INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Department of Chemical Engineering

CH5115: Parameter and State Estimation

Assignment 4 Solutions

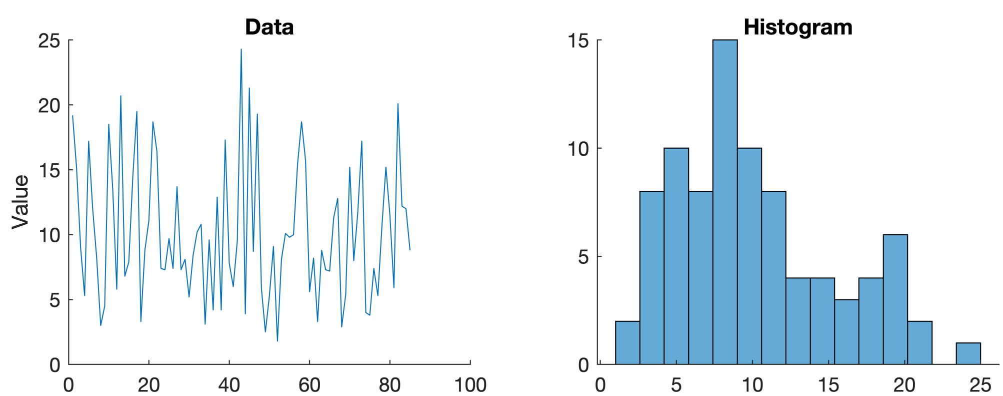
Question 1

Part a

```
load("qdist_data.mat")
```

Data and Histogram

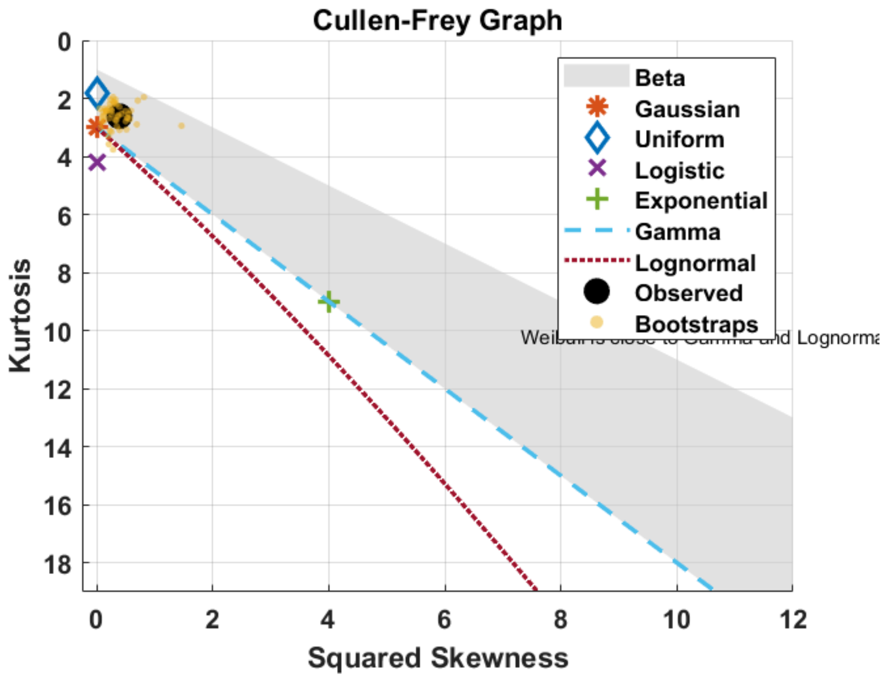
```
figure("Position", [100, 100, 600, 200])
subplot(1, 2, 1)
plot(Ydata);
box OFF
title('Data')
ylabel('Value')
subplot(1, 2, 2)
histogram(Ydata, 15)
box OFF
title('Histogram')
```



The data is positive (time to publish) and continuous. The histogram is unimodal and not symmetric. Possible distributions are exponential, chi-squared, f-distribution, gamma, and log-normal.

Kurtosis-Skewness Based Analysis

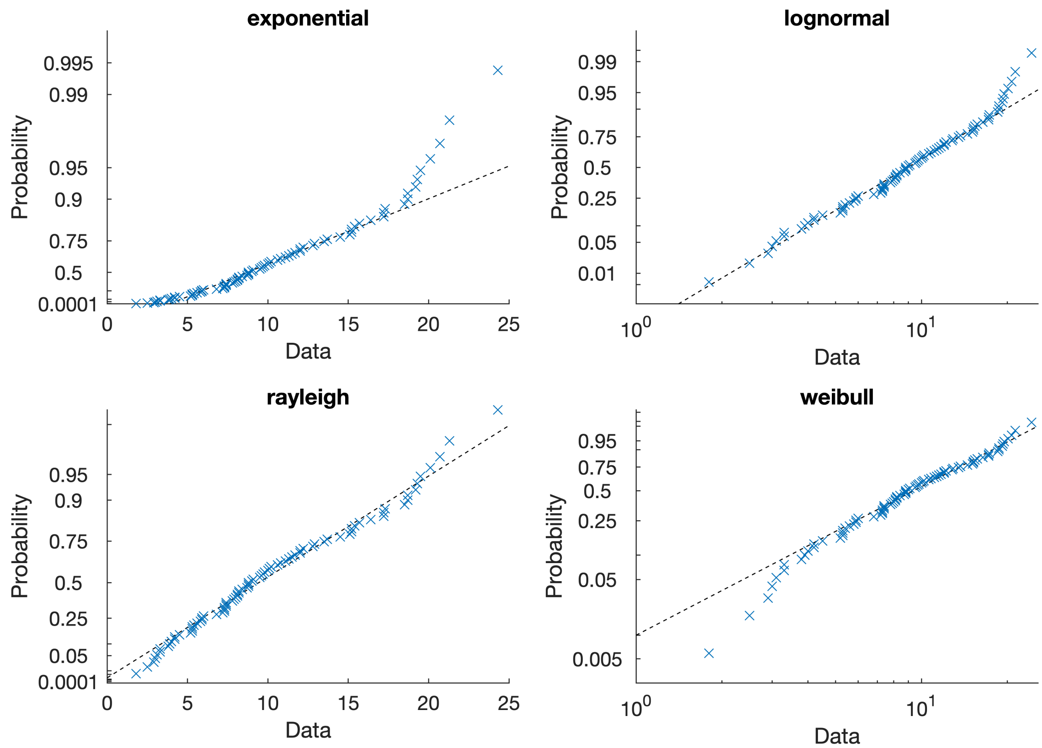
We use the Cullen-Frey graph to compare the estimated kurtosis and skewness with standard distributions.



Based on kurtosis-skew values, none of the compared distributions can be clearly selected. The Beta and Gamma family of distributions appear promising.

Probability (Q-Q) Plots

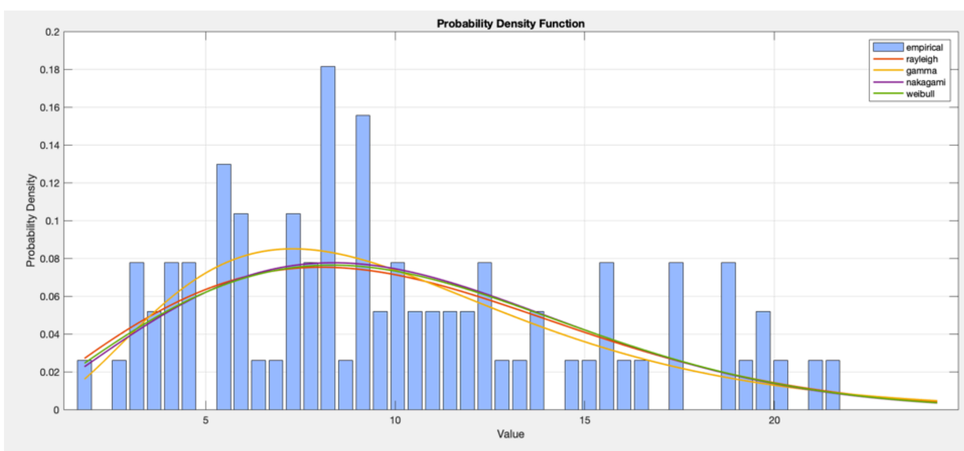
```
distributions = {'exponential', 'lognormal', 'rayleigh', 'weibull'};
figure("Position", [100, 100, 600, 400])
box OFF
for i = 1:length(distributions)
    subplot(2, 2, i)
    probplot(distributions{i}, Ydata)
    title(distributions{i})
end
```



The data seems to closely follow the **Rayleigh distribution** (special case of Weibull with shape parameter=2). We use the distribution fitting toolbox to further confirm this.

GUI Fitting Tool

```
writematrix(Ydata,'data.txt')
% Main_FitDistribution_GUI
```



	Distribution 1	Distribution 2	Distribution 3	Distribution 4
Dist Name	rayleigh	gamma	nakagami	weibull
NLogL	255.367	254.7378	255.2024	255.2957
BIC	515.1767	518.3608	519.29	519.4768
AIC	512.734	513.4755	514.4047	514.5915
AICc	512.7822	513.6219	514.5511	514.7378
Params Nemas	B,	a, b,	mu, omega,	A, B,
Params Values	8.0573	3.5767 2.8311	1.08208 129.8399	11.4721 2.06503

From the above analysis, the **Rayleigh distribution** has the lowest AIC and BIC and is selected.

Hypothesis Test

We test this using the Anderson-Darling Test (for Weibull)

```
h = adtest(Ydata, "Distribution" , "weibull");
if h==0
    disp('Failed to Reject Null ==> Assumed Distribution is not Inco
else
    disp('Null Rejected ==> Assumed Distribution is Incorrect')
end
```

```
Failed to Reject Null ==> Assumed Distribution is not Incorrect
```

Parameter Estimation

We estimate the parameter and its confidence interval for the selected distribution

```
[phat, pci] = raylfit(Ydata)
```

```
phat = 8.0573
pci = 2×1
    7.2843
    9.0153
```

Interpretation

The Rayleigh distribution can model the lifetime of an object based on its age. This can be interpreted as the time to complete the work (and publish) based on the amount of time spent on the work till that point.

Part b

i) Systemic Errors: They change the scale and location of the distribution. The distribution might not change, but the estimated parameters will have a bias. Systemic errors are difficult to rectify without prior knowledge.

ii) Random Errors: The effect of random errors depends to a large extent on the signal-to-noise ratio. For low SNRs, the noise distribution will dominate over the data distribution. Generally, random errors will affect the convergence of the estimates to the true solution.

Question 2

We have

$$f(y;\theta) = \begin{cases} \frac{2y}{\theta^2} & 0 \le y \le \theta \\ 0 & \text{Otherwise} \end{cases}$$

Part a

We compute the log-likelihood (assuming the sample is iid) as

$$N \log 2 - 2N \log \theta + \sum_{y_i \in [0,\theta]} \log y_i$$

The log-likelihood does not have a clear maximum. We want the value of θ to be as small as possible (to reduce the 2nd term) but to be greater than all y_i (to increase the 3rd term). Thus,

$$\hat{\theta}_{MLE} = \max(y_i)$$

We need to find $E\left[\hat{\theta}_{MLE}\right]$. For this, we need the pdf of $\hat{\theta}_{\text{MLE}}$.

Let the CDF of $\hat{\theta}_{MLE} = \max(y_i)$ be $F_{max(y_i)}(y)$

$$\begin{aligned} F_{\max(\text{yi})}(y) &= P(\max(y_i) \le y) \\ &= P(y_i \le y \,\forall\, i = 1, \ldots N) \\ &= \prod_{i=1 \ldots N} P(y_i \le y) \\ &= \left(\frac{y^2}{\theta^2}\right)^N \end{aligned}$$

We now compute the pdf using

$$\begin{aligned} f(y) &= \frac{\text{dF}}{\text{dy}} \\ &= 2N \frac{y^{2N-1}}{\theta^{2N}} \end{aligned}$$

$$\begin{aligned} E\left[\hat{\theta}_{\text{MLE}}\right] &= \int_0^\theta 2N \frac{y^{2N}}{\theta^{2N}} \\ &= \frac{2N}{2N+1}\theta \end{aligned}$$

$E\left[\widehat{\theta}_{MLE}\right] \neq \theta$.

Therefore it is **biased.**

The unbiased modification is

$$\widehat{\theta}_{ub} = \frac{2N+1}{2N}\widehat{\theta}_{MLE}$$

Part b

At median, CDF=0.5

$$\frac{y^2}{\theta^2} = \frac{1}{2}$$

$$y_{\text{med}} = \frac{\theta}{\sqrt{2}}$$

By functional invariance, the MLE estimate of the median(M) is

$$\widehat{M}_{\text{MLE}} = \frac{\widehat{\theta}_{\text{MLE}}}{\sqrt{2}} = \frac{\max(y_i)}{\sqrt{2}}$$

Part c

The estimate is asymptotically unbiased.

$$\lim_{N\to\infty} \frac{2N}{2N+1}\frac{\theta}{\sqrt{2}} = \lim_{N\to\infty} \frac{1}{1+\frac{1}{2N}}\frac{\theta}{\sqrt{2}} = \frac{\theta}{\sqrt{2}}$$

We find the variance of the MLE estimate of θ

$$E\left[\left(\widehat{\theta}_{\text{MLE}}\right)^2\right] = \int_0^\theta 2N\frac{y^{2N+1}}{\theta^{2N}}$$

$$= \frac{2N}{2N+2}\theta^2$$

$$\text{var}\left[\widehat{\theta}_{\text{MLE}}\right] = E\left[\left(\widehat{\theta}_{\text{MLE}}\right)^2\right] - \left(E\left[\widehat{\theta}_{\text{MLE}}\right]\right)^2$$

$$= \theta^2\left(\frac{1}{1+N^{-1}} - \frac{1}{1+N^{-1}+(2N)^{-2}}\right)$$

Since, $\widehat{M}_{\text{MLE}} = \dfrac{\widehat{\theta}_{\text{MLE}}}{\sqrt{2}}$

$$\mathrm{var}\left[\widehat{M}_{\mathrm{MLE}}\right] = \frac{\mathrm{var}\left(\widehat{\theta}_{\mathrm{MLE}}\right)}{2} = \frac{\theta^2}{2}\left(\frac{1}{1+N^{-1}} - \frac{1}{1+N^{-1}+(2N)^{-2}}\right)$$

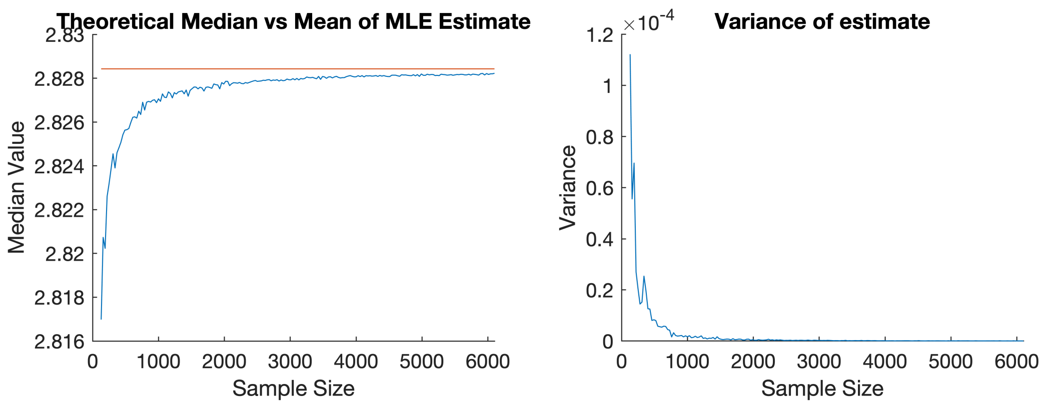
$$\lim_{N\to\infty} \mathrm{var}\left[\widehat{M}_{\mathrm{MLE}}\right] = 0$$

Therefore the estimate shows mean square converge and is consistent.

Part d

```
% Set value of theta
theta = 4;

% Theoretical Median
median_th = theta/sqrt(2);

pdf = makedist('Triangular', 'a', 0, 'b', theta, 'c', theta);

% 100 realizations of increasing sample size
median_mean = [];
median_var = [];
R = 200;
Ns = [];
for r=1:R
    N = 100 + 30*r;
    Ns = [Ns, N];
    K = 100;
    median_est = [];
    for k=1:K
        % Sample from distribution
        yk = pdf.random(N, 1);
        % Median estimate
        median_est = [median_est, max(yk)/sqrt(2)];
    end
    % Sample mean of median estimates
    median_mean = [median_mean, mean(median_est)];


    % Sample variance of median estimates
    median_var = [median_var, var(median_est)];
end

figure("Position", [100, 100, 600, 200])
```

```
subplot(1, 2, 1)
plot(Ns, median_mean)
hold on
plot([Ns(1), Ns(end)], [median_th, median_th])
title('Theoretical Median vs Mean of MLE Estimate')
xlabel('Sample Size')
ylabel('Median Value')
box OFF
hold off

subplot(1, 2, 2)
plot(Ns, median_var)
box OFF
title('Variance of estimate')
xlabel('Sample Size')
ylabel('Variance')
```



The variance approaches zero, and the mean approaches the truth as we increase the sample size. Thus, we verify that the estimate is consistent.

However, the distribution of the estimate is not symmetric. The support depends on the true parameter. Hence it is **not asymptotically Gaussian.** We verify this with the AD test.

```
% Set value of theta
theta = 4;

% Theoretical Median
median_th = theta/sqrt(2);

pdf = makedist('Triangular', 'a', 0, 'b', theta, 'c', theta);
N = 500;
```

```
median_est = [];

for r=1:R
    % Sample from distribution
    yk = pdf.random(N, 1);
    % Median estimate
    median_est = [median_est, max(yk)/sqrt(2)];
end

h = adtest(median_est);
```

```
Warning: P is less than the smallest tabulated value, returning
 0.0005.
```

```
if h==0
    disp('Failed to Reject Null ==> Normal Distribution is not Incor
else
    disp('Null Rejected ==> Normal Distribution is Incorrect')
end
```

```
Null Rejected ==> Normal Distribution is Incorrect
```

Question 3

Part a

```
load('stack_loss.mat')
X = [yvec, psi1vec, psi2vec, psi3vec];
correl = corr(X);
disp(['Correlation with regressors = ', num2str(correl(1, 2:end))])
```

```
Correlation with regressors = 0.91966      0.8755      0.39983
```

Regressors 1 and 2 are strongly correlated with the response variable while regressor 3 is not, It might not be useful in the linear model.

Part b and Part c

```
linMod1 = fitlm([psi1vec, psi2vec, psi3vec], yvec)
```

```
linMod1 =
Linear regression model:
    y ~ 1 + x1 + x2 + x3

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-39.92	11.896	-3.3557	0.0037503
x1	0.71564	0.13486	5.3066	5.799e-05
x2	1.2953	0.36802	3.5196	0.0026301
x3	-0.15212	0.15629	-0.97331	0.34405

```
Number of observations: 21, Error degrees of freedom: 17
Root Mean Squared Error: 3.24
R-squared: 0.914,  Adjusted R-Squared: 0.898
F-statistic vs. constant model: 59.9, p-value = 3.02e-09
```

The p-value indicates that the 3rd regressor is not useful. We modify the model and estimate again.

```
linMod2 = fitlm([psi1vec, psi2vec], yvec)
```

```
linMod2 =
Linear regression model:
    y ~ 1 + x1 + x2

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-50.359	5.1383	-9.8006	1.2165e-08
x1	0.67115	0.12669	5.2976	4.898e-05
x2	1.2954	0.36749	3.5249	0.0024191

```
Number of observations: 21, Error degrees of freedom: 18
Root Mean Squared Error: 3.24
R-squared: 0.909,  Adjusted R-Squared: 0.899
F-statistic vs. constant model: 89.6, p-value = 4.38e-10
```

The R^2 and adjusted R^2 are reported. The test for regression (F-test between intercept-only model vs regression model) rejects the null hypothesis that the intercept-only model is sufficient. The p-values of the estimates of the regression coefficients indicate that the estimates are not close to zero and hence useful. We test the normality assumption of the residuals below.

```
ek = linMod2.Residuals.Raw;
h = adtest(ek);
if h==0
    disp('Failed to Reject Null ==> Normal Distribution is not Incor
else
    disp('Null Rejected ==> Normal Distribution is Incorrect')
end
```

```
Failed to Reject Null ==> Normal Distribution is not Incorrect
```

Part d

```
Xnew = [80, 25];
[yPred, CI] = predict(linMod2, Xnew, 'Prediction', 'curve')
```

```
yPred = 35.7173
CI = 1×2
   31.9927   39.4419
```

Part e

```
[yPred, PI] = predict(linMod2, Xnew, 'Prediction', 'observation');
PI
```

```
PI = 1×2
   27.9605   43.4741
```

We see that the prediction intervals are larger than the confidence intervals as they account for the noise of the model too.

Question 4

Part a

```
load('engine_thrust.mat')

linMod1 = fitlm(Phi, y)
```

```
linMod1 =
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6
```

```
Estimated Coefficients:
                   Estimate        SE         tStat       pValue
                   _________    ________    ________    __________

    (Intercept)        -4738      2444.7      -1.938      0.061213
    x1                1.1185     0.28647      3.9045    0.00044089
    x2             -0.030184    0.038234    -0.78946       0.43548
    x3               0.23062     0.11803      1.9539      0.059231
    x4                3.8495      2.6862      1.4331       0.16125
    x5               0.82186     0.35075      2.3432      0.025298
    x6               -16.946      2.6201     -6.4679    2.4504e-07


Number of observations: 40, Error degrees of freedom: 33
Root Mean Squared Error: 26.5
R-squared: 0.998,  Adjusted R-Squared: 0.997
F-statistic vs. constant model: 2.35e+03, p-value = 6.07e-42
```

```
[~, ~, ic] = aicbic(linMod1.LogLikelihood, linMod1.NumEstimatedCoeff
```

```
ic = struct with fields:
      aic: 382.0183
      bic: 368.0183
     aicc: 366.0183
     caic: 375.0183
      hqc: -Inf
```

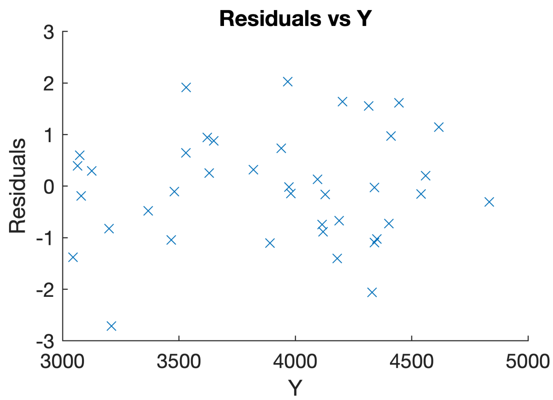
The t-test on parameters idicates that not all terms are useful in the model. The f-test confirms that the model is better than constant.

Residual Analysis

```
ek = linMod1.Residuals.Standardized;
[ySort, sortInd] = sort(y);
```

Residuals vs Response Variable

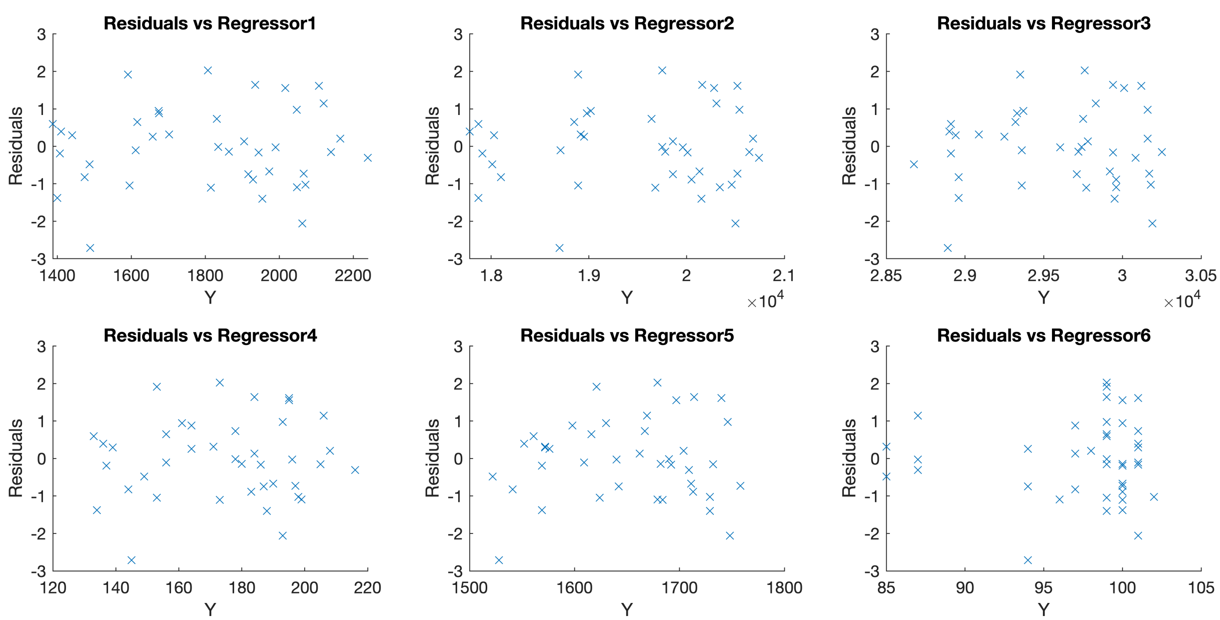
```
figure("Position", [100, 100, 300, 200])
plot(ySort, ek(sortInd), 'x')
box OFF
title('Residuals vs Y')
ylabel('Residuals')
xlabel('Y')
```



We don't see any obvious patterns. The model does not have an underfit.

Residuals vs Regressors

```
figure("Position", [0, 0, 900, 400])
for i=1:size(Phi, 2)
    subplot(2, 3, i)
    [PhiSort, sortInd] = sort(Phi(:, i));
    plot(PhiSort, ek(sortInd), 'x')
    box OFF
    title(strcat('Residuals vs Regressor ', num2str(i)))
    ylabel('Residuals')
    xlabel('Y')
end
```



We assume the residuals to be uncorrelated with the residuals. However, we find some trends in Regressor 5 which might make this assumption invalid. This indicates an underfit and we might have to add non-linear terms to overcome this.

Uncorrelated Regressors

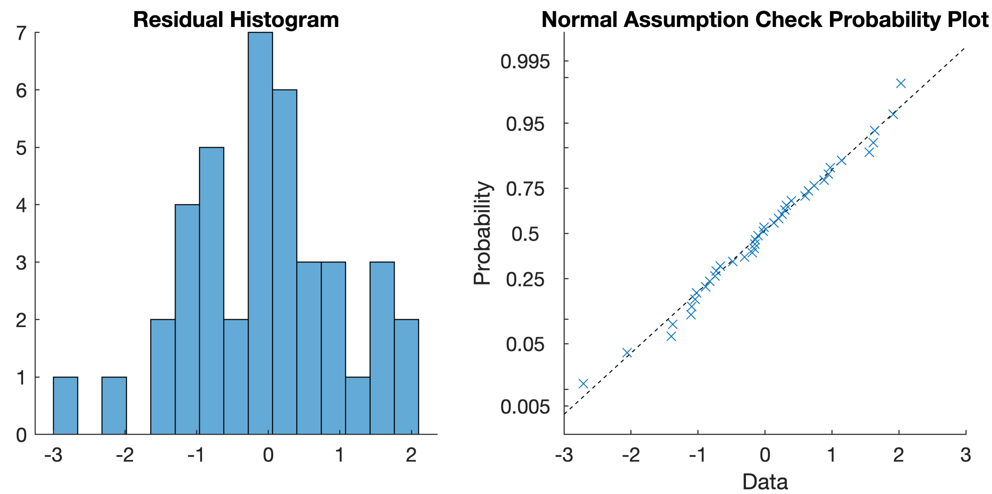
```
corr(Phi)
```

```
ans = 6×6
    1.0000    0.9852    0.9504    0.9940    0.8940   -0.0727
    0.9852    1.0000    0.9723    0.9719    0.9267    0.0171
    0.9504    0.9723    1.0000    0.9203    0.9758    0.2165
    0.9940    0.9719    0.9203    1.0000    0.8519   -0.1534
    0.8940    0.9267    0.9758    0.8519    1.0000    0.3018
   -0.0727    0.0171    0.2165   -0.1534    0.3018    1.0000
```

We assume that the data does not have multicollinearity. But we find that some of the regressors are strongly correlated with each other. This assumption is clearly invalid. We need to either perform feature selection or a dimensionality reduction technique like Principal Component Analysis to overcome multicollinearity in the regressors.

Normal Residuals

```
figure("Position", [100, 100, 600, 250])
subplot(1, 2, 1)
histogram(ek, 15)
box OFF
title('Residual Histogram')

subplot(1, 2, 2)
probplot(ek)
box OFF
title('Normal Assumption Check Probability Plot')
```




```
h = adtest(ek);
if h==0
    disp('Failed to Reject Null ==> Normal Distribution is not Incor
else
    disp('Null Rejected ==> Normal Distribution is Incorrect')
end
```

```
Failed to Reject Null ==> Normal Distribution is not Incorrect
```

We assume the residuals to be iid from a normal distribution. We see that the normality assumption is valid based on the Q-Q plot and the AD test.

Uncorrelated Residuals

```
h = lbqtest(ek);
if h==0
    disp('Not enough evidence to believe that the residuals are seri
else
    disp('Null Rejected ==> Residuals are serially correlated')
end
```

```
Not enough evidence to believe that the residuals are serially corre
```

We assume that the residuals are uncorrelated (iid and uncorrelated mean the same for a normal random variable). The lbqtest fails to reject the null that the residuals are uncorrelated. Hence this assumption holds.

The assumption that the regressors are noise-free can not be tested and depends on the nature of the data collection procedure.

Part b

Based on the t-test we eliminate regressors 2, 3, 4 and the intercept.

```
selected = [1, 5, 6];
linMod2 = fitlm(Phi, y, 'Intercept', false, 'PredictorVars', selecte
```

```
linMod2 = 
Linear regression model:
    y ~ x1 + x5 + x6

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
x1	1.6101	0.043065	37.388	5.3825e-31
x5	1.5025	0.13287	11.308	1.4513e-13
x6	-15.318	1.5483	-9.8936	6.1356e-12

```
Number of observations: 40, Error degrees of freedom: 37
Root Mean Squared Error: 28.8
```

```
linMod2.Rsquared
```

```
ans = struct with fields:
    Ordinary: 0.9969
    Adjusted: 0.9967
```

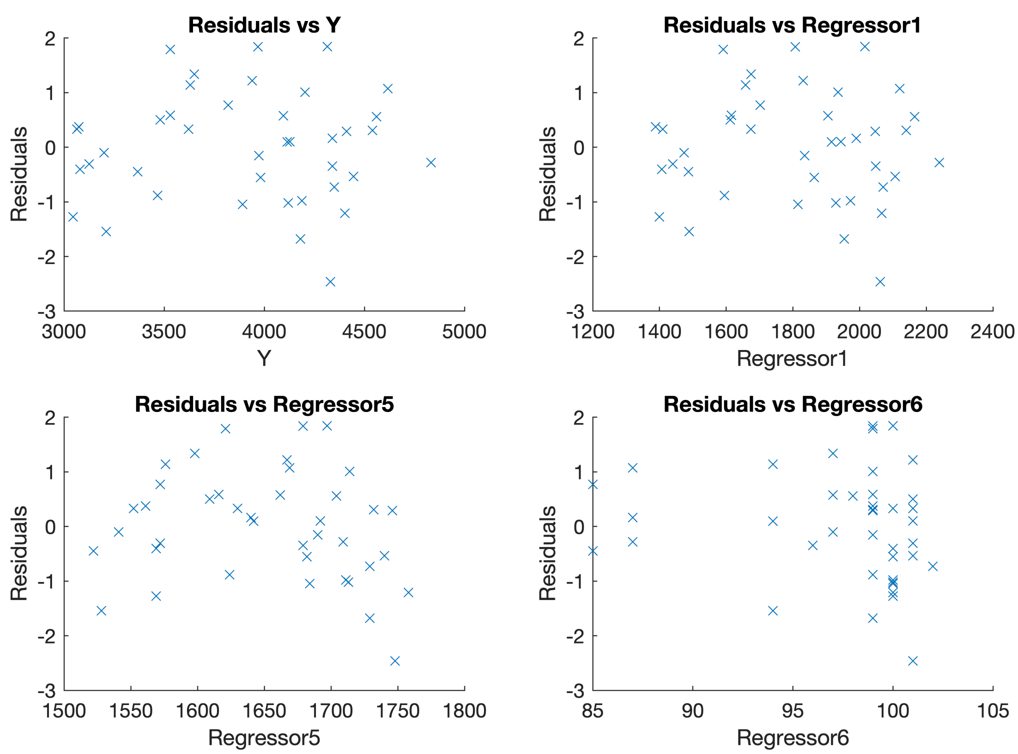
```
[~, ~, ic] = aicbic(linMod2.LogLikelihood, linMod2.NumEstimatedCoeff
```

```
ic = struct with fields:
     aic: 385.2788
     bic: 379.2788
    aicc: 377.2788
    caic: 382.2788
     hqc: -Inf
```

This model has lower R^2 values and higher AIC, BIC values than the model in Part a.

```
ek = linMod2.Residuals.Standardized;
[ySort, sortInd] = sort(y);
```

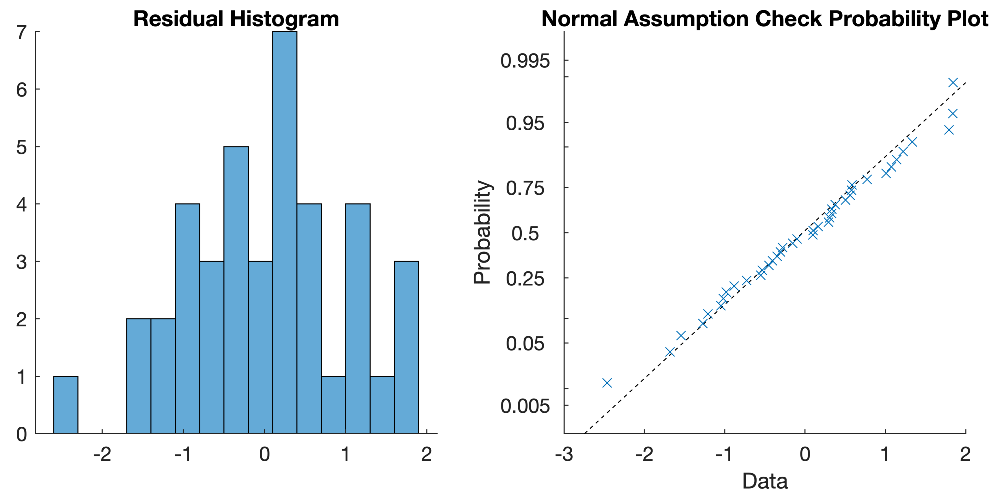
```
figure("Position", [100, 100, 600, 400])
subplot(2, 2, 1)
plot(ySort, ek(sortInd), 'x')
box OFF
title('Residuals vs Y')
ylabel('Residuals')
xlabel('Y')
for i=1:length(selected)
    subplot(2, 2, i+1)
    [PhiSort, sortInd] = sort(Phi(:, selected(i)));
    plot(PhiSort, ek(sortInd), 'x')
    box OFF
    title(strcat('Residuals vs Regressor ', num2str(selected(i))))
    ylabel('Residuals')
    xlabel(strcat('Regressor ', num2str(selected(i))))
end
```



The residuals do not show any patterns when plotted against the response variable. But we observe some patterns in the residual vs regressor plots for regressors 1 and 5. This indicates the presence of non-linear trends with these regressors.

```
figure("Position", [100, 100, 600, 250])
subplot(1, 2, 1)
```

```
histogram(ek, 15)
box OFF
title('Residual Histogram')

subplot(1, 2, 2)
probplot(ek)
box OFF
title('Normal Assumption Check Probability Plot')
```




```
h = adtest(ek);
if h==0
    disp('Failed to Reject Null ==> Normal Distribution is not Incor
else
    disp('Null Rejected ==> Normal Distribution is Incorrect')
end
```

```
Failed to Reject Null ==> Normal Distribution is not Incorrect
```

The residuals can be concluded to be from a normal distribution.

```
h = lbqtest(ek);
if h==0
    disp('Not enough evidence to believe that the residuals are seri
else
    disp('Null Rejected ==> Residuals are serially correlated')
end
```

```
Not enough evidence to believe that the residuals are serially corre
```

The lbqtest fails to reject the null that the residuals are uncorrelated. Hence this assumption holds.

Part c

Based on the information criteria values, 4a seems to be the better model.

Fitting a sequential model.

```
stepWiseMod = stepwiselm(Phi, y, 'PEnter', 0.1, 'PRemove', 0.15)
```

```
1. Adding x4, FStat = 3858.7533, pValue = 8.0120836e-40
2. Adding x1, FStat = 15.6283, pValue = 0.000334495
3. Adding x6, FStat = 21.5125, pValue = 4.51418e-05
4. Adding x5, FStat = 22.5907, pValue = 3.37585e-05
5. Adding x1:x5, FStat = 5.6735, pValue = 0.022962
6. Removing x4, FStat = 0.064481, pValue = 0.80108
stepWiseMod = 
Linear regression model:
    y ~ 1 + x6 + x1*x5

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-2420.2	939.31	-2.5765	0.01435
x1	3.3026	0.53177	6.2105	4.0813e-0
x5	2.7787	0.60178	4.6174	5.0663e-0
x6	-13.693	1.5055	-9.0955	9.5355e-1
x1:x5	-0.00096816	0.00032037	-3.022	0.004672

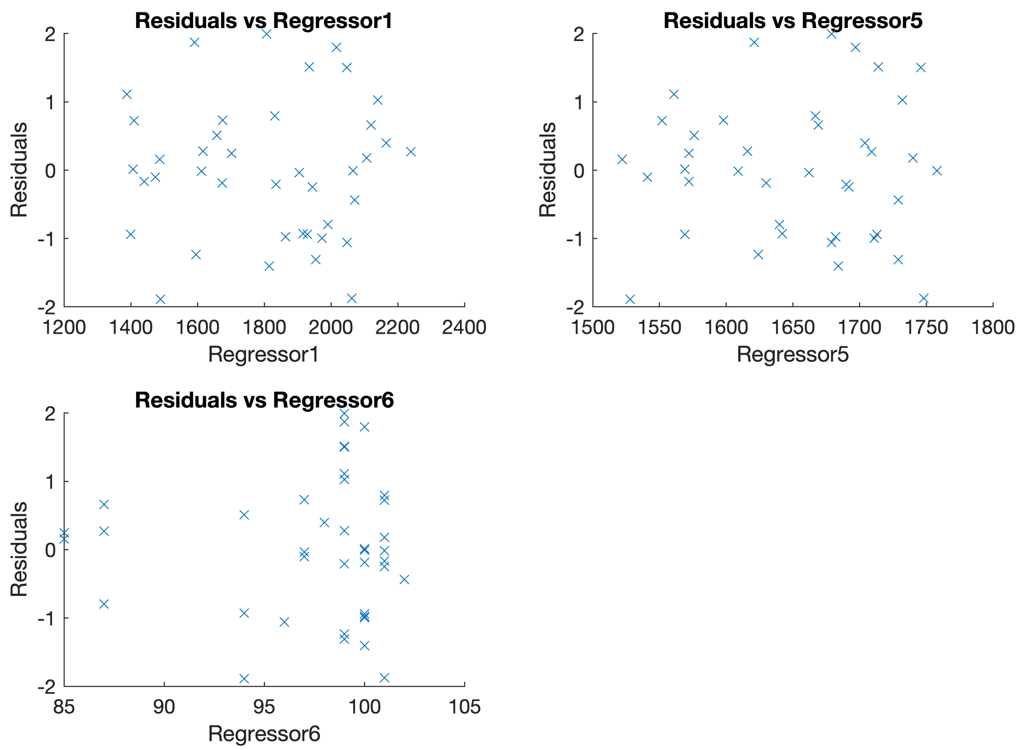
```
Number of observations: 40, Error degrees of freedom: 35
Root Mean Squared Error: 25.2
R-squared: 0.998,  Adjusted R-Squared: 0.998
F-statistic vs. constant model: 3.9e+03, p-value = 7.82e-46
```

The obtained model is similar to the model obtained in Part b but has an intercept. This model has the highest AIC/BIC and R^2 values.

Part d

Plotting the residuals against the regressors

```
ek = stepWiseMod.Residuals.Standardized;
[ySort, sortInd] = sort(y);

figure("Position", [100, 100, 600, 400])
for i=1:length(selected)
    subplot(2, 2, i)
    [PhiSort, sortInd] = sort(Phi(:, selected(i)));
    plot(PhiSort, ek(sortInd), 'x')
    box OFF
    title(strcat('Residuals vs Regressor ', num2str(selected(i))))
    ylabel('Residuals')
    xlabel(strcat('Regressor ', num2str(selected(i))))
end
```



We see some kind of quadratic pattern in regressors 1 and 5. We add these quadratic terms to the model and reestimate.

```
mod = fitlm(Phi, y, 'y ~ x1 + x5 + x6 + x1^2 + x5^2 - 1')
```

```
mod =
Linear regression model:
    y ~ x1 + x5 + x6 + x1^2 + x5^2
```

```
Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
x1	2.258	0.3227	6.9973	3.8513e-08
x5	1.0009	0.34341	2.9146	0.0061717
x6	-13.567	1.6065	-8.4454	5.7935e-10
x1^2	-0.00014641	8.5135e-05	-1.7197	0.09432
x5^2	-9.4455e-06	0.00010711	-0.088184	0.93023

```
Number of observations: 40, Error degrees of freedom: 35
Root Mean Squared Error: 26.9
```

```
mod.Rsquared
```

```
ans = struct with fields:
    Ordinary: 0.9975
    Adjusted: 0.9972
```

```
[~, ~, ic] = aicbic(mod.LogLikelihood, mod.NumEstimatedCoefficients)
```

```
ic = struct with fields:
     aic: 381.5439
     bic: 371.5439
    aicc: 369.5439
    caic: 376.5439
     hqc: -Inf
```

The t-test shows that the two added terms are not significant at 95% confidence levels.

The model also has lower AIC/BIC and R^2values. Thus the non-linear model is not an improvement over the linear model.

However, we have only explored homogenous quadratic terms. A non-linear model can have interaction terms too. A visual inspection of the residuals vs regressors plot will not help us identify these terms. We instead calculate the correlation coefficient between the different non-linear terms and the response variable.

```
% Quadratic Terms
PhiNonLin = [Phi(:, selected), Phi(:, selected).^2];

% Get Regressor Names
```

```
PhiNames = {};
PhiNames2 = {};
for i=1:length(selected)
    k = selected(i);
    PhiNames{end+1} = strcat('Phi', num2str(k));
    PhiNames2{end+1} = strcat('Phi', num2str(k), '^2');
end
PhiNames = [PhiNames, PhiNames2];

% Interaction Terms
for i=1:length(selected)-1
    for j=i+1:length(selected)
        k1 = selected(i);
        k2 = selected(j);
        PhiNonLin = [PhiNonLin, Phi(:, k1).*Phi(:, k2)];
        PhiNames{end+1} = strcat('Phi', num2str(k1), 'Phi', num2str(
    end
end

% Correlations
for i=1:size(PhiNonLin, 2)
    disp([PhiNames{i} , ' : ', num2str(corr(PhiNonLin(:, i), y))])
end
```

```
Phi1 : 0.99501
Phi5 : 0.87169
Phi6 : -0.14744
Phi1^2 : 0.9923
Phi5^2 : 0.87055
Phi6^2 : -0.14678
Phi1Phi5 : 0.98445
Phi1Phi6 : 0.90792
Phi5Phi6 : 0.40557
```

Based on this we select [Phi1, Phi5, Phi1^2, Phi5^2, Phi1Phi5, Phi1Phi6]

```
selected = [1, 2, 4, 5, 7, 8];
fitlm(PhiNonLin(:, selected), y, "VarNames", [PhiNames(selected), {'
```

```
ans =
Linear regression model:
    y ~ 1 + Phi1 + Phi5 + Phi1^2 + Phi5^2 + Phi1Phi5 + Phi1Phi6
```

```
Estimated Coefficients:
                      Estimate        SE         tStat       pValue
                     __________    __________    _______    __________

    (Intercept)          -33000         14900    -2.2147      0.033795
    Phi1                -9.8429        6.5017    -1.5139       0.13957
    Phi5                 53.534        25.138     2.1296      0.040754
    Phi1^2           -0.0016351    0.00072832    -2.2451      0.031585
    Phi5^2            -0.021948      0.010609    -2.0687      0.046479
    Phi1Phi5            0.01115     0.0055832     1.9971      0.054116
    Phi1Phi6          -0.010042     0.0015587    -6.4422    2.6407e-07


Number of observations: 40, Error degrees of freedom: 33
Root Mean Squared Error: 25.2
R-squared: 0.998,  Adjusted R-Squared: 0.998
F-statistic vs. constant model: 2.6e+03, p-value = 1.12e-42
```

We see that Phi1 is not useful, modifying the model accordingly

```
selected = [2, 4, 5, 7, 8];
finalMod = fitlm(PhiNonLin(:, selected), y, "VarNames", [PhiNames(se
```

```
finalMod =
Linear regression model:
    y ~ 1 + Phi5 + Phi1^2 + Phi5^2 + Phi1Phi5 + Phi1Phi6

Estimated Coefficients:
                      Estimate         SE         tStat       pValue
                     ___________    __________    _______    __________

    (Intercept)           -10820        2766.6    -3.9111     0.0004174
    Phi5                  15.805         3.355     4.7109      4.059e-0
    Phi1^2           -0.00054825    0.00012488    -4.3902     0.0001044
    Phi5^2            -0.0059552    0.00099947    -5.9583     9.7542e-0
    Phi1Phi5           0.0027082    0.00028211     9.5995     3.2873e-1
    Phi1Phi6          -0.0080262     0.0008259    -9.7181     2.4137e-1


Number of observations: 40, Error degrees of freedom: 34
Root Mean Squared Error: 25.7
```

```
R-squared: 0.998,  Adjusted R-Squared: 0.997
F-statistic vs. constant model: 3.01e+03, p-value = 5.85e-44
```

```
[~, ~, ic] = aicbic(finalMod.LogLikelihood, finalMod.NumEstimatedCoe
```

```
ic = struct with fields:
      aic: 378.6093
      bic: 366.6093
     aicc: 364.6093
     caic: 372.6093
      hqc: -Inf
```

This model is better than all our previous models and is thus selected. The final model structure is

y ~ Phi5 + Phi1^2 + Phi5^2 + Phi1Phi5 + Phi1Phi6

Question 5

Part a

The likelihood of the data is given by

$$f(\mathbf{y}|\theta) = \prod_k \frac{1}{\sqrt{2\pi\,\sigma_e^2}} \exp\left(-\frac{\left(y[k] - \varphi[k]^T\theta\right)^2}{2\sigma_e^2}\right)$$

The negative log-likelihood is

$$-\mathrm{LL} = \frac{N}{2}\log\left(2\pi\,\sigma_e^2\right) + \sum_k \frac{\left(y[k] - \varphi[k]^T\theta\right)^2}{2\sigma_e^2}$$

The Gaussian prior for θ is given by

$$f(\theta) = \frac{1}{\sqrt{2\pi\sigma_\beta^2}} \exp\left(-\frac{\theta^T\theta}{2\sigma_\beta^2}\right)$$

The negative log of the prior is given by

$$-\log(f(\theta)) = -\frac{1}{2}\log\left(2\pi\,\sigma_\beta^2\right) + \frac{\|\theta\|_2^2}{2\sigma_\beta^2}$$

The negative log of the posterior can be obtained as the sum of the negative logs of the prior, the likelihood and a constant.

$$-\log(f(\theta|\boldsymbol{y})) = \sum_k \frac{(y[k] - \varphi[k]^T\theta)^2}{2\sigma_e^2} + \frac{\|\theta\|_2^2}{2\sigma_\beta^2} + c$$

This gives us the objective function

$$\min_\theta \left(\sum_k (y[k] - \varphi[k]^T\theta)^2 + \frac{\sigma_e^2}{\sigma_\beta^2} \|\theta\|_2^2 \right)$$

Which is equivalent to the Ridge Regression objective with

$$\lambda = \frac{\sigma_e^2}{\sigma_\beta^2}$$

Part b

Expanding the vectors is the elastic-net objective function, we have

$$\min_\Theta \sum_k (y[k] - \varphi[k]^T\Theta) + \lambda\alpha \sum_{j=1\ldots p} \theta_j + \lambda(1-\alpha) \sum_{j=1\ldots p} |\theta_j|$$

where $\Theta = \begin{bmatrix}\theta_1 & \ldots & \theta_p\end{bmatrix}$, $\phi = [\varphi[1] \ \ \ldots \ \ \varphi[N]]^T$ and $\boldsymbol{y} = [y[1] \ \ \ldots \ \ y[N]]$

Combining the first two summations and defining

$$\boldsymbol{y}^\star = \begin{bmatrix} \boldsymbol{y} \\ 0_{\text{px1}} \end{bmatrix} \text{ and } \phi^\star = \begin{bmatrix} \phi \\ \sqrt{\lambda\,\alpha}\, I_{\text{pxp}} \end{bmatrix}$$

We have

$$\min_\Theta \|\boldsymbol{y}^\star - \phi^\star\Theta\|_2^2 + \lambda(1-\alpha)\|\Theta\|_2^2$$

which is equivalent to the LASSO problem objective with

$$\beta = \lambda(1-\alpha)$$

Question 6

Part a

The likelihood is given by

$$f(\boldsymbol{y}|\theta) = \frac{\theta^{\Sigma y[k]} e^{-N\theta}}{\prod_k y[k]!}$$

The log-likelihood is

$$\mathrm{LL} = \log(\theta)\sum_k y[k] - N\theta - \sum_k \log(y[k]!)$$

We compute the second derivate of the log-likelihood as

$$\frac{\partial^2}{\partial\theta^2}\log(f(\boldsymbol{y}|\theta)) = -\frac{\Sigma y[k]}{\theta^2}$$

The Fischer Information is obtained as

$$-E\left[\frac{\partial^2}{\partial\theta^2}\log(f(\boldsymbol{y}|\theta))\right] = I(\theta) = \frac{N}{\theta} \quad (\text{Use } E(y[k]) = \theta)$$

Thus, the Jeffreys' prior is

$$J(\theta) \propto I(\theta)^{\frac{1}{2}} \propto \theta^{-\frac{1}{2}}$$

Hence $\pi(\theta)$ is a class of Jeffreys' prior

Part b

The posterior pdf given by the product of the prior and likelihood is

$$\begin{aligned} f(2N\theta|\boldsymbol{y}) &\propto \frac{\theta^{\Sigma y[k]} e^{-N\theta}}{\prod_k y[k]!}\theta^{-\frac{1}{2}} \\ &\propto \theta^{\Sigma y[k]-\frac{1}{2}} e^{-\frac{2N\theta}{2}} \\ &\propto (2N\theta)^{\left(\frac{2N\bar{y}+1}{2}-1\right)} e^{-\frac{2N\theta}{2}} \end{aligned}$$

Which resembles a $\chi^2(k)$ pdf with $x = 2N\theta$ and $k = 2N\bar{y} + 1$

Part c

The MMSE estimate is given by the conditional expectation

$E(2N\theta) = k = 2N\overline{y} + 1$

Therefore the MMSE estimate of θ is

$$\hat{\theta} = \overline{y} + \frac{1}{2N}$$

Part d

Using $2N\theta \sim \chi^2(2N\overline{y} + 1)$ and the MMSE as the estimator,

We can solve the CDF equations given below

$$\frac{1}{\Gamma(N\overline{y} + 0.5)} \gamma\left(N\overline{y} + 0.5, \frac{\mathrm{CI}_{\mathrm{lower}}}{2N}\right) = \frac{\alpha}{2}$$

and

$$\frac{1}{\Gamma(N\overline{y} + 0.5)} \gamma\left(N\overline{y} + 0.5, \frac{\mathrm{CI}_{\mathrm{upper}}}{2N}\right) = 1 - \frac{\alpha}{2}$$

where $\Gamma(.)$ and $\gamma(.)$ are the complete gamma function and lower incomplete gamma functions respectively.

Since there is no closed-form expression for the inverse CDF of the χ^2 distribution, we need to use a numerical method to solve the above equations. The *chi2inv* function in MATLAB can solve the above equations.

The credible confidence interval for θ can hence be written as

$$\left(\frac{\chi^2_{\frac{\alpha}{2}}(k)}{2N}, \frac{\chi^2_{1-\frac{\alpha}{2}}(k)}{2N}\right) \text{ where } k = 2N\overline{y} + 1$$