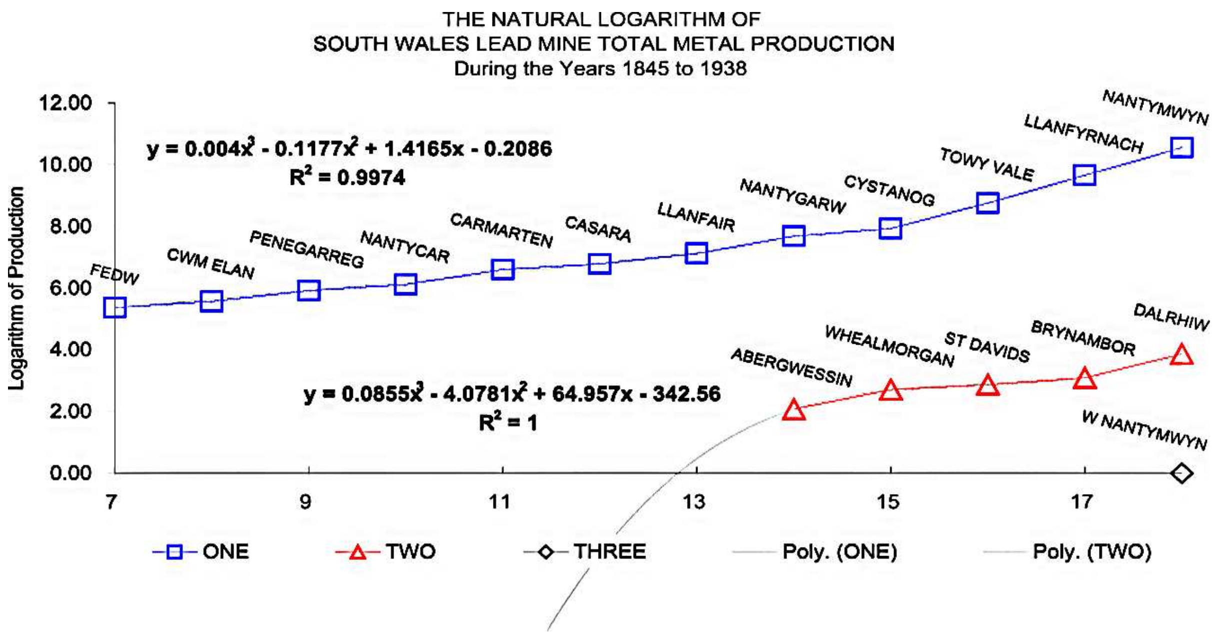
(1) STATISTICAL PROVINCE GRAPHS

Headed "The Natural Logarithm of Total .... Lead Production ...." these illustrate the segregation of mines into Classes when ranked by Log Production.

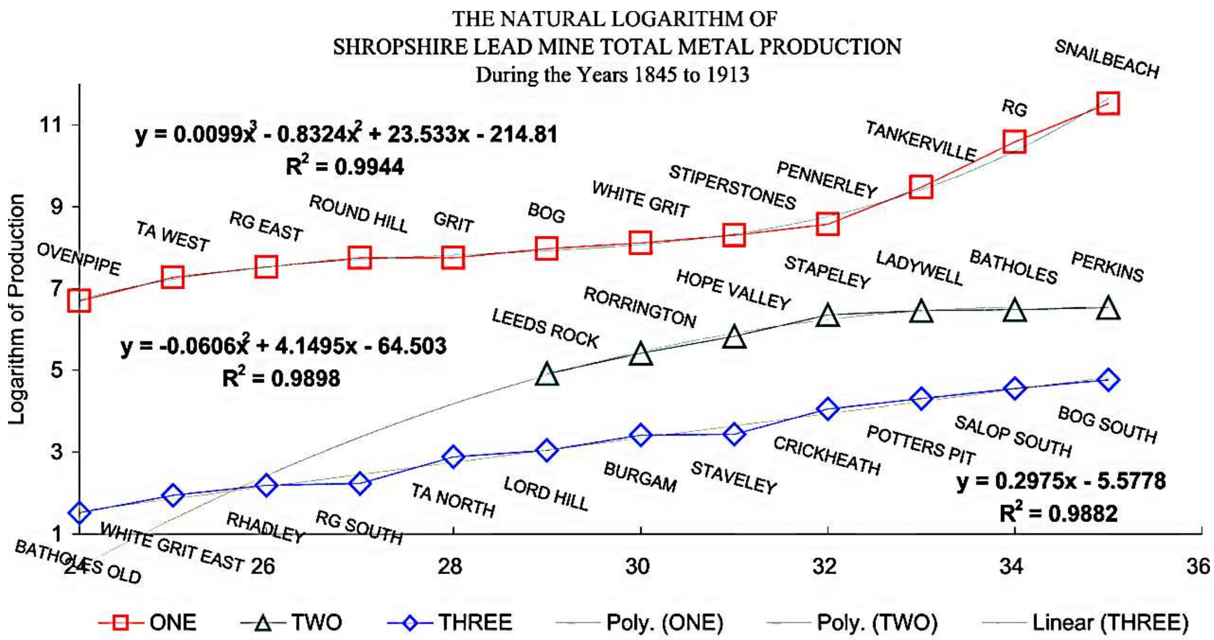
Note the two clear cubic provinces on the South Wales plot, the apparent cubic, quadratic and linear orders of the Shropshire rankings, and the several separate cubic or quadratic classes for Montgomeryshire.

The Correlation Coefficients for Montgomeryshire are all based on cubic fitments and in the Shropshire case the four smallest, erratic mines have been suppressed.

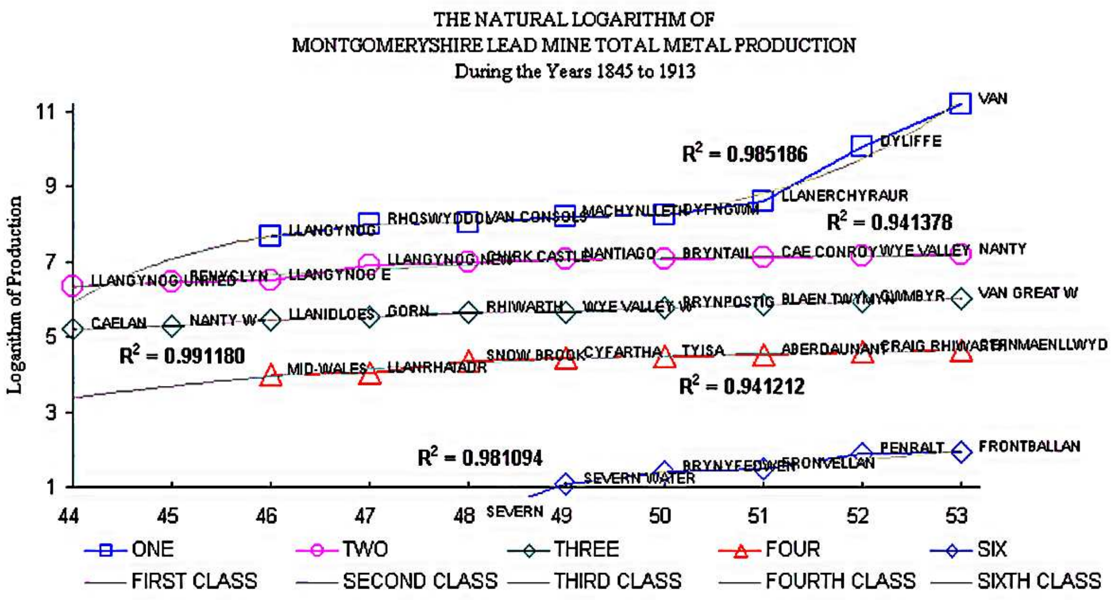
The reason why the CC figures on the following text-body plots disagree slightly with those in the text is that I used the EXCEL regression engine to compute the figures in these late internet diagrams, that appear below for reader convenience:-



THE NATURAL LOGARITHM OF
SOUTH WALES LEAD MINE TOTAL METAL PRODUCTION
During the Years 1845 to 1938
Logarithm of Production
y = 0.004x³ - 0.1177x² + 1.4165x - 0.2086
R² = 0.9974
y = 0.0855x³ - 4.0781x² + 64.957x - 342.56
R² = 1
FEDW
CWM ELAN
PENEGARREG
NANTYCAR
CARMARTEN
CASARA
LLANFAIR
NANTYGARW
CYSTANOG
TOWY VALE
LLANFYRNACH
NANTYMWYN
ABERGWESSIN
WHEALMORGAN
ST DAVIDS
BRYNAMBOR
DALRHIW
W NANTYMWYN
ONE
TWO
THREE
Poly. (ONE)
Poly. (TWO)



THE NATURAL LOGARITHM OF
SHROPSHIRE LEAD MINE TOTAL METAL PRODUCTION
During the Years 1845 to 1913
Logarithm of Production
y = 0.0099x³ - 0.8324x² + 23.533x - 214.81
R² = 0.9944
y = -0.0606x² + 4.1495x - 64.503
R² = 0.9898
y = 0.2975x - 5.5778
R² = 0.9882
OVENPIPE
TA WEST
RG EAST
ROUND HILL
GRIT
BOG
WHITE GRIT
STIPERSTONES
PENNERLEY
TANKERVILLE
RG
SNAILBEACH
LEEDS ROCK
RORRINGTON
HOPE VALLEY
STAPELEY
LADYWELL
BATHOLES
PERKINS
BATHOLES OLD
WHITE GRIT EAST
RHADLEY
RG SOUTH
TA NORTH
LORD HILL
BURGAM
STAVELEY
CRICKHEATH
POTTERS PIT
SALOP SOUTH
BOG SOUTH
ONE
TWO
THREE
Poly. (ONE)
Poly. (TWO)
Linear (THREE)



(2) NORMALISED SOURCE DATA

These worksheet printouts *.CSV list Mine Name, Normalised Rank, and Normalised Log Production for each province.

(3) POLYNOMIAL REGRESSION COEFFICIENTS

The Class Regression Coefficient Files *.COF are shown consolidated upon display reports for each province.

(4) TABULATED MODEL FITMENTS

These key reports *.MOD compare the quality of both the least-squares polynomial regressions concordance with the source data; and also Fourier-switched modellings with the limiting polynomials' approximation.

The Fourier models were tabulated for v = 64, 128, 256, 512 and 1024.

Summary metrics of correlation are also listed.

(5) MODEL AGREEMENT GRAPHS

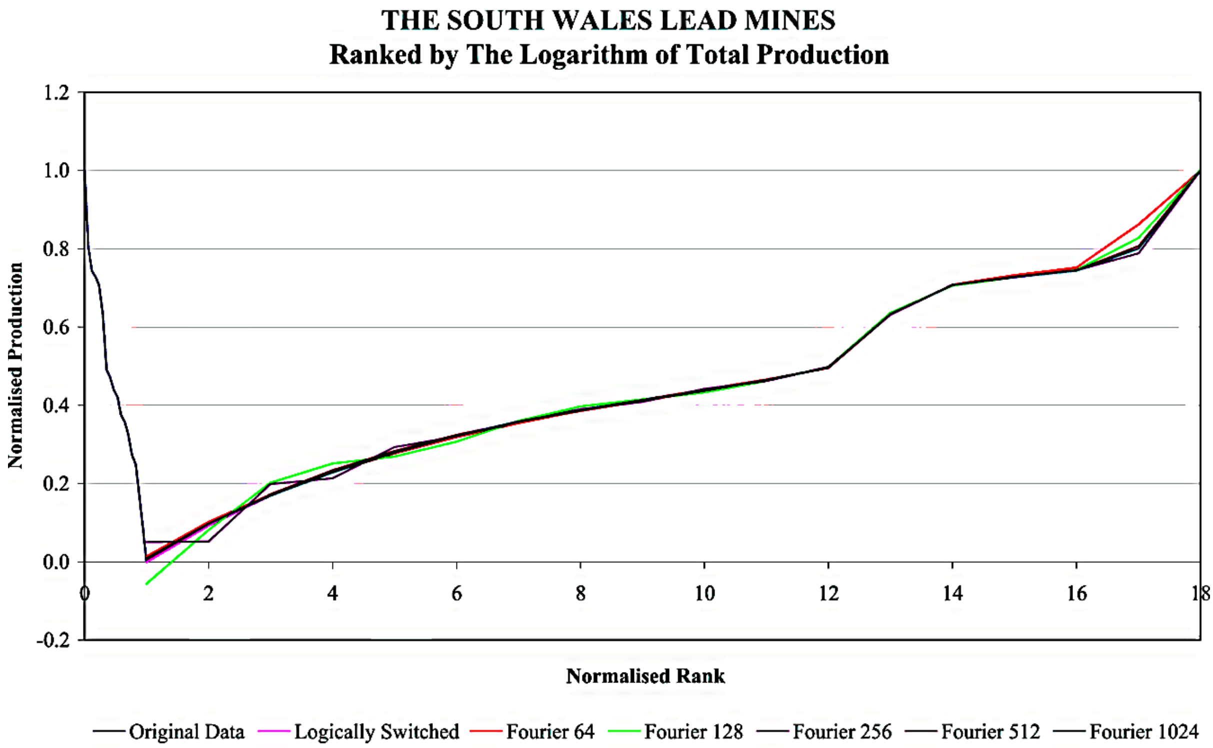
These graphs use the Fitment Series to illustrate the degree of fidelity with which each fitted model follows the curve of the normalised Original Data.

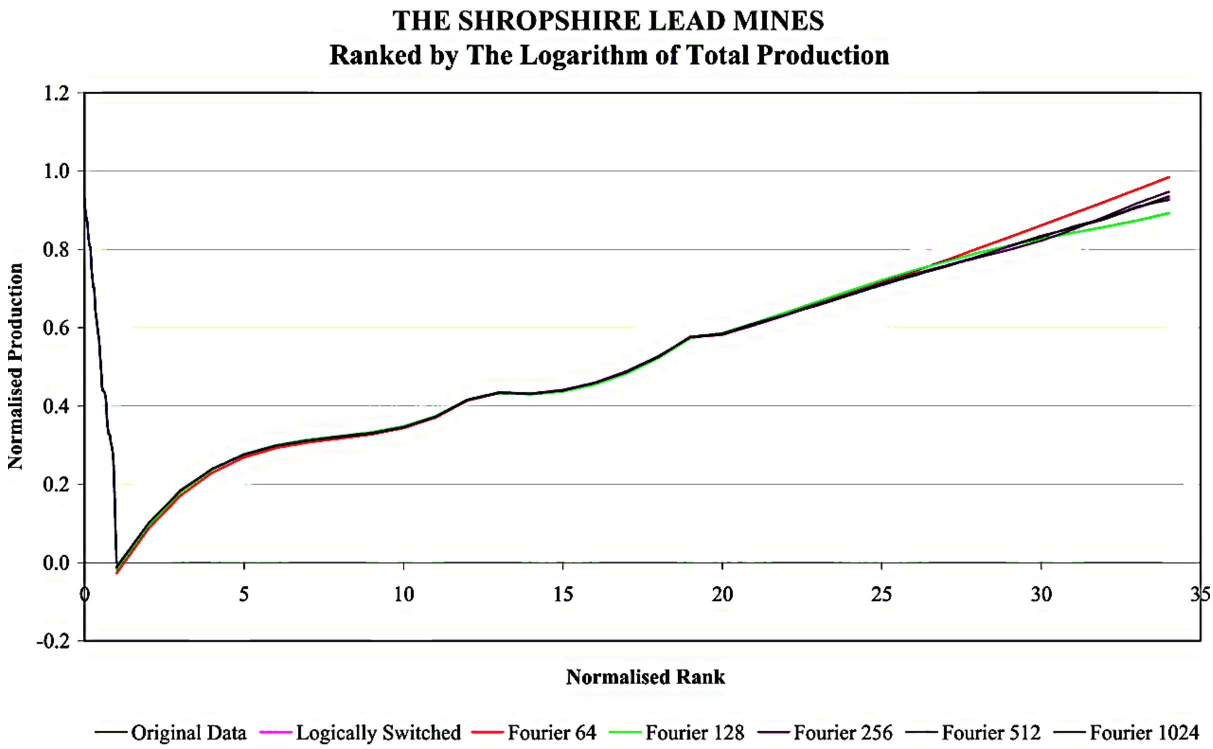
They make three striking demonstrations of the linear convergence of the logically switched and Fourier Series models set against the original data.

For Fourier modeling note how approximations to the data improve as the number of Fourier Series components rises from 64 to 1024.

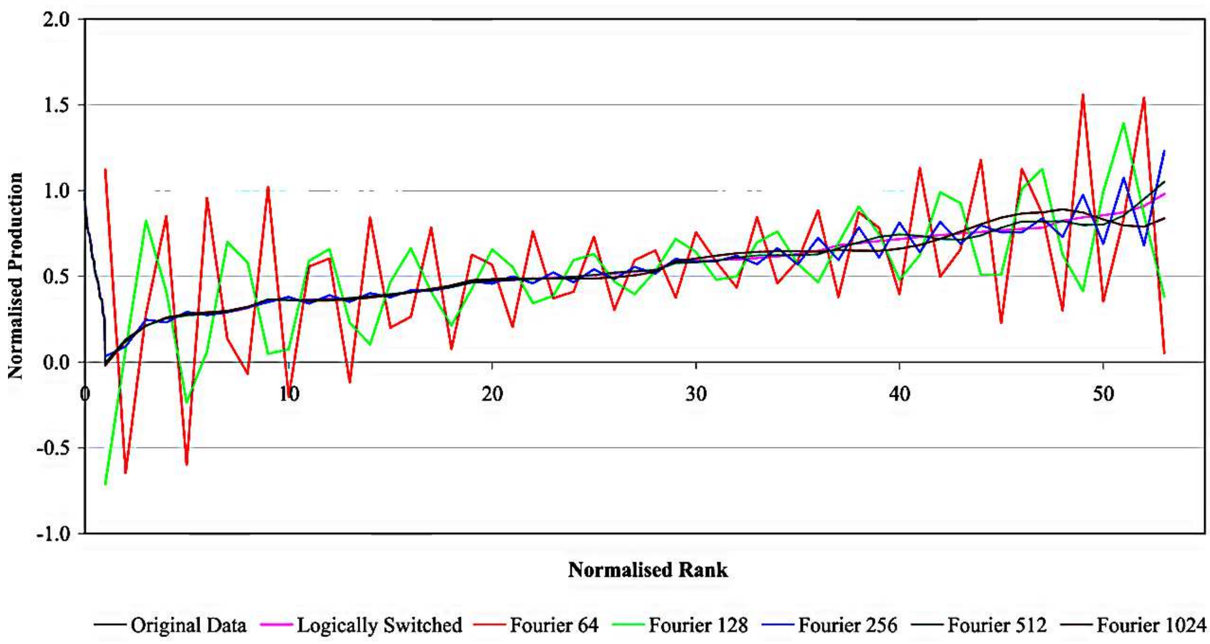
The Montgomeryshire convergence is perhaps the most impressive of the three.

These Model Agreement Graphs also appear in the Appendices and EXCEL renditions are given below for reader convenience:-





THE MONTGOMERYSHIRE LEAD MINES
Ranked by The Logarithm of Total Production



Program POLYWORK.BAS

Appendix Four records the source code of POLYWORK.BAS whose first option was invoked to generate for each Class the Least Squares Polynomial Regression coefficient files *.COF.

Program MINEMODL.BAS

Appendix Five contains the MINEMODL.BAS source code. This program utilised the coefficient files *.COF with Class and Fourier definitional keyboard data to generate the model fitments *.MOD.

A Review of Results

Southern Wales

The Statistical Province Graph illustrates the superb segregation of the lead mines into cubic Classes or Statistical Provinces. I admit that the West Nantymwyn trial could conform to any function but there is little reason why we should not allow it to qualify as a degenerate cubic!

Sensationally, the Second Class Cubic Fitment accounts for

99.99993% of the data variation whilst the First Class Fitment accounted for 99.87%. Readers should place scant credence upon the Third Class fit quality as it is entirely notional.

The Determination due to polynomial approximations is 99.94%.

The best Fourier fitments are v = 64 and v = 512 whilst the Model Agreement Graph shows good approximations for middle ranks though low-v fits wander at high and low extremes.

Shropshire

The Statistical Province graph shows another interesting three-class segregation, this time into a Cubic First, a Quadratic Second and a Linear Third Class. It is of course true that Quadratic and Linear polynomials are special cases of the Cubic.

Fitments are not significantly inferior to C_d = 99% for any class and the Determination due to polynomial approximations is 99.86%.

Fourier fitments improve with steady convergence until v = 1024 (and perhaps beyond?) when the residual error is measurable in millionths.

The Model Agreement Graph shows Fourier fitments are generally excellent for the First and Second Classes though low-v series diverge from the required line for the Third Class.

Montgomeryshire

The mine population of Montgomeryshire is large but is characterised by three or four large mines dominating a province in which fifty very middling workings show very similar low yields.

In any event, the simple elegance of the Southern Wales and Shropshire distributions does not apply and the Montgomeryshire field resolves into either six or nine indistinct Classes. The chosen study illustrates six Cubic Classes.

For Classes Two and Four C_d's are as low as 94% but the overall Determination due to polynomial approximations is, however, 99.9%.

Fourier fitments are of poor quality and even that for v = 1024 shows a residual error an order of magnitude away from the logically-switched series.

Study of the Model Agreement Graph discloses Fourier fitments radically more ill-behaved than those for either Southern Wales or Shropshire and characterised by unstable oscillations across the field, especially at the extremes.

I have neither a compiler nor a math co-processor on my old 20MHz

Tiko 386. Accordingly MINEMODL.BAS was elaborated as interpreted source. Both the Southern Wales and the Shropshire modeling took about half an hour to complete and I guessed the Montgomery exercise should take an hour and a quarter on that basis: It took nearly two hours to produce an inferior result, though for unknown reasons.

The Absolute Magnitude of Total Mine Productions

This study is of course couched in relative terms in order to facilitate inter-provincial comparisons and dimensionless computations.

If you wish to re-construct absolute data you will need figures for the Total Production of Metal or Concentrate for a defined mine in each province.

In practical terms, Table One specifies accumulated Production in Metric Tonnes for the leading mine:-

Province	Mine Name	Production	Substance
Southern Wales	Nantymwyn	34888.0571386	Concentrate
Shropshire	Snailbeach	91302.51954	Metal
Montgomeryshire	Van	67479.84826	Metal

A Formal Statement of The Cubic Class Rule

Equation One implies that:-

$$p_i = \frac{\mathrm{Log}_n(Y_{max}) - \mathrm{Log}_n(Y_i)}{\mathrm{Log}_n(Y_{max})} = 1 - \frac{\mathrm{Log}_n(Y_i)}{\mathrm{Log}_n(Y_{max})} \qquad \textbf{Eqn.20}$$

I have demonstrated from recorded data that:-

$$p_{i,k} = a_{0,k} + a_{1,k}\, r_i + a_{2,k}\, r_i^2 + a_{3,k}\, r_i^3 \qquad \textbf{Eqn.21}$$

For Class k let:-

$$a_k' = 1 - a_{0,k} \qquad \textbf{Eqn.22}$$

Then it follows that:-

$$\frac{Log_n(Y_i)}{Log_n(Y_{max})} = 1 - a_{0,k} - a_{1,k}\, r_i - a_{2,k}\, r_i^2 - a_{3,k}\, r_i^3$$
$$= a_k^{'} - (a_{1,k}\, r + a_{2,k}\, r^2 + a_{3,k}\, r^3) \qquad \textbf{Eqn.23}$$

or:-

$$\frac{Log_n(Y_i)}{Log_n(Y_{max})} = a_k^{'} - U_k \qquad \textbf{Eqn.24}$$

Discussion

The Number of Fourier Series Components, v

When I ran the Model Fitments I incremented v in steps of 2^i. As a doctoral student I learned that several relatively-efficient computational schemes (such as some FFT's) depended upon contrived or natural data populations of extent 2^i.

The computational scheme of MINEMODL.BAS does not require this restriction but I nevertheless considered it good policy to use 2^i increments in case it should facilitate extended operations.

The Mixed Character of Modelling

Equation Three defines the structure of the mixed Fourier Series - Polynomial Regressions models I have essayed.

By now many readers will be wondering why I did not discuss the logically-switched model alone. To be sure, the determinate action of the bistable multivibrator sets the limiting accuracy of the logically-switched model to the collective fidelity of the least-squares regressions. And the program has shown that Determination to exceed 99.8% in each province. I guess that many practical applications of The Cubic Class Rule could be made using only logically-switched modelling.

My motive for using the hybrid is that several orthogonal series of generated terms populate a multidimensional matrix which you may sum across many different vectors in order to create different expressions of Equation Three which may facilitate both theoretical and computational analysis of the underlying statistical situation.

Many other readers will take the contrary tack and wonder why I did not discountenance the regressions and develop pure Fourier Series models for the entirety of the ranked productions in each province.

Aside from my very uncertain skills at the calculus proving the usual handicap a more respectable reason for rejecting the pure model is that it would ignore The Cubic Class Rule and once again I needed to *contemplate* that discovery rather than *ignore* it.

Residual Errors in Modelling

Cursory examination of the five-member Residual Error series for each province suggests that such error may conform to a damped oscillatory rule of the kind typical for series errors.

The Selection of v

As aforementioned, Tabulated Model Fitments were computed for v = 64, 128, 256, 512 and 1024 or in general:-

$$v = 2^i \qquad \textbf{Eqn.25}$$

where i = 6, 7, 8, 9 and 10.
Note that for even i v is a perfect square defined by:-

$$v = 2^{2j} \qquad \textbf{Eqn.26}$$

where j = 3, 4 and 5
and the Square Root of v is given by:-

$$\sqrt{v} = 2^j \qquad \textbf{Eqn.27}$$

This is of interest because there is a numerical rule, valid in many instances, which asserts that the most accurate sum of a series is determined when it can be and is summed as n ranks of n separately-totalled data elements. This rule is of course predicated upon having a data strip whose population is a perfect square, a condition satisfied for alternate series in this study.
Such a summation may formally be defined by:-

$$S = \sum_{i=1}^{2^i} \sum_{j=1}^{2^j} x_{i,j} \qquad \textbf{Eqn.28}$$

Reference to the Tabulated Model Fitments confirms that Model Error Metrics for even i are not generally inferior to those of odd i. I therefore consider that savings may be made by computing even i v's only and accuracy enhanced by re-configuring the computational program to execute a two-phase summation according to Equation Twenty-Eight.

Further Research

Though I am a trained geologist I have no idea why this Cubic Class Rule should apply to the yield distributions of Victorian lead mines in and about the British Lower Palaeozoic.

An obvious diversification would explore the relation both for other kinds of mine and for the more typical British lead mines in the Dinantian blue limestone.

It is very clear that The Zipfian Paradigm of the linear relation of Size to Rank is violated by these objects.

I suspect that the observed distributions are as much the result of socio-environmental as of scientific factors and the published Victorian and Edwardian records already furnish some statistics, for instance regarding employment, prices and continuity of operation, which may supply multiple regression and other correlative studies with the data to resolve the relevant determinants of productivity.

Nevertheless, some of the data collated by The University of Exeter and published in computer-composited handbooks should be treated with circumspection. I used the occasional grid references of the Shropshire mines to map them by Class. It was clear from the fall of the plotted position that many co-ordinates were notional or highly-inaccurate. Allowing for that there was, however, clearly no correlation of Class with location within that province.

In the Dinantian region of Derbyshire a great many non-yielding mines were registered in the nineteenth-century for apparently fiscal motives[7] and except for two or three high-yielders only desultory lead mining took place during the period.

Notation

α	The Lower Bounding Rank of Class k
a	A Pulse Half-Width
a_0	The Fourier Series Constant Coefficient
a_i	A Polynomial Equation Coefficient
$a_{i,k}$	The Polynomial Equation Coefficient for the ith. Term of Class k
a_n	The First Fourier Series Coefficient
b	A Pulse Shift
b_n	The Second Fourier Series Coefficient
C_d	The Coefficient of Determination
ε	The Residual Error in Approximating Two Series
i	The First Series Counter

j	The Second Series Counter
k	The Number of Mine Classes in the Region
L	A Fourier Series Period
μ_T	The Mean of The Logarithms of Total Production
m	The Number of Mines in the Region
v	The Number of Fourier Series Components
n	The Number of Data
n	The Number of Polynomial Equation Terms
π	The Ludolphine Constant
p	A Pulse Height
p_i	The Normalised Logarithm of Total Production for Mine i
P	A Generalised Polynomial Function
$P(t)$	A Generalised Fourier Series Function
r_i	The Normalised Distributional Rank
R_α	The Rank of the Class Most Productive Mine
R_ω	The Rank of the Class Least Productive Mine
R_i	The Distributional Rank
S	A Series Sum
T	The Napierian Logarithm of Total Production
T_i	The Total Production of the ith. Mine
T_{max}	The Total Production of the Most Prolific Mine
T_v	A Fourier-Switched Log Production Computed from Polynomial Coefficients
T_{Pb}	The Napierian Logarithm of the Total Production of Metallic Lead
T_{PbS}	The Napierian Logarithm of the Total Production of Lead Ore Concentrate
T_{syn}	A (Logically-Switched) Log Production computed from Polynomial Coefficients
U_k	The Origin-Interceptive Cubic $a_{1,k}r^1+a_{2,k}r^2+a_{3,k}r^3$ for Class k
w	The Fourier Series Square-Pulse Width
x	An Arbitrary Datum
X	A Switching Function
Y	The Total Production in Metric Tonnes
$Y_{i,k}$	The Total Production of Mine i, a member of Class k
Y_{max}	The Total Production of the Leading Mine
ω	The Upper Bounding Rank of Class k
Ω	The Fourier Square-Wave Offset

z The Relative Range Constant

References

1 "Metal Mines of Southern Wales"
G W Hall 1993
Griffin Publications of Kington
No ISBN

2 "The Mines of Shropshire and Montgomeryshire with Cheshire and Staffordshire"
R Burt, P Waite and R Burnley 1990
University of Exeter Press
ISBN 0-85989-343-X

3 "Introduction to Computational Mathematics"
M Beilby and G McCauley 1986
Scottish Academic Press of Edinburgh
ISBN 0-7073-0495-4

4 "The FFT: Fundamentals and Concepts"
Robert W Ramirez 1985
Prentice-Hall Incorporated
ISBN 0-13-314386-4

5 "VNR Concise Encyclopedia of Mathematics"
Eds: W Gellert, H Kustner, M Hellwich and H Kastner 1975
Van Nostrand Reinhold and Company of New York
ISBN 0-442-22646-2

6 "Engineering Mathematics Handbook"
Second Edition 1979
Jan J Tuma
McGraw-Hill
ISBN 0-07-065429-8

7 "The Derbyshire Mineral Statistics"
R Burt, P Waite, M Atkinson and R Burnley 1981

The University of Exeter Press
ISBN 0-9507624-0-7

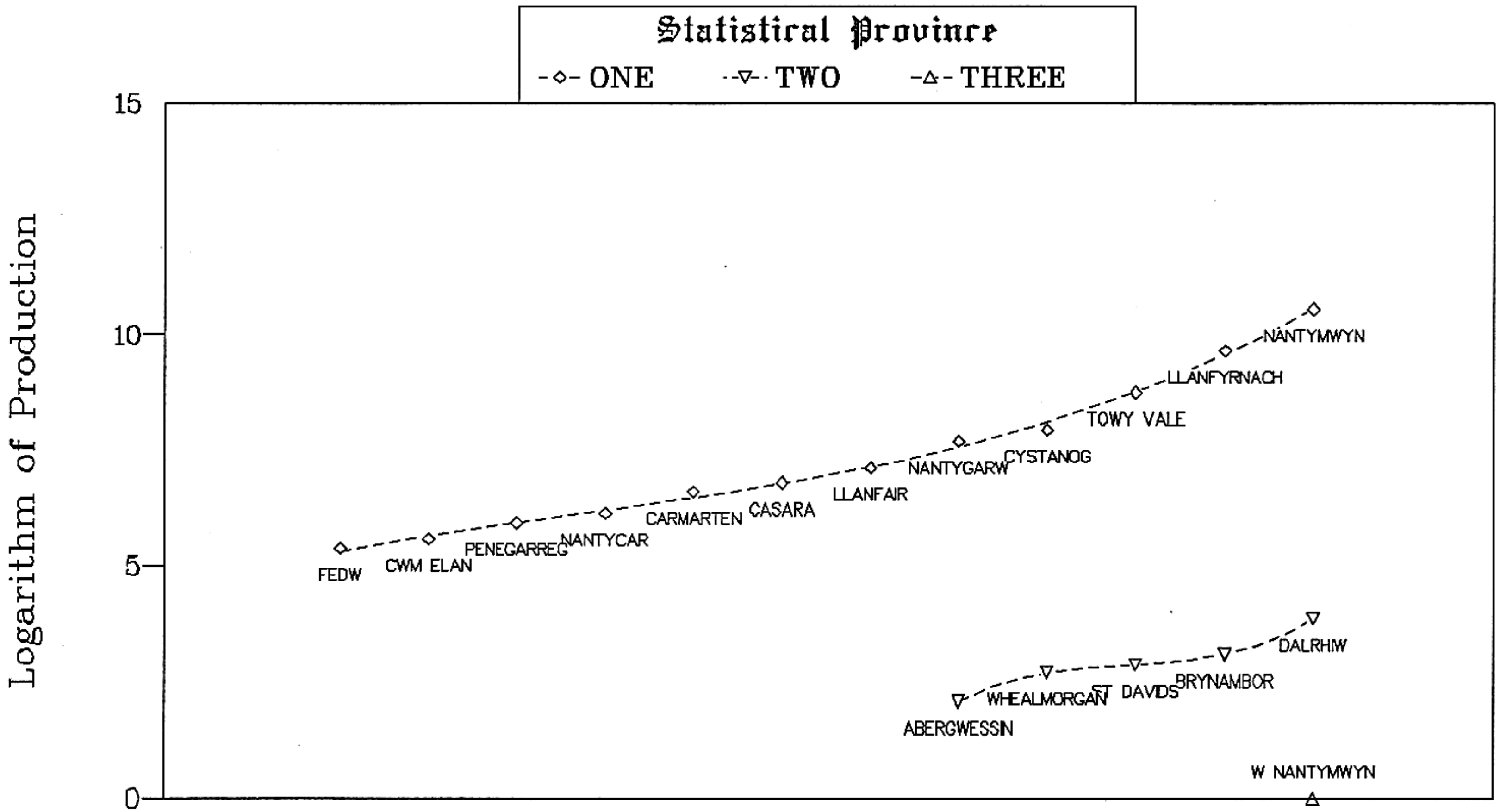
APPENDIX ONE

Province Report for Southern Wales

THE NATURAL LOGARITHM OF THE TOTAL LEAD CONCENTRATE PRODUCTION FOR THE LEAD MINES OF SOUTH WALES

During the Years 1845 to 1938

A NON-ZIPFIAN INTERPRETATION



Drafted by James R Warren

SWALES.CSV

THE SOUTH WALES LEAD MINES

Ranked by The Logarithm of Total Production

Normalised Rank

Normalised Production

18

NANTYMWYN	1.0000000	.0000000
LLANFYRNACH	.9411765	.0851440
TOWY VALE	.8823529	.1710357
CYSTANOG	.8235294	.2480279
NANTYGARW	.7647059	.2720017
LLANFAIR	.7058824	.3253142
CASARA	.6470588	.3567280
CARMARTEN	.5882353	.3749819
NANTYCAR	.5294118	.4199669
PENEGARREG	.4705882	.4389746
CWM ELAN	.4117647	.4720195
FEDW	.3529412	.4908477
DALRHIW	.2941176	.6333156
BRYNAMBOR	.2352941	.7072130
ST DAVIDS	.1764706	.7275456
WHEALMORGAN	.1176471	.7434905
ABERGWESSIN	.0588235	.8030330
W NANTYMWYN	.0000000	1.0000000

SWALE1.CSV
FIRST CLASS SOUTH WALES LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production

4	
0	.8963591
1	-1.9213202
2	2.8915149
3	-1.8677488

COEFFICIENT OF DETERMINATION = .9973554
COEFFICIENT OF CORRELATION = .9986768
STANDARD ERROR OF THE ESTIMATE = .0093637

SWALE2.CSV
SECOND CLASS SOUTH WALES LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production

4	
0	.9551120
1	-3.6460390
2	20.3767520
3	-39.7799635

COEFFICIENT OF DETERMINATION = .9999987
COEFFICIENT OF CORRELATION = .9999993
STANDARD ERROR OF THE ESTIMATE = .0001425

SWALE3.CSV
THIRD CLASS SOUTH WALES LEAD MINE
Cubic Segment with a Single Member
Normalised Rank
Normalised Production

4	
0	1.0000000
1	.0000000
2	.0000000
3	.0000000

COEFFICIENT OF DETERMINATION = 1.0000000
COEFFICIENT OF CORRELATION = 1.0000000
STANDARD ERROR OF THE ESTIMATE = .0000000

****
A WORKSHEET TO PRESENT FOURIER-POLYNOMIAL
MODEL FITMENTS TO LEAD MINE
LOG PRODUCTION VERSUS RANK DATA
****

JAMES R WARREN
SWALES.MOD
07-11-1995

THE SOUTH WALES LEAD MINES
Ranked by The Logarithm of Total Production

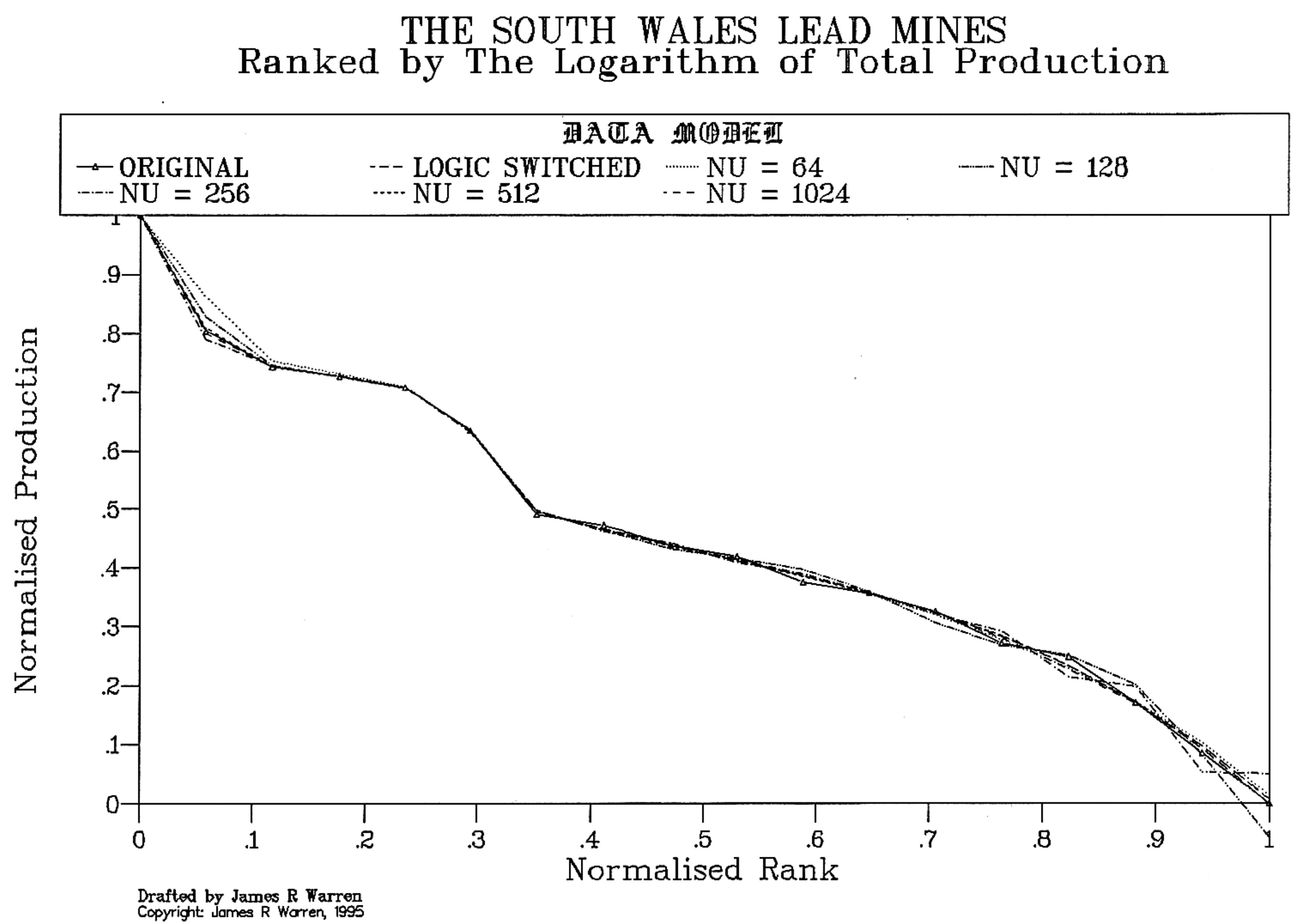
NUMBER OF CLASSES		3
ORIGINAL DATA	SWALES.CSV	
NUMBER OF MINES		18

POLYNOMIAL SEGMENTS

REGRESSION FILENAME	LOWER RANK	UPPER RANK
SWALE1.COF	1	12
SWALE2.COF	13	17
SWALE3.COF	18	18

THE TABLE OF MODEL APPROXIMATIONS

	MINE NAME	ORIGINAL DATA	LOGICALLY SWITCHED MODEL	FOURIER SWITCHED				
NUMBER OF TERMS =				64	128	256	512	1024
	NANTYMWYN	.000000	-.001195	.014208	-.056954	.050809	.005234	.008595
	LLANFYRNACH	.085144	.092245	.101863	.081969	.051843	.097183	.097531
	TOWY VALE	.171036	.169200	.171673	.201690	.197952	.172497	.168541
	CYSTANOG	.248028	.231951	.228804	.250510	.213692	.233719	.227863
	NANTYGARW	.272002	.282778	.276969	.268953	.292438	.283295	.278874
	LLANFAIR	.325314	.323963	.318472	.307292	.320692	.323579	.322409
	CASARA	.356728	.357787	.354563	.360045	.356862	.356853	.358526
	CARMARTEN	.374982	.386530	.386013	.397218	.389729	.385354	.388164
	NANTYCAR	.419967	.412474	.413761	.415162	.408501	.411298	.413659
	PENEGARREG	.438975	.437900	.439457	.433109	.441634	.436904	.438170
	CWM ELAN	.472019	.465088	.465660	.462057	.462150	.464406	.464775
	FEDW	.490848	.496319	.495189	.497416	.498032	.496171	.496030
	DALRHIW	.633316	.633333	.632569	.635655	.631656	.633229	.633099
	BRYNAMBOR	.707213	.707145	.708054	.706704	.708266	.707479	.707170
	ST DAVIDS	.727546	.727648	.731723	.726624	.726620	.728309	.727805
	WHEALMORGAN	.743490	.743422	.751782	.745678	.744883	.744351	.743596
	ABERGWESSIN	.803033	.803050	.862124	.827193	.788593	.807406	.800022
	W NANTYMWYN	1.000000	1.000000	.999568	1.001427	.999539	1.000032	1.000154
DETERMINATION COEF.		1.000000	.999419					
MODEL ERROR METRICS		.000000	.000140	.000944	.001166	.001349	.000161	.000200


THE SOUTH WALES LEAD MINES
Ranked by The Logarithm of Total Production
DATA MODEL
ORIGINAL
LOGIC SWITCHED
NU = 64
NU = 128
NU = 256
NU = 512
NU = 1024
Normalised Production
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1
Normalised Rank
Drafted by James R Warren
Copyright: James R Warren, 1995

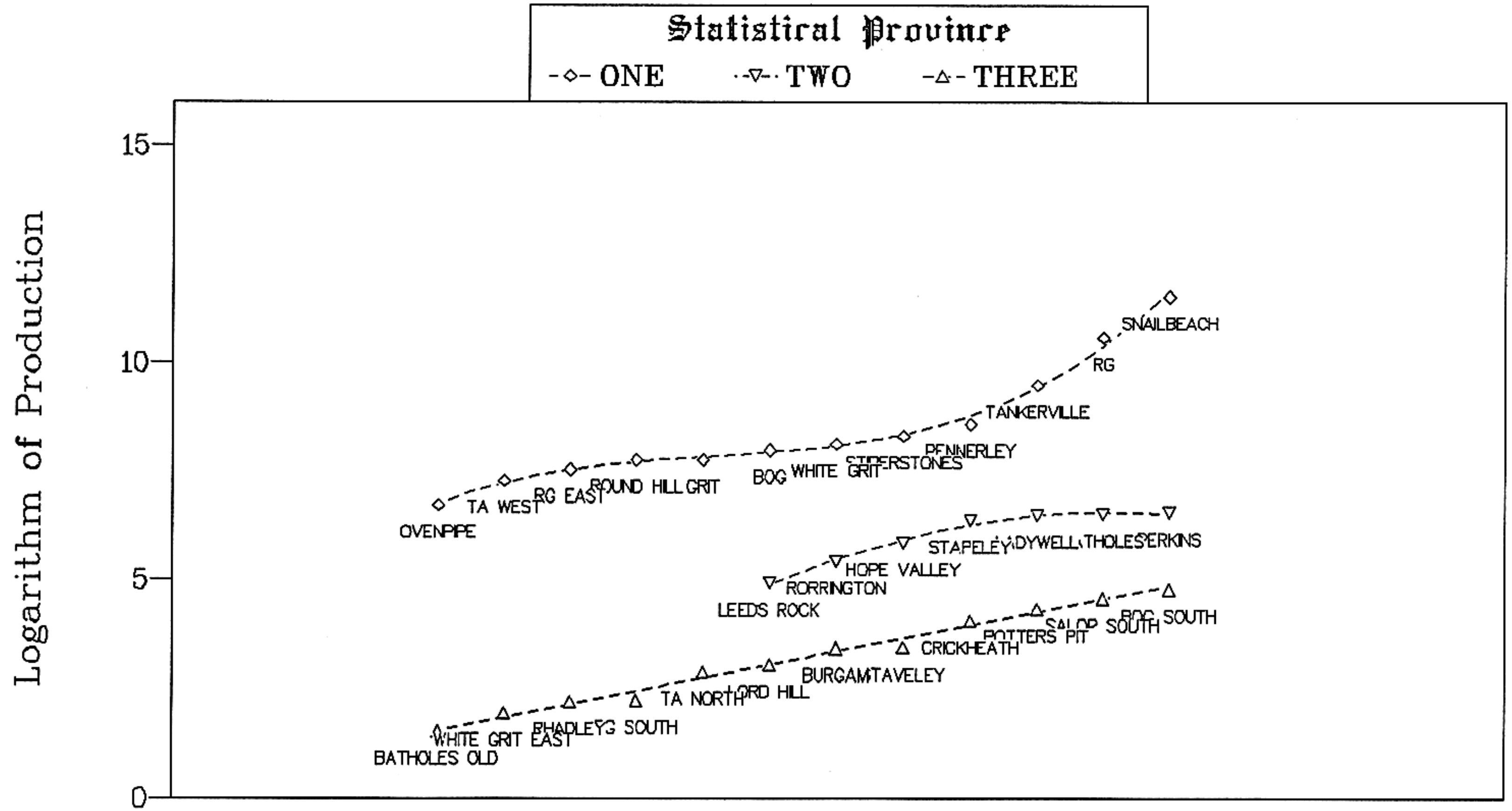
APPENDIX TWO

Province Report for Shropshire

THE NATURAL LOGARITHM OF THE TOTAL METALLIC LEAD PRODUCTION FOR THE SHROPSHIRE LEAD MINES

During the Years 1845 to 1913

A NON-ZIPFIAN INTERPRETATION



Drafted by James R Warren

SHROP.CSV

THE SHROPSHIRE LEAD MINES

Ranked by The Logarithm of Total Production

Normalised Rank

Normalised Production

34

SNAILBEACH	1.0000000	.0000000
ROMAN GRAVELS	.9696970	.0812912
TANKERVILLE	.9393939	.1772952
PENNERLEY	.9090909	.2560921
STIPERSTONES	.8787879	.2790697
WHITE GRIT	.8484848	.2953760
BOG	.8181818	.3078523
GRIT	.7878788	.3273448
ROUND HILL	.7575758	.3274422
ROMAN GRAVELS EAST	.7272727	.3468171
TANKERVILLE WEST	.6969697	.3686854
OVENPIPE	.6666667	.4186914
PERKINS BEACH	.6363636	.4327732
BATHOLES	.6060606	.4371849
LADYWELL	.5757576	.4390224
STAPELEY	.5454545	.4482985
HOPE VALLEY	.5151515	.4930467
RORRINGTON	.4848485	.5297870
LEEDS ROCK HOUSE	.4545455	.5734661
BOG SOUTH	.4242424	.5862231
SALOP SOUTH	.3939394	.6046756
POTTERS PIT	.3636364	.6256609
CRICKHEATH	.3333333	.6478107
STAVELEY	.3030303	.7018938
BURGAM	.2727273	.7034478
LORD HILL	.2424242	.7357035
TANKERVILLE NORTH	.2121212	.7495690
ROMAN GRAVELS SOUTH	.1818182	.8064113
RHADLEY	.1515152	.8092578
WHITE GRIT EAST	.1212121	.8310746
BATHOLES OLD	.0909091	.8675223
ROMAN GRAVELS WEST	.0606061	.8796551
SNAILBEACH NEW WEST	.0303030	.8873666
CALLOW HILL	.0000000	.9296027

```
 SHROP1.CSV
FIRST CLASS SHROPSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                           4
                                           0  15.8442245
                                           1 -58.2147732
                                           2  73.1043996
                                           3 -30.7449865
COEFFICIENT OF DETERMINATION =                  .9943634
COEFFICIENT OF CORRELATION =                    .9971777
STANDARD ERROR OF THE ESTIMATE =                .0107347


SHROP2.CSV
SECOND CLASS SHROPSHIRE LEAD MINES
Quadratic Segment
Normalised Rank
Normalised Production
                                           3
                                           0   2.5872400
                                           1  -7.0276515
                                           2   5.7275660
COEFFICIENT OF DETERMINATION =                  .9898382
COEFFICIENT OF CORRELATION =                    .9949061
STANDARD ERROR OF THE ESTIMATE =                .0067860


SHROP3.CSV
THIRD CLASS SHROPSHIRE LEAD MINES
Linear Segment
Normalised Rank
Normalised Production
                                           2
                                           0    .9306872
                                           1   -.8153935
COEFFICIENT OF DETERMINATION =                  .9901369
COEFFICIENT OF CORRELATION =                    .9950562
STANDARD ERROR OF THE ESTIMATE =                .0114451
```

****

A WORKSHEET TO PRESENT FOURIER-POLYNOMIAL
MODEL FITMENTS TO LEAD MINE
LOG PRODUCTION VERSUS RANK DATA

****

JAMES R WARREN
SHROP.MOD
07-11-1995

THE SHROPSHIRE LEAD MINES
Ranked by The Logarithm of Total Production

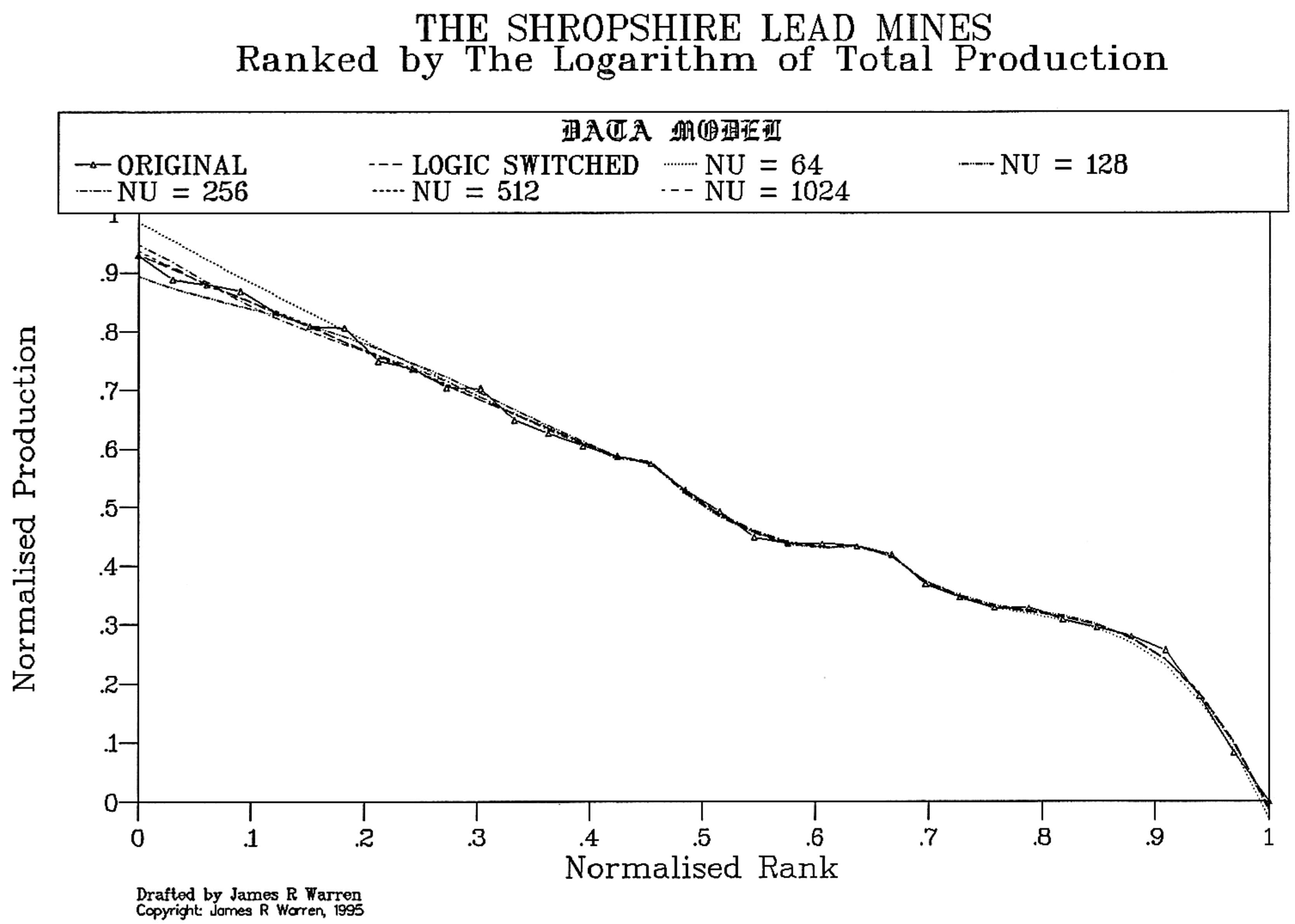
NUMBER OF CLASSES		3
ORIGINAL DATA	SHROP.CSV	
NUMBER OF MINES		34

POLYNOMIAL SEGMENTS

REGRESSION FILENAME	LOWER RANK	UPPER RANK	
SHROP1.COF		1	12
SHROP2.COF		13	19
SHROP3.COF		20	34

THE TABLE OF MODEL APPROXIMATIONS

	MINE NAME	ORIGINAL DATA	LOGICALLY SWITCHED MODEL	FOURIER SWITCHED				
NUMBER OF TERMS =				64	128	256	512	1024
	SNAILBEACH	.000000	-.011136	-.027475	-.016999	-.010486	-.012864	-.010905
	ROMAN GRAVELS	.081291	.100666	.086953	.096058	.100941	.100525	.100486
	TANKERVILLE	.177295	.182466	.171043	.179421	.182125	.183641	.182589
	PENNERLEY	.256092	.239399	.229963	.237920	.238712	.239688	.239334
	STIPERSTONES	.279070	.276597	.268866	.276484	.276062	.275802	.276610
	WHITE GRIT	.295376	.299193	.292902	.300126	.299177	.298847	.299228
	BOG	.307852	.312321	.307218	.313931	.312844	.312858	.312246
	GRIT	.327345	.321114	.316963	.323031	.321864	.321490	.321225
	ROUND HILL	.327442	.330704	.327287	.332586	.331235	.330356	.330563
	ROMAN GRAVELS EAST	.346817	.346226	.343351	.347775	.346212	.345825	.346394
	TANKERVILLE WEST	.368685	.372811	.370339	.373801	.372247	.372995	.372625
	OVENPIPE	.418691	.415594	.413717	.415954	.414912	.415926	.415763
	PERKINS BEACH	.432773	.434526	.432974	.432794	.433462	.434423	.434099
	BATHOLES	.437185	.431846	.429776	.429766	.431806	.431215	.432198
	LADYWELL	.439022	.439686	.437597	.436830	.440655	.439526	.439319
	STAPELEY	.448298	.458045	.456044	.454600	.459736	.458669	.458436
	HOPE VALLEY	.493047	.486922	.485181	.483257	.488618	.487316	.486508
	RORRINGTON	.529787	.526319	.525216	.522900	.527110	.525695	.526757
	LEEDS ROCK HOUSE	.573466	.576234	.577409	.573133	.575464	.575845	.575721
	BOG SOUTH	.586223	.584763	.581223	.585792	.582661	.585035	.585036
	SALOP SOUTH	.604676	.609472	.608462	.612175	.606662	.610714	.609184
	POTTERS PIT	.625661	.634181	.634907	.639596	.632280	.634051	.634403
	CRICKHEATH	.647811	.658889	.661565	.667226	.659382	.657368	.658777
	STAVELEY	.701894	.683598	.688610	.694562	.687026	.683315	.683558
	BURGAM	.703448	.708307	.716115	.721068	.713678	.710099	.708547
	LORD HILL	.735703	.733016	.744118	.746207	.737906	.733912	.732529
	TANKERVILLE NORTH	.749569	.757725	.772637	.769513	.759239	.755784	.758510
	ROMAN GRAVELS SOUTH	.806411	.782434	.801669	.790679	.778728	.780724	.781303
	RHADLEY	.809258	.807143	.831197	.809645	.798783	.809017	.808666
	WHITE GRIT EAST	.831075	.831852	.861182	.826667	.822184	.834559	.829897
	BATHOLES OLD	.867522	.856561	.891563	.842356	.850601	.855080	.858978
	ROMAN GRAVELS WEST	.879655	.881269	.922259	.857678	.883345	.877430	.878370
	SNAILBEACH NEW WEST	.887367	.905978	.953162	.873896	.917111	.906614	.909364
	CALLOW HILL	.929603	.930687	.984135	.892467	.947137	.935708	.926833
DETERMINATION COEF.		1.000000	.998588					
MODEL ERROR METRICS		.000000	.000246	.001256	.000497	.000368	.000266	.000260


THE SHROPSHIRE LEAD MINES
Ranked by The Logarithm of Total Production
DATA MODEL
ORIGINAL
LOGIC SWITCHED
NU = 64
NU = 128
NU = 256
NU = 512
NU = 1024
Normalised Production
1
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1
Normalised Rank
Drafted by James R Warren
Copyright: James R Warren, 1995

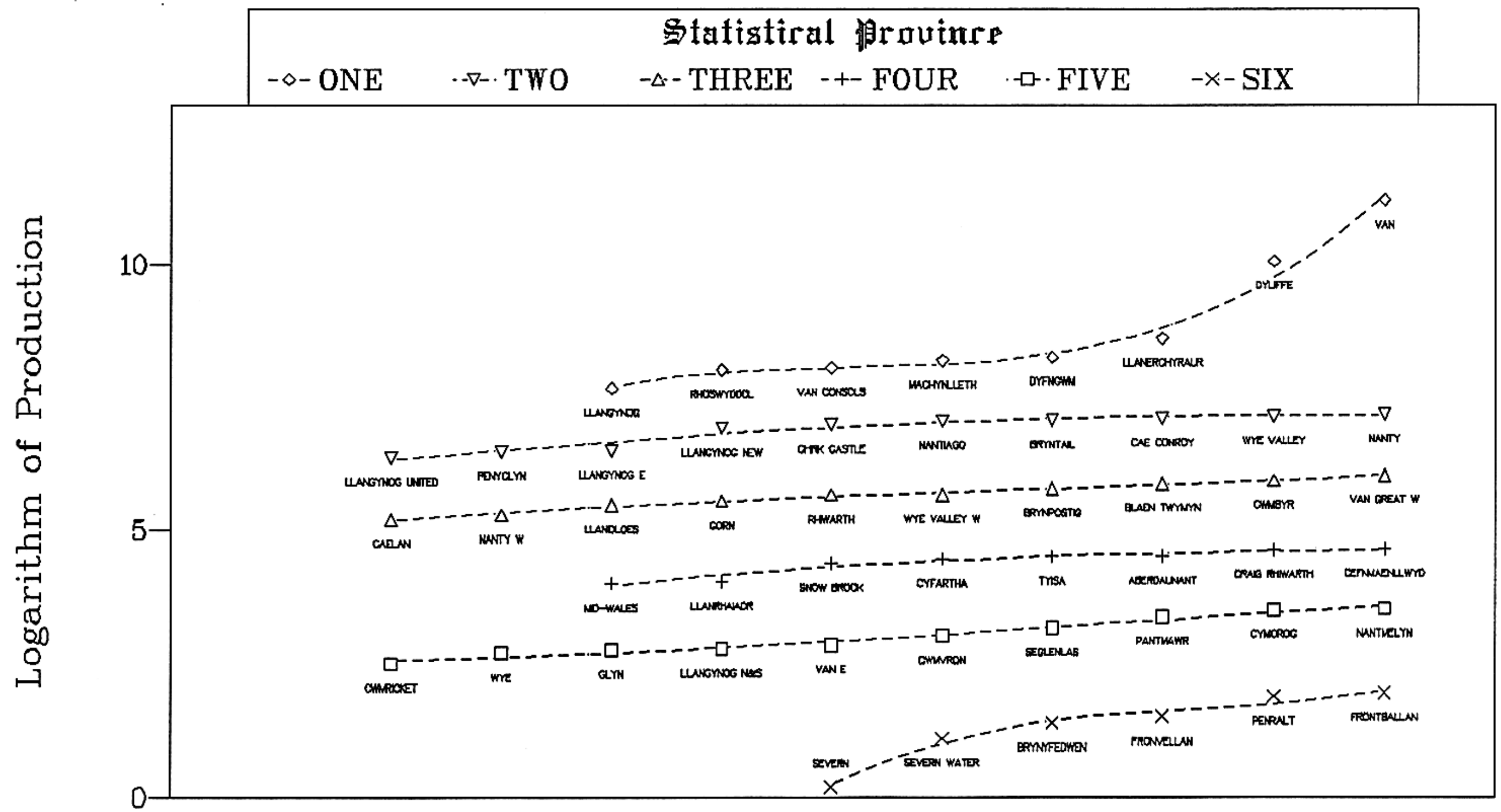
APPENDIX THREE

Province Report for Montgomeryshire

THE NATURAL LOGARITHM OF THE TOTAL METALLIC LEAD PRODUCTION FOR THE MONTGOMERYSHIRE LEAD MINES

During the Years 1845 to 1913

A NON-ZIPFIAN INTERPRETATION



Drafted by James R Warren

MONT.CSV
THE MONTGOMERYSHIRE LEAD MINES
Ranked by The Logarithm of Total Production
Normalised Rank
Normalised Production

	53		
VAN		1.0000000	.0000000
DYLIFFE		.9807692	.1026285
LLANERCHYRAUR		.9615385	.2323050
DYFNGWM		.9423077	.2640170
MACHYNLLETH		.9230769	.2697373
VAN CONSOLS		.9038462	.2821909
RHOSWYDDOL		.8846154	.2855862
LLANGYNOG		.8653846	.3169295
NANTY		.8461538	.3603565
WYE VALLEY		.8269231	.3639355
CAE CONROY		.8076923	.3671545
BRYNTAIL		.7884615	.3706166
NANTIAGO		.7692308	.3709772
CHIRK CASTLE		.7500000	.3776816
LLANGYNOG NEW		.7307692	.3840458
LLANGYNOG EAST		.7115385	.4211464
PENYCLYN		.6923077	.4229186
LLANGYNOG UNITED		.6730769	.4329949
VAN GREAT WEST		.6538462	.4634831
CWMBYR		.6346154	.4713034
BLAEN TWYMYN		.6153846	.4770017
BRYNPOSTIG		.5961538	.4855012
WYE VALLEY WEST		.5769231	.4957970
RHIWARTH		.5576923	.4963600
GORN		.5384615	.5065198
LLANIDLOES		.5192308	.5126329
NANTY WEST		.5000000	.5294537
CAELAN		.4807692	.5368477
CEFNMAENLLWYD		.4615385	.5868118
CRAIG RHIWARTH		.4423077	.5885599
ABERDAUNANT		.4230769	.5986421
TYISA		.4038462	.5988399
CYFARTHA		.3846154	.6035171
SNOW BROOK		.3653846	.6105747
LLANRHAIADR		.3461538	.6411381
MID-WALES		.3269231	.6443803
NANTMELYN		.3076923	.6856234
CYMOROG		.2884615	.6882848
PANTMAWR		.2692308	.6998041
SEGLENLAS		.2500000	.7185520
CWMVRON		.2307692	.7320420
VAN EAST		.2115385	.7474178
LLANGYNOG NORTH AND SOUTH		.1923077	.7528225
GLYN		.1730769	.7545099
WYE		.1538462	.7603772
CWMRICKET		.1346154	.7777296
BWLCH CREOLAN		.1153846	.7854198
FRONTBALLAN		.0961538	.8265212
PENRALT		.0769231	.8331280
FRONVELLAN		.0576923	.8659108
BRYNYFEDWEN		.0384615	.8764112
SEVERN WATER		.0192308	.9020582
SEVERN		.0000000	.9837460

```
MONT1.CSV
FIRST CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0    248.6703917
                                        1   -821.2882844
                                        2    905.4359860
                                        3   -332.8276986
COEFFICIENT OF DETERMINATION =                 .9851862
COEFFICIENT OF CORRELATION =                   .9925655
STANDARD ERROR OF THE ESTIMATE =               .0176690
```

```
MONT2.CSV
SECOND CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0       .4906415
                                        1      2.7079542
                                        2     -7.1088455
                                        3      4.4089503
COEFFICIENT OF DETERMINATION =                 .9413778
COEFFICIENT OF CORRELATION =                   .9702463
STANDARD ERROR OF THE ESTIMATE =               .0081676
```

```
MONT3.CSV
THIRD CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0      2.7311909
                                        1    -10.7035444
                                        2     17.5258964
                                        3     -9.8814080
COEFFICIENT OF DETERMINATION =                 .9911795
COEFFICIENT OF CORRELATION =                   .9955800
STANDARD ERROR OF THE ESTIMATE =               .0027890
```

```
MONT4.CSV
FOURTH CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0      2.2016088
                                        1    -10.0736595
                                        2     21.2169946
                                        3    -15.1006389
COEFFICIENT OF DETERMINATION =                 .9412121
COEFFICIENT OF CORRELATION =                   .9701608
STANDARD ERROR OF THE ESTIMATE =               .0071081
```

```
MONT5.CSV
FIFTH CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0       .8156357
                                        1      -.1915763
                                        2      -.7505970
                                        3      -.1817925
COEFFICIENT OF DETERMINATION =                 .9787581
COEFFICIENT OF CORRELATION =                   .9893220
STANDARD ERROR OF THE ESTIMATE =               .0060281
```

```
MONT6.CSV
SIXTH CLASS MONTGOMERYSHIRE LEAD MINES
Cubic Segment
Normalised Rank
Normalised Production
                                        4
                                        0       .9808819
                                        1     -4.8268752
                                        2     65.8730648
                                        3   -340.5972992
COEFFICIENT OF DETERMINATION =                 .9810942
COEFFICIENT OF CORRELATION =                   .9905020
STANDARD ERROR OF THE ESTIMATE =               .0124919
```

****
A WORKSHEET TO PRESENT FOURIER-POLYNOMIAL
MODEL FITMENTS TO LEAD MINE
LOG PRODUCTION VERSUS RANK DATA
****

JAMES R WARREN
MONT.MOD
07-11-1995

THE MONTGOMERYSHIRE LEAD MINES
Ranked by The Logarithm of Total Production

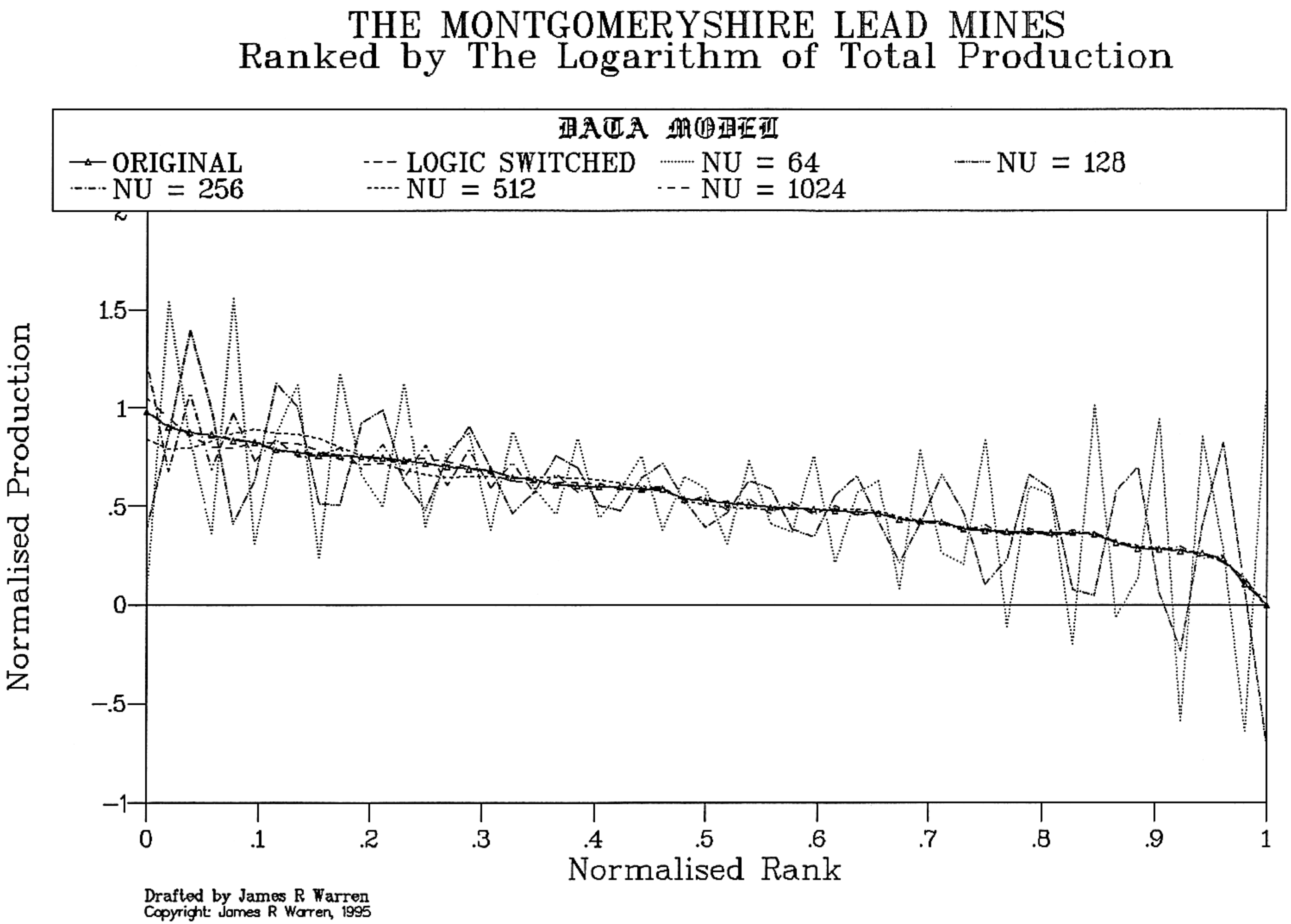
NUMBER OF CLASSES		6
ORIGINAL DATA	MONT.CSV	
NUMBER OF MINES		53

POLYNOMIAL SEGMENTS

REGRESSION FILENAME	LOWER RANK	UPPER RANK
MONT1.COF	1	8
MONT2.COF	9	18
MONT3.COF	19	28
MONT4.COF	29	36
MONT5.COF	37	47
MONT6.COF	48	53

THE TABLE OF MODEL APPROXIMATIONS

	MINE NAME	ORIGINAL DATA	LOGICALLY SWITCHED MODEL	FOURIER SWITCHED				
NUMBER OF TERMS =				64	128	256	512	1024
	VAN	.000000	-.009605	1.121817	-.712757	.033719	-.019791	-.004281
	DYLIFFE	.102628	.129493	-.646181	.070106	.089960	.121456	.132810
	LLANERCHYRAUR	.232305	.213972	.280489	.823385	.247448	.210020	.213258
	DYFNGWM	.264017	.258035	.850882	.412898	.232177	.258846	.254109
	MACHYNLLETH	.269737	.275882	-.596257	-.234508	.293244	.280887	.271622
	VAN CONSOLS	.282191	.281717	.955116	.059406	.273095	.289351	.280013
	RHOSWYDDOL	.285586	.289742	.135699	.700827	.289940	.297877	.291552
	LLANGYNOG	.316929	.314159	-.068315	.579482	.321626	.320630	.317929
	NANTY	.360356	.363280	1.019838	.047896	.350356	.366238	.366404
	WYE VALLEY	.363935	.361918	-.204987	.075633	.380310	.360640	.362016
	CAE CONROY	.367155	.363388	.556586	.588775	.341791	.357989	.360888
	BRYNTAIL	.370617	.367501	.602873	.657781	.390463	.359163	.364607
	NANTIAGO	.370977	.374071	-.116340	.229824	.351556	.364765	.373014
	CHIRK CASTLE	.377682	.382907	.842544	.103540	.403258	.374979	.384136
	LLANGYNOG NEW	.384046	.393824	.201362	.466589	.377183	.389463	.395824
	LLANGYNOG EAST	.421146	.406631	.263326	.663218	.418253	.407353	.407497
	PENYCLYN	.422919	.421142	.783388	.409483	.415564	.427369	.420405
	LLANGYNOG UNITED	.432995	.437167	.076042	.212666	.435925	.448011	.436094
	VAN GREAT WEST	.463483	.463160	.627065	.425946	.470799	.476719	.463530
	CWMBYR	.471303	.471359	.567448	.657205	.456657	.484371	.472779
	BLAEN TWYMYN	.477002	.478607	.205849	.555812	.499688	.487894	.479291
	BRYNPOSTIG	.485501	.485324	.760789	.343553	.458793	.488074	.483596
	WYE VALLEY WEST	.495797	.491933	.371565	.386272	.522631	.486498	.488540
	RHIWARTH	.496360	.498855	.410573	.593281	.465604	.485306	.497010
	GORN	.506520	.506512	.730286	.629903	.540418	.486859	.509074
	LLANIDLOES	.512633	.515325	.304213	.470246	.482887	.493318	.521443
	NANTY WEST	.529454	.525717	.593288	.395237	.554384	.506195	.530418
	CAELAN	.536848	.538108	.651555	.531049	.516053	.525823	.536053
	CEFNMAENLLWYD	.586812	.587191	.376429	.717169	.602591	.587485	.578299
	CRAIG RHIWARTH	.588560	.590083	.757290	.642067	.585161	.603474	.580518
	ABERDAUNANT	.598642	.593847	.584333	.478759	.586778	.619648	.592391
	TYISA	.598840	.599128	.435681	.499655	.619470	.633361	.609141
	CYFARTHA	.603517	.606571	.844629	.698025	.572173	.642655	.621543
	SNOW BROOK	.610575	.616819	.459241	.760192	.665437	.646736	.624186
	LLANRHAIADR	.641138	.630516	.591405	.572697	.568215	.646441	.621962
	MID-WALES	.644380	.648309	.885150	.465246	.722998	.644197	.628350
	NANTMELYN	.685623	.680331	.380551	.693321	.595743	.655018	.664552
	CYMOROG	.688285	.693553	.872858	.907574	.785500	.648888	.697033
	PANTMAWR	.699804	.706103	.781990	.746876	.610606	.649488	.729323
	SEGLENLAS	.718552	.717989	.396456	.480174	.812613	.660525	.743896
	CWMVRON	.732042	.729219	1.130463	.624827	.640434	.683773	.735369
	VAN EAST	.747418	.739801	.498129	.990304	.817397	.718344	.717435
	LLANGYNOG NORTH AND SOUTH	.752822	.749742	.657505	.926589	.688852	.760517	.713860
	GLYN	.754510	.759051	1.178673	.509465	.797807	.804191	.739000
	WYE	.760377	.767735	.228649	.510945	.756176	.841938	.783573
	CWMRICKET	.777730	.775801	1.124013	1.007987	.755782	.866502	.819174
	BWLCH CREOLAN	.785420	.783258	.866766	1.125781	.838038	.872409	.820407
	FRONTBALLAN	.826521	.823003	.299633	.627064	.730631	.890537	.820607
	PENRALT	.833128	.844337	1.557972	.413581	.974468	.872147	.798755
	FRONVELLAN	.865911	.856258	.352564	.993179	.689515	.832659	.801017
	BRYNYFEDWEN	.876411	.873300	.836718	1.391328	1.073532	.796461	.855069
	SEVERN WATER	.902058	.909996	1.539676	.857370	.681459	.789671	.949688
	SEVERN	.983746	.980882	.052573	.381433	1.230470	.837798	1.051869
DETERMINATION COEF.		1.000000	.998958					
MODEL ERROR METRICS		.000000	.000136	.504105	.195727	.016252	.005112	.001354


THE MONTGOMERYSHIRE LEAD MINES
Ranked by The Logarithm of Total Production
DATA MODEL
ORIGINAL
LOGIC SWITCHED
NU = 64
NU = 128
NU = 256
NU = 512
NU = 1024
Normalised Production
Normalised Rank
Drafted by James R Warren
Copyright: James R Warren, 1995

APPENDIX FOUR

Program POLYWORK.BAS

```
'       PROGRAM POLYWORK.BAS
'       A PROGRAM TO GENERATE A POLYNOMIAL REGRESSION EQUATION
'       OR USE A POLYNOMIAL EQUATION TO GENERATE A CURVE OR DISTRIBUTION.
'       THE INPUT AND OUTPUT DATA ARE STORED IN COMMA SEPARATED VARIABLE
'       FILES AMENABLE TO SPREADSHEET MANIPULATION.
'       THESE FILES BEAR THE EXTENSIONS SHOWN:-
'
'              REGRESSION INPUT DATA                    .CSV
'              POLYNOMIAL REGRESSION COEFFICIENTS       .COF
'              FITTED CURVE CO-ORDINATES                .CUR
'              FREQUENCY DISTRIBUTION DATA              .DIS
'
'       WRITTEN BY:-
'
'          JAMES R WARREN BSc MSc PhD PGCE
'          "SOUTHGATE"
'          31 VICTORIA AVENUE
'          BLOXWICH
'          WS3 3HS
'          UNITED KINGDOM
'
'          21 MAY 1995
'
'       THIS PROGRAM IS WRITTEN IN MICROSOFT QBASIC
'
' VARIABLE TYPE DEFAULTS
       DEFDBL A-H, O-R, T-Z
       DEFSTR S
       DEFINT I-K, M-N
       DEFLNG L
' SEGMENT DECLARATIONS
       DECLARE SUB CSVIN (ID, SF)
       DECLARE SUB COEFOUT (IV, CD, CC, CE)
       DECLARE SUB COFIN (SF, IN, DB, UB)
       DECLARE SUB DATATRANS (ISW, IU, IV, N, X(), Y())
       DECLARE SUB DEFINITE (ID, FD, VM, DB, UB, C(), A())
       DECLARE SUB DISIN (SF, IP, DB, UB, JN)
       DECLARE SUB GETTER ()
       DECLARE SUB HEADERIN (IU, SF, SHT, SST, SXT, SYT)
       DECLARE SUB HEADEROUT (IV, SF, SHT, SST, SXT, SYT)
       DECLARE SUB INTEGRALOUT (IV, FD, VM)
       DECLARE SUB MENU (IC)
       DECLARE SUB MOMENTSOUT (IV, DM, DS, ZT)
       DECLARE SUB NOTE (S)
       DECLARE SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
       DECLARE SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
       DECLARE SUB POLYDIST (ID, IP, JN, N, DB, UB, DM, DS, ZT, C(), A(), X(), Y())
       DECLARE SUB POLYREG (ID, N, X(), Y(), C(), A(), CD, CC, CE)
       DECLARE SUB POLYSIMP (ID, FS, N, X(), Y())
       DECLARE SUB POLYSYN (ID, N, IN, DB, UB, C(), A(), X(), Y())
       DECLARE SUB RDUMP (SPATH, R(), ID, J)
       DECLARE SUB SIMPSON (FS, N, X(), Y())
' COMMON VARIABLES
       COMMON SHARED IA, SA
       COMMON SHARED IZ, IERR
       COMMON SHARED SC, SM
       COMMON SHARED S6, S8, S9, S10, S14, S15
       COMMON SHARED SE, SD, SN, SW, SS, SO, SL, SU
       COMMON SHARED SA8, SI1, SI2, SI4, SF13.6
' STATIC ARRAY DEFINITIONS
       DIM C(12), A(12), X(1000), Y(1000)
' DYNAMIC ARRAY DEFINITIONS
'      ( none )
' DEVICE ATTRIBUTIONS
       SCREEN 12: WINDOW (1, 1)-(640, 480)
' LOGICAL UNIT AND PATHNAME SETTINGS
       IU = 1: IV = 2: SP = "C:\QBASIC\QBFILES\"
' FORMAT DEFINITIONS
       SI1 = "#": SI2 = "##": SI4 = "####": SF13.6 = "######.######"
' NUMERICAL CONSTANT DEFINITIONS
'      ( none )
' STRING CONSTANT DEFINITIONS
       SC = ":": SM = ","
       S6 = SPACE$(6): S8 = SPACE$(8): S9 = SPACE$(9): S10 = SPACE$(10)
       S14 = SPACE$(14): S15 = SPACE$(15)
' TEXT VARIABLE DEFINITIONS
       SE = "ENTER THE ": SD = "DEGREE OF ": SN = "NUMBER OF ": SW = "FILENAME ( WITHOUT
EXTENSION )"
       SO = "ORDINAL ": SS = "ABSCISSAL ": SL = "LOWER BOUND ": SU = "UPPER BOUND "
'
```

```
' ** THE ALGORITHM **
'
       DO
          IERR = 0
          ERASE C, A, X, Y
          MENU IC
          SELECT CASE IC
             CASE 1
                CSVIN ID, SF
                OPEN "I", IU, SP + SF + ".CSV"
                OPEN "O", IV, SP + SF + ".COF"
                HEADERIN IU, SF, SHT, SST, SXT, SYT
                DATATRANS 1, IU, IV, N, X(), Y()
                IF ID < 13 AND N - ID > 1 THEN
                      POLYREG ID, N, X(), Y(), C(), A(), CD, CC, CE
                   ELSE
                      CLS : COLOR 4: PRINT "INSUFFICIENT DATA"
                      PRINT "OR NUMBER OF DEGREES EXCEEDS 12"
                      IERR = 1
                      BEEP: SLEEP 2
                END IF
                IF IZ = 1 AND IERR = 0 THEN
                      HEADEROUT IV, SF, SHT, SST, SXT, SYT
                      DATATRANS 2, IU, IV, ID + 1, C(), A()
                      COEFOUT IV, CD, CC, CE
                   ELSE
                      IF IZ = 0 THEN
                         CLS : COLOR 4: PRINT "NO UNIQUE SOLUTION"
                         IERR = 1
                         BEEP: SLEEP 2
                      END IF
                END IF
                CLOSE IU, IV
             CASE 2
                COFIN SF, IN, DB, UB
                OPEN "I", IU, SP + SF + ".COF"
                OPEN "O", IV, SP + SF + ".CUR"
                HEADERIN IU, SF, SHT, SST, SXT, SYT
                DATATRANS 1, IU, IV, IDD, C(), A()
                ID = IDD - 1
                POLYSYN ID, N, IN, DB, UB, C(), A(), X(), Y()
                DEFINITE ID, FD, VM, DB, UB, C(), A()
                HEADEROUT IV, SF, SHT, SST, SXT, SYT
                DATATRANS 2, IU, IV, N, X(), Y()
                INTEGRALOUT IV, FD, VM
                CLOSE IU, IV
             CASE 3
                DISIN SF, IP, DB, UB, JN
                OPEN "I", IU, SP + SF + ".COF"
                OPEN "O", IV, SP + SF + ".DIS"
                HEADERIN IU, SF, SHT, SST, SXT, SYT
                DATATRANS 1, IU, IV, IDD, C(), A()
                ID = IDD - 1
                POLYDIST ID, IP, JN, N, DB, UB, DM, DS, ZT, C(), A(), X(), Y()
                HEADEROUT IV, SF, SHT, SST, SXT, SYT
                DATATRANS 2, IU, IV, N, X(), Y()
                MOMENTSOUT IV, DM, DS, ZT
                CLOSE IU, IV
          END SELECT
       LOOP UNTIL IC = 4
       CLS
       END

      SUB COEFOUT (IV, CD, CC, CE)
' A SUBROUTINE TO WRITE THE POLYNOMIAL REGRESSION SUMMARY CORRELATIVE METRICS
'     ARGUMENTS:
'        IV     THE OUTPUT LOGICAL UNIT NUMBER
'        CD     THE COEFFICIENT OF DETERMINATION
'        CC     THE COEFFICIENT OF CORRELATION
'        CE     THE STANDARD ERROR OF THE ESTIMATE
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
      PRINT #IV, "COEFFICIENT OF DETERMINATION = "; SM; CD
      PRINT #IV, "COEFFICIENT OF CORRELATION = "; SM; CC
      PRINT #IV, "STANDARD ERROR OF THE ESTIMATE = "; SM; CE
      END SUB

       SUB COFIN (SF, IN, DB, UB)
' A SUBROUTINE TO OBTAIN KEYBOARD DATA FOR SEGMENT POLYSYN
'      ARGUMENTS:
'         SF     THE GENERIC FILENAME
'         IN     THE NUMBER OF ABSCISSAL INTERVALS
```

```
'         DB      THE ABSCISSAL LOWER BOUND
'         UB      THE ABSCISSAL UPPER BOUND
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
      CLS : COLOR 2
      LOCATE 8, 33: PRINT "SEGMENT POLYSYN"
      LOCATE 10, 19: PRINT "A Segment to Synthesise The Curve Expressing"
      LOCATE 11, 13: PRINT "The Polynomial Equation Defined by Coefficients in *.COF"
      LOCATE 12, 18: PRINT "And to Record the Curve Co-ordinates in *.CUR"
      LOCATE 14, 14: PRINT SE + SW + SC
      LOCATE 15, 14: PRINT SE + SN + SS + "INTERVALS " + SC
      LOCATE 16, 14: PRINT SE + SS + SL + S8 + SC
      LOCATE 17, 14: PRINT SE + SS + SU + S8 + SC
      PICK 4, 0, 0, 0, SF, 14, 56, 3, SA8
      PICK 1, IN, 0, 0, S, 15, 56, 3, SI4
      PICK 3, 0, 0, DB, S, 16, 56, 3, SF13.6
      PICK 3, 0, 0, UB, S, 17, 56, 3, SF13.6
      END SUB

      SUB CSVIN (ID, SF)
' A SUBROUTINE TO OBTAIN KEYBOARD DATA FOR SEGMENT POLYREG
'     ARGUMENTS:
'         ID      THE DEGREE OF THE POLYNOMIAL TO BE FITTED
'         SF      THE GENERIC FILENAME
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
      CLS : COLOR 2
      LOCATE 10, 33: PRINT "SEGMENT POLYREG"
      LOCATE 12, 12: PRINT "A Segment to Perform an ID-degree Polynomial Regression"
      LOCATE 13, 17: PRINT "On a Curve Defined by Co-ordinates in File *.CSV"
      LOCATE 15, 17: PRINT SE + SD + "THE FITTED EQUATION " + SC
      LOCATE 16, 17: PRINT SE + SW + SC
      PICK 1, ID, 0, 0, S, 15, 59, 3, SI2
      PICK 4, 0, 0, 0, SF, 16, 59, 3, SA8
      END SUB

      SUB DATATRANS (ISW, IU, IV, N, X(), Y())
' A SUBROUTINE TO READ OR WRITE THE CO-ORDINATES DATA LIST
'     ARGUMENTS:
'         ISW     THE TRANSPUT SELECTOR SWITCH
'                   1   READ
'                   2   WRITE
'         IU      THE INPUT  LOGICAL UNIT NUMBER
'         IV      THE OUTPUT LOCICAL UNIT NUMBER
'         N       THE NUMBER OF DATA CO-ORDINATES
'         X()     THE ARRAY OF ABSCISSAL CO-ORDINATES
'         Y()     THE ARRAY OF ORDINAL   CO-ORDINATES
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
      IF ISW = 1 THEN
            INPUT #IU, N
            FOR I = 1 TO N: INPUT #IU, X(I), Y(I): NEXT I
         ELSE
            PRINT #IV, N
            FOR I = 1 TO N: PRINT #IV, X(I); SM; Y(I): NEXT I
      END IF
      END SUB

      SUB DEFINITE (ID, FD, VM, DB, UB, C(), A())
' A SUBROUTINE TO COMPUTE THE DEFINITE INTEGRAL OF
' A POLYNOMIAL EQUATION BETWEEN DB AND UB
' AND TO ESTABLISH THE INTEGRAL MEAN VALUE BETWEEN
' THOSE BOUNDS
'     ARGUMENTS:
'         ID      THE DEGREE OF THE POLYNOMIAL
'         FD      THE DEFINITE INTEGRAL
'         VM      THE INTEGRAL MEAN VALUE
'         DB      THE INTERVAL LOWER BOUND
'         UB      THE INTERVAL UPPER BOUND
'         C()     THE ARRAY OF TERM EXPONENTS
'         A()     THE ARRAY OF TERM COEFFICIENTS
'
      FOR I = 1 TO ID + 1
         T1 = T1 + (A(I) * DB ^ I) / I: T2 = T2 + (A(I) * UB ^ I) / I
      NEXT I
      FD = T2 - T1: VM = FD / (UB - DB)
      END SUB

      SUB DISIN (SF, IP, DB, UB, JN)
' A SUBROUTINE TO OBTAIN KEYBOARD DATA FOR SEGMENT POLYDIST
'     ARGUMENTS:
'         SF      THE GENERIC FILENAME
```

```
'          IP      THE NUMBER OF SYNTHETIC ABSCISSAL POINTS
'          DB      THE ABSCISSAL LOWER BOUND
'          UB      THE ABSCISSAL UPPER BOUND
'          JN      THE NUMBER OF ORDINAL INCREMENTS
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
       CLS : COLOR 2: LOCATE 6, 33: PRINT "SEGMENT POLYDIST"
       LOCATE 8, 14: PRINT "A Segment to Generate a Grouped Frequency Distribution"
       LOCATE 9, 7: PRINT "From Log(Quantity) vs. Rank Data Expressed by a Polynomial
Equation"
       LOCATE 10, 22: PRINT "Whose Co-efficients are Read from *.COF"
       LOCATE 11, 23: PRINT "The Distribution is Recorded in *.DIS"
       LOCATE 13, 10: PRINT SE + SW + S6 + SC
       LOCATE 14, 10: PRINT SE + SN + "SYNTHETIC " + SS + "POINTS" + SC
       LOCATE 15, 10: PRINT SE + SS + SL + S14 + SC
       LOCATE 16, 10: PRINT SE + SS + SU + S14 + SC
       LOCATE 17, 10: PRINT SE + SN + SO + "INCREMENTS" + S8 + SC
       PICK 4, 0, 0, 0, SF, 13, 58, 3, SA8
       PICK 1, IP, 0, 0, S, 14, 58, 3, SI4
       PICK 3, 0, 0, DB, S, 15, 58, 3, SF13.6
       PICK 3, 0, 0, UB, S, 16, 58, 3, SF13.6
       PICK 1, JN, 0, 0, S, 17, 58, 3, SI2
       END SUB

       SUB GETTER
' A SUBROUTINE TO ACCEPT A KEYSTROKE AS SA AND TO YIELD ITS ASCII CODE AS IA
'      ( THE ARGUMENTS SA AND IA ARE COMMON SHARED )
'
       DO
          SA = INKEY$
       LOOP UNTIL SA <> ""
       IA = ASC(SA)
       END SUB

       SUB HEADERIN (IU, SF, SHT, SST, SXT, SYT)
' A SUBROUTINE TO READPOLYNOMIAL FILE HEADER DATA
'      ARGUMENTS:
'          IU      THE INPUT FILE LOGICAL UNIT NUMBER
'          SF      THE GENERIC FILE NAME
'          SHT     THE FILE MAIN HEADING
'          SST     THE FILE SUB  HEADING
'          SXT     THE ABSCISSAL DESCRIPTOR
'          SYT     THE ORDINAL DESCRIPTOR
'
       INPUT #IU, SF
       INPUT #IU, SHT
       INPUT #IU, SST
       INPUT #IU, SXT
       INPUT #IU, SYT
       END SUB

       SUB HEADEROUT (IV, SF, SHT, SST, SXT, SYT)
' A SUBROUTINE TO WRITE POLYNOMIAL FILE HEADER DATA
'      ARGUMENTS:
'          IV      THE OUTPUT FILE LOGICAL UNIT NUMBER
'          SF      THE GENERIC FILENAME
'          SHT     THE FILE MAIN HEADING
'          SST     THE FILE SUB  HEADING
'          SXT     THE ABSCISSAL DESCRIPTOR
'          SYT     THE ORDINAL   DESCRIPTOR
'
       PRINT #IV, SF
       PRINT #IV, SHT
       PRINT #IV, SST
       PRINT #IV, SXT
       PRINT #IV, SYT
       END SUB

       SUB INTEGRALOUT (IV, FD, VM)
' A SUBROUTINE TO WRITE THE DEFINITE INTEGRAL
' AN THE INTEGRAL MEAN VALUE
'      ARGUMENTS:
'          IV      THE OUTPUT LOGICAL UNIT NUMBER
'          FD      THE DEFINITE INTEGRAL
'          VM      THE INTEGRAL MEAN VALUE
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
       PRINT #IV, "DEFINITE INTEGRAL = "; SM; FD
       PRINT #IV, "MEAN VALUE = "; SM; VM
       END SUB

       SUB MENU (IC)
```

```
' A SUBROUTINE TO OBTAIN THE FUNCTION CHOICE FROM THE USER
'       ARGUMENT:
'          IC     THE USER FUNCTION CHOICE
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
       CLS : COLOR 2
       LOCATE 8, 31: PRINT "PROGRAM POLYWORK.BAS"
       LOCATE 10, 14: PRINT "A Program to Generate a Polynomial Regression Equation"
       LOCATE 11, 9: PRINT "Or Use a Polynomial Equation to Generate a Curve or Distribution"
       LOCATE 13, 3: PRINT "1    Generate a Polynomial Regression Equation"
       LOCATE 14, 3: PRINT "2    Generate XY-Plot Data from Polynomial Coefficients"
       LOCATE 15, 3: PRINT "3    Generate a Grouped Frequency Distribution using Polynomial
Coefficients"
       LOCATE 16, 3: PRINT "4    Quit"
       LOCATE 18, 27: PRINT "Select Your Option by Number"
       GETTER
       IC = IA - 48
       END SUB

       SUB MOMENTSOUT (IV, DM, DS, ZT)
' A SUBROUTINE TO WRITE THE FREQUENCY DISTRIBUTION SUMMARY METRICS
'       ARGUMENTS:
'          IV     THE OUTPUT LOGICAL UNIT NUMBER
'          DM     THE MEAN               OF THE DISTRIBUTION
'          DS     THE STANDARD DEVIATION OF THE DISTRIBUTION
'          ZT     THE SUM OF THE ORDINAL QUANTITIES
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
       PRINT #IV, "DISTRIBUTION MEAN = "; SM; DM
       PRINT #IV, "DISTRIBUTION STANDARD DEVIATION = "; SM; DS
       PRINT #IV, "SUM OF ORDINAL QUANTITIES = "; SM; ZT
       END SUB

       SUB NOTE (S)
' A SUBROUTINE TO SOUND A NOTE UPON THE COMPUTER SPEAKER
'       ARGUMENT:
'          S     THE NOTE SPECIFIER STRING "IN$NL" e.g. "2B0506"
'                   I   THE OCTAVE NUMBER        ( 0-6 )
'                   N$  THE NOTE LETTER          ( ABCDEFG )
'                   N   THE NOTE NUMBER          ( 0-84 )
'                   L   THE LENGTH OF THE NOTE   ( 1-64 )
'
       I = VAL(MID$(S, 1, 1)): N$ = MID$(S, 2, 1)
       N = VAL(MID$(S, 3, 2)): L = VAL(MID$(S, 5, 2))
       PLAY "O" + STR$(I) + "N" + STR$(N) + "L" + STR$(L) + "X" + VARPTR$(N$)
       END SUB

       SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
' A SUBROUTINE TO OBTAIN A VARIABLE OF TYPE KT AT SCREEN POSITION IY,IX
'       ARGUMENTS:
'          KT     THE DATUM TYPE CHOICE
'                    1   SHORT INTEGER
'                    2   LONG  INTEGER
'                    3   DOUBLE PRECISION REAL
'                    4   STRING
'          I      THE SHORT INTEGER
'          L      THE LONG INTEGER TO BE OBTAINED  ( OPTION )
'          R      THE REAL         TO BE OBTAINED  ( OPTION )
'          S      THE STRING       TO BE OBTAINED  ( OPTION )
'          IY     THE STARTING SCREEN ROW
'          IX     THE STARTING SCREEN COLUMN
'          MK     THE PRINTING COLOR
'          SF     THE PROPER ( OR DEFAULT ) PRINTING FORMAT
'                 ( "SAnn" TRUNCATES A STRING TO nn CHARACTERS )
'
       SDEL = CHR$(0) + CHR$(83)
       COLOR MK: LOCATE IY, IX: IO = I: LO = L: RO = R: SOL = S: L1 = LEN(SF)
       SELECT CASE KT
          CASE 1
             IF LEN(STR$(I)) > L1 THEN L1 = LEN(STR$(I))
          CASE 2
             IF LEN(STR$(L)) > L1 THEN L1 = LEN(STR$(L))
          CASE 3
             IF LEN(STR$(R)) > L1 THEN L1 = LEN(STR$(R))
          CASE 4
             L1 = LEN(S)
       END SELECT
       PRINT SPACE$(L1)
       DO
          LOCATE IY, IX: PRINT "."
          FOR II = 1 TO 10: NEXT II
          LOCATE IY, IX: PRINT " "
```

```
            SA = INKEY$
         LOOP UNTIL SA <> ""
         IA = ASC(SA)
         IF IA <> 13 THEN
            SCON = "": IT = IX: LOCATE IY, IX
               DO
                  IF SA = SDEL THEN
                        IT = IT - 1
                        LOCATE IY, IT: PRINT SPACE$(1)
                        LOCATE IY, IT: SCON = LEFT$(SCON, LEN(SCON) - 1)
                     ELSE
                        SELECT CASE KT
                           CASE 1 TO 3
                              IF IA > 47 AND IA < 58 OR IA = 46 OR IA = 45 THEN
                                    SCON = SCON + SA
                                    PRINT SA;
                                    IT = IT + 1
                                 ELSE
                                    IF IA <> 13 THEN NOTE "2B0506"
                              END IF
                           CASE 4
                              IF IA > 31 AND IA < 127 THEN
                                    SCON = SCON + SA
                                    PRINT SA;
                                    IT = IT + 1
                                 ELSE
                                    IF IA <> 13 THEN NOTE "2B0506"
                              END IF
                        END SELECT
                  END IF
                  GETTER
               LOOP UNTIL IA = 13
               SELECT CASE KT
                  CASE 1
                     I = INT(VAL(SCON) + .5)
                  CASE 2
                     L = INT(VAL(SCON) + .5)
                  CASE 3
                     R = VAL(SCON)
               END SELECT
            ELSE
               I = IO: L = LO: R = RO
         END IF
         L1 = LEN(SF): IF LEN(SCON) > L1 THEN L1 = LEN(SCON): PRINT SPACE$(L1)
         LOCATE IY, IX: PRINT SPACE$(L1): LOCATE IY, IX
         SELECT CASE KT
            CASE 1
               PRINT USING SF; I
            CASE 2
               PRINT USING SF; L
            CASE 3
               PRINT USING SF; R
            CASE 4
               LOCATE IY, IX: PRINT SPACE$(LEN(SCON)): LOCATE IY, IX
               IF SOL <> "" AND SCON = "" THEN SCON = SOL
               IF LEFT$(SF, 2) = "SA" THEN
                     S = LEFT$(SCON, VAL(MID$(SF, 3)))
                  ELSE
                     S = SCON
               END IF
               PRINT S
         END SELECT
         END SUB

         SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
    ' A SUBROUTINE TO PLACE A VARIABLE OF TYPE KT AT SCREEN POSITION IY, IX
    ' OR ALTERNATIVELY PLACE THE VARIABLE WITHIN A PADDED REPORT FILE
    '    ARGUMENTS:
    '       KT     THE DATUM TYPE CHOICE
    '                 1   SHORT INTEGER
    '                 2   LONG INTEGER
    '                 3   DOUBLE PRECISION REAL
    '                 4   STRING
    '       I      THE SHORT INTEGER TO BE PRINTED  ( OPTION )
    '       L      THE LONG  INTEGER TO BE PRINTED  ( OPTION )
    '       R      THE REAL          TO BE PRINTED  ( OPTION )
    '       S      THE STRING        TO BE PRINTED  ( OPTION )
    '       IY     THE STARTING SCREEN ROW
    '       IX     THE STARTING SCREEN COLUMN
    '       IU     THE LOGICAL UNIT NUMBER
    '                 1   PRINT TO THE SCREEN
    '                 2   PRINT TO A REPORT FILE
```

```
'         MK      THE NOMINAL PRINTING COLOR
'         JF      THE LINE FEED SUPPRESSOR SWITCH
'                    0   FOLLOW WITH A LINE FEED
'                    1   DO NOT FOLLOW WITH A LINE FEED
'         SF      THE REQUIRED PRINTING FORMAT
'         IB      THE NUMBER OF FORWARD PADDING SPACES
'
      IF SF = "" THEN SF = "#########"
      COLOR MK
      SELECT CASE IU
         CASE 1
            LOCATE IY, IX
               SELECT CASE KT
                  CASE 1
                     PRINT USING SF; I
                  CASE 2
                     PRINT USING SF; L
                  CASE 3
                     PRINT USING SF; R
                  CASE 4
                     PRINT S
               END SELECT
         CASE 2
            IF IB > 0 THEN PRINT #IU, SPACE$(IB);
            IF JF = 1 THEN SCC = ";" ELSE SCC = CHR$(13) + CHR$(10)
               SELECT CASE KT
                  CASE 1
                     PRINT #IU, USING SF; I; SCC
                  CASE 2
                     PRINT #IU, USING SF; L; SCC
                  CASE 3
                     PRINT #IU, USING SF; R; SCC
                  CASE 4
                     PRINT #IU, S
               END SELECT
      IF JF = 0 THEN PRINT #LU,
      END SELECT
      END SUB

      SUB POLYDIST (ID, IP, JN, N, DB, UB, DM, DS, ZT, C(), A(), X(), Y())
' A SUBROUTINE TO GENERATE A GROUPED FREQUENCY DISTRIBUTION
' FROM DATA EXPRESSED BY A POLYNOMIAL EQUATION
'     ARGUMENTS:
'        ID    THE DEGREE OF THE REGRESSION POLYNOMIAL
'        IP    THE NUMBER OF SYNTHETIC ABSCISSAL POINTS
'        JN    THE NUMBER OF ORDINAL INCREMENTS
'        N     THE NUMBER OF DATA PAIRS
'        DB    THE ABSCISSAL LOWER BOUND
'        UB    THE ABSCISSAL UPPER BOUND
'        DM    THE MEAN               OF THE DISTRIBUTION
'        DS    THE STANDARD DEVIATION OF THE DITRIBUTION
'        ZT    THE SUM OF THE ORDINAL QUANTITIES
'        C()   THE ARRAY OF TERM EXPONENTS
'        A()   THE ARRAY OF TERM COEFFICIENTS
'        X()   THE ARRAY OF ABSCISSAL CO-ORDINATES
'        Y()   THE ARRAY OF ORDINAL CO-ORDINATES
'
      DIM Z(5000)
      IF IP > 5000 THEN IP = 5000
      N = JN + 1: R = UB - DB: ZT = 0!
      ZX = -9.999999999999999D+33: ZN = 9.999999999999999D+33
      COLOR 13
      LOCATE 19, 29: PRINT "POINTS DEFINED  :"
      LOCATE 20, 29: PRINT "POINTS ALLOCATED:"
      COLOR 4
      XI = R / (IP - 1): XX = DB - XI
      FOR I = 1 TO IP
         XX = XX + XI
         FOR J = 1 TO ID + 1: Z(I) = Z(I) + A(J) * XX ^ C(J): NEXT J
         Z(I) = EXP(Z(I))
         IF Z(I) < ZN THEN ZN = Z(I)
         IF Z(I) > ZX THEN ZX = Z(I)
         ZT = ZT + Z(I)
         LOCATE 19, 47: PRINT I
      NEXT I
      RY = ZX - ZN: YI = RY / JN
      X(1) = ZN: FOR I = 2 TO N: X(I) = X(I - 1) + YI: NEXT I
      COLOR 6
      FOR I = 1 TO IP
         LOCATE 22, 1: J = INT((Z(I) - ZN) / YI) + 1: Y(J) = Y(J) + 1
         LOCATE 20, 47: PRINT I
      NEXT I
```

```
      TF = 0!: TFX = 0!: TFX2 = 0!
      FOR I = 1 TO JN
         XM = X(I) + YI / 2
         TF = TF + Y(I): TFX = TFX + Y(I) * XM
         TFX2 = TFX2 + Y(I) * XM ^ 2
      NEXT I
      DM = TFX / TF: DS = SQR(TFX2 / TF - DM * DM)
      END SUB

      SUB POLYREG (ID, N, X(), Y(), C(), A(), CD, CC, CE)
' A SUBROUTINE TO PERFORM A POLYNOMIAL REGRESSION
'     ARGUMENTS:
'        ID     THE DEGREE OF THE REGRESSION POLYNOMIAL
'        N      THE NUMBER OF DATA CO-ORDINATES
'        X()    THE ARRAY OF ABSCISSAL CO-ORDINATES
'        Y()    THE ARRAY OF ORDINAL   CO-ORDINATES
'        C()    THE ARRAY OF TERM EXPONENTS
'        A()    THE ARRAY OF TERM CO-EFFICIENTS
'        CD     THE COEFFICIENT OF DETERMINATION
'        CC     THE COEFFICIENT OF CORRELATION
'        CE     THE STANDARD ERROR OF THE ESTIMATE
' ( IZ AND IERR ARE COMMON SHARED )
'
      DIM AA(25), R(13, 14), T(14)
      KD = ID + 1: KKD = ID + 2
' POPULATE THE SOLUTION MATRICES WITH A SYSTEM OF EQUATIONS
      AA(1) = N
      FOR I = 1 TO N
         FOR J = 2 TO 2 * ID + 1: AA(J) = AA(J) + X(I) ^ (J - 1): NEXT J
         FOR K = 1 TO KD
            R(K, KKD) = T(K) + Y(I) * X(I) ^ (K - 1): T(K) = R(K, KKD)
         NEXT K
         T(KKD) = T(KKD) + Y(I) ^ 2
      NEXT I
' SOLVE THE SYSTEM OF EQUATIONS IN THE MATRICES
      FOR J = 1 TO KD: FOR K = 1 TO KD: R(J, K) = AA(J + K - 1): NEXT K: NEXT J
      J = 0
      DO
         J = J + 1: K = J
         IZ = 0: IERR = 1
            DO
               IF R(K, J) <> 0 THEN
                  FOR I = 1 TO KKD: RS = R(J, I): R(J, I) = R(K, I): R(K, I) = RS: NEXT I
                  Z = 1 / R(J, J): FOR I = 1 TO KKD: R(J, I) = Z * R(J, I): NEXT I
                  FOR KK = 1 TO J - 1
                     Z = -R(KK, J): FOR I = 1 TO KKD: R(KK, I) = R(KK, I) + Z * R(J, I): NEXT
I
                  NEXT KK
                  FOR KK = J + 1 TO KD
                     Z = -R(KK, J): FOR I = 1 TO KKD: R(KK, I) = R(KK, I) + Z * R(J, I): NEXT
I
                  NEXT KK
                  IZ = 1: IERR = 0
               END IF
               K = K + 1
            LOOP UNTIL K = KD OR IZ = 1
      LOOP UNTIL J = KD OR IERR = 1
' LOAD THE RESULTS ARRAYS
      FOR I = 1 TO KD
         C(I) = I - 1: A(I) = R(I, KKD)
      NEXT I
' COMPUTE THE REGRESSION ANALYSIS
      IF IERR = 0 THEN
         P = 0: FOR J = 2 TO KD: P = P + R(J, KKD) * (T(J) - AA(J) * T(1) / N): NEXT J
         Q = T(KKD) - T(1) ^ 2 / N: Z = Q - P: I = N - KD
         CD = P / Q: CC = SQR(CD)
         IF Z < 0 THEN Z = 0
         CE = SQR(Z / I)
      END IF
      END SUB

      SUB POLYSIMP (ID, FS, N, X(), Y())
' A SUBROUTINE TO PERFORM A SIMPSON'S RULE INTEGRATION
' UPON THE CURVE COMPUTED BY SEGMENT POLYSYN.
' IF THE POLYNOMIAL DEGREE IS LESS THAN FOUR
' THE SHORT FORM OF THE INTEGRATING ALGORITHM IS EMPLOYED.
'     ARGUMENTS:
'        ID     THE DEGREE OF THE REGRESSION POLYNOMIAL
'        FS     THE SIMPSONIAN INTEGRAL
'        N      THE NUMBER OF DATA CO-ORDINATES
'        X()    THE ARRAY OF ABSCISSAL CO-ORDINATES
'        Y()    THE ARRAY OF ORDINAL   CO-ORDINATES
```

```
'
      IF ID < 4 THEN
            FS = ((X(N) - X(1)) / 6) * (Y(1) + 4 * Y(INT(N / 2) + 1) + Y(N))
         ELSE
            SIMPSON FS, N, X(), Y()
      END IF
      END SUB

      SUB POLYSYN (ID, N, IN, DB, UB, C(), A(), X(), Y())
' A SUBROUTINE TO SYNTHESISE THE CURVE EXPRESSING A POLYNOMIAL EQUATION
'     ARGUMENTS:
'        ID     THE DEGREE OF THE REGRESSION POLYNOMIAL
'        N      THE NUMBER OF DATA CO-ORDINATES
'        IN     THE NUMBER OF ABSCISSAL INTERVALS
'        DB     THE ABSCISSAL LOWER BOUND
'        UB     THE ABSCISSAL UPPER BOUND
'        C()    THE ARRAY OF TERM EXPONENTS
'        A()    THE ARRAY OF TERM CO-EFFICIENTS
'        X()    THE ARRAY OF ABSCISSAL CO-ORDINATES
'        Y()    THE ARRAY OF ORDINAL   CO-ORDINATES
'
      N = IN + 1: AI = (UB - DB) / IN: X(0) = DB - AI
      FOR I = 1 TO N
         X(I) = X(I - 1) + AI: Y(I) = 0!
         FOR J = 1 TO ID + 1
            Y(I) = Y(I) + A(J) * X(I) ^ (J - 1)
         NEXT J
      NEXT I
      END SUB

      SUB RDUMP (SPATH, R(), ID, J)
      OPEN "A", 4, SPATH
      PRINT #4, "J = "; J
      PRINT #4,
      FOR JJ = 1 TO ID + 2
      FOR KK = 1 TO ID + 2
      PRINT #4, USING "######.######"; R(JJ, KK);
      PRINT #4, " ";
      NEXT KK
      PRINT #4,
      NEXT JJ
      CLOSE 4
      END SUB

      SUB SIMPSON (FS, N, X(), Y())
' A SUBROUTINE TO PERFORM A SIMPSON'S RULE INTEGRATION
' UPON THE CURVE COMPUTED BY SEGMENT POLYSYN
'     ARGUMENTS:
'        FS     THE SIMPSONIAN INTEGRAL
'        N      THE NUMBER OF DATA CO-ORDINATES
'        X()    THE ARRAY OF ABSCISSAL CO-ORDINATES
'        Y()    THE ARRAY OF ORDINAL   CO-ORDINATES
'
      IN = N - 1
      IF IN MOD 2 = 0 THEN
            J = -2: K = 2: T = 0!
            FOR I = 2 TO IN
               J = -J: K = K + J: T = T + K * Y(I)
            NEXT I
            FS = ((X(N) - X(1)) / (3 * IN)) * (Y(1) + Y(N) + T)
         ELSE
            CLS : COLOR 4
            PRINT "ODD NUMBER OF ABSCISSAL INTERVALS:"
            PRINT "SIMPSONIAN INTEGRAL SET TO ZERO"
            IERR = 1
            FS = 0
            BEEP: SLEEP 2
      END IF
      END SUB
```

APPENDIX FIVE

Program MINEMODL.BAS

```
'       PROGRAM MINEMODL.BAS
'       A PROGRAM TO GENERATE A SERIES OF FOURIER-POLYNOMIAL MODEL FITMENTS
'       TO LEAD MINE LOG PRODUCTION VERSUS RANK DATA AND TO OUTPUT THESE
'       FITMENTS AS A COMPARATIVE TABLE.
'       THE INPUT AND OUTPUT DATA ARE STORED IN COMMA SEPARATED VARIABLE
'       FILES AMENABLE TO SPREADSHEET MANIPULATION.
'       THESE FILES BEAR THE EXTENSIONS SHOWN:-
'
'               ORIGINAL RANKED DATA                    .CSV
'               POLYNOMIAL REGRESSION COEFFICIENTS      .COF
'               MODEL FITMENTS REPORT                   .MOD
'
'       WRITTEN BY:-
'
'          JAMES R WARREN BSc MSc PhD PGCE
'          "SOUTHGATE"
'          31 VICTORIA AVENUE
'          BLOXWICH
'          WS3 3HS
'          UNITED KINGDOM
'
'          11 JULY 1995
'
'       THIS PROGRAM IS WRITTEN IN MICROSOFT QBASIC
'
' VARIABLE TYPE DEFAULTS
       DEFDBL A-H, O-R, T-Z
       DEFSTR S
       DEFINT I-K, M-N
       DEFLNG L
' SEGMENT DECLARATIONS
       DECLARE SUB DATATRANS (ISW, IU, IV, N, X(), Y())
       DECLARE SUB DETERMINATION (M, JC, JF, JE, W(), CD())
       DECLARE SUB ERRORMETRIC (M, JC, JF, JE, W(), ER())
       DECLARE SUB ERRORSET (SER())
       DECLARE SUB ERRORSHOW (I, IC, SER())
       DECLARE SUB FOURSERIES (M, M1, M2, N, X(), Y())
       DECLARE SUB GETTER ()
       DECLARE SUB HEADERIN (IU, SF, SHT, SST, SXT, SYT)
       DECLARE SUB INSCREEN (SFO, SHT, SST, K, NM, NX, SFN(), IR())
       DECLARE SUB LINEBLANKER (I, J)
       DECLARE SUB META (M, SM)
       DECLARE SUB MODELCALC (IU, K, M, NM, NX, SFO, SFN(), IR(), W(), CD(), ER(), SMN())
       DECLARE SUB MODELRPT (ISW, IV, SFO, SFN(), SHT, SST, K, M, IR(), NM, NX, W(), CD(),
ER(), SMN())
       DECLARE SUB NOSEG (SK)
       DECLARE SUB NOTE (S)
       DECLARE SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
       DECLARE SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
' COMMON VARIABLES
       COMMON SHARED IA, SA
       COMMON SHARED PI
       COMMON SHARED SC, SM, SCR
       COMMON SHARED SE, SN, SX, SU, SH, SO, SD, SL, ST, SR, SG, SJ, SW, SDA
       COMMON SHARED SP, SXV, SXC, SXM
       COMMON SHARED SA8, SI2, SFF
' STATIC ARRAY DEFINITIONS
       DIM SFN(12), C(12), CD(12), ER(12), A(12), IR(2, 12), SMN(100), W(12, 100)
' DYNAMIC ARRAY DEFINITIONS
'      ( none )
' DEVICE ATTRIBUTIONS
       SCREEN 12: WINDOW (1, 1)-(640, 480)
' LOGICAL UNIT, EXTENSION AND PATHNAME SETTINGS
       IU = 1: IV = 2
       SXV = ".CSV": SXC = ".COF": SXM = ".MOD"
       SP = "C:\QBASIC\QBFILES\"
' FORMAT DEFINITIONS
       SI2 = "##"
       SFF = "#.###############"
' NUMERICAL CONSTANT DEFINITIONS
       PI = 3.141592653589793#
' STRING CONSTANT DEFINITIONS
       SC = ":": SM = ",": SCR = CHR$(13) + CHR$(10)
' TEXT VARIABLE DEFINITIONS
       SE = "ENTER THE ": SN = "NUMBER OF ": SX = "INDEX FOR THE "
       ST = "FOURIER TERMS": SH = "HEADER": SO = "ORIGINAL ": SD = "MODEL "
       SL = "LOWER ": SU = "UPPER ": SR = "RANK ": SG = "REGRESSION "
       SJ = "FILENAME": SW = "SWITCHED": SDA = "DATA"
'
' ** THE ALGORITHM **
```

```
'
       INSCREEN SFO, SHT, SST, K, NM, NX, SFN(), IR()
       MODELCALC IU, K, M, NM, NX, SFO, SFN(), IR(), W(), CD(), ER(), SMN()
       MODELRPT 1, IV, SFO, SFN(), SHT, SST, K, M, IR(), NM, NX, W(), CD(), ER(), SMN()
       NOTE "3C0506"
       NOTE "4D0509"
       NOTE "5E0512"
       NOTE "6F0515"
       END

       SUB DATATRANS (ISW, IU, IV, N, X(), Y())
' A SUBROUTINE TO READ OR WRITE THE CO-ORDINATES DATA LIST
'      ARGUMENTS:
'         ISW    THE TRANSPUT SELECTOR SWITCH
'                   1   READ
'                   2   WRITE
'         IU     THE INPUT  LOGICAL UNIT NUMBER
'         IV     THE OUTPUT LOCICAL UNIT NUMBER
'         N      THE NUMBER OF DATA CO-ORDINATES
'         X()    THE ARRAY OF ABSCISSAL CO-ORDINATES
'         Y()    THE ARRAY OF ORDINAL   CO-ORDINATES
' ( EMPLOYED CLICHES ARE COMMON SHARED )
'
       IF ISW = 1 THEN
            INPUT #IU, N
            FOR I = 1 TO N: INPUT #IU, X(I), Y(I): NEXT I
          ELSE
            PRINT #IV, N
            FOR I = 1 TO N: PRINT #IV, X(I); SM; Y(I): NEXT I
       END IF
       END SUB

       SUB DETERMINATION (M, JC, JF, JE, W(), CD())
' A SUBROUTINE TO COMPUTE COEFFICIENTS OF DETERMINATION
' BETWEEN COLUMNS OF DATA IN ARRAY W(J,I)
'      ARGUMENTS:
'         M      THE NUMBER OF MINES
'         JC     THE DETERMINATION COEFFICIENT SERIAL NUMBER
'         JF     THE COLUMN OF FIDUCIAL REFERENCE DATA
'         JE     THE COLUMN OF MODEL ESSAY DATA
'         W()    THE MODEL SERIES MATRIX
'         CD()   THE ARRAY OF DETERMINATION COEFFICIENTS
'
       FOR I = 1 TO M: T3 = T3 + W(JF, I): NEXT I: T3 = T3 / M
       FOR I = 1 TO M: U1 = W(JE, I) - T3: U2 = W(JF, I) - T3: T1 = T1 + U1 * U1: T2 = T2 + U2
* U2: NEXT I
       CD(JC) = T1 / T2
       END SUB

       SUB ERRORMETRIC (M, JC, JF, JE, W(), ER())
' A SUBROUTINE TO COMPUTE MODEL ERROR METRICS
' BETWEEN COLUMNS OF DATA IN ARRAY W(J,I)
'      ARGUMENTS:
'         M      THE NUMBER OF MINES
'         JC     THE ERROR METRIC SERIAL NUMBER
'         JF     THE COLUMN OF FIDUCIAL REFERENCE DATA
'         JE     THE COLUMN OF MODEL ESSAY DATA
'         W()    THE MODEL SERIES MATRIX
'         ER()   THE ARRAY OF MODEL ERROR METRICS
'
       FOR I = 1 TO M
          D1 = W(JF, I): D2 = (D1 - W(JE, I))
          T2 = T2 + D2 * D2: T1 = T1 + D1 * D1
       NEXT I
       ER(JC) = T2 / T1
       END SUB

       SUB ERRORSET (SER())
' A SUBROUTINE TO DEFINE STANDARD ERROR FLASHES
'
       SER(1) = "ERROR ONE IS UNDEFINED"
       SER(2) = "ERROR TWO IS UNDEFINED"
       SER(3) = "ERROR THREE IS UNDEFINED"
       SER(4) = "ERROR FOUR IS UNDEFINED"
       SER(5) = "ERROR FIVE IS UNDEFINED"
       SER(6) = "ERROR SIX IS UNDEFINED"
       END SUB

       SUB ERRORSHOW (I, IC, SER())
' A SUBROUTINE TO EXHIBIT THE ERROR MESSAGE SER(I) WITH A WARNING BLEEP
'      ARGUMENT:
'         I      THE ERROR MESSAGE SERIAL NUMBER
```

```
'         IC      THE COLOR DEFINITION NUMBER
'         SER()  THE ARRAY OF ERROR DESCRIPTORS
'
      COLOR IC
      LINEBLANKER 24, 24
      LOCATE 24, 1: PRINT "ERROR " + STR$(I) + ": " + SER(I);
      NOTE "2B0506"
      SLEEP 2
      LINEBLANKER 24, 24
      END SUB

      SUB FOURSERIES (M, M1, M2, N, X(), Y())
' A SUBROUTINE TO COMPUTE A SQUARE-WAVE FOURIER SERIES
'      ARGUMENTS:
'         M      THE NUMBER OF RANKED DATA
'         M1     THE FIRST RANK FOR WHICH f(x)=1
'         M2     THE LAST  RANK FOR WHICH f(x)=1
'         N      THE NUMBER OF FOURIER SERIES TERMS
'         X()    THE NORMALISED INDEPENDENT VARIABLE
'         Y()    THE FUNCTION OF X() AT X(I)
' ( PI IS COMMON SHARED )
'
      Z = (M2 - M1 + 1) / (2 * M)
      FOR I = 1 TO M
         Y(I) = 2 * Z
         FOR J = 1 TO N
         Y(I) = Y(I) + (4 * SIN(J * PI * Z) * COS(J * PI * (Z + (M1 - 1) / M)) * COS(J * PI *
(1 - X(I))) / (J * PI))
         NEXT J
      NEXT I
      END SUB

      SUB GETTER
' A SUBROUTINE TO ACCEPT A KEYSTROKE AS SA AND TO YIELD ITS ASCII CODE AS IA
'      ( THE ARGUMENTS SA AND IA ARE COMMON SHARED )
'
      DO
         SA = INKEY$
      LOOP UNTIL SA <> ""
      IA = ASC(SA)
      END SUB

      SUB HEADERIN (IU, SF, SHT, SST, SXT, SYT)
' A SUBROUTINE TO READPOLYNOMIAL FILE HEADER DATA
'      ARGUMENTS:
'         IU      THE INPUT FILE LOGICAL UNIT NUMBER
'         SF      THE GENERIC FILE NAME
'         SHT     THE FILE MAIN HEADING
'         SST     THE FILE SUB  HEADING
'         SXT     THE ABSCISSAL DESCRIPTOR
'         SYT     THE ORDINAL DESCRIPTOR
'
      INPUT #IU, SF
      INPUT #IU, SHT
      INPUT #IU, SST
      INPUT #IU, SXT
      INPUT #IU, SYT
      END SUB

      SUB INSCREEN (SFO, SHT, SST, K, NM, NX, SFN(), IR())
' A SUBROUTINE TO ACCEPT FUNDAMENTAL RUN DATA
'      ARGUMENTS:
'         SFO     THE ORIGINAL DATA FILE NAME
'         SHT     THE OUTPUT MAIN HEADING
'         SST     THE OUTPUT SUB  HEADING
'         K       THE NUMBER OF CLASSES
'         NM      THE EXPONENT FOR THE LOWER NUMBER OF FOURIER SERIES TERMS
'         NX      THE EXPONENT FOR THE UPPER NUMBER OF FOURIER SERIES TERMS
'         SFN()  THE ARRAY OF POLYNOMIAL COEFFICIENT FILENAMES
'         IR()    THE ARRAY OF TERMINATION RANK POSITIONS
' ( SC AND ALL CLICHES ARE COMMON SHARED )
'
      S1 = "MINEMODL FUNDAMENTAL RUN " + SDA: S2 = STRING$(29, 45)
      S3 = SE + SJ + "S WITHOUT EXTENSIONS": S4 = SE + SJ + " FOR THE " + SO + SDA + SC
      S5 = SE + "MAIN " + SH + SC: S6 = SE + "SUB  " + SH + SC
      S7 = SE + SN + "CLASSES" + SC: S8 = SE + SX + "MINIMUM " + SN + ST + SC
      S9 = SE + SX + "MAXIMUM " + SN + ST + SC: S10 = "CLASS INFORMATION": S11 = STRING$(17,
45)
      PLACE 4, 0, 0, 0, S1, 2, 26, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S2, 3, 26, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S3, 5, 21, 1, 14, 0, SF, 0
      PLACE 4, 0, 0, 0, S4, 7, 9, 1, 2, 0, SF, 0
```

```
      PLACE 4, 0, 0, 0, S5, 8, 9, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S6, 9, 9, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S7, 10, 9, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S8, 11, 9, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S9, 12, 9, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S10, 15, 32, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, S11, 16, 32, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SG, 18, 26, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SL, 18, 41, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SL, 18, 41, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SU, 18, 51, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SJ, 19, 27, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SR, 19, 41, 1, 2, 0, SF, 0
      PLACE 4, 0, 0, 0, SR, 19, 51, 1, 2, 0, SF, 0
      PICK 4, 0, 0, 0, SFO, 7, 53, 3, SA8
      PICK 4, 0, 0, 0, SHT, 8, 32, 3, SA40
      PICK 4, 0, 0, 0, SST, 9, 32, 3, SA40
      PICK 1, K, 0, 0, S, 10, 38, 3, SI2
      PICK 1, NM, 0, 0, S, 11, 66, 3, SI2
      PICK 1, NX, 0, 0, S, 12, 66, 3, SI2
      FOR I = 1 TO K
         IW = I + 20
         PICK 4, 0, 0, 0, SFN(I), IW, 27, 3, SA8
         PICK 1, IR(1, I), 0, 0, S, IW, 42, 3, SI2
         PICK 1, IR(2, I), 0, 0, S, IW, 52, 3, SI2
      NEXT I
      END SUB

      SUB LINEBLANKER (I, J)
' A SUBROUTINE TO BLANK SCREEN LINES I TO J INCLUSIVE
'     ARGUMENTS:
'        I      THE START  LINE
'        J      THE FINISH LINE
      FOR K = I TO J: LOCATE K, 1: PRINT SPACE$(80); : NEXT K
      END SUB

      SUB META (M, SM)
' A SUBROUTINE TO NOTIFY THE CURRENT STAGE OF ELABORATION
'     ARGUMENTS:
'        M     THE ARBITRARY STAGE NUMBER
'              ( NEGATIVE FOR OMISSION )
'        SM    THE STAGE DESCRIPTOR
'
      IF M > -1 THEN
         LOCATE 1, 1: COLOR 2: PRINT "META"
         LOCATE 1, 6: COLOR 14: PRINT M
         LOCATE 1, 10: PRINT SPACE$(80): LOCATE 1, 10: COLOR 12: PRINT SM
      END IF
      END SUB

      SUB MODELCALC (IU, K, M, NM, NX, SFO, SFN(), IR(), W(), CD(), ER(), SMN())
' A SUBROUTINE TO CONSTRUCT THE FOURIER-POLYNOMIAL MODEL FITMENTS
' WITH THEIR RESPECTIVE MODEL ERROR METRICS
'     ARGUMENTS:
'        IU     THE INPUT LOGICAL UNIT NUMBER
'        K      THE NUMBER OF CLASSES
'        M      THE NUMBER OF MINES
'        NM     THE EXPONENT FOR THE LOWER NUMBER OF FOURIER SERIES TERMS
'        NX     THE EXPONENT FOR THE UPPER NUMBER OF FOURIER SERIES TERMS
'        SFO    THE ORIGINAL DATA FILE NAME
'        SFN()  THE ARRAY OF POLYNOMIAL COEFFICIENT FILENAMES
'        IR()   THE ARRAY OF TERMINATING RANK POSITIONS
'        W()    THE MODEL SERIES MATRIX
'        CD()   THE ARRAY OF DETERMINATION COEFFICIENTS
'        ER()   THE ARRAY OF MODEL ERROR METRICS
'        SMN()  THE ARRAY OF MINE NAMES
'
' DIMENSION THE LOCAL ARRAYS
      DIM ID(12), CC(12, 12), X(100), Y(100)
' LOAD THE ORIGINAL RANK VERSUS LOG PRODUCTION DATA
      OPEN "I", IU, SP + SFO + SXV
      HEADERIN IU, SF, SHT, SST, SXT, SYT
      INPUT #IU, M
      FOR I = 1 TO M: INPUT #IU, SMN(I), X(I), Y(I): NEXT I
      CLOSE IU
      FOR I = 1 TO M: W(1, I) = Y(I): NEXT I
' LOAD THE POLYNOMIAL REGRESSION DEGREES AND COEFFICIENTS
      FOR I = 1 TO K
         OPEN "I", IU, SP + SFN(I) + SXC
         HEADERIN IU, SF, SHT, SST, SXT, SYT
         DATATRANS 1, IU, IV, ID(I), X(), Y()
         CLOSE IU
```

```
         FOR J = 1 TO ID(I): CC(J, I) = Y(J): NEXT J
      NEXT I
' DEFINE THE ARRAY OF NORMALISED RANKS
      FOR I = 1 TO M: X(I) = (M - I) / (M - 1): NEXT I
' ESTABLISH THE POLYNOMIAL SEGMENTS BASE MODEL
      FOR I = 1 TO K
         FOR J = IR(1, I) TO IR(2, I)
            W(2, J) = 0!: FOR JJ = 1 TO ID(I): W(2, J) = W(2, J) + CC(JJ, I) * X(J) ^ (JJ -
1): NEXT JJ
         NEXT J
      NEXT I
' COMPUTE THE DETERMINATION COEFFICIENT AND THE MODEL ERROR METRIC
' ON POLYNOMIALS ALONE
      DETERMINATION M, 0, 1, 2, W(), CD()
      ERRORMETRIC M, 0, 1, 2, W(), ER()
' COMPUTE THE FOURIER-POLYNOMIAL MODEL FITMENTS
' AND THEIR ERROR METRICS
      II = 2
      FOR I = 1 TO M: X(I) = (M - I) / (M - 1): NEXT I
      FOR I = NM TO NX
         II = II + 1: NU = 2 ^ I
         FOR J = 1 TO K
            FOURSERIES M, IR(1, J), IR(2, J), NU, X(), Y()
            FOR JJ = 1 TO M
               Z = 0!: FOR KK = 1 TO ID(J): Z = Z + CC(KK, J) * X(JJ) ^ (KK - 1): NEXT KK
               W(II, JJ) = W(II, JJ) + Z * Y(JJ)
            NEXT JJ
         NEXT J
         ERRORMETRIC M, II - 2, 1, II, W(), ER()
      NEXT I
      END SUB

      SUB MODELRPT (ISW, IV, SFO, SFN(), SHT, SST, K, M, IR(), NM, NX, W(), CD(), ER(),
SMN())
' A SUBROUTINE TO OUTPUT THE FOURIER POLYNOMIAL FITMENTS
'     ARGUMENTS:
'        ISW    THE OUTPUT DESTINATION SWITCH
'                  0   OUTPUT TO THE SCREEN
'                  1   OUTPUT TO THE NOMINATED FILE
'        IV     THE OUTPUT LOGICAL UNIT NUMBER
'        SFO    THE ORIGINAL DATA FILE NAME
'        SFN()  THE ARRAY OF POLYNOMIAL COEFFICIENT FILENAMES
'        SHT    THE OUTPUT MAIN HEADING
'        SST    THE OUTPUT SUB  HEADING
'        K      THE NUMBER OF CLASSES
'        M      THE NUMBER OF MINES
'        IR()   THE ARRAY OF TERMINATING RANK POSITIONS
'        NM     THE EXPONENT FOR THE LOWER NUMBER OF FOURIER SERIES TERMS
'        NX     THE EXPONENT FOR THE UPPER NUMBER OF FOURIER SERIES TERMS
'        W()    THE MODEL SERIES MATRIX
'        CD()   THE ARRAY OF DETERMINATION COEFFICIENTS
'        ER()   THE ARRAY OF MODEL ERROR METRICS
'        SMN()  THE ARRAY OF MINE NAMES
' ( SP, SC, SM, SCR AND ALL CLICHES ARE COMMON SHARED )
'
      IF ISW = 0 THEN SOU = "CON" ELSE SOU = SP + SFO + SXM
      OPEN "O", IV, SOU
      SS = STRING$(41, 42)
      PRINT #IV, SS
      PRINT #IV, "A WORKSHEET TO PRESENT FOURIER-POLYNOMIAL"
      PRINT #IV, "       MODEL FITMENTS TO LEAD MINE"
      PRINT #IV, "     LOG PRODUCTION VERSUS RANK " + SDA
      PRINT #IV, SS + SCR + SCR
      PRINT #IV, "JAMES R WARREN" + SCR + SFO + SXM + SCR + DATE$ + SCR + SCR
      PRINT #IV, SHT + SCR + SST + SCR + SCR
      PRINT #IV, SN + "CLASSES" + SM; K
      PRINT #IV, SCR + SO + SDA + SM + SFO + SXV
      PRINT #IV, SN + "MINES" + SM; M
      PRINT #IV, SCR + SCR + "POLYNOMIAL SEGMENTS" + SCR + STRING$(19, 45) + SCR + SCR
      PRINT #IV, SG + SM + SL + SM + SU
      PRINT #IV, SJ + SM + SR + SM + SR
      PRINT #IV,
      FOR I = 1 TO K
         PRINT #IV, SFN(I) + SXC + SM; IR(1, I); SM; IR(2, I)
      NEXT I
      PRINT #IV, SCR + SCR + "THE TABLE OF MODEL APPROXIMATIONS" + SCR + STRING$(33, 45) +
SCR + SCR
      PRINT #IV, SM + "MINE" + SM + SO + SM + "LOGICALLY" + SM + "FOURIER"
      PRINT #IV, SM + "NAME" + SM + SDA + SM + SW + SM + SW
      PRINT #IV, SM + SM + SM + SD
      PRINT #IV, "NUMBER OF TERMS = " + SM + SM + SM;
      FOR I = NM TO NX
```

```
         PRINT #IV, SM; 2 ^ I;
      NEXT I
      PRINT #IV, SCR
      JJ = NX - NM + 3
      FOR I = 1 TO M
         PRINT #IV, SM + SMN(I);
         FOR J = 1 TO JJ
            PRINT #IV, SM;
            PRINT #IV, USING SFF; W(J, I);
         NEXT J
         PRINT #IV,
      NEXT I
      PRINT #IV, SCR + SCR
      PRINT #IV, "DETERMINATION COEF." + SM + SM;
      PRINT #IV, USING SFF; 1;
      PRINT #IV, SM;
      PRINT #IV, USING SFF; CD(0)
      PRINT #IV, "MODEL ERROR METRICS" + SM + SM;
      PRINT #IV, USING SFF; 0;
      FOR I = 0 TO NX - NM + 1
         PRINT #IV, SM;
         PRINT #IV, USING SFF; ER(I);
      NEXT I
      PRINT #IV,
      CLOSE IV
      END SUB

      SUB NOSEG (SK)
' A SUBROUTINE TO FLASH THE CAPTION "SEGMENT SK UNAVAILABLE"
' BEFORE A WARNING BUZZ
'     ARGUMENT:
'        SK     THE MISSING SEGMENT OR PROCESS DESCRIPTOR
'
      LINEBLANKER 24, 24
      COLOR 6
      LOCATE 24, 1: PRINT "SEGMENT ";
      COLOR 14: PRINT SK;
      COLOR 6: PRINT " UNAVAILABLE";
      NOTE "2B0506"
      SLEEP 2
      LINEBLANKER 24, 24
      END SUB

      SUB NOTE (S)
' A SUBROUTINE TO SOUND A NOTE UPON THE COMPUTER SPEAKER
'     ARGUMENT:
'        S     THE NOTE SPECIFIER STRING "IN$NL" e.g. "2B0506"
'                 I   THE OCTAVE NUMBER        ( 0-6 )
'                 N$  THE NOTE LETTER          ( ABCDEFG )
'                 N   THE NOTE NUMBER          ( 0-84 )
'                 L   THE LENGTH OF THE NOTE   ( 1-64 )
'
      I = VAL(MID$(S, 1, 1)): N$ = MID$(S, 2, 1)
      N = VAL(MID$(S, 3, 2)): L = VAL(MID$(S, 5, 2))
      PLAY "O" + STR$(I) + "N" + STR$(N) + "L" + STR$(L) + "X" + VARPTR$(N$)
      END SUB

      SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
' A SUBROUTINE TO OBTAIN A VARIABLE OF TYPE KT AT SCREEN POSITION IY,IX
'     ARGUMENTS:
'        KT     THE DATUM TYPE CHOICE
'                  1   SHORT INTEGER
'                  2   LONG  INTEGER
'                  3   DOUBLE PRECISION REAL
'                  4   STRING
'        I      THE SHORT INTEGER
'        L      THE LONG INTEGER TO BE OBTAINED  ( OPTION )
'        R      THE REAL         TO BE OBTAINED  ( OPTION )
'        S      THE STRING       TO BE OBTAINED  ( OPTION )
'        IY     THE STARTING SCREEN ROW
'        IX     THE STARTING SCREEN COLUMN
'        MK     THE PRINTING COLOR
'        SF     THE PROPER ( OR DEFAULT ) PRINTING FORMAT
'               ( "SAnn" TRUNCATES A STRING TO nn CHARACTERS )
'
      SDEL = CHR$(0) + CHR$(83)
      COLOR MK: LOCATE IY, IX: IO = I: LO = L: RO = R: SOL = S: L1 = LEN(SF)
      SELECT CASE KT
         CASE 1
            IF LEN(STR$(I)) > L1 THEN L1 = LEN(STR$(I))
         CASE 2
            IF LEN(STR$(L)) > L1 THEN L1 = LEN(STR$(L))
```

```
         CASE 3
            IF LEN(STR$(R)) > L1 THEN L1 = LEN(STR$(R))
         CASE 4
            L1 = LEN(S)
      END SELECT
      PRINT SPACE$(L1)
      DO
         LOCATE IY, IX: PRINT "."
         FOR II = 1 TO 10: NEXT II
         LOCATE IY, IX: PRINT " "
         SA = INKEY$
      LOOP UNTIL SA <> ""
      IA = ASC(SA)
      IF IA <> 13 THEN
         SCON = "": IT = IX: LOCATE IY, IX
            DO
               IF SA = SDEL THEN
                     IT = IT - 1
                     LOCATE IY, IT: PRINT SPACE$(1)
                     LOCATE IY, IT: SCON = LEFT$(SCON, LEN(SCON) - 1)
                  ELSE
                     SELECT CASE KT
                        CASE 1 TO 3
                           IF IA > 47 AND IA < 58 OR IA = 46 OR IA = 45 THEN
                                 SCON = SCON + SA
                                 PRINT SA;
                                 IT = IT + 1
                              ELSE
                                 IF IA <> 13 THEN NOTE "2B0506"
                           END IF
                        CASE 4
                           IF IA > 31 AND IA < 127 THEN
                                 SCON = SCON + SA
                                 PRINT SA;
                                 IT = IT + 1
                              ELSE
                                 IF IA <> 13 THEN NOTE "2B0506"
                           END IF
                     END SELECT
               END IF
               GETTER
            LOOP UNTIL IA = 13
            SELECT CASE KT
               CASE 1
                  I = INT(VAL(SCON) + .5)
               CASE 2
                  L = INT(VAL(SCON) + .5)
               CASE 3
                  R = VAL(SCON)
            END SELECT
         ELSE
            I = IO: L = LO: R = RO
      END IF
      L1 = LEN(SF): IF LEN(SCON) > L1 THEN L1 = LEN(SCON)
      LOCATE IY, IX: PRINT SPACE$(L1): LOCATE IY, IX
      SELECT CASE KT
         CASE 1
            PRINT USING SF; I
         CASE 2
            PRINT USING SF; L
         CASE 3
            PRINT USING SF; R
         CASE 4
            LOCATE IY, IX: PRINT SPACE$(LEN(SCON)): LOCATE IY, IX
            IF SOL <> "" AND SCON = "" THEN SCON = SOL
            IF LEFT$(SF, 2) = "SA" THEN
                  S = LEFT$(SCON, VAL(MID$(SF, 3)))
               ELSE
                  S = SCON
            END IF
            PRINT S
      END SELECT
      END SUB

      SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
' A SUBROUTINE TO PLACE A VARIABLE OF TYPE KT AT SCREEN POSITION IY, IX
' OR ALTERNATIVELY PLACE THE VARIABLE WITHIN A PADDED REPORT FILE
'     ARGUMENTS:
'        KT     THE DATUM TYPE CHOICE
'                  1   SHORT INTEGER
'                  2   LONG INTEGER
'                  3   DOUBLE PRECISION REAL
```

```
'                   4   STRING
'         I      THE SHORT INTEGER TO BE PRINTED  ( OPTION )
'         L      THE LONG  INTEGER TO BE PRINTED  ( OPTION )
'         R      THE REAL          TO BE PRINTED  ( OPTION )
'         S      THE STRING        TO BE PRINTED  ( OPTION )
'         IY     THE STARTING SCREEN ROW
'         IX     THE STARTING SCREEN COLUMN
'         IU     THE LOGICAL UNIT NUMBER
'                   1   PRINT TO THE SCREEN
'                   2   PRINT TO A REPORT FILE
'         MK     THE NOMINAL PRINTING COLOR
'         JF     THE LINE FEED SUPPRESSOR SWITCH
'                   0   FOLLOW WITH A LINE FEED
'                   1   DO NOT FOLLOW WITH A LINE FEED
'         SF     THE REQUIRED PRINTING FORMAT
'         IB     THE NUMBER OF FORWARD PADDING SPACES
'
      IF SF = "" THEN SF = "#########"
      COLOR MK
      SELECT CASE IU
         CASE 1
            LOCATE IY, IX
               SELECT CASE KT
                  CASE 1
                     PRINT USING SF; I
                  CASE 2
                     PRINT USING SF; L
                  CASE 3
                     PRINT USING SF; R
                  CASE 4
                     PRINT S
               END SELECT
         CASE 2
            IF IB > 0 THEN PRINT #IU, SPACE$(IB);
            IF JF = 1 THEN SCC = ";" ELSE SCC = CHR$(13) + CHR$(10)
               SELECT CASE KT
                  CASE 1
                     PRINT #IU, USING SF; I; SCC
                  CASE 2
                     PRINT #IU, USING SF; L; SCC
                  CASE 3
                     PRINT #IU, USING SF; R; SCC
                  CASE 4
                     PRINT #IU, S
               END SELECT
      IF JF = 0 THEN PRINT #LU,
      END SELECT
      END SUB
```