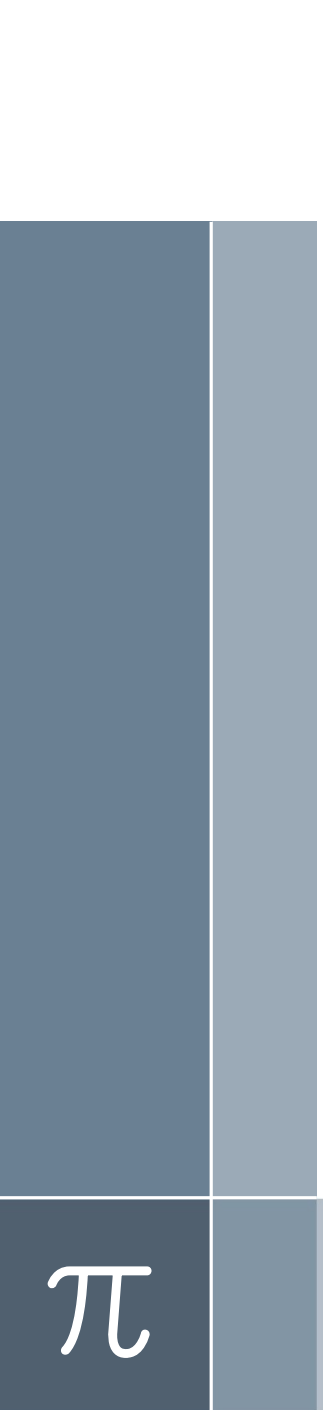
MTH686 - PROJECT

Alok Kumar
210101



Dataset and Description

1. Consider the data set you receive in your e-mail, which is of the form $(t, y(t))$. Try to fit the following model to the given data set:

$$y(t) = \alpha_1 + \alpha_2 e^{\beta t} + \epsilon(t).$$

Let us assume that $\{\epsilon(t)\}$ is a sequence of *i.i.d.* normal random variable with mean zero and finite variance. Analyze the data keeping the following points in mind, and write a report based on that.

1. Plot the data.
2. Plot the residual sum of squares as a function of β.
3. Find the least squares estimators of α_1, α_2 and β based on the Gauss-Newton method. Clearly mention which initial value you are taking? Does your result affect by the choice of the initial guess?
4. Find the estimate of σ^2.
5. Find the associated confidence intervals based on the Fisher information matrix.
6. Plot the residuals.
7. Test whether it satisfies the normality assumption or not?
8. Use any standard package (say R) and try to obtain the least squares estimators of the unknown parameters based on three dimensional optimization problem. Repeat all the above questions. Indicate which initial values you are taking and why?
9. What will you do to fit the following model

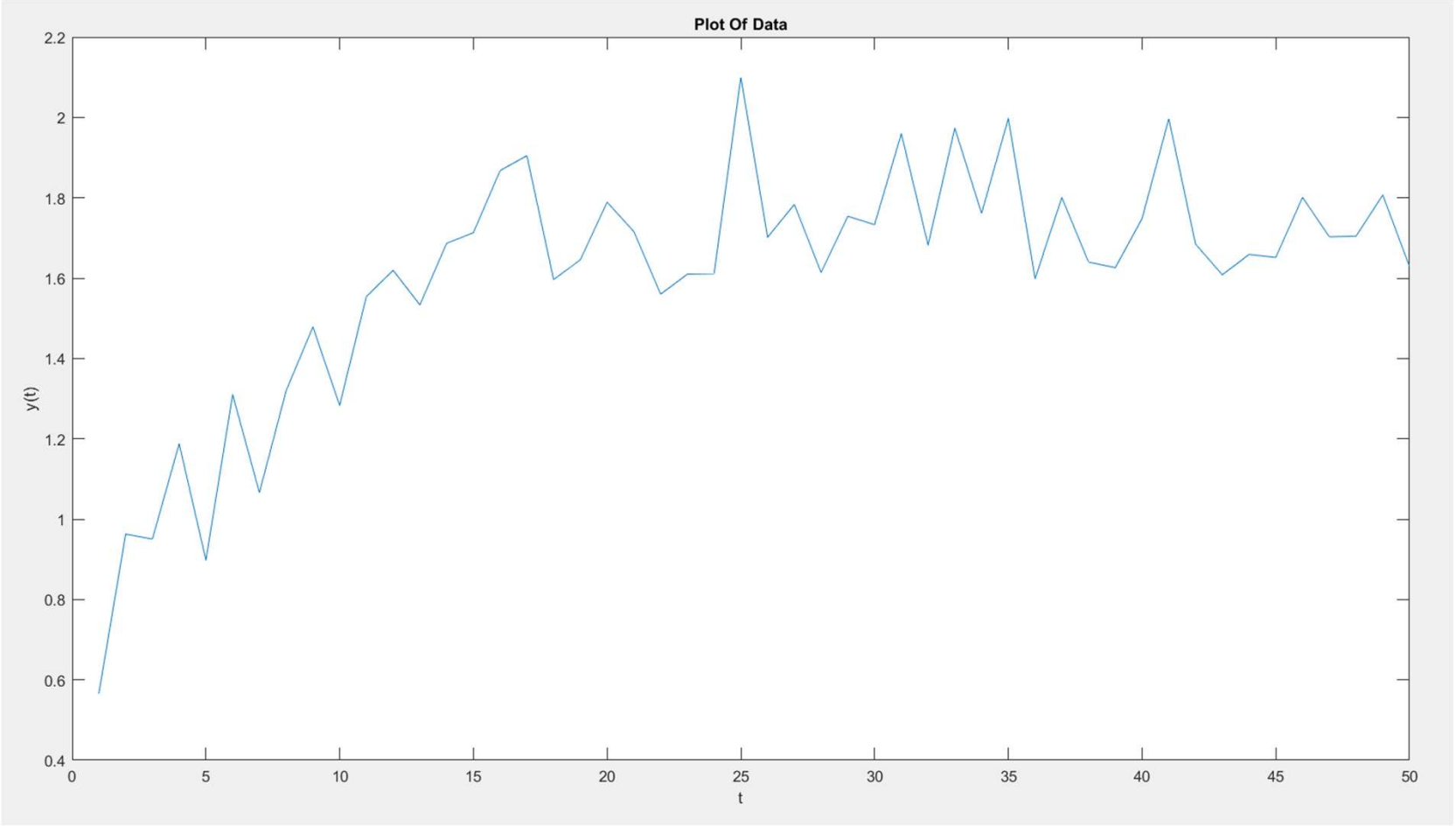
$$y(t) = \alpha_1 e^{\beta_1 t} + \alpha_2 e^{\beta_2 t} + \epsilon(t),$$

to the same data set? Let us assume that $\epsilon(t)$ satisfies the same assumption as above.

{DATA-27}

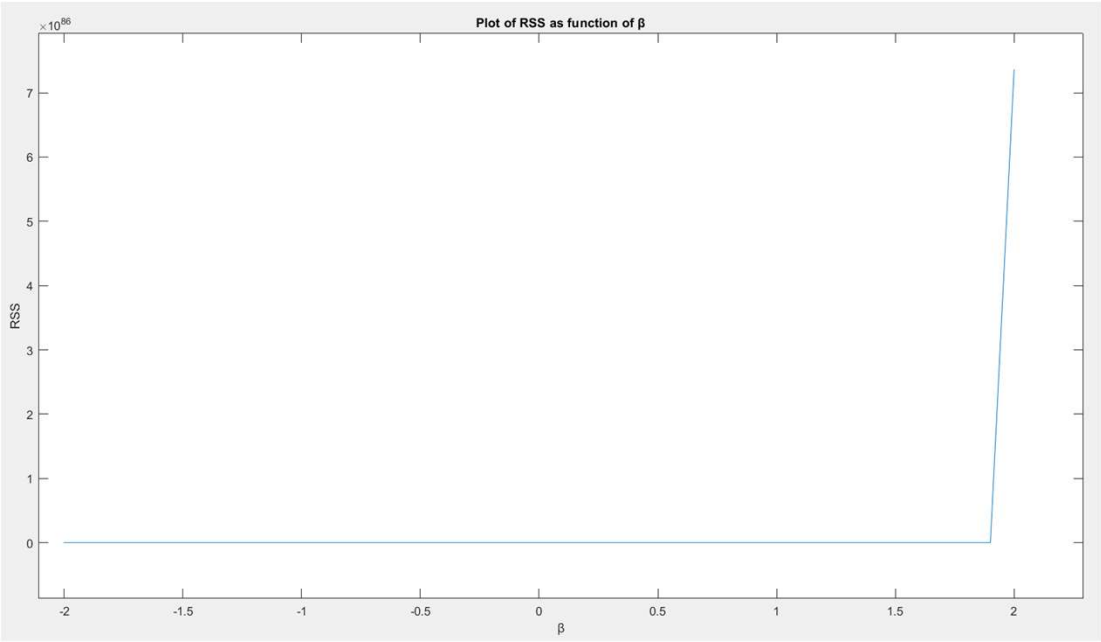
1	0.56715775	26	1.70169806
2	0.96274519	27	1.78355527
3	0.95040822	28	1.61399376
4	1.18756568	29	1.75389171
5	0.89818621	30	1.73321986
6	1.30984819	31	1.95956850
7	1.06628180	32	1.68216753
8	1.31979942	33	1.97339118
9	1.47856700	34	1.76096809
10	1.28328013	35	1.99731207
11	1.55431867	36	1.59826410
12	1.61940098	37	1.80070817
13	1.53331769	38	1.63988161
14	1.68663585	39	1.62601376
15	1.71317244	40	1.74806416
16	1.86777782	41	1.99668121
17	1.90480912	42	1.68470097
18	1.59638488	43	1.60826361
19	1.64562643	44	1.65886891
20	1.78914320	45	1.65140200
21	1.71568429	46	1.80100572
22	1.56004906	47	1.70292771
23	1.60996628	48	1.70441115
24	1.61110640	49	1.80686963
25	2.10022306	50	1.62870157

DATA PLOT


Plot Of Data
y(t)
t
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
5
10
15
20
25
30
35
40
45
50

RSS as a function of β

Here, We assumed that **α1=0.1 & α2=1** underlying the condition for **β** that it exists between **(-2,2)**. Thus, The plot will help us to tell the nature of RSS as a function of **β** and that the graph will be **NOT SCALED.**



CODE USED

```
>> betas = -2:0.1:2;
alpha1 = 0.1;
alpha2 = 1;
RSS = zeros(size(betas));
for i = 1:length(betas)
    y_predicted = alpha1 + alpha2 * exp(betas(i) * t);
    residuals = y - y_predicted;
    RSS(i) = sum(residuals.^2);
end
plot(betas, RSS);
```

LSE of α1, α2 and β using G-N Method

Before Moving ahead to Gauss Newton Method, let us hop into fitting the equation to a linear model by taking **log(y(t)) vs t** as a model and then find out the initial parameter guess by the help of following code :

CODE EXECUTED →

```
>> initial_alpha1 = mean(y);
initial_alpha2 = (y(end) - y(1)) / (t(end) - t(1));
beta_initial_guess = polyfit(t, log(y), 1);
initial_beta = beta_initial_guess(1);
```

INITIAL PARAMETERS →

```
Initial guess for alpha1: 1.6030
Initial guess for alpha2: 0.0217
Initial guess for beta: 0.0102
```

Yes, Choice of Initial Guess of parameters will surely affect the parameters estimated and hence, Initial Guess should be appropriate and should done by proper methods.

LSE of α1, α2 and β using G-N Method

Now, that we know initial guesses, let us **iterate** the model with the help of following **code** underlined :

```
>> model = @(params, t) params(1) + params(2) * exp(params(3) * t);
rss = @(params, t, y) sum((y - model(params, t)).^2);
initial_guess = [initial_alpha1, initial_alpha2, initial_beta];
options = optimset('Display', 'iter', 'TolFun', 1e-6, 'TolX', 1e-6);
estimated_params = fminsearch(@(params) rss(params, t, y), initial_guess, options);
estimated_alpha1 = estimated_params(1);
estimated_alpha2 = estimated_params(2);
estimated_beta = estimated_params(3);
```

After **373** Successful Iterations, The Compiler concluded that **min.RSS = 2.661554**, and so estimated our parameters.

```
Current function value: 2.661554

Estimated alpha1: -30.8341
Estimated alpha2: 32.0997
Estimated beta: 0.0004
```

Estimate of σ^2

Now, that we have estimated the parameters we can also estimate the variance with the help of following code:

```
>> y_predicted = estimated_alpha1 + estimated_alpha2 * exp(estimated_beta * t);
    residuals = y - y_predicted;
    RSS = sum(residuals.^2);
n = length(y);
p = 3;
sigma_squared = RSS / (n - p);
fprintf('Sigma Squared: %.4f\n', sigma_squared);
```

Upon executing the code we get our σ^2 estimated value as given below ;

```
Sigma Squared: 0.0566
```

Associated Confidence Interval

Let us first find the **Fischer Information Matrix** with the help of **Jacobian Matrix** using following code :

```
>> J = [ones(length(t), 1), exp(estimated_beta * t), estimated_alpha2 * t .* exp(estimated_alpha2 * t)];
FisherInformation = (J.' * J);
```

Now that we found the F.I matrix let us find the **covariance matrix** and **sqrt** of which **diagonal elements** will give us **standard errors** associated :

```
covarianceMatrix = inv(FisherInformation);
standardErrors = sqrt(diag(covarianceMatrix));
```

Now. Let us calculate **Z-score** for 95% Confidence Interval :

```
confidenceLevel = 0.95;
alpha = 1 - confidenceLevel;
z = norminv(1 - alpha/2);
```

And using these we can calculate **confidence Intervals** for parameters using following code :

```
lowerBounds = estimatedParameters - z * standardErrors;
upperBounds = estimatedParameters + z * standardErrors;

confidenceIntervals = [lowerBounds, upperBounds];
```

Associated Confidence Interval

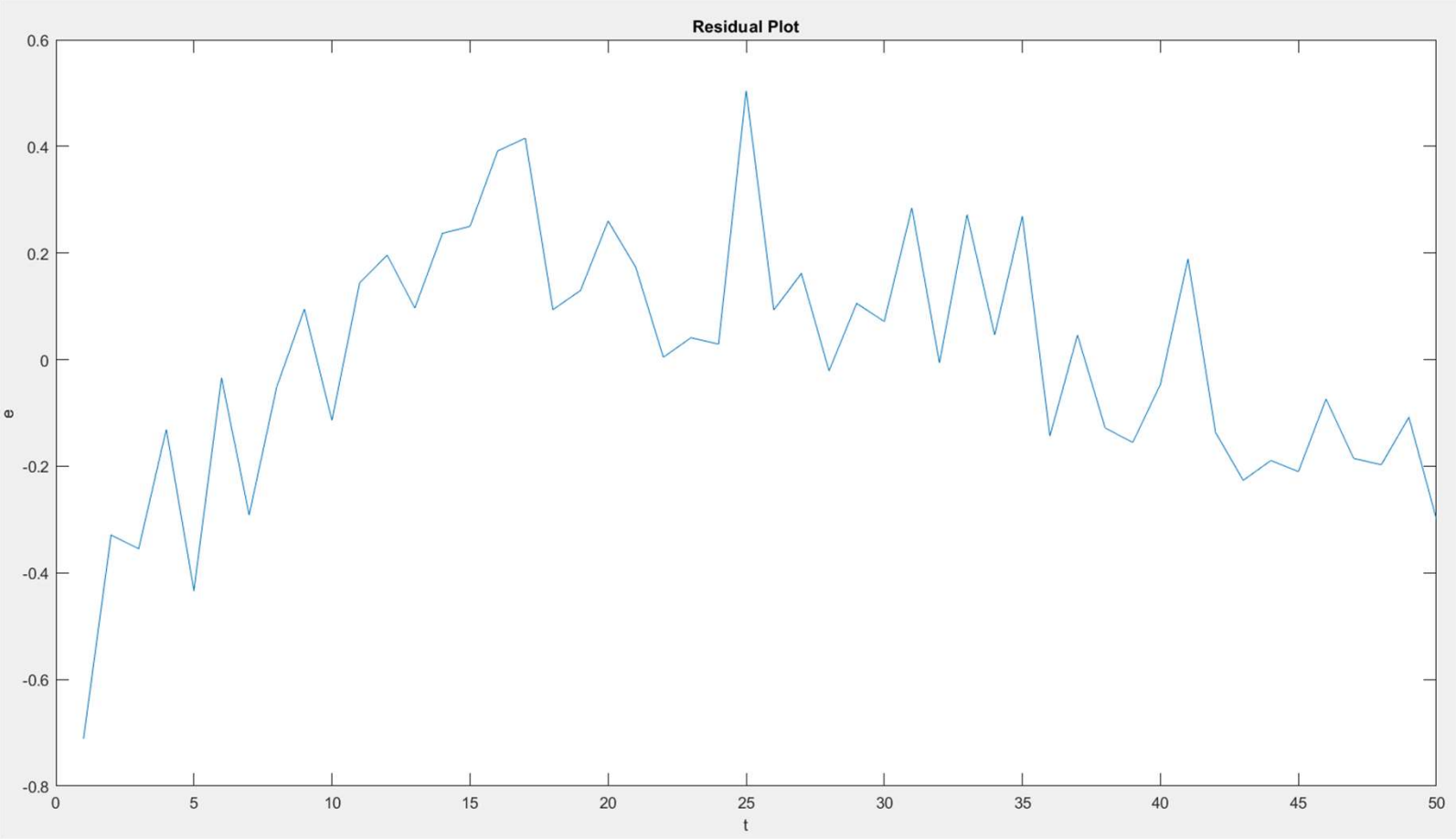
Upon solving we get the following results;

$\alpha 1 = (-77.8299, 16.1617)$

$\alpha 2 = (-14.8961, 79.0956)$

$\beta = (-46.9954, 46.5074)$

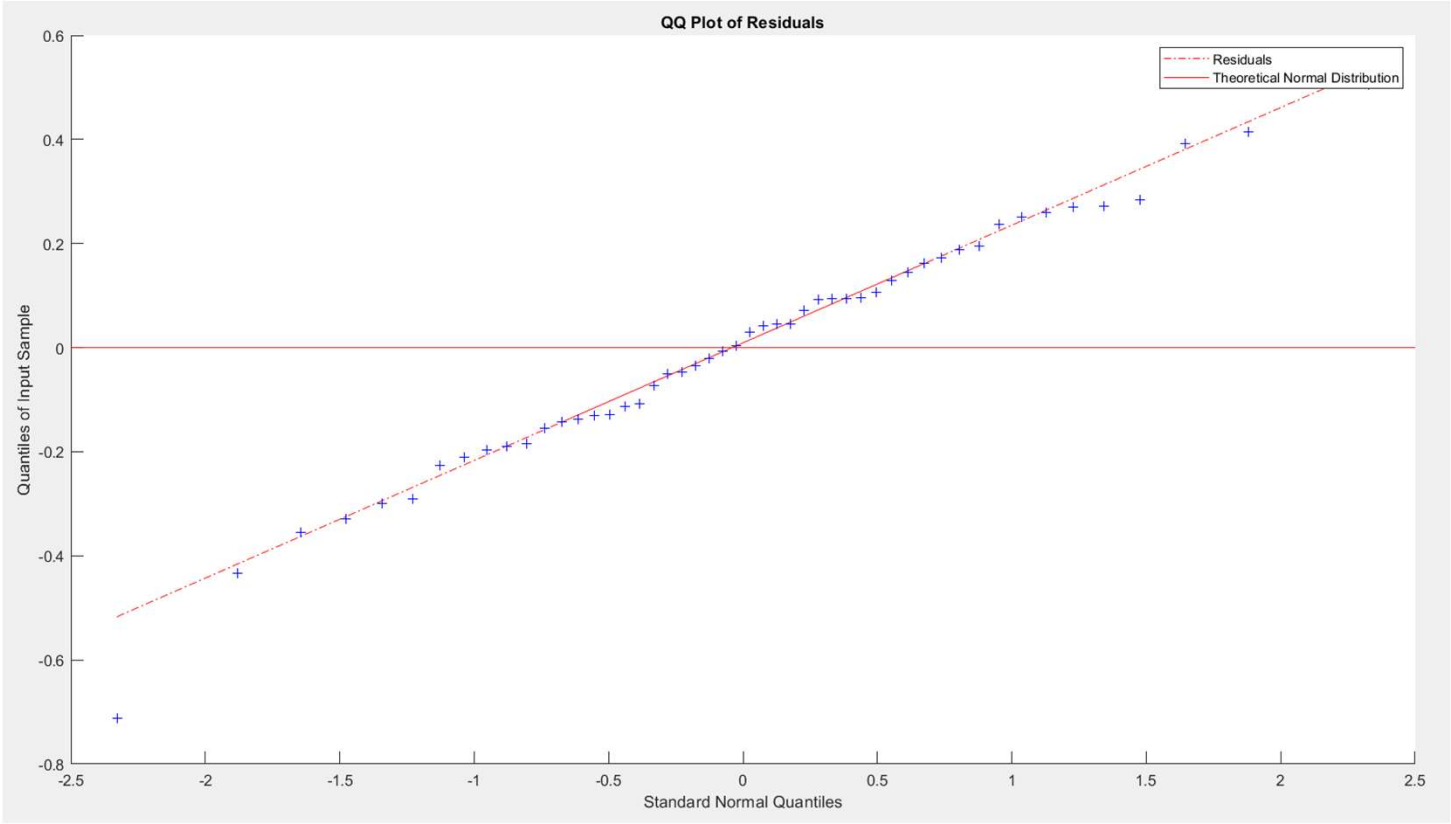
Residual Plot



Normality Assumption Test

We are going to use the Q-Q Plot to determine Normality assumption test :

As, Marked Points are close to the diagonal line thus we conclude that It follows normal distribution and passes the N-A Test



Using R

The DATA PLOT and the RSS as a function of β, Normality assumption will be the same irrespective of what Language we tend to choose whether R, Pandas or say numPy.

The only major change will occur in initializing the value and thus will affect our Estimated Parameters and respective related problems. Let us solve them with help of R and see what changes we tend to see in our conclusions;

Here, we will initialise the parameters by following code :

```
linear_model <- lm(y ~ t)
initial_alpha1 <- coef(linear_model)[1]
initial_alpha2 <- coef(linear_model)[2]

log_y <- log(y)
log_linear_model <- lm(log_y ~ t)
initial_beta <- coef(log_linear_model)[2]
```

Which will give us the initial parameters as follows ;

Alpha1 = 1.5379

Alpha2 = 0.0237

Beta = 0.0115

Clearly our values are very close to our previous initials, so we expect slight change in further proceedings.

Using R

.Now upon solving, using R codes we get the following final results of the data given,

Estimated alpha_1 = -31.2357

Estimated alpha_2 = 30.9980

Estimated beta = 0.0004

Estimated sigma _squared = 0.0593

95% CONFIDENCE INTERVALS :

alpha_1 = (-73.8354, 17.8963)

alpha_2 = (-15.5074, 77.2914)

Beta = (-46.9764, -43.2789)

Every data concluded here has been evaluated using **StatR.**

What if $y(t) = \alpha 1\ e^{\beta 1t} + \alpha 2\ e^{\beta 2t} + e(t)$?

.If we are changing the model from $y(t) = \alpha 1 + \alpha 2\ e^{\beta 2t} + e(t)$ to $y(t) = \alpha 1\ e^{\beta 1t} + \alpha 2\ e^{\beta 2t} + e(t)$,

then, every procedure we did previously will be same except for altering the model in MATLAB code, i.e, replacing,

```
>> model = @(params,t) params(1) + params(2) * exp(params(3) * t);
```

INTO

```
>> model = @(params,t) params(1) * exp(params(2) * t) + params(3) * exp(params(4) * t);
```

and upon doing so, we just need to proceed with same procedure that we have done to obtain our answers and that's all.

Thanking You ☺