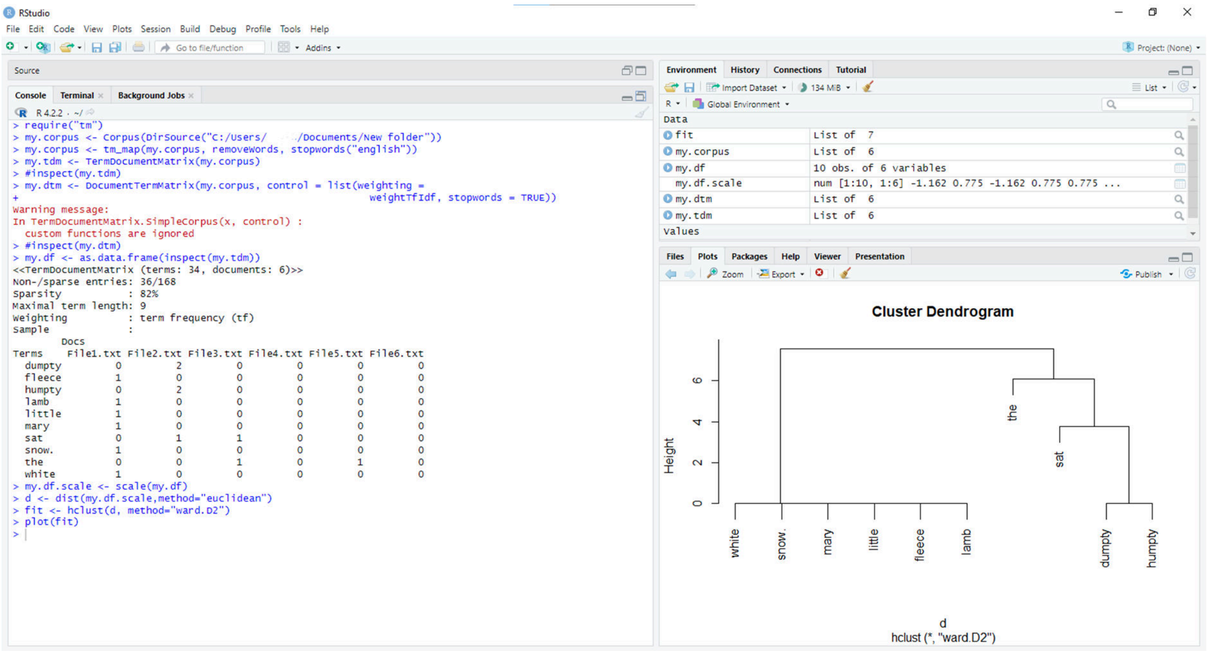
Practical 7: Measuring Similarity Among Documents and Detecting Passages Which Have Been Reused

Codes:

```r
# Install necessary packages
install.packages("tm")
require("tm")
install.packages("ggplot2")
install.packages("textreuse")
install.packages("devtools")

# Load in corpus and preprocess text
my.corpus <- Corpus(DirSource("C:/Users/asif0/Documents/New folder"))  # Load in corpus from directory
my.corpus <- tm_map(my.corpus, removeWords, stopwords("english"))  # Remove stop words from corpus

# Create term-document matrix
my.tdm <- TermDocumentMatrix(my.corpus)  # Create term-document matrix from corpus
#inspect(my.tdm)  # Inspect term-document matrix (optional)

# Create document-term matrix
my.dtm <- DocumentTermMatrix(my.corpus, control = list(weighting = weightTfIdf, stopwords = TRUE))  # Create document-term matrix from corpus, using TF-IDF weighting and removing stop words
#inspect(my.dtm)  # Inspect document-term matrix (optional)

# Convert document-term matrix to data frame and scale data
my.df <- as.data.frame(inspect(my.tdm))  # Convert document-term matrix to data frame
my.df.scale <- scale(my.df)  # Scale data using z-score normalization

# Perform hierarchical clustering and plot dendrogram
d <- dist(my.df.scale, method = "euclidean")  # Calculate distance matrix using Euclidean distance
fit <- hclust(d, method = "ward")  # Perform hierarchical clustering using Ward's method
plot(fit)  # Plot dendrogram
```

OUTPUT:



```
> require("tm")
> my.corpus <- Corpus(DirSource("C:/Users/    /Documents/New folder"))
> my.corpus <- tm_map(my.corpus, removeWords, stopwords("english"))
> my.tdm <- TermDocumentMatrix(my.corpus)
> #inspect(my.tdm)
> my.dtm <- DocumentTermMatrix(my.corpus, control = list(weighting =
+                                                         weightTfIdf, stopwords = TRUE))
Warning message:
In TermDocumentMatrix.SimpleCorpus(x, control) :
  custom functions are ignored
> #inspect(my.dtm)
> my.df <- as.data.frame(inspect(my.tdm))
<<TermDocumentMatrix (terms: 34, documents: 6)>>
Non-/sparse entries: 36/168
Sparsity           : 82%
Maximal term length: 9
Weighting          : term frequency (tf)
Sample             :
        Docs
Terms    File1.txt File2.txt File3.txt File4.txt File5.txt File6.txt
  dumpty         0         2         0         0         0         0
  fleece         1         0         0         0         0         0
  humpty         0         2         0         0         0         0
  lamb           1         0         0         0         0         0
  little         1         0         0         0         0         0
  mary           1         0         0         0         0         0
  sat            0         1         1         0         0         0
  snow.          1         0         0         0         0         0
  the            0         0         1         0         1         0
  white          1         0         0         0         0         0
> my.df.scale <- scale(my.df)
> d <- dist(my.df.scale,method="euclidean")
> fit <- hclust(d, method="ward.D2")
> plot(fit)
> 
```

```
Data
fit            List of  7
my.corpus      List of  6
my.df          10 obs. of 6 variables
my.df.scale    num [1:10, 1:6] -1.162 0.775 -1.162 0.775 0.775 ...
my.dtm         List of  6
my.tdm         List of  6
Values
```