

Andrew Marschall

## 1. Social Media (Twitter) Data Scrapping

This data set is from twitter, scraping data from the hashtag #solarenergy.

The results show the amount of tweets scraped from 2022-01-20

The screenshot displays a Windows 10 desktop environment. The primary focus is on a PowerShell script editor, which is part of the RStudio application. The script, named 'solarenergy.ps1', is a REST client for the Twitter API. It defines several variables for headers, base URL, and endpoints. The script then uses the 'Invoke-WebRequest' cmdlet to send a GET request to the Twitter API, searching for tweets related to 'solar energy' from January 20, 2022, to January 24, 2022. The script includes a 'while' loop to iterate through the results, displaying the tweet ID, user name, and tweet text. The terminal window at the bottom shows the execution of the script, with the output of the REST client commands visible.

```

# PowerShell Script: solarenergy.ps1

# Headers
$headers = @{
    'Authorization' = 'Bearer '
    'Content-Type' = 'application/json'
}

# Base URL
$baseUrl = 'https://api.twitter.com/2'

# Search endpoint
$searchEndpoint = '$baseUrl/tweets/search/recent'

# Parameters
$params = @{
    'query' = '#solarenergy'
    'start_time' = '2022-01-20T00:00:00Z'
    'end_time' = '2022-01-24T23:59:59Z'
    'tweet_fields' = 'id,author_id,created_at,public_metrics'
    'expansion_fields' = 'attachments,author_id,context_annotations,conversation_id,created_at,entities,mentions,reply_id,reply_count,retweet_count,source'
}

# Send GET request
$response = Invoke-WebRequest -Uri $searchEndpoint -Headers $headers -Body $params

# Parse JSON response
$jsonResponse = $response.Content | ConvertFrom-Json

# Iterate through results
while ($jsonResponse.data.Count -gt 0) {
    $tweet = $jsonResponse.data[0]

    # Display tweet details
    Write-Host "Tweet ID: $($tweet.id)"
    Write-Host "User: $($tweet.author_id)"
    Write-Host "Text: $($tweet.text)"

    # Display tweet expansions
    $expansions = $tweet.expansions
    if ($expansions) {
        $expansions | ForEach-Object {
            Write-Host "Expansion: $_"
        }
    }

    # Move to the next tweet
    $jsonResponse = $jsonResponse.next
}

```

## 2. Data Cleaning

This data set is from twitter and the tv show #ICanSeeYourVoice

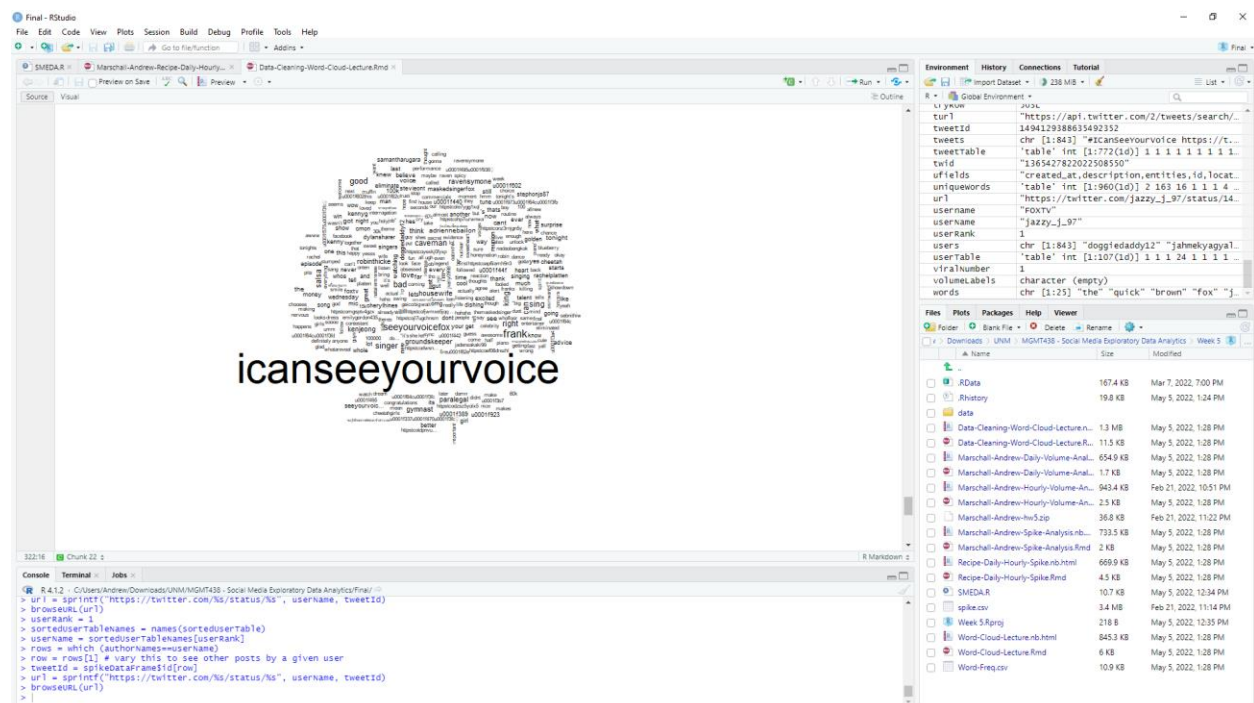
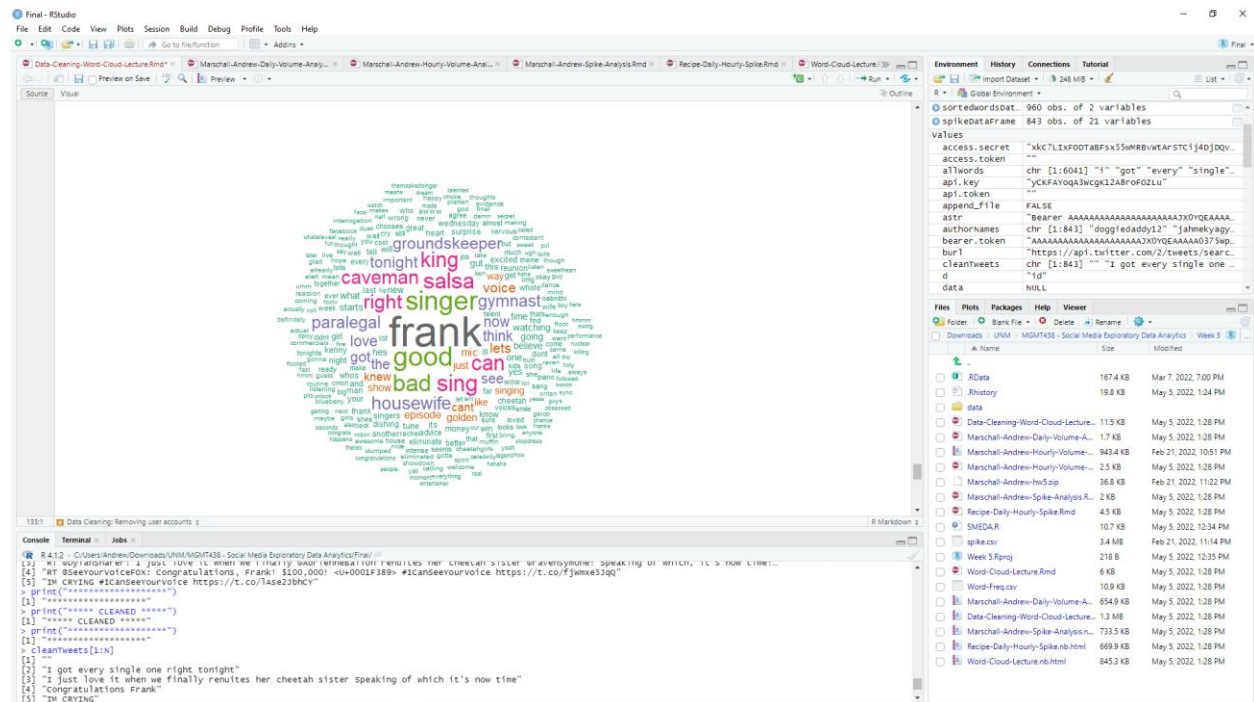
The results show that we are removing data so we can analyze it more clearly. In this case we are removing words from tweets over the past seven days to transform the data. This makes the word clouds easier to read. We are using the `gsub` and `grep` functions to make this possible.

[illegible]

### 3. Word Clouds

This data set is still from #ICanSeeYourVoice on twitter.

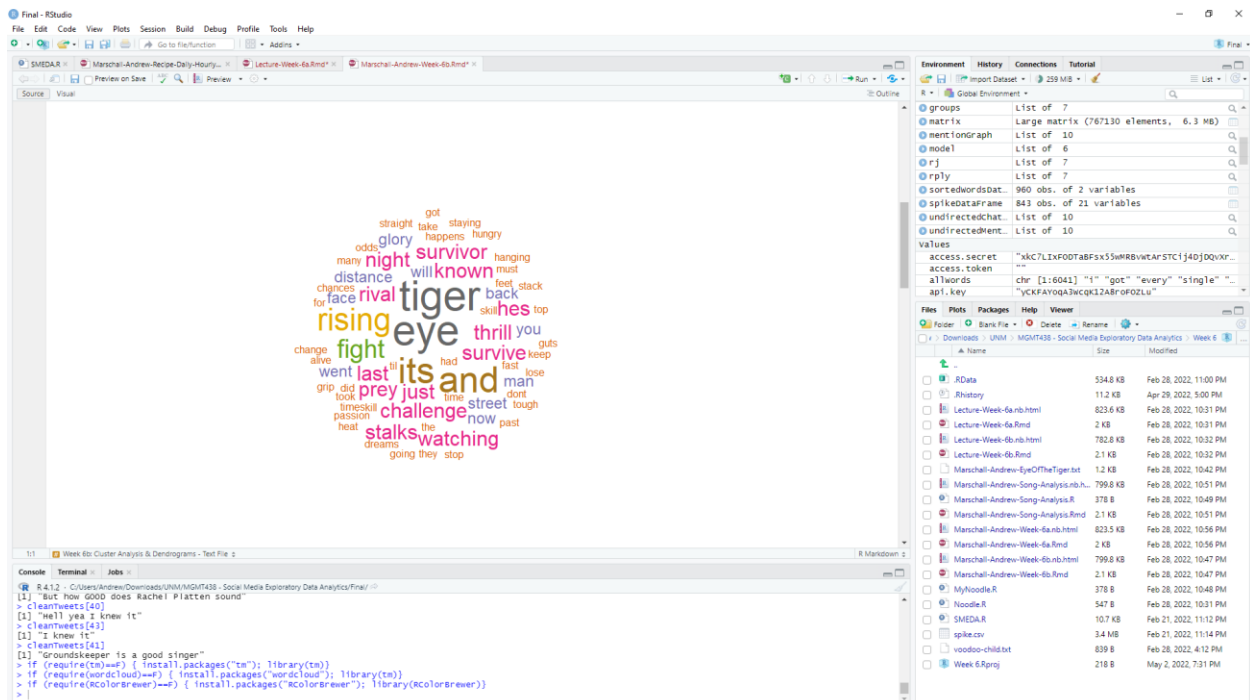
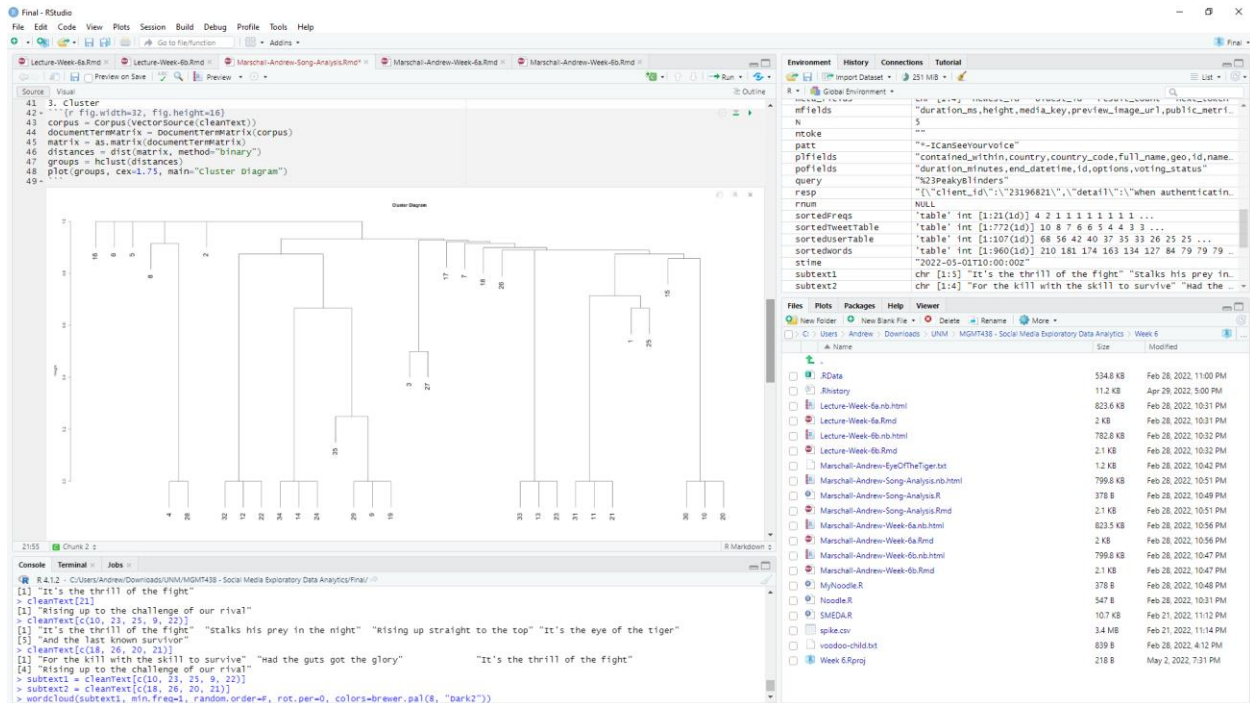
The results show which words were used the most frequently within tweets over the past seven days.



## 4. Hierarchical Clustering & Dendrograms

This data set is from a text file with the lyrics from Eye Of The Tiger.

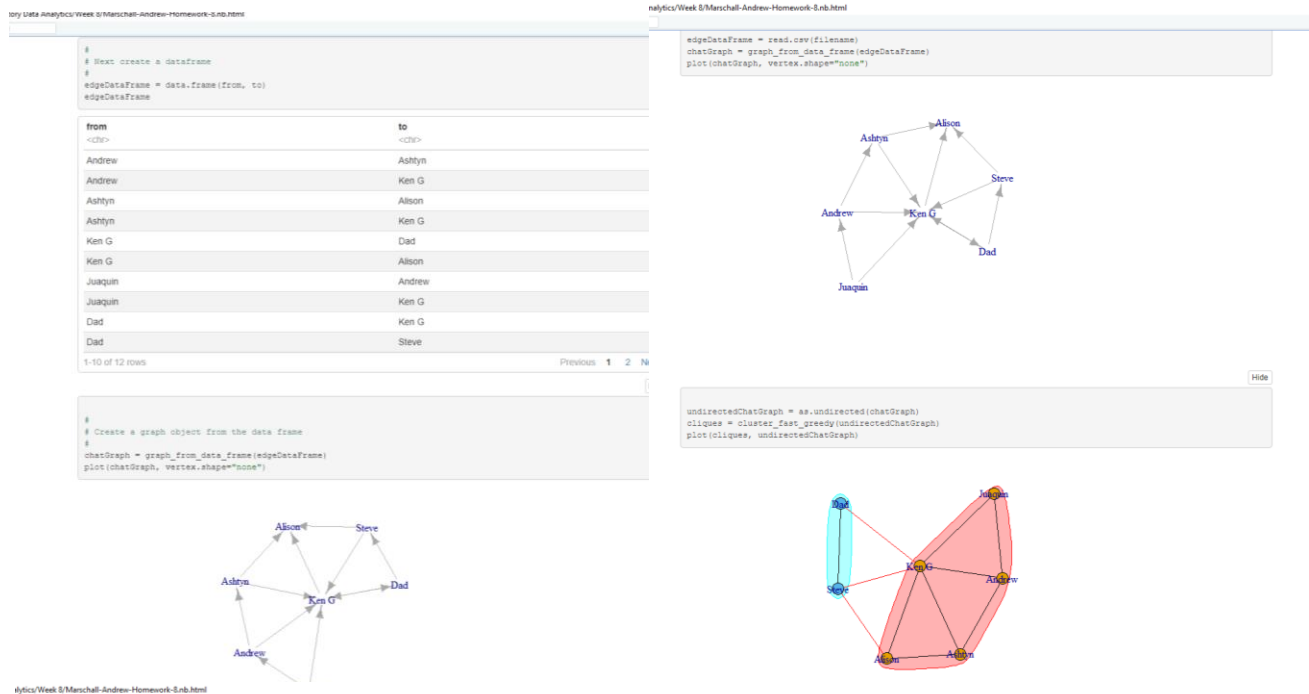
The results show the song lyrics converted into a word cloud as well as clustering the data as a dendrogram.



## 5. Social Networking Analysis

This data set is from a basic chat network of people I know, as well as from a twitter spike showing a more complex network of users.

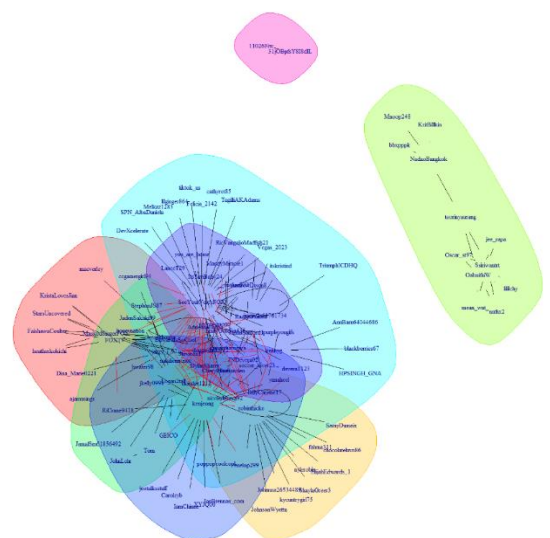
The results show who is messaging who, on both a smaller and larger scale.



### Spike Clique Network Code Chunk

```
library(igraph) # Assumes: install.packages("igraph") once
library(jonlite) # Assumes: install.packages("jonlite") once
filename = "spike.csv"
spikeDataFrame = read.csv(filename, encoding="UTF-8")
#
# The hard part, generating the from-to vectors
#
users = apply(spikeDataFrame$users, function(u) {
  from = NULL
  to = NULL
  for (i in 1:nrow(spikeDataFrame)) {
    entity = from300(spikeDataFrame$entities[i])
    if (is.na(entity$mentions)) {
      for (username in entity$mentions[[1]]$username) {
        from = c(from, users[i])
        to = c(to, username)
      }
    }
  }
}, MARGIN=2)
names(from) = NULL
names(to) = NULL
#
# create the edge data frame
#
edgeDataFrame = data.frame(from, to)
edgeDataFrame = unique(edgeDataFrame)
mentionGraph = graph_from_data_frame(edgeDataFrame)
undirectedMentionGraph = as.undirected(mentionGraph)
communities = cluster_fast_greedy(undirectedMentionGraph)
plot(communities, undirectedMentionGraph, vertex.shape="none")
```

```
communities = cluster_fast_greedy(undirectedMentionGraph)
plot(communities, undirectedMentionGraph, vertex.shape="none")
```



## 6. Linear Regression

This data set is from...

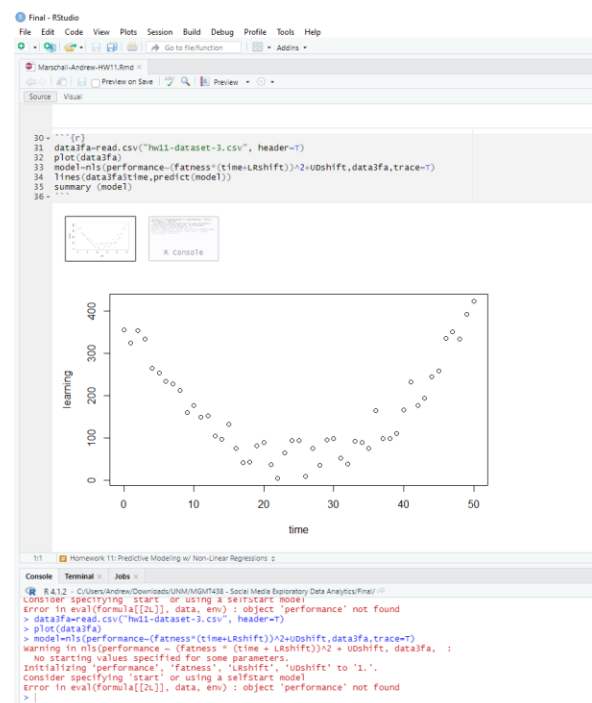
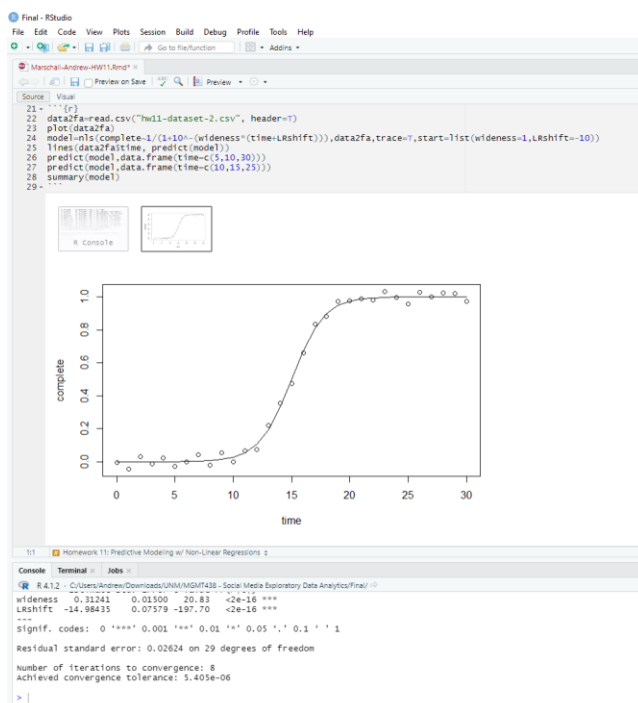
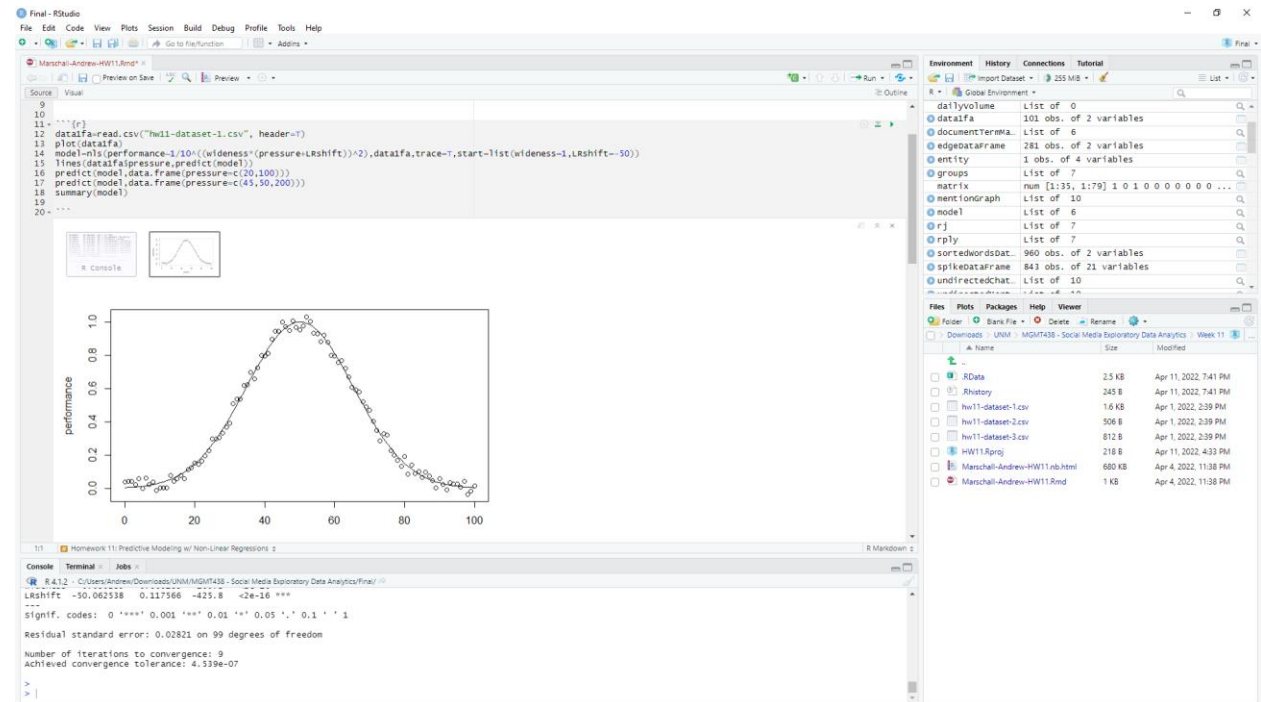
The results show that...



## 7. Non-Linear Regression

This data set is from three csv files with data on variables such as pressure and performance, or time and completion / learning.

The first result shows that as pressure increases, performance will also increase up to a certain point.



## 8. Neural Network



## 9. Bayesian Network

