Clustering – kmeans

The dataset for k-means clustering has been taken from:
[https://opendata.socrata.com/Government/Airplane-Crashes-and-Fatalities-Since-1908/q2te-8cvq](https://opendata.socrata.com/Government/Airplane-Crashes-and-Fatalities-Since-1908/q2te-8cvq)
The dataset contains: 13 attributes and 5268 instances. Here is the list of attributes in the dataset:

| Column Name | Description | Type |
|---|---|---|
| Date | | Date & Time (with timezone) |
| Time | | Plain Text T |
| Location | | Plain Text T |
| Operator | | Plain Text T |
| Flight # | | Plain Text T |
| Route | | Plain Text T |
| Type | | Plain Text T |
| Registration | | Plain Text T |
| cn/In | | Plain Text T |
| Aboard | | Number # |
| Fatalities | | Number # |
| Ground | | Number # |
| Summary | | Plain Text T |

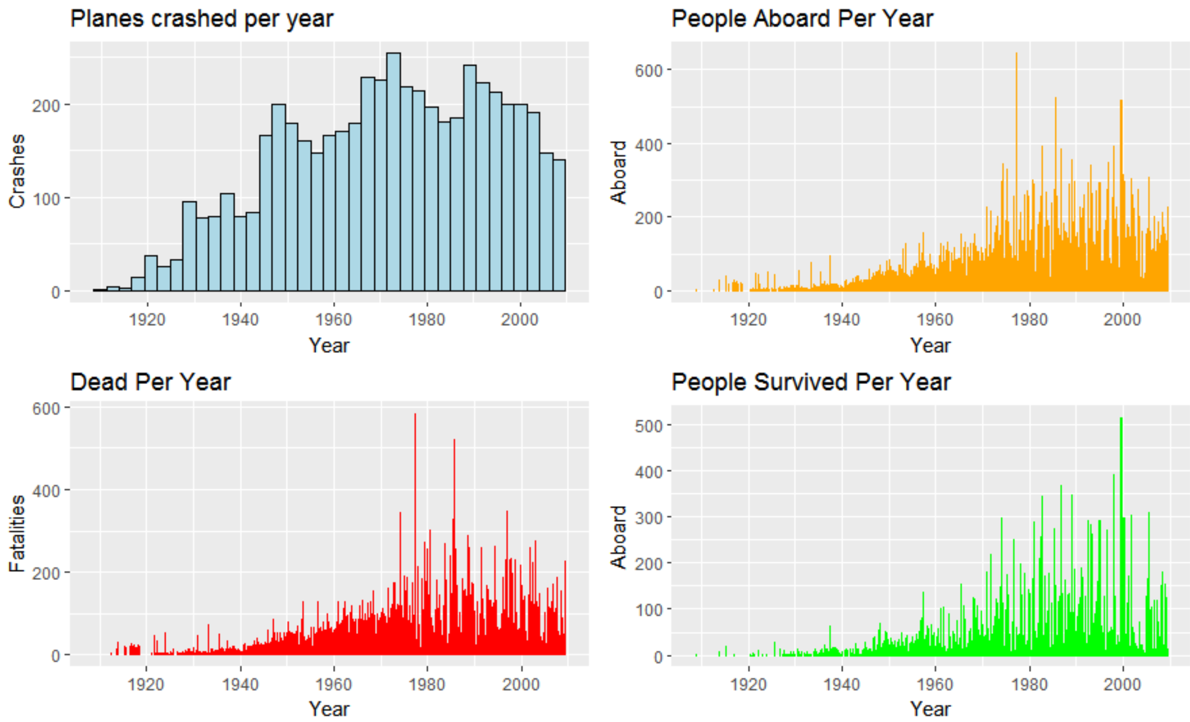
*Figure 1. Dataset attributes*

More information about the dataset can be found on the link provided above.

Basic EDA: The dataset is loaded in R, and after some pre-processing, some basic graphs are plotted for the dataset. For an in-depth data analysis of the dataset, refer to the R script attached in the folder (crash_EDA.R)

```
Air$Survived = Air$Aboard - Air$Fatalities
#Basic EDA
dead_per_year = ggplot(Air, aes(Date, Fatalities)) +
  geom_bar(na.rm = TRUE, stat="identity", position="identity", colour="red") +
  scale_x_date() + xlab("Year") + ylab("Fatalities") + ggtitle("Dead Per Year")
crash_per_year = ggplot(Air, aes(Date)) +
  geom_histogram(binwidth=1000, fill="lightblue", col="black") +
  scale_x_date() + xlab("Year") +
  ylab("Crashes") + ggtitle("Planes crashed per year")
aboard_per_year = ggplot(Air, aes(Date, Aboard)) +
  geom_bar(na.rm = TRUE, stat="identity", position="identity", colour="orange") +
  scale_x_date() + xlab("Year") + ylab("Aboard") + ggtitle("People Aboard Per Year")
survived_per_year = ggplot(Air, aes(Date, Survived)) +
  geom_bar(na.rm = TRUE, stat="identity", position="identity", colour="green") +
  scale_x_date() + xlab("Year") + ylab("Aboard") + ggtitle("People Survived Per Year")
grid.arrange(crash_per_year, aboard_per_year, dead_per_year, survived_per_year, ncol=2)
```

Which gives out the following plots:




The dataset, as seen from the description also, contains an attribute called 'Summary'. The idea here is to use the instances of the attribute to create a corpus and then apply k-means on the words found in the corpus, to see due to what reasons the flight crashed. The following code does the preprocessing:

```
> corpus = VCorpus(VectorSource(Air$Summary))
> corpus = tm_map(corpus, tolower)
> corpus = tm_map(corpus, PlainTextDocument)
> corpus = tm_map(corpus, removePunctuation)
> corpus = tm_map(corpus, removeWords, stopwords("english"))
> dtm = DocumentTermMatrix(corpus)
> dtm
<<DocumentTermMatrix (documents: 5268, terms: 9876)>>
Non-/sparse entries: 84276/51942492
Sparsity           : 100%
Maximal term length: 32
Weighting          : term frequency (tf)
> dtm = removeSparseTerms(dtm, 0.95)
```

As it can be observed that the document matrix is a sparse matrix, 95% sparsity of the matrix is removed and now it contains:

```
> str(dtm)
List of 6
 $ i       : int [1:23702] 1 1 1 1 1 1 1 2 2 3 ...
 $ j       : int [1:23702] 2 9 16 18 21 26 36 4 18 1 ...
 $ v       : num [1:23702] 1 1 1 2 1 1 1 1 1 1 ...
 $ nrow    : int 5268
 $ ncol    : int 41
 $ dimnames:List of 2
  ..$ Docs : chr [1:5268] "character(0)" "character(0)" "character(0)" "character(0)" ...
  ..$ Terms: chr [1:41] "accident" "aircraft" "airport" "altitude" ...
 - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

It contains some words that are too obvious, so again these are removed using stopwords removal, document matrix is filtered and the sparsity is removed, this time at 97%.

```
> dtm = DocumentTermMatrix(corpus, control = list(stopwords = c("aircraft", "plane", "crashed", "crash", "flight", "flew"
, "killed", "due", "resulted", "cause", "caused", "one", "two")))
> dtm
<<DocumentTermMatrix (documents: 5268, terms: 9863)>>
Non-/sparse entries: 75187/51883097
Sparsity           : 100%
Maximal term length: 32
Weighting          : term frequency (tf)
```

The 100 most frequent words in the document matrix were observed, and the results of this matrix is different from the previous one. Empty documents from the matrix are removed and data is pre-processed for k-means.

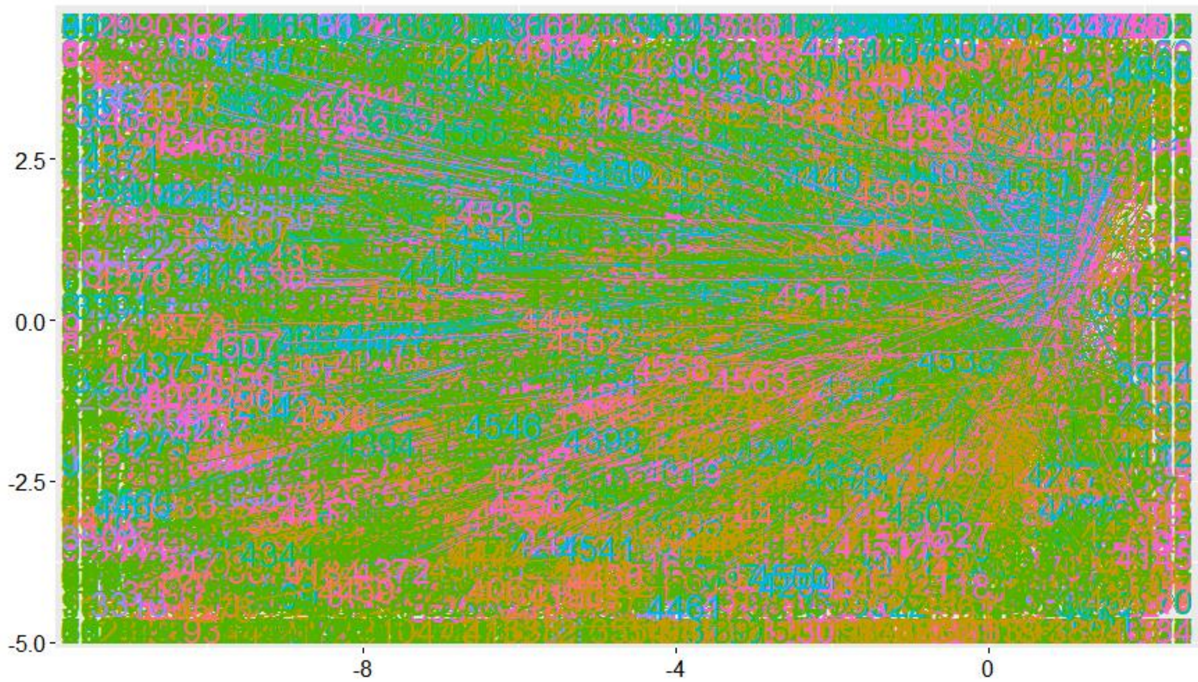
```
[1] "100 most frequent terms:"
> for( i in freq_terms)
+   cat(i, " ")
accident  air  airport  altitude  approach  area  attempting  cargo  conditions  continued  control  crew  descent  emerg
ency  engine  error  failed  failure  feet  fire  flames  flying  fog  fuel  ground  heavy  high  hit  improper  land  la
nding  left  loss  lost  low  maintain  miles  minutes  mountain  pilot  pilots  poor  power  rain  right  route  runway
 sea  short  shortly  stalled  struck  takeoff  taking  terrain  trees  turn  vfr  visibility  weather  wing
> nRows = apply(dtm , 1, sum)
> dtm = dtm[nRows> 0, ]
> dtm_tfxidf = weightTfIdf(dtm)
> m = as.matrix(dtm_tfxidf)
> rownames(m) = 1:nrow(m)
> preproc = preProcess(m)
> m_norm = predict(preproc, m)
```

k-means is implemented by taking the centre parameter as 7:

```
> cl = kmeans(m_norm, centers = 7, iter.max = 50, nstart = 10)
> print('Clusters:')
[1] "Clusters:"
> table(cl$cluster)

   1    2    3    4    5    6    7 
 292  586 2059  304  301  170  858 
```

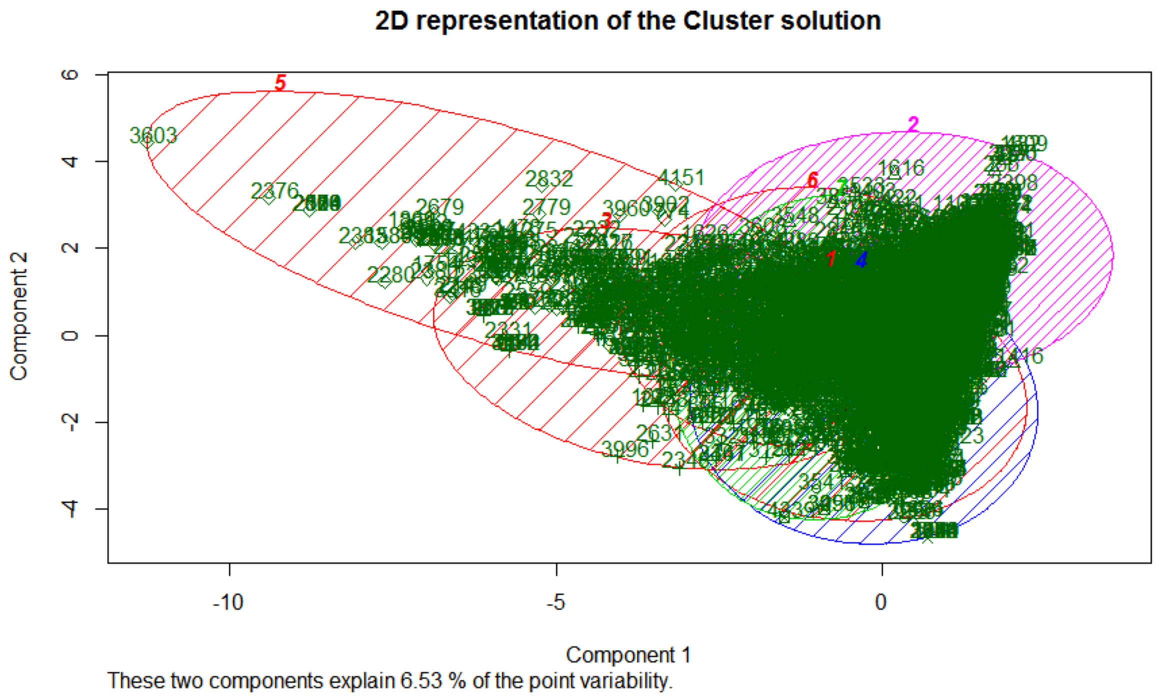
7 clusters are formed. The function fviz_cluster() is used to plot the cluster that is generated, but the result cannot be interpreted.



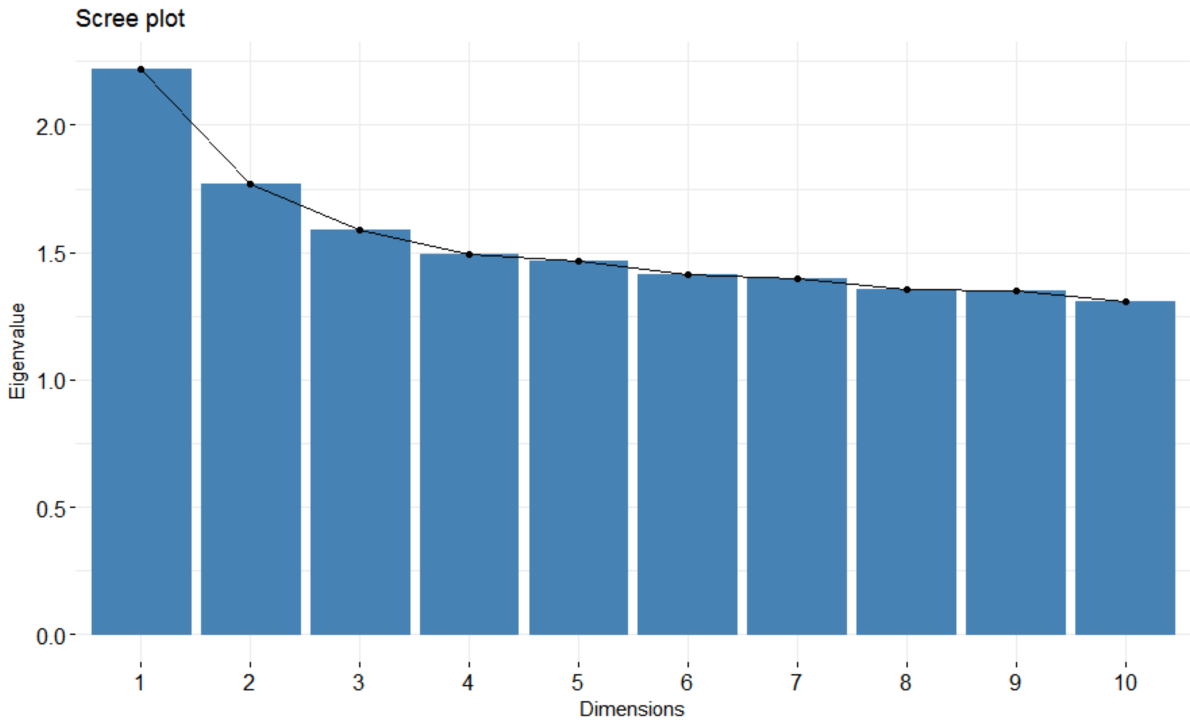
Here is the code snippet:

```
fviz_cluster(cl, data = m_norm, geom = "text", show.clust.cent = FALSE, repel = TRUE, labelsize = 20) +
  theme(legend.position = "none") +
  labs(title = "", x = "", y = "")
```

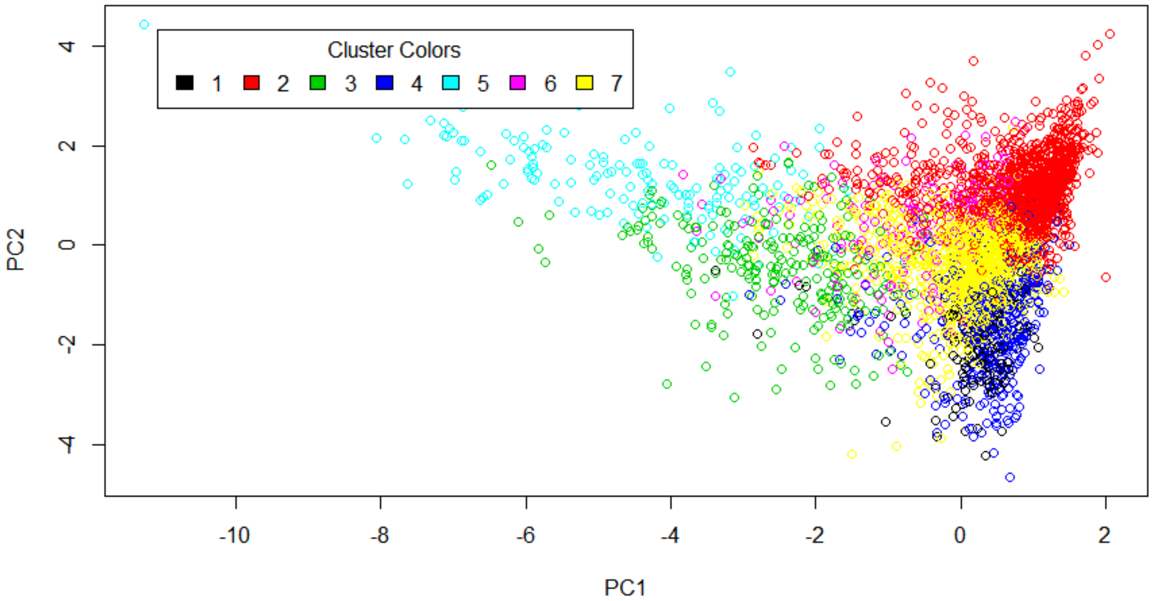
A second function is used to see the visualization created by k-means, but again the results cannot be interpreted.




Finally, principle component analysis is done on the dataset which is then used to visualize the clusters created.




Here, we can see some clarity in the clusters.

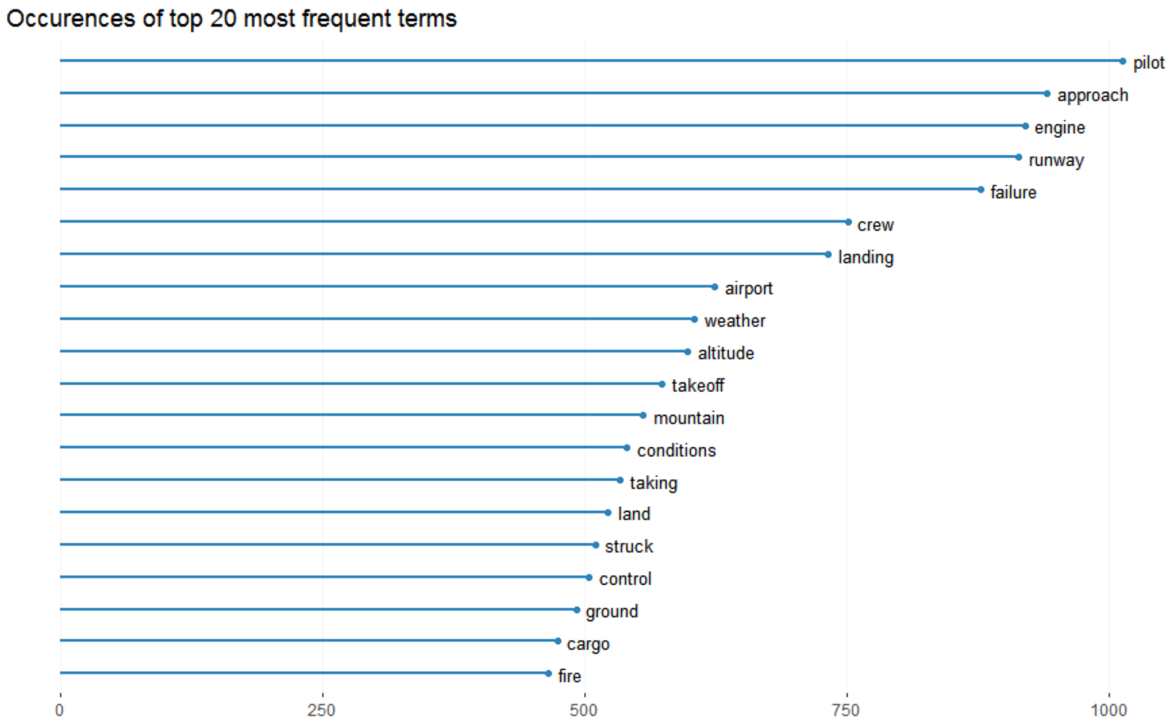



Now we determine the most frequent words in each of the clusters and also determine the number of fatalities. This is done so that there can be a better understanding as to what are the causes of the plane crash and which flight crashed due to a certain keyword or a combination of them is more fatal.

```
> freq_terms_1 = findFreqTerms(dtm[cl$cluster==1,], 50)
> freq_terms_2 = findFreqTerms(dtm[cl$cluster==2,], 50)
> freq_terms_3 = findFreqTerms(dtm[cl$cluster==3,], 50)
> freq_terms_4 = findFreqTerms(dtm[cl$cluster==4,], 50)
> freq_terms_5 = findFreqTerms(dtm[cl$cluster==5,], 50)
> freq_terms_6 = findFreqTerms(dtm[cl$cluster==6,], 50)
> freq_terms_7 = findFreqTerms(dtm[cl$cluster==7,], 50)
> print('50 most frequnt terms in cluster 1:')
[1] "50 most frequnt terms in cluster 1:"
> for( i in freq_terms_1)
+   cat(i, " ")
attempting  land
> print('50 most frequnt terms in cluster 2:')
[1] "50 most frequnt terms in cluster 2:"
> for( i in freq_terms_2)
+   cat(i, " ")
accident  air  airport  altitude  approach  area  attempting  cargo  conditions  control  crew  emergency  engine  failed
  failure  feet  fire  flames  flying  fuel  ground  improper  land  landing  left  loss  lost  maintain  miles  minutes
 pilot  pilots  power  right  route  runway  sea  shortly  stalled  struck  takeoff  taking  terrain  trees  turn  weathe
r  wing
> print('50 most frequnt terms in cluster 3:')
[1] "50 most frequnt terms in cluster 3:"
> for( i in freq_terms_3)
+   cat(i, " ")
approach  conditions  mountain  pilot  poor  visibility  weather
> print('50 most frequnt terms in cluster 4:')
[1] "50 most frequnt terms in cluster 4:"
> for( i in freq_terms_4)
+   cat(i, " ")
approach  cargo  crew  failure  land  landing  pilot  runway  short  takeoff
> print('50 most frequnt terms in cluster 5:')
[1] "50 most frequnt terms in cluster 5:"
> for( i in freq_terms_5)
+   cat(i, " ")
conditions  continued  pilot  vfr  weather
> print('50 most frequnt terms in cluster 6:')
[1] "50 most frequnt terms in cluster 6:"
> for( i in freq_terms_6)
+   cat(i, " ")
altitude  flying  low  pilot
```

```
> print('Fatalities in cluster 1:')
[1] "Fatalities in cluster 1:"
> sum(Air$Fatalities[which(cl$cluster==1)])
[1] 4428
> print('Fatalities in cluster 2:')
[1] "Fatalities in cluster 2:"
> sum(Air$Fatalities[which(cl$cluster==2)])
[1] 31224
> print('Fatalities in cluster 3:')
[1] "Fatalities in cluster 3:"
> sum(Air$Fatalities[which(cl$cluster==3)])
[1] 6048
> print('Fatalities in cluster 4:')
[1] "Fatalities in cluster 4:"
> sum(Air$Fatalities[which(cl$cluster==4)])
[1] 9420
> print('Fatalities in cluster 5:')
[1] "Fatalities in cluster 5:"
> sum(Air$Fatalities[which(cl$cluster==5)])
[1] 3521
> print('Fatalities in cluster 6:')
[1] "Fatalities in cluster 6:"
> sum(Air$Fatalities[which(cl$cluster==6)])
[1] 4635
> print('Fatalities in cluster 7:')
[1] "Fatalities in cluster 7:"
> sum(Air$Fatalities[which(cl$cluster==7)])
[1] 33420
```

To complete this first semantic analysis, we can look at the most frequent terms, and their correlation with other terms. We begin by plotting the 20 most frequent terms. All of them are obviously included in the above cluster analysis, but here we get a sense of their frequency relatively to each other.




Some hypothesis we can do on the top 5 terms:

- Pilot: is it only because this is a generic term, or indicating that pilot is the cause?
- Approach: this suggest that accidents often happen in the runway approach phase
- Engine: probably one of the most common causes
- Runway: relates to the approach phase
- Failure: this is too generic to draw conclusions, we'll some more context

To add more context to the list, we have to look at which terms are most correlated with these 20 frequent terms.

```
### Terms correlation
assocs <- findAssocs(dtm, as.character(freq[1:20, 1]), corlimit = 0.17)
print(assocs)
```

```
$pilot
turn
0.17

$approach
descent   short
   0.23    0.22

$engine
    right     power     left emergency     loss    failed
     0.26      0.25     0.24      0.23     0.21      0.19

$runway
short
  0.4
```

```
$failure
maintain   pilots accident 
    0.27     0.23     0.21 

$crew
numeric(0)

$landing
emergency 
     0.31 

$airport
miles 
 0.21 

$weather
     poor       vfr continued 
     0.41      0.33      0.24 

$altitude
    feet maintain      low  descent 
    0.21     0.20     0.19     0.18 

$takeoff
numeric(0)

$mountain
numeric(0)

$conditions
      vfr continued    pilots      poor   terrain 
     0.42      0.25      0.19      0.19      0.19 

$taking
shortly minutes 
   0.43    0.23 

$land
attempting 
      0.53 

$struck
numeric(0)

$control
loss lost 
0.46 0.23 

$ground
high 
0.17 

$cargo
numeric(0)

$fire
emergency      left 
     0.17      0.17 
```

This is quite enlightening. Let's look at some of the terms associations:

- Pilot: 'error' is one of the most correlated words, which is consistent with the fact that 60% of crashes are due to pilot errors
- Approach: the accidents in final approach phase seem to be often caused by confusion in reading instruments and low visibility ('ils', 'instruments', 'visual', 'missed')
- Engine seems related to shutdown of engine and/or loss of power
- Runway is associated with 'short', 'end' and 'overran', that could be as well in take-off or landing phases
- Failure: we have more context here, suggesting that it can be pilot, maintenance, procedure or system failures
- Landing: this shows that it is not necessarily about the standard landing phase, but rather about landing gears, or emergency landings
- Weather and Conditions suggest that visibility is one of the most important crashes factors in bad weather