# Group Project

CMPT-318: Cybersecurity Fall 2021

|| Group 1 ||
Daniel Tham - 301358046
Sunayani Sarkar - 301376589

## Table of Contents

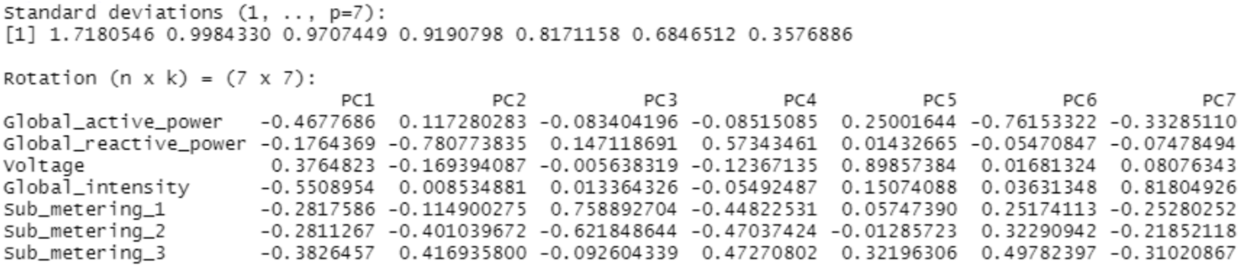
## Abstract

This project report shows our process of experimental analysis and interpretation of the result for the electricity consumption datasets we were given. We will apply PCA tests on a subset of training data to determine the features we will use to train our multivariate Hidden Markov models. By using the Hidden Markov Model, we are finding the log-likelihood and BIC values for different numbers of Hidden Markov Model states. We then find the normalized training log-likelihood and test log-likelihood to show how well our model fits the data. We then find the log-likelihood of the three anomalous datasets using the same time window in which each of the three datasets will be interpreted.

## Part 1

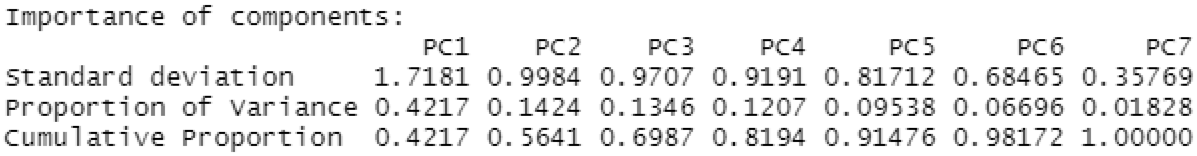
We decided to choose Global_intensity and Sub_metering_3 as our final variables for training our models. Our judgement came from a combination of the PCA results and our scatter plots using ggbiplot. With the three year dataset, we used the first two years to run our PCA tests.

Initially, we did the PCA test using every datapoint within the two year portioned data set. We noticed PC1 only had 42% of the variance explained and Global_intensity and Global_active_power had the highest absolute value in the PCA results matrix. For the time being, we were planning to use those as our training features.



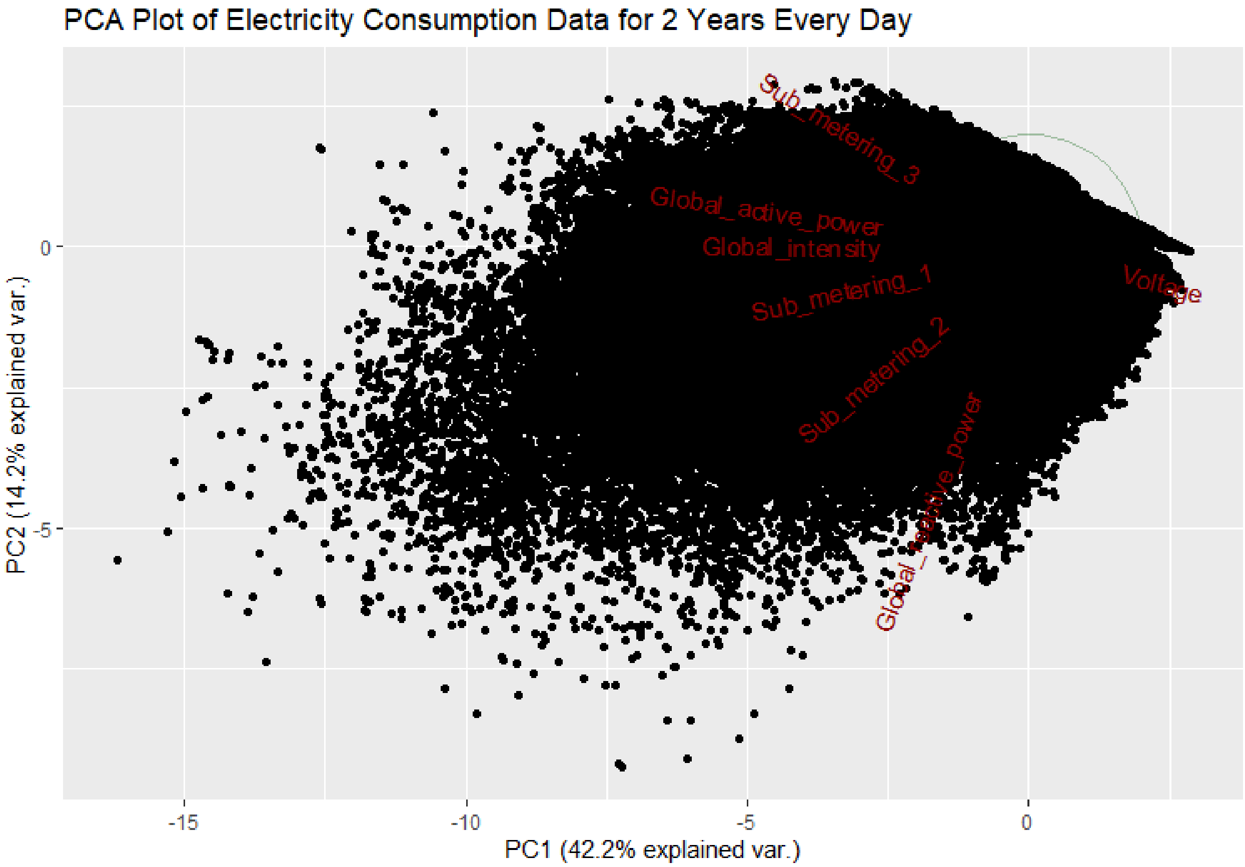
```
Standard deviations (1, .., p=7):
[1] 1.7180546 0.9984330 0.9707449 0.9190798 0.8171158 0.6846512 0.3576886

Rotation (n x k) = (7 x 7):
                             PC1          PC2          PC3         PC4         PC5         PC6         PC7
Global_active_power   -0.4677686  0.117280283 -0.083404196 -0.08515085  0.25001644 -0.76153322 -0.33285110
Global_reactive_power -0.1764369 -0.780773835  0.147118691  0.57343461  0.01432665 -0.05470847 -0.07478494
Voltage                0.3764823 -0.169394087 -0.005638319 -0.12367135  0.89857384  0.01681324  0.08076343
Global_intensity      -0.5508954  0.008534881  0.013364326 -0.05492487  0.15074088  0.03631348  0.81804926
Sub_metering_1        -0.2817586 -0.114900275  0.758892704 -0.44822531  0.05747390  0.25174113 -0.25280252
Sub_metering_2        -0.2811267 -0.401039672 -0.621848644 -0.47037424 -0.01285723  0.32290942 -0.21852118
Sub_metering_3        -0.3826457  0.416935800 -0.092604339  0.47270802  0.32196306  0.49782397 -0.31020867
```

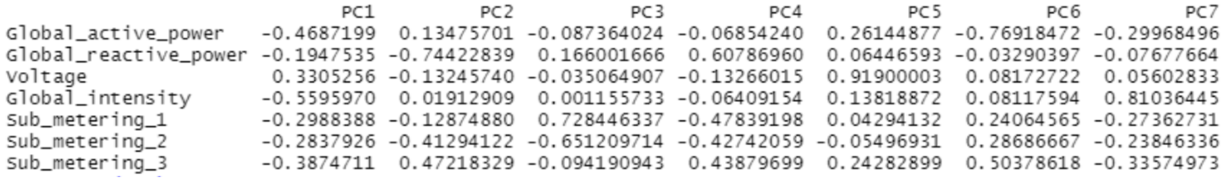


```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.7181 0.9984 0.9707 0.9191 0.81712 0.68465 0.35769
Proportion of Variance 0.4217 0.1424 0.1346 0.1207 0.09538 0.06696 0.01828
Cumulative Proportion  0.4217 0.5641 0.6987 0.8194 0.91476 0.98172 1.00000
```

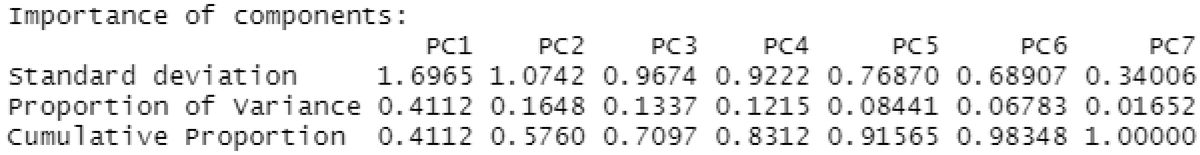
We plotted these values with respect to PC1 and PC2 which did not give us much information to go off of because there were too many data points clustered which muddled the results as seen below.



The only visible information we can see from the scatter plot is that Global_active_power and Global_intensity are the only two closest features based on the placement of the text on the plot. We did not think we had sufficient rationale to conclude to use these two features. However we decided to then take the two years of data we already had and portion them into weeks of specific time windows. We chose Fridays from 5pm to 8pm which will be explained during the Hidden Markov Model section. This way, we will get more narrowed down data for a certain timeframe.

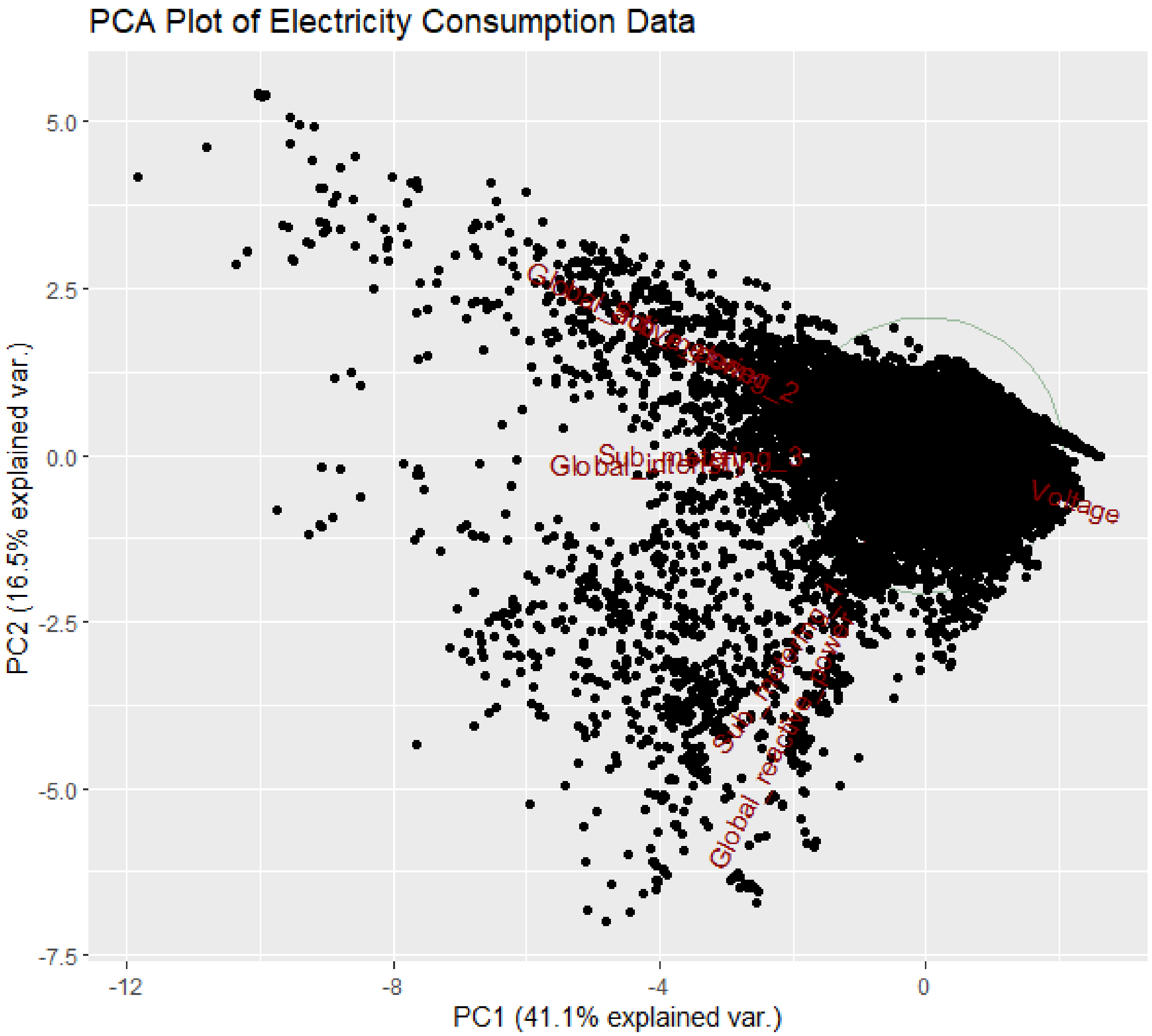


```
                            PC1         PC2          PC3         PC4         PC5         PC6         PC7
Global_active_power   -0.4687199  0.13475701 -0.087364024 -0.06854240  0.26144877 -0.76918472 -0.29968496
Global_reactive_power -0.1947535 -0.74422839  0.166001666  0.60786960  0.06446593 -0.03290397 -0.07677664
Voltage                0.3305256 -0.13245740 -0.035064907 -0.13266015  0.91900003  0.08172722  0.05602833
Global_intensity      -0.5595970  0.01912909  0.001155733 -0.06409154  0.13818872  0.08117594  0.81036445
Sub_metering_1        -0.2988388 -0.12874880  0.728446337 -0.47839198  0.04294132  0.24064565 -0.27362731
Sub_metering_2        -0.2837926 -0.41294122 -0.651209714 -0.42742059 -0.05496931  0.28686667 -0.23846336
Sub_metering_3        -0.3874711  0.47218329 -0.094190943  0.43879699  0.24282899  0.50378618 -0.33574973
```



```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.6965 1.0742 0.9674 0.9222 0.76870 0.68907 0.34006
Proportion of Variance 0.4112 0.1648 0.1337 0.1215 0.08441 0.06783 0.01652
Cumulative Proportion  0.4112 0.5760 0.7097 0.8312 0.91565 0.98348 1.00000
```

Upon further comparison, the results of our second PCA test with our portioned off dataset seems to be very similar to the results of the PCA test with every data point. The plot for the second PCA test did give us more workable information.



There are three clusters of features which are grouped as follows from top to bottom: global_active_power and sub_metering_2, global_intensity and sub_metering_3, and global_reactive_power and sub_metering_1.

We noticed that with respect to both PC1 and PC2 of this PCA test, global_intensity and sub_metering_3 appealed to us to use as features for a couple reasons.

1. In the plot, their positions seem very consistent in relation to each other. They were essentially grouped together in a flat line.

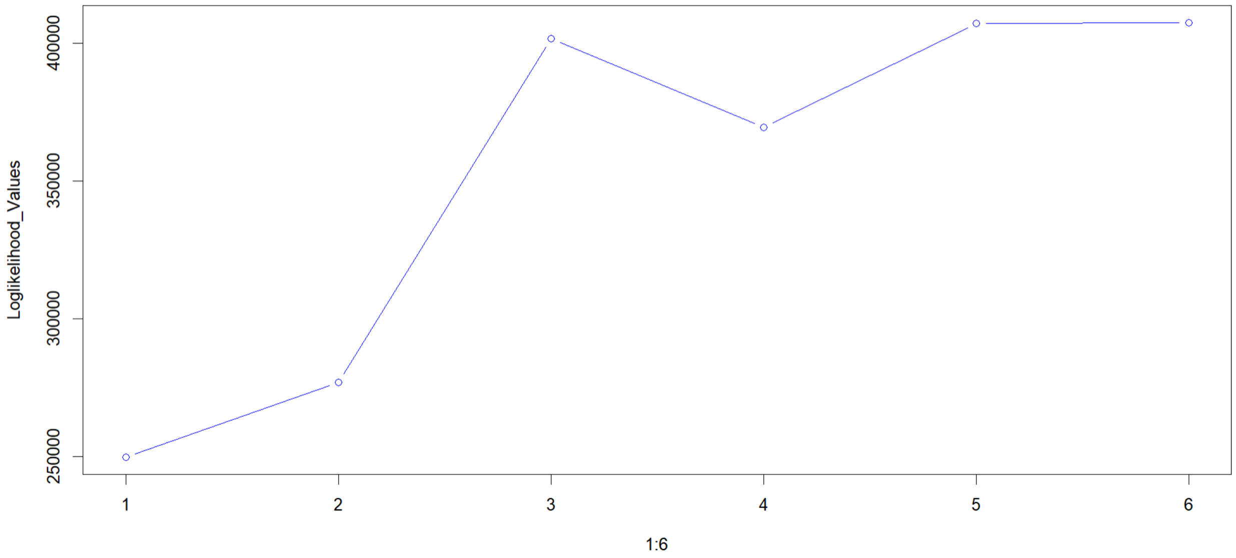
2. In the PCA matrix, global_intensity had the highest absolute value in PC1 which encompassed 42% of the data, while sub_metering_3 had the second highest absolute value in PC2.
3. We did not choose global_reactive_power and sub_metering_2 because both had low values in PC1 and only if you count PC2, then global_reactive_power has the highest absolute value.  Voltage also had low values in both PC1 and PC2.
4. Global_intensity also had the highest value in our first PCA test so half of our observation was still in line with our reasoning with our second PCA test results.

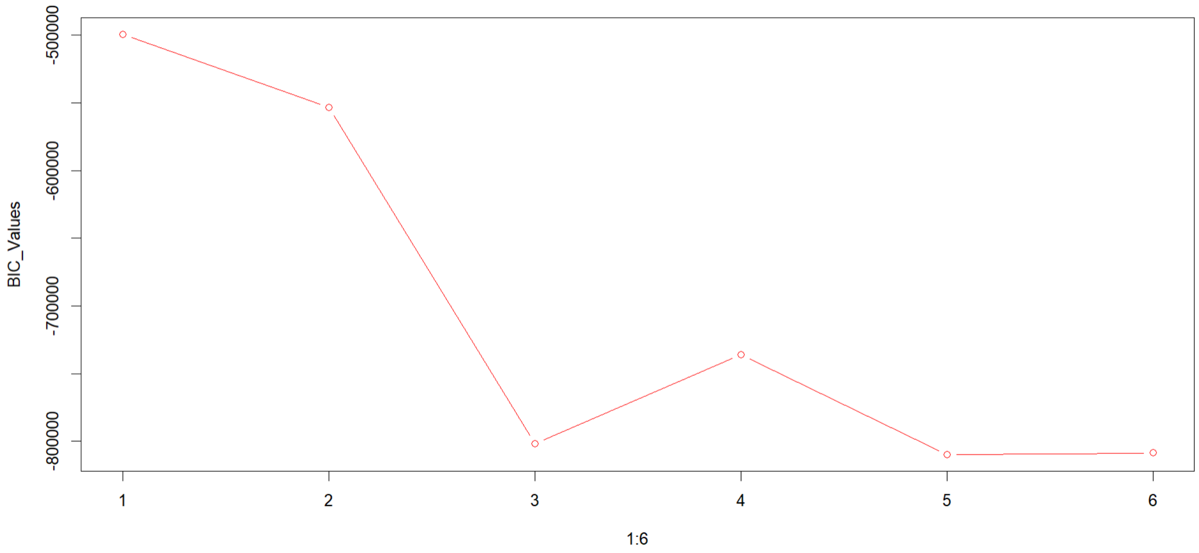
## Part 2

The observation time window for the analysis was chosen based on the fact that Friday evenings were considered the beginning of the weekend. Most people came back home at around 5 pm from their workplaces and spent the evening together with their family and friends. We wanted to see the trend in power usage. Hence, we chose Friday evenings between 5 pm and 8 pm.

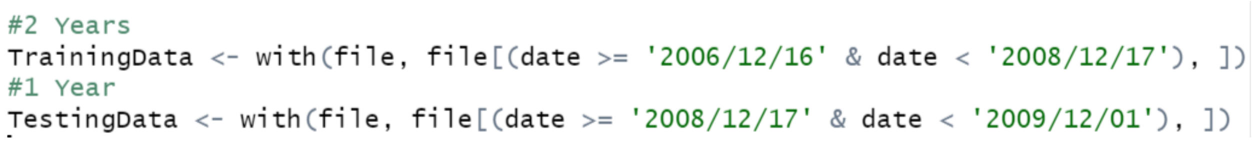
### Discussion:

We used 6 states to map the following Log-likelihood and BIC plots, with nstates values being equal to 4, 8, 12, 16, 20 and 24, respectively. At nstates = 12, that is at point 3 on the x-axis in the plots below, we see that the log-likelihood is high and the BIC value is low. The log-likelihood value for nstates = 12 is high but not so high that it would encourage over-fitting. Also, to counter overfitting, BIC value shows us that the value at nstates = 12 is quite low at -801696.8 but not as low as that of nstates = 20 and 24. Hence, we chose nstates value as 12 and the 3rd fitted model fm3 as our final model.





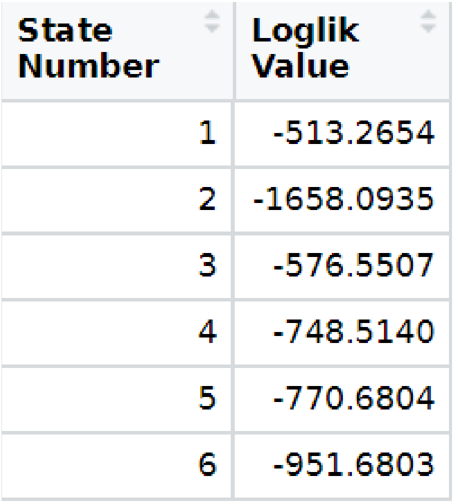
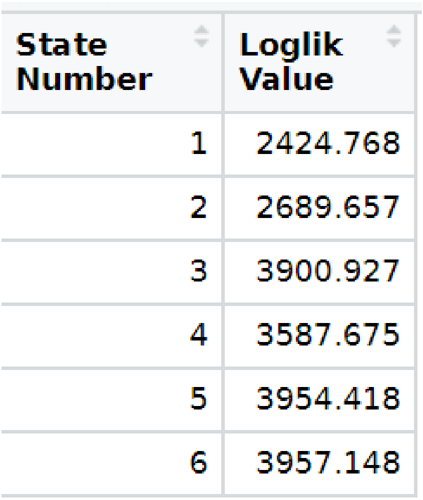
We divided our data into 2 years of training data and 1 year of testing data as shown below. Larger training datasets guarantee that model performance is calculated more accurately.



```
#2 Years
TrainingData <- with(file, file[(date >= '2006/12/16' & date < '2008/12/17'), ])
#1 Year
TestingData <- with(file, file[(date >= '2008/12/17' & date < '2009/12/01'), ])
```

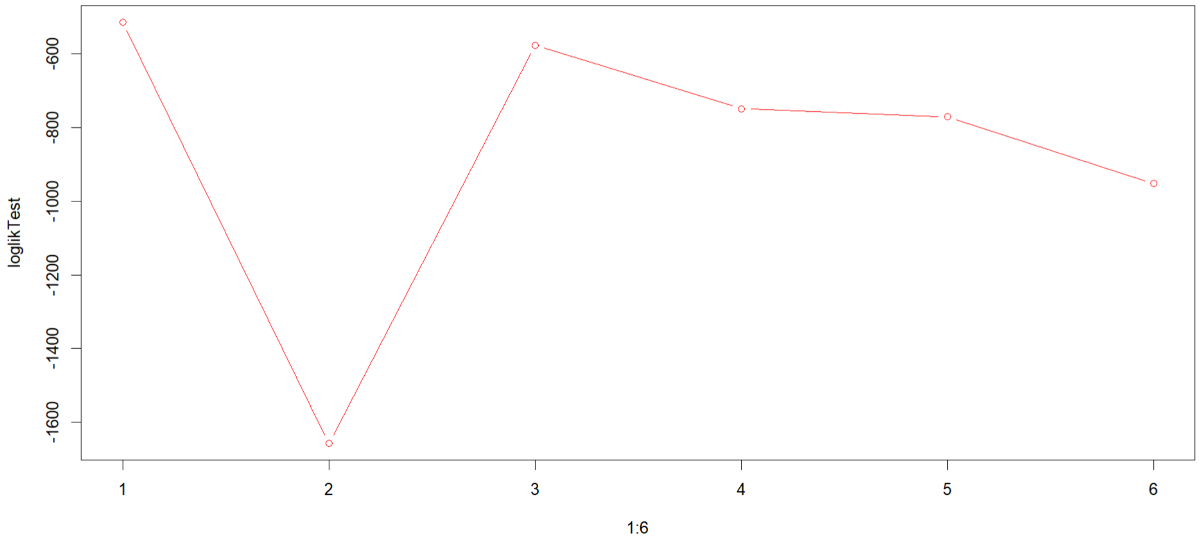
Below are the tables and plots showing the values of the training log-likelihood and test log-likelihood values.

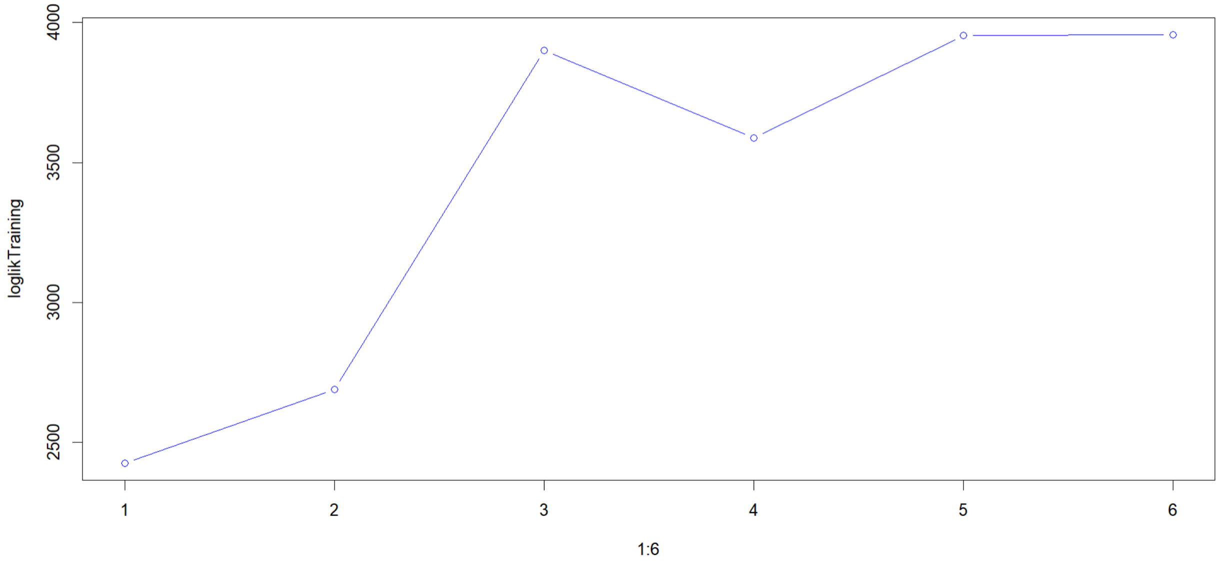
Comparison of Test likelihood vs Training likelihood

<table>
<tr><th>Test Likelihood</th><th>Training Likelihood</th></tr>
<tr>
<td>
<table>
<tr><th>State Number</th><th>Loglik Value</th></tr>
<tr><td>1</td><td>-513.2654</td></tr>
<tr><td>2</td><td>-1658.0935</td></tr>
<tr><td>3</td><td>-576.5507</td></tr>
<tr><td>4</td><td>-748.5140</td></tr>
<tr><td>5</td><td>-770.6804</td></tr>
<tr><td>6</td><td>-951.6803</td></tr>
</table></td>
<td>
<table>
<tr><th>State Number</th><th>Loglik Value</th></tr>
<tr><td>1</td><td>2424.768</td></tr>
<tr><td>2</td><td>2689.657</td></tr>
<tr><td>3</td><td>3900.927</td></tr>
<tr><td>4</td><td>3587.675</td></tr>
<tr><td>5</td><td>3954.418</td></tr>
<tr><td>6</td><td>3957.148</td></tr>
</table></td>
</tr>
</table>

In the above table, the training data is larger and hence the outcome of probability densities must be continuous. Hence, the value of the log-likelihood is positive. Whereas in the case of the testing dataset, the probabilities were more discrete as Markov Models only depend on the present value of the state. Hence, the log-likelihood value is a sum of logs of probabilities that are smaller than 1, resulting in a negative value.

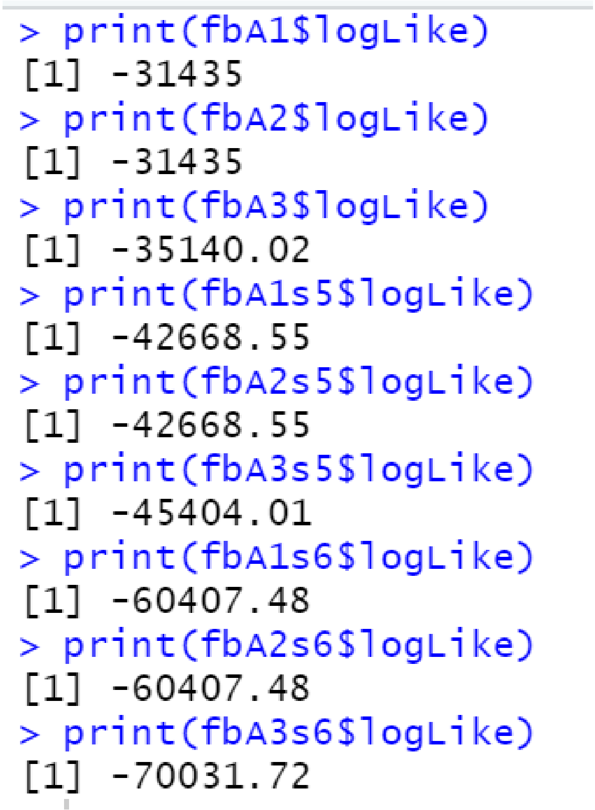
As can be seen in the plot below, the log-likelihood of the training data and testing data follow similar trends except at nstates = 4, i.e. point 1 on the x-axis.



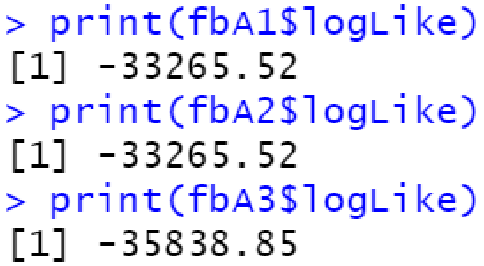


## Part 3

We chose nstates = 12, 20 and 24 and calculated the log-likelihood of the 3 anomalous datasets. The value of log-likelihood of nstates = 12 is the highest and hence we choose nstates = 12 model for the anomalous dataset readings. Here are two different instances of the anomalous behaviour when tested based on two separately trained models on the same dataset. We see that anomalous datasets 1 and 2 have the same log-likelihood values for all 4 tests (two with the 3rd nstates = 12 and one each with the 5th and 6th points where the n-states = 20 and 24, respectively.



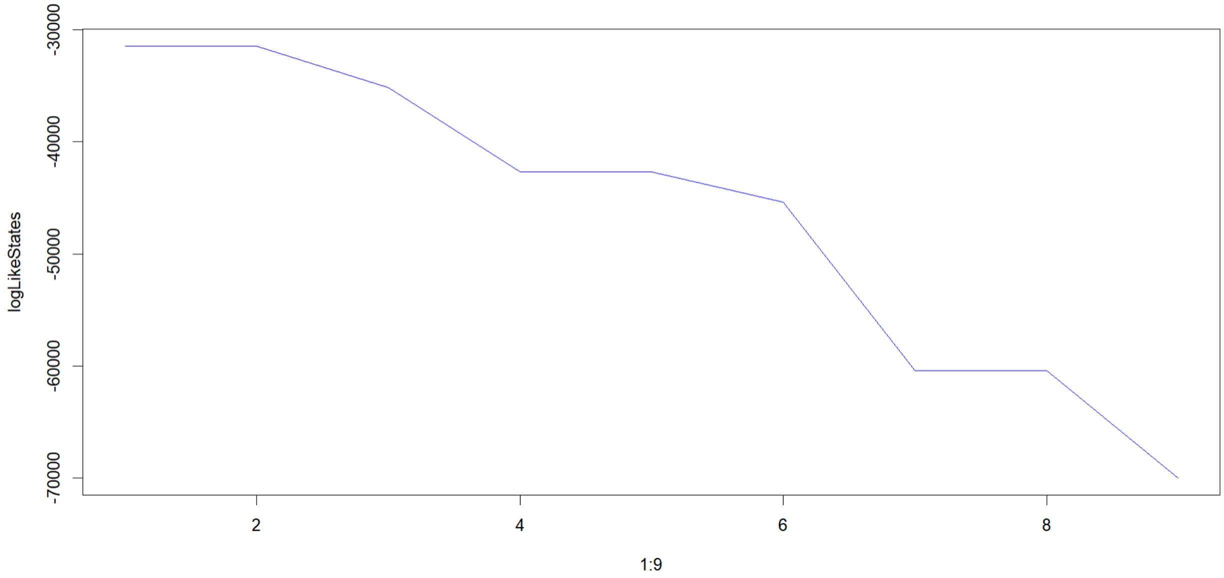
```
> print(fbA1$logLike)
[1] -31435
> print(fbA2$logLike)
[1] -31435
> print(fbA3$logLike)
[1] -35140.02
> print(fbA1s5$logLike)
[1] -42668.55
> print(fbA2s5$logLike)
[1] -42668.55
> print(fbA3s5$logLike)
[1] -45404.01
> print(fbA1s6$logLike)
[1] -60407.48
> print(fbA2s6$logLike)
[1] -60407.48
> print(fbA3s6$logLike)
[1] -70031.72
```



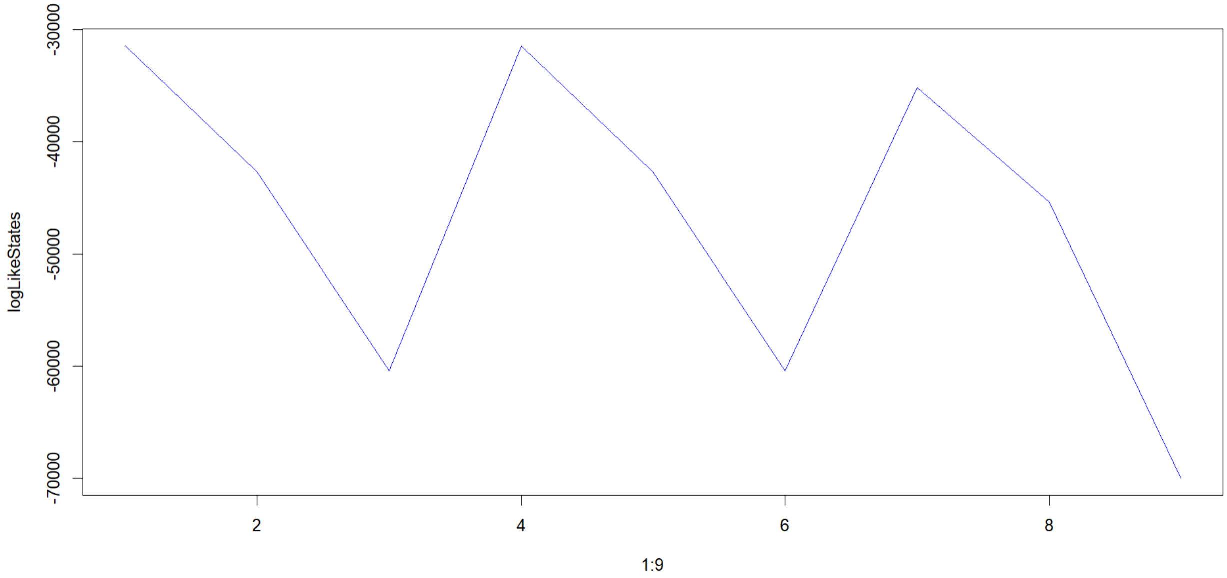
```
> print(fbA1$logLike)
[1] -33265.52
> print(fbA2$logLike)
[1] -33265.52
> print(fbA3$logLike)
[1] -35838.85
```

The first 3 points on the x-axis reflect the log-likelihood values for nstates = 12. The Anomaly dataset number 3 seems to have a higher degree of anomaly (more visible) than the other two datasets.

The first graph shows the trend in log-likelihood of the three anomaly datasets with respect to the 3 n-states values, 12, 20 and 24.

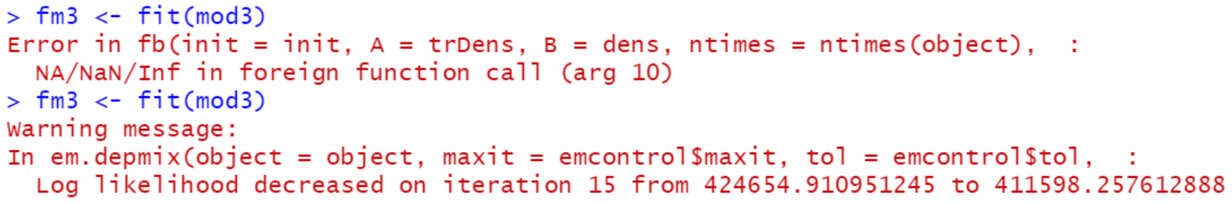


The graph below shows that the peak value of the first and second anomaly dataset are equal and they are higher than the peak value for the third anomaly. Similarly, the trough value for the third anomaly is lower than the first and second anomaly datasets, which are equal.



## Challenges

While fitting the model, we noticed that a number of times the fit values would not work. They produced an error as shown below. But when we ran the same line of code again, soon after, we got the code working but with the latter warnings:



```
> fm3 <- fit(mod3)
Error in fb(init = init, A = trDens, B = dens, ntimes = ntimes(object),  : 
  NA/NaN/Inf in foreign function call (arg 10)
> fm3 <- fit(mod3)
Warning message:
In em.depmix(object = object, maxit = emcontrol$maxit, tol = emcontrol$tol,  :
  Log likelihood decreased on iteration 15 from 424654.910951245 to 411598.257612888
```

This made us realise that we were facing this issue due to the manner in which the EM algorithm works in an HMM, with a Gaussian response distribution. It is possible for the variance for a given state to decrease to zero when the state models a single valued observation. It is also possible that the likelihood of state-occupancy drops to the point where a state doesn't "appear" in the whole time series. The state is undefined in the scenario. Both will result in numerical problems. Whether such mistakes arise in a given run of the EM method depends on the initial values when the model is not specified per se. By default, depmixS4 assigns observations randomly to states before then conducting a maximisation step for this random assignment. The EM algorithm may be re-run until one or more converging runs are obtained. To maximise the possibilities of achieving a global optimum, it is often recommended to execute the algorithm many times with varied starting variables. Another option is to utilise numerical optimization instead of the EM technique.

A second challenge we faced was the illustration of the anomalies in the three different data sets with injected anomalies. We could compare and interpret the three datasets using log-likelihood values of the three n-states (12, 20 and 24), where the training model showed that the log-likelihood values were the same. However, we were unable to plot a graph showing clusters of data to find the outliers and anomalies in the three datasets provided.

## References:

*Interpret the key results for Principal Components Analysis - Minitab*. (2019). (C) Minitab, LLC. All Rights Reserved. 2019. https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/multivariate/how-to/principal-components/interpret-the-results/key-results/

J. (2020). *`NA/NaN/Inf in foreign function call` from `fit()` - githubmemory*. Githubmemory.Com. https://githubmemory.com/repo/depmix/depmixS4/issues/2

Starmer, S. W. J. (2017, November 27). *StatQuest: PCA in R* [Video]. YouTube. https://www.youtube.com/watch?v=0Jp4gsfOLMs&feature=youtu.be

Towers, S. (2019, March 18). *Testing if one model fits the data significantly better than another model | Polymatheia*. Sherry Towers. http://sherrytowers.com/2019/03/18/determining-which-model-fits-the-data-significantly-better/

Yeh, A. (2019, June 13). *A Simple Way to Detect Anomaly - Towards Data Science*. Medium. https://towardsdatascience.com/a-simple-way-to-detect-anomaly-3d5a48c0dae0

## Contributions

| Tasklist | Name of the Person |
|---|---|
| CODING - Part 1: PCA | Daniel Tham |
| CODING - Part 2: HMM - Training and Testing Datasets | Sunayani Sarkar |
| CODING - Part 3: Anomaly Detection using the designated model | Sunayani Sarkar |
| CODING - Maintainability: Code clean up | Sunayani Sarkar |
| REPORT - Part 1 | Daniel Tham |
| REPORT - Part 2 | Sunayani Sarkar |
| REPORT - Part 3 | Sunayani Sarkar |
| PRESENTATION - Part 1 | Daniel Tham |
| PRESENTATION - Part 2 | Daniel Tham |
| PRESENTATION - Part 3 | Daniel Tham |