BOSTON UNIVERSITY

AD699

Data Mining for Business Analytics

Spring 2021

Professor Greg Page

Assignment 5

Kunfei Chen

U15575304

Hierarchical Clustering

Task 1

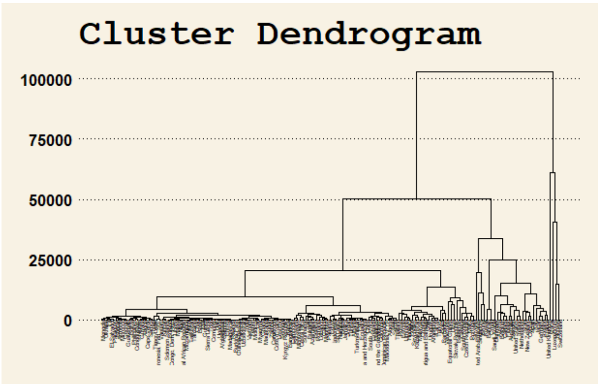
The data file "Country-data.csv" is downloaded from the class blackboard site and imported to R-studio via "read.csv ()" function.

Task 2

- **a**

Firstly, the Euclidean distance matrix for every pair of country is calculated via "dist ()" function. Then a hierarchical clustering model is built by "hclust ()" function with the "average" method and plotted by "fviz_dend ()" function (See Figure 1).

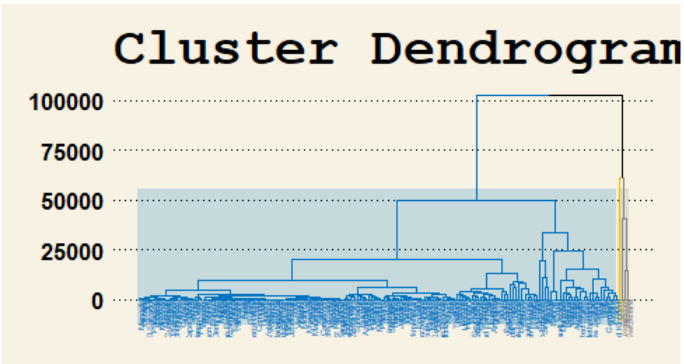
Figure 1



- **b**

As shown in Figure 2, by choosing the cutoff distance on the y-axis to be just above 50000, there will be 3 clusters.

Figure 2

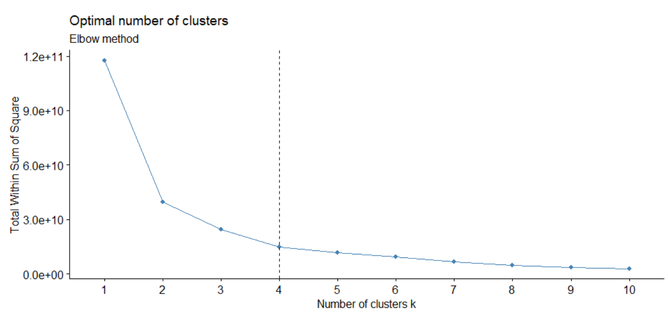


- **c**

The “fviz_nbclust ()” function is applied to identify the suitable number of clusters. According to “Elbow” method, the sum of squares for error (SSE) between each cluster’s centroid and its samples is called distortion. The distortion will decrease along with the increase of cluster numbers. Usually, the distortion will be greatly decreased at the critical point and then slowly decreases, this critical point can be considered as the point with better clustering performance. As shown in Figure 3, by setting “FUNcluster = hcut” in “fviz_nbclust ()” function, the decreasing trend significantly decrease at value 4, and after which the slop becomes low. Therefore, the optimal number of clusters is identified to be 4.

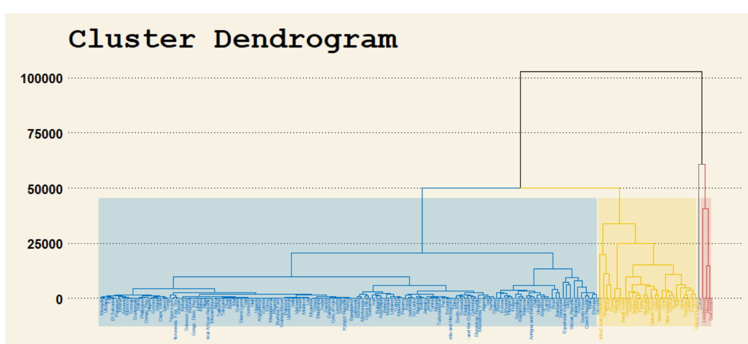
Figure 3

```
fviz_nbclust(data, FUNcluster = hcut, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



The cluster dendrogram is shown in Figure 4.

Figure 4



The resulting cluster assignments for each country is achieved by “cutree ()” function and shown in Figure 5.

Figure 5

	V1
Qatar	4
Luxembourg	3
Norway	3
Switzerland	3
Australia	2
Austria	2
Bahrain	2
Belgium	2
Brunei	2

- **d**

The assigned cluster number is then attached to the original dataset, and for each cluster its variables' mean values are generated by "group_by ()" function and "summarise_all ()" function (See Figure 4).

Figure 4

	cluster	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
1	1	45.705882	37.47279	6.448824	46.87254	9743.11	8.654228	68.38309	3.186176	5507.456
2	2	5.762963	52.88519	8.541481	44.75556	44577.78	3.915704	79.95185	1.919259	39937.037
3	3	3.500000	92.90000	9.583333	74.60000	69833.33	3.295667	81.50000	1.700000	89133.333
4	4	9.000000	62.30000	1.810000	23.80000	125000.00	6.980000	79.50000	2.070000	70300.000

In order to identify which variables strongly impact the cluster assignments, the standard variance of each column is calculated and ordered (See Figure 5). It can be seen that the "income", "gdpp", "exports", and "imports" are the four variables that greatly influence the clustering.

Figure 5

```
> impact_sort_1 <- sapply(summary_stats_1[,c(-1)],sd) %>% sort(decreasing=TRUE)
> impact_sort_1
      income         gdpp      exports      imports   child_mort   life_expec       health    inflation
4.852609e+04 3.659877e+04 2.336716e+01 2.084886e+01 1.993729e+01 6.028241e+00 3.446569e+00 2.538364e+00
   total_fer
6.625297e-01
```

Task 3

The dataset should be scaled because the computed distances are highly influenced by the scale of each variable. As analyzed before, the reason why variables "income", "gdpp", "exports" and "imports" have high standard variance is because of their high scales. Therefore, z-score normalization can eliminate the difference of variables on scales.

Task 4

The “scale ()” function is applied to standardize the data (See Figure 6).

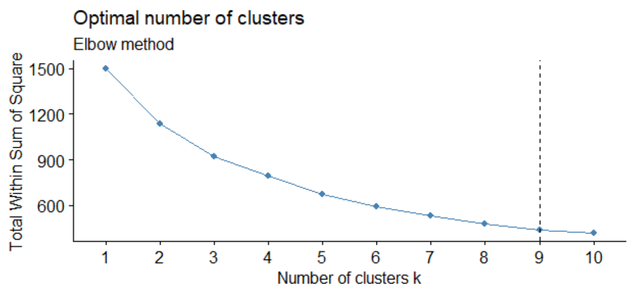
Figure 6

```
> # 4
> data_norm <- sapply(data, scale) %>% data.frame
> row.names(data_norm) <- row.names(data)
> data_norm
                      child_mort      exports      health     imports      income
Afghanistan           1.28765971 -1.1348666486  0.27825140 -0.08220771 -0.80582187
Albania              -0.53733286 -0.4782201668 -0.09672528  0.07062429 -0.37424335
Algeria              -0.27201464 -0.0988244217 -0.96317624 -0.63983800 -0.22018227
Angola                2.00178723  0.7730561847 -1.44372888 -0.16481961 -0.58328920
Antigua and Barbuda  -0.69354825  0.1601861350 -0.28603389  0.49607554  0.10142673
```

Task 5

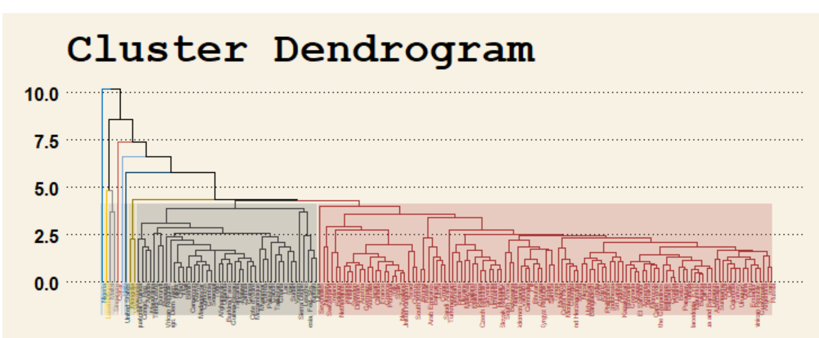
As for the new processed data, number 9 can be considered as the optimal number of clusters (See Figure 7).

Figure 7



A new hierarchical cluster dendrogram is created (See Figure 8). Compared with the formal dendrogram, there are so many differences. For example, the cluster number increases from 4 to 9. One possible reason could be that the common scale of each variable makes the distinguishability between them less significant.

Figure 8



In addition, Figure 9 and Figure 10 shows that after scaling, the top 4 variables that have high influence on clustering become “inflation”, “exports”, “income” and “imports”.

Figure 9

	cluster	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
1	1	1.2693657	-0.48793286	-0.16419681	-0.18605463	-0.7117564	0.2308320	-1.13446834	1.3778228	-0.6186544
2	2	-0.5198659	0.08179533	0.06191250	-0.01509469	0.1718676	-0.2092275	0.45008896	-0.5246547	0.1527641
3	3	4.2086397	-0.94152074	0.03433453	0.73565004	-0.8115278	-0.2205939	-4.32418141	0.2523609	-0.6711961
4	4	-0.8795190	4.88439277	0.34742185	3.92859974	3.8673643	-0.3937138	1.20815289	-0.8706055	5.0214047
5	5	-0.8299268	4.93911331	-0.18591876	4.83733058	1.7146590	-0.5571845	1.23064206	-1.1183187	1.1395156
6	6	-0.4133524	-0.12800871	-0.59730094	-0.40232880	-0.2614208	3.2891062	0.02747179	-0.2595797	-0.2667486
7	7	2.2745443	-0.57671714	-0.63552672	-1.21812126	-0.6221935	9.1023425	-1.13072014	1.9103878	-0.5801913
8	8	-0.7257832	0.77305618	-1.82234611	-0.95376320	5.5947159	-0.0758542	1.00575042	-0.5799554	3.1281995
9	9	-0.7679365	-1.04731378	4.03529928	-1.28421077	1.6731610	-0.6207564	0.91579376	-0.6724350	1.9333523

Figure 10

```
> clusters_2 <- cutree(dendrogram_2,k=9)
> View(clusters_2)
> data_2 <- data_norm
> data_2$cluster <- clusters_2
> summary_stats_2 <- data_2 %>% group_by(cluster) %>% summarise_all(c("mean"))
> View(summary_stats_2)
> impact_sort_2 <- sapply(summary_stats_2[,c(-1)],sd) %>% sort(decreasing=TRUE)
> impact_sort_2
inflation   exports    income   imports      gdpp child_mort life_expec    health
 3.207765  2.376501  2.267365  2.245096  1.994293   1.805009   1.801073  1.597827
total_fer
 1.044347
```

- **a**

As shown in Figure 11, country Norway is assigned to cluster 2 which is the second biggest cluster represented by gray color in Figure 8. Whereas in the first clustering, Norway was grouped with cluster 3 which is a very small cluster, in which besides Norway only Luxembourg and Switzerland are assigned in this cluster. One of the reasons for such a change could be that the common scale of each variable changes the order of importance on clustering, so as to change the results (See Figure 10).

Figure 11

	V1
Morocco	2
Namibia	2
Nepal	2
Netherlands	2
New Zealand	2
Norway	2
Oman	2
Panama	2
Paraguay	2
Peru	2

Task 6

As mentioned before, the units or scales of variables can have a huge influence on the clustering results. However, unequal weighting should be considered if we want the clusters to depend more on certain measurements and less on others. This is more practical because different variables should have different level of effects on clustering.

Task 7

There are totally 9 variables for each country. If these variables are weighted and the sum of weights is set to be 1, the average weight should be 1/9 = 0.11. Therefore, for the weight of each variable, if the value is greater than 1.11, it means the variable is relatively considered to be more important, otherwise, it is considered to be less important. From my perspective, the GDP per capita and total health spending per capita are the most two important variables for clustering, this is because great economy and health conditions can generally represent the will-being level of the country's people. Therefore, I assigned the following weights to each variable (See Figure 12).

Figure 12

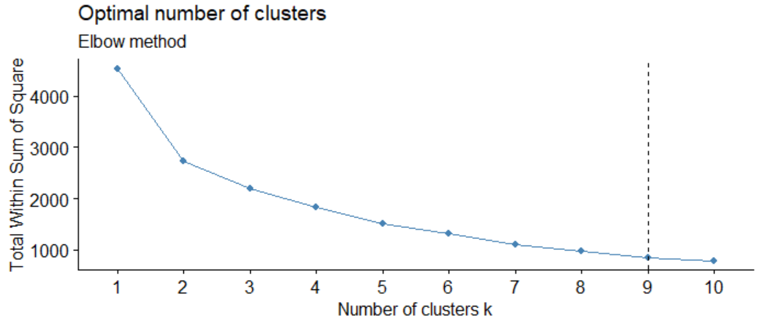
```
> data_3 <- data_norm
> data_3$gdpp <- data_3$gdpp * 3
> data_3$income <- data_3$income * 2
> data_3$inflation <- data_3$inflation * 2
> data_3$exports <- data_3$exports * 0.5
> data_3$imports <- data_3$imports * 0.5
> data_3$health <- data_3$health* 3
> data_3$total_fer <- data_3$child_mort * 0.5
> data_3$child_mort <- data_3$child_mort * 0.5
> data_3$life_expec <- data_3$life_expec * 0.5
> data_3
                         child_mort       exports      health      imports       income
Afghanistan             0.643829854 -0.5674333243  0.83475420 -0.041103857 -1.61164375
Albania                -0.268666428 -0.2391100834 -0.29017584  0.035312145 -0.74848670
Algeria                -0.136007321 -0.0494122109 -2.88952873 -0.319919000 -0.44036453
Angola                  1.000893617  0.3865280924 -4.33118664 -0.082409804 -1.16657839
```

Task 8

- **a**

Based on the newly-rescaled set of variables, the analysis of number of clusters still shows that value 9 is the optimal number (See Figure 13).

Figure 13



Then a new dendrogram and the cluster assignment result is generated (See Figure 14, Figure 15).

Figure 14

```
> dist_3 <- dist(data_3, method = "euclidean")
> dendrogram_3 <- hclust(dist_3,method = "average")
> fviz_dend(dendrogram_3, cex = 0.3, lwd = 0.7, k=9,
+           rect=TRUE, k_colors="jco",rect_border="jco",
+           rect_fill=TRUE, ggtheme = theme_wsj())
> clusters_3 <- cutree(dendrogram_3, k=9)
> view(clusters_3)
```

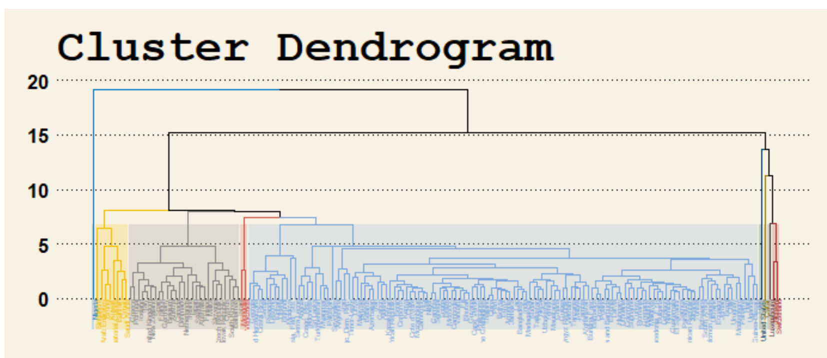


Figure 15

	V1
United States	9
Qatar	8
Norway	7
Switzerland	7
Nigeria	6
Mongolia	5
Venezuela	5
Luxembourg	4

- **b**

The assigned cluster number is then attached to the original dataset, and for each cluster its variables' mean values are generated by "group_by ()" function and "summarise_all ()" function (See Figure 16).

Figure 16

```
> data_3$cluster <- clusters_3
> summary_stats_3 <- data_3 %>% group_by(cluster) %>% summarise_all(c("mean"))
> View(summary_stats_3)
```

cluster	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
1	0.1140033	-0.10054370	-0.5183855	-0.01764176	-0.9066281	-0.0121352	-0.1535726	0.1140033	-1.4183481
2	-0.4167542	0.13846164	3.0105897	0.01611253	1.8969682	-1.2628787	0.5418148	-0.4167542	3.9135201
3	-0.1959829	0.80605223	-3.4028463	0.27075604	4.0699423	1.0226221	0.2681446	-0.1959829	2.6206180
4	-0.4397595	2.44219638	1.0422656	1.96429987	7.7347287	-0.7874277	0.6040764	-0.4397595	15.0642141
5	-0.2066762	-0.06400435	-1.7919028	-0.20116440	-0.5228417	6.5782124	0.0137359	-0.2066762	-0.8002457
6	1.1372721	-0.28835857	-1.9065802	-0.60906063	-1.2443870	18.2046851	-0.5653601	1.1372721	-1.7405740
7	-0.4267415	0.19591821	4.0129546	-0.12371575	4.3318980	-0.8794745	0.6209433	-0.4267415	11.1686851
8	-0.3628916	0.38652809	-5.4670383	-0.47688160	11.1894317	-0.1517084	0.5028752	-0.3628916	9.3845984
9	-0.3839683	-0.52365689	12.1058978	-0.64210539	3.3463220	-1.2415128	0.4578969	-0.3839683	5.8000570

The standard variance of each column shows that after the weighting system applied, the importance of variables "gdpp" and "health" are significantly improved respectively from the 5th to the 2nd and the 8th to the 3rd (See Figure 17).

Figure 17

```
> impact_sort_3 <- sapply(summary_stats_3[,c(-1)],sd) %>% sort(decreasing=TRUE)
> impact_sort_2
 inflation    exports     income    imports       gdpp child_mort life_expec     health 
  3.207765   2.376501   2.267365   2.245096   1.994293   1.805009   1.801073   1.597827 
 total_fer 
  1.044347 
> impact_sort_3
 inflation      gdpp     health     income    exports    imports child_mort  total_fer 
6.4134857 5.9813081 5.1938870 4.1607582 0.8801639 0.7901877 0.5075317 0.5075317 
life_expec 
 0.4091060 
```

- **c**

According to Figure 15, Norway is assigned to cluster 7 which is a group with small size. Switzerland is the only anoter country in this cluster. It can be seen from the Figure 16 that cluster 7 has relatively high total health spending per capita and high GDP per capita. This is why the two countries are assigned in cluster 7 and are known for their developed economy, health care, and social welfare.

In addition, annother finding is that United States is assigned to the cluster 9 by itsself due to its highest health spending per capita among all the countries.

Text Mining

Task 1

The gutenbergy package is lodaed into the R evnironment.

Task 2

text number of twice seed values which is 2*30 =60 is imported into the environment as well (See Figure 18).

Figure 18

```
> seed <- 30
> book <- gutenberg_download(seed*2)
```

Task 3

The “View ()” function shows that there are so many texts in the loaded object.

Figure 19

```
> # 3
> View(book)
```

	gutenberg_id	text
1	60	The Scarlet Pimpernel
2	60	
3	60	by Baroness Orczy
4	60	
5	60	
6	60	Contents
7	60	
8	60	I. PARIS: SEPTEMBER, 1792
9	60	II. DOVER: "THE FISHERMAN'S REST"
10	60	III. THE REFUGEES
11	60	IV. THE LEAGUE OF THE SCARLET PIMPERNEL
12	60	V. MARGUERITE
13	60	VI. AN EXQUISITE OF '92
14	60	VII. THE SECRET ORCHARD
15	60	VIII. THE ACCREDITED AGENT

Task 4

- **a**

The “unnest_tokens ()” function is then applied to the data (See Figure 20).

Figure 20

```
> words <- book %>% unnest_tokens(word, text)
> View(words)
```

- **b**

As shown in Figure 21, all the texts are converted into single words.

Figure 21

	gutenberg_id	word
1	60	the
2	60	scarlet
3	60	pimpernel
4	60	by
5	60	baroness
6	60	orczy

Task 5

- **a**

After using “anti_join ()” functon, the stopwords are removed. Then the reamined meaninful words are counted and the top 10 words are shown in Figuer 22.

Figure 22

```
> meaningful_words <- words %>% anti_join(stop_words)
Joining, by = "word"
> sorted_words <- meaningful_words %>% count(word, sort = TRUE)
> sorted_words[1:10,1:2]
# A tibble: 10 x 2
   word           n
   <chr>      <int>
 1 sir          343
 2 marguerite   325
 3 chauvelin    301
 4 percy        202
 5 andrew       178
 6 moment       163
 7 blakeney     159
 8 eyes         140
 9 lord         139
10 time         131
```

- **b**

The “unnest_tokens ()” function is applied with bigrams and the number of words is set to be 2. It can be seent from the Figure 23 that each bigram has two words.

Figure 23

```
> word_pairs <- book %>% unnest_tokens(bigram, text, token='ngrams',n=2) %>% drop_na()
> word_pairs
# A tibble: 77,511 x 2
   gutenberg_id bigram
          <int> <chr>
 1           60 the scarlet
 2           60 scarlet pimpernel
 3           60 by baroness
 4           60 baroness orczy
 5           60 i paris
 6           60 paris september
 7           60 september 1792
 8           60 ii dover
 9           60 dover the
10           60 the fisherman's
```

- **c**

The list of the most frequently-used words in Figure 22, to some extent, can reflect the topic of the article and the author’s writing style.

Task 6

- **a**

In order tot do sentiment analysis, the bing lexicon is joined with the counted words. The final top 10 words made the biggest sentiment contributions is the text is shown in Figure 24.

Figure 24

```
> words_sentiment_count <- sorted_words %>% inner_join(get_sentiments("bing"))
Joining, by = "word"
> top_sentiment <- words_sentiment_count %>% top_n(10,n)
> top_sentiment
# A tibble: 10 x 3
   word          n sentiment
   <chr>     <int> <chr>
 1 love         75 positive
 2 ready        59 positive
 3 death        54 negative
 4 doubt        47 negative
 5 stranger     46 negative
 6 loved        43 positive
 7 pretty       41 positive
 8 beautiful    39 positive
 9 fear         39 negative
10 smile        38 positive
```

- **b**

Of these top 10 words, 6 of them are positive and 4 of them are negative.

- **c**

The lis of top 10 words might indicateds that this article’s topic is related to different attitude towards life becaues the top list consist of words like “love”, “fear”, “smile” and “death”.

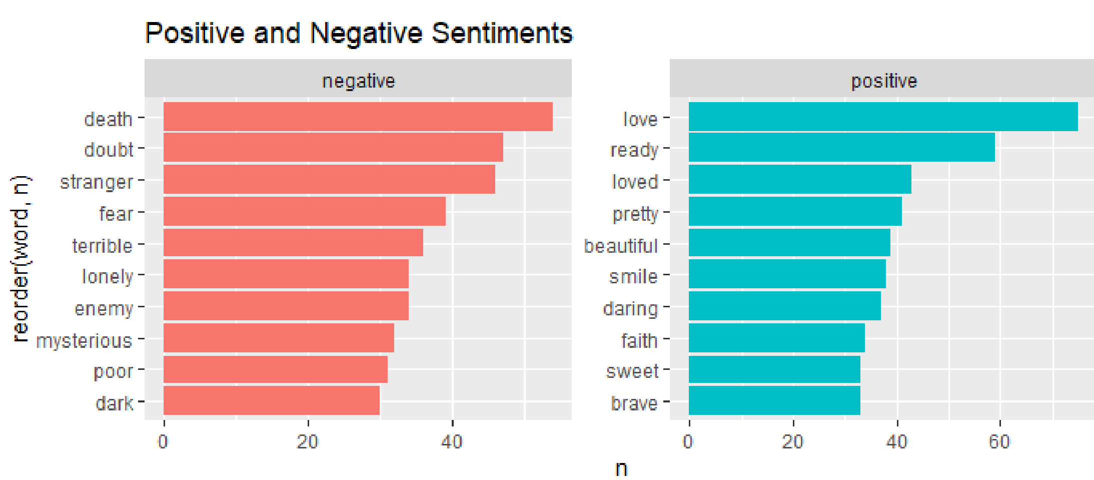
Task 7

- **a**

The top 10 negative words and 10 positive words are selected by “filter ()” function and ploted by “ggplot ()” function (See Figure 25). Generally, the count distribution of both top positive words and negative words are similar, but the value of positive words count is relatively higher than that of negative words. The top negative words consis of “death”, “doubt”, “stranger”, “fear”, “terrible”, “lonely”, “enemy”, “mysterious”, “poor”, “dark”. The top positive words consist of “love”, “ready”, “loved”, “pretty”, “beautiful”, “smile”, “daring”, “fath”, “sweet”, “brave”.

Figure 25

```
> ggplot(sentiment_20_words, aes(x=reorder(word,n), y=n, fill = sentiment)) + geom_col(show.legend = FALSE) +
+   facet_wrap(~sentiment, scales = "free") +  coord_flip() + ggtitle("Positive and Negative Sentiments")
> negative_words <- words_sentiment_count %>% filter(sentiment=="negative")
> negative_10_words <- negative_words[1:10,1:3]
> positive_words <- words_sentiment_count %>% filter(sentiment=="positive")
> positive_10_words <- positive_words[1:10,1:3]
> sentiment_20_words <- rbind(negative_10_words,positive_10_words)
> ggplot(sentiment_20_words, aes(x=reorder(word,n), y=n, fill = sentiment)) + geom_col(show.legend = FALSE) +
+   facet_wrap(~sentiment, scales = "free") +  coord_flip() + ggtitle("Positive and Negative Sentiments")
```



Task 8

- **a**

The afinn lexicon lexicon is loaded into the environment and joined with the meaningful words by “get_sentiments ()” function and “inner_join ()” function (See Figure 26). Then the sum of all the values is calculated to be 235.

Figure 26

```
> sentiment_score <- meaningful_words %>% inner_join(get_sentiments("afinn"))
Joining, by = "word"
> sum(sentiment_score$value)
[1] 235
```

- **b**

The sum of sentiment value equals 235 means that there are more positive words than negative words in this book, which to some extent can help to predict whether the topic of this book is more nagative or positive. However, this method might be incomplete or even mislaeading because it misses the efffect of negative sentence pattern. For example, if a sentence is like “The man is not lonely”, although the word “lonely” is negative, but the whole sentence conveys positive meanings.