39_Project2_Unit2

Tilak Poudel

2025-03-29

```r
setwd("/home/tilak/projects/tilak/mds1/R-programming/assignment/project2/MDS503P2")
```

Import covnep_252days.csv file into R studio as covnep_252days data frame

```r
# load the csv as data frame
covnep_252days <- read.csv("covnep_252days.csv")
```

```r
# Display the first few rows of the data frame
head(covnep_252days)
```

```
##        date totalCases newCases totalRecoveries newRecoveries totalDeaths
## 1 1/23/2020          1        1               0             0           0
## 2 1/24/2020          0        0               0             0           0
## 3 1/25/2020          0        0               0             0           0
## 4 1/26/2020          0        0               0             0           0
## 5 1/27/2020          0        0               0             0           0
## 6 1/28/2020          0        0               0             0           0
##   newDeaths
## 1         0
## 2         0
## 3         0
## 4         0
## 5         0
## 6         0
```

```r
# Check the structure of the data frame
str(covnep_252days)
```

```
## 'data.frame':    252 obs. of  7 variables:
##  $ date           : chr  "1/23/2020" "1/24/2020" "1/25/2020" "1/26/2020" ...
##  $ totalCases     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ newCases       : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ totalRecoveries: int  0 0 0 0 0 0 0 0 1 1 ...
##  $ newRecoveries  : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ totalDeaths    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ newDeaths      : int  0 0 0 0 0 0 0 0 0 0 ...
```

First, we loads the data from the CSV file into a dataframe called "covnep_252days". The head() function then displays the first six(default) rows of this dataframe, giving a nature of the dataset's structure. The str() function provides information about the structure of the dataframe, such as number of variables(7), the data types and the number of observations(252).

Covert the date (character date) variable as date variable (date2) using as.Date function (covnep_252days data frame)

```
# Convert the date variable to Date type
covnep_252days$date2 <- as.Date(covnep_252days$date, format = "%m/%d/%Y")

head(covnep_252days)
```

```
##        date totalCases newCases totalRecoveries newRecoveries totalDeaths
## 1 1/23/2020          1        1               0             0           0
## 2 1/24/2020          0        0               0             0           0
## 3 1/25/2020          0        0               0             0           0
## 4 1/26/2020          0        0               0             0           0
## 5 1/27/2020          0        0               0             0           0
## 6 1/28/2020          0        0               0             0           0
##   newDeaths      date2
## 1         0 2020-01-23
## 2         0 2020-01-24
## 3         0 2020-01-25
## 4         0 2020-01-26
## 5         0 2020-01-27
## 6         0 2020-01-28
```

```
# check the structure of the data frame again
str(covnep_252days)
```

```
## 'data.frame':    252 obs. of  8 variables:
##  $ date           : chr  "1/23/2020" "1/24/2020" "1/25/2020" "1/26/2020" ...
##  $ totalCases     : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ newCases       : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ totalRecoveries: int  0 0 0 0 0 0 0 0 1 1 ...
##  $ newRecoveries  : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ totalDeaths    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ newDeaths      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ date2          : Date, format: "2020-01-23" "2020-01-24" ...
```

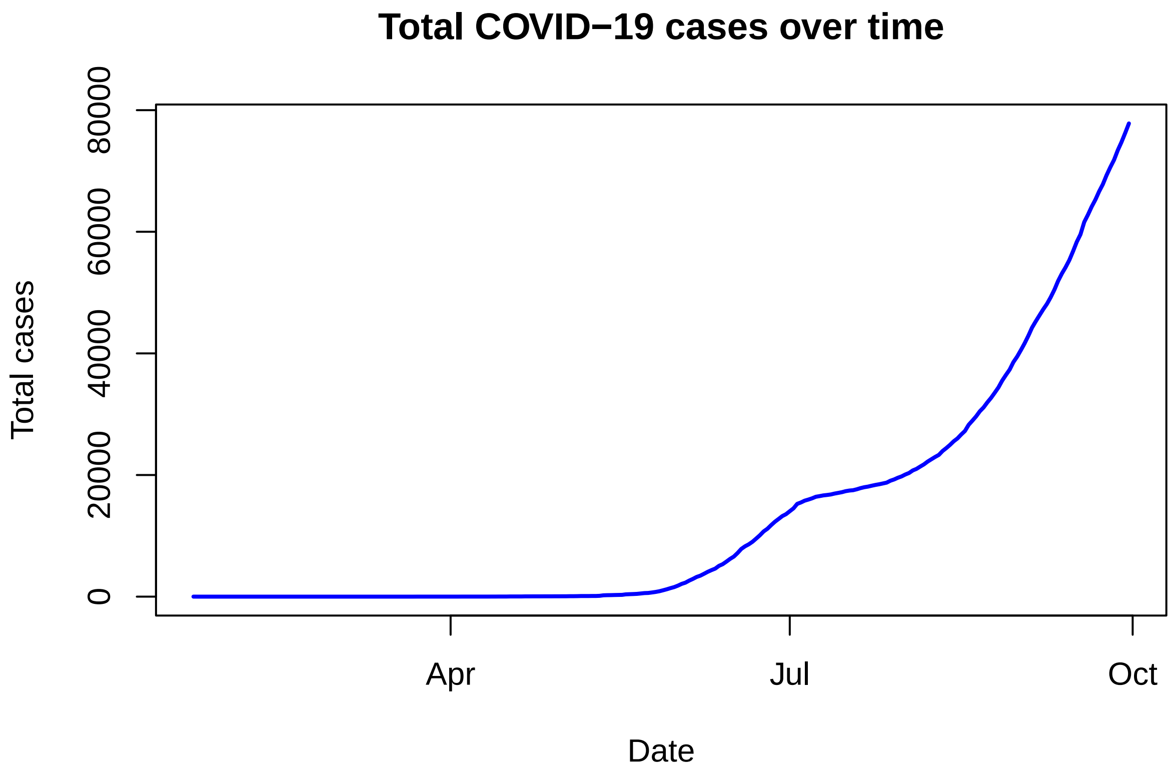
Now, we created a new column date2 of type Date converting from data type char. We can see the type of date2 is Date.

Create line chart of date2 and totalCases variables and interpret it carefully (covnep_252days data frame)

```
Sys.setlocale("LC_TIME", "en_US.UTF-8")  # set locale to english
```

```
## [1] "en_US.UTF-8"
```

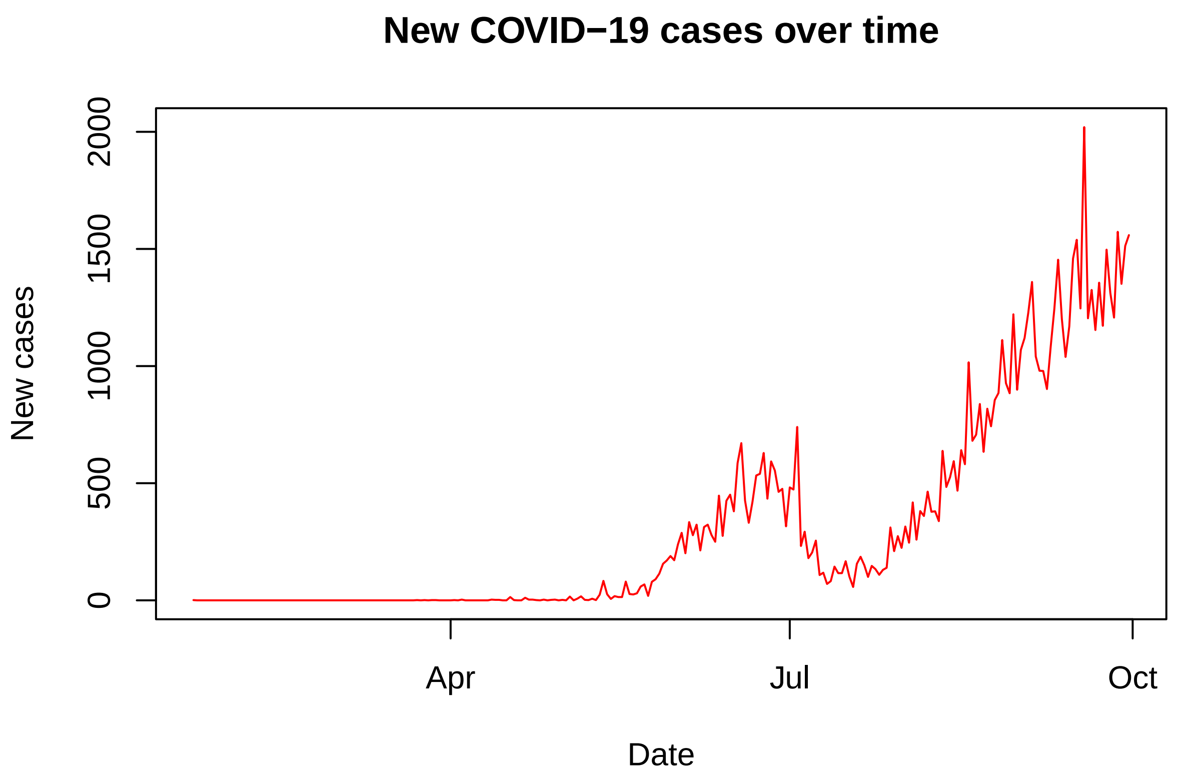
```
plot(
    covnep_252days$date2,
    covnep_252days$totalCases,
    type="l",
    col="blue",
    lwd=2,
    xlab="Date",
    ylab="Total cases",
    main="Total COVID-19 cases over time",
)
```



We can observe that the case seems to be almost none up to May, and has increased rapidly from end of August.

Create line chart of date2 and newCases variables and interpret it carefully (covnep_252days data frame)

```
plot(
    covnep_252days$date2,
    covnep_252days$newCases,
    type="l",
    col="red",
    lwd=1,
    xlab="Date",
    ylab="New cases",
    main="New COVID-19 cases over time",
)
```



Interpretation:

1. Early Period (Before April)

- Very few or no reported new cases.
- Possibly due to low transmission, limited testing, or early containment.

2. Gradual Increase (April – July)

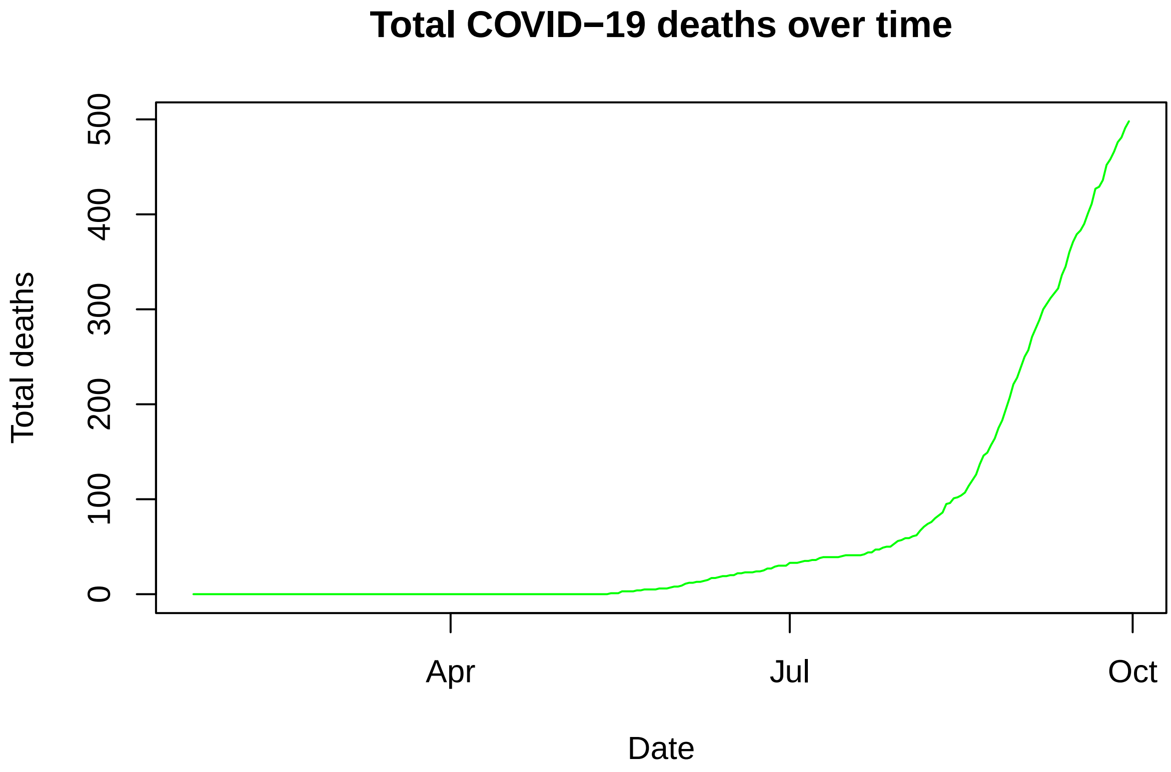
- The number of new cases starts rising.
- Fluctuations suggest intermittent outbreaks.

3. Rapid Surge (After July – October)

- A sharp increase in new cases, indicating widespread transmission.
- Peaks and dips suggest waves of infections.
- The highest peak exceeds 2000 cases per day, signaling a significant outbreak.

Create line chart of date2 and totalDeaths variables and interpret it carefully (covnep_252days data frame)

```
plot(
    covnep_252days$date2,
    covnep_252days$totalDeaths,
    type="l",
    col="green",
    lwd=1,
    xlab="Date",
    ylab="Total deaths",
    main="Total COVID-19 deaths over time",
)
```

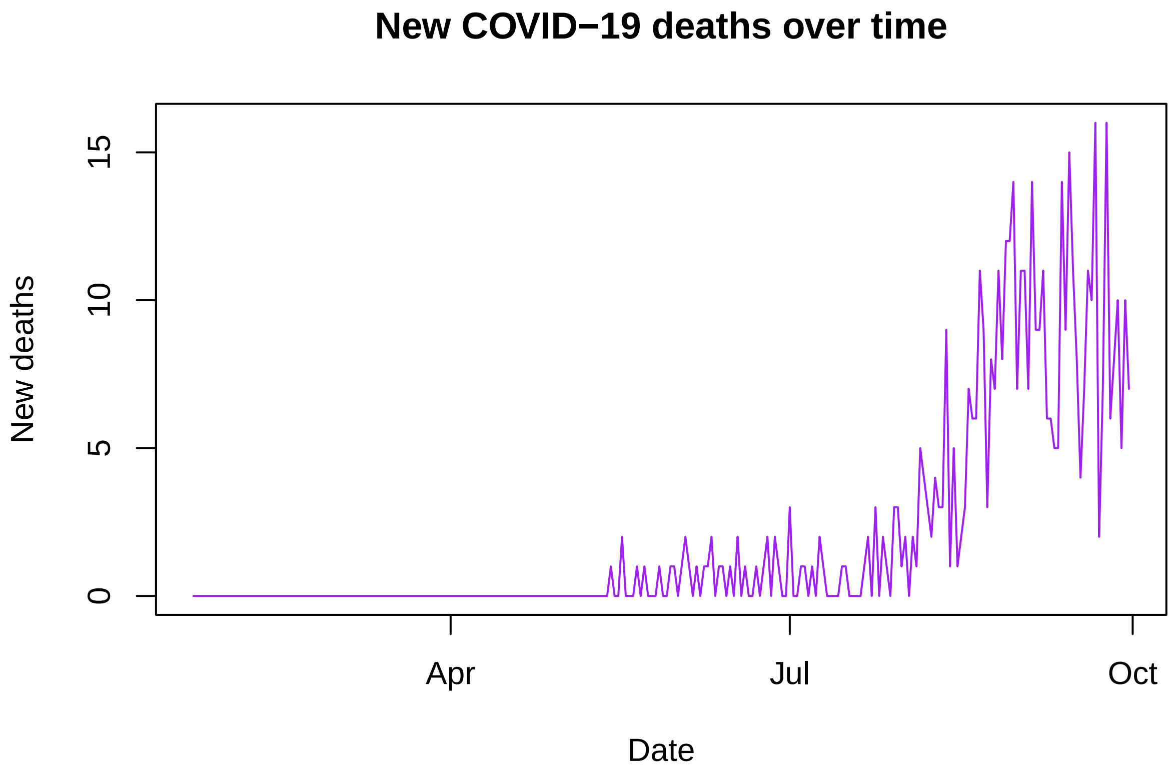


Interpretation:

We can say that the number of deaths are none till may and gradually increase till July. The total deaths seems to rise rapidly after July reaching more than 500 per day.

Create line chart of date2 and newDeaths variable and interpret it carefully (covnep_252days data frame)

```
plot(
    covnep_252days$date2,
    covnep_252days$newDeaths,
    type="l",
    col="purple",
    lwd=1,
    xlab="Date",
    ylab="New deaths",
    main="New COVID-19 deaths over time",
)
```



Key Observations from the plot It can be interpreted in 3 phases:

1. Early Period (Before April)

- No recorded COVID-19 deaths.
- Likely due to low infections, effective containment, or delays in reporting fatalities.

2. Gradual Increase (April – July)

- A small but noticeable rise in deaths.
- The fluctuations suggest periodic outbreaks, possibly linked to localized surges.

3. Significant Surge (After July – October)

- A sharp increase in deaths, correlating with the earlier observed rise in new cases.
- The highest peaks exceed 15 daily deaths, indicating a worsening outbreak.
- Large fluctuations suggest variability in fatality rates, possibly due to hospital capacity, treatment improvements, or reporting delays.

Compute summary measures of totalCases, newCases, totalRecoveries, newRecoveries, totalDeaths and newDeaths variables using an appropriate apply family of functions (covnep_252days data frame)

```
summary_measures <- sapply(covnep_252days[, c("totalCases", "newCases", "totalRecoveries", "newRecoveri
  c(
    mean = mean(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    sd = sd(x, na.rm = TRUE),
    min = min(x, na.rm = TRUE),
    max = max(x, na.rm = TRUE)
  )
})

summary_measures
```

```
##        totalCases  newCases totalRecoveries newRecoveries totalDeaths newDeaths
## mean     13376.33  308.7976        8380.341      223.3413     66.6746  1.976190
## median     963.00   82.5000         182.000        3.5000      6.0000  0.000000
## sd       19629.60  439.2922       13785.458      424.2460    122.7278  3.625857
## min          0.00    0.0000           0.000        0.0000      0.0000  0.000000
## max      77816.00 2020.0000       56282.000     2287.0000    498.0000 16.000000
```

Here sapply() function from apply function is used to compute the summary of the measures of totalCases, newCases, totalRecoveries, newRecoveries, totalDeaths and newDeaths. The null values are removed with na.rm and there after the mean, median, standard deviation, min and max is computed.

The data shows significant variation in total cases, recoveries, and deaths across observations, with a highly skewed distribution. While the average total cases are 13,376, the median is much lower (963), indicating that a few high-case regions are inflating the mean. Recoveries follow a similar trend, averaging 8,380 but with a median of 182. Deaths are relatively low, with an average of 67 but a median of 6, and most observations report zero new deaths. The high standard deviations confirm substantial disparities, with some locations experiencing extreme spikes in cases, recoveries, and deaths.

Import MR_Drugs.xlxs file into R studio as MR_Drugs data frame and create given table and interpret response percentage and percentage of cases carefully

```
library(readxl)
MR_Drugs <- read_excel("MR_Drugs.xlsx")

head(MR_Drugs)
```

```
## # A tibble: 6 x 27
##      id   sex  city inco1 inco2 inco3 inco4 inco5 inco6 inco7 pinco1 pinco2
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1  1001     2     1     0     0     0     0     0     1     0      6     -1
## 2  1002     2     1     0     1     0     0     0     0     0      2     -1
## 3  1003     2     1     0     0     0     0     0     1     0      6     -1
## 4  1004     2     1     0     1     0     0     0     0     0      2     -1
## 5  1005     2     1     0     0     0     0     0     0     1      7     -1
## 6  1006     2     1     1     1     0     0     0     0     0      2      1
## # i 15 more variables: pinco3 <dbl>, pinco4 <dbl>, pinco5 <dbl>, pinco6 <dbl>,
## #   sinco1 <chr>, sinco2 <chr>, sinco3 <chr>, sinco4 <chr>, sinco5 <chr>,
## #   sinco6 <chr>, crime1 <dbl>, crime2 <dbl>, crime3 <dbl>, crime4 <dbl>,
## #   crime5 <dbl>
```

```
# income columns
drugs_data <- MR_Drugs[, c("inco1", "inco2", "inco3", "inco4", "inco5", "inco6", "inco7")]
# get sum of every column
colSums(drugs_data)
```

```
## inco1 inco2 inco3 inco4 inco5 inco6 inco7
##   226   607   293    50    82   151   352
```

```
# total sum of all columns
total_sum <- sum(colSums(drugs_data))
total_sum
```

```
## [1] 1761
```

```
# get percentage of each column across whole data
each_column_percentage_on_all_data <- round(as.numeric(colSums(drugs_data) / total_sum * 100), 1)
each_column_percentage_on_all_data
```

```
## [1] 12.8 34.5 16.6  2.8  4.7  8.6 20.0
```

```
# get percentage of each column across whole data
round(as.numeric(colSums(!is.na(drugs_data)) / total_sum * 100), 1)
```

```
## [1] 55.2 55.2 55.2 55.2 55.2 55.2 55.2
```

```
# column names
names(drugs_data)
```

```
## [1] "inco1" "inco2" "inco3" "inco4" "inco5" "inco6" "inco7"
```

```
colSums(!is.na(drugs_data))
```

```
## inco1 inco2 inco3 inco4 inco5 inco6 inco7
##   972   972   972   972   972   972   972
```

```
levels <- c(names(drugs_data))
levels
```

```
## [1] "inco1" "inco2" "inco3" "inco4" "inco5" "inco6" "inco7"
```

```
income_frequencies <- data.frame(
  levels = c(names(drugs_data)),
  N = colSums(drugs_data),
  Percent = round(as.numeric(colSums(drugs_data) / (total_sum) * 100), 1),
  Percent_of_cases = round(as.numeric(colSums(drugs_data) / colSums(!is.na(drugs_data)) * 100), 1)
)

income_frequencies
```

```
##       levels   N Percent Percent_of_cases
## inco1  inco1 226    12.8             23.3
## inco2  inco2 607    34.5             62.4
## inco3  inco3 293    16.6             30.1
## inco4  inco4  50     2.8              5.1
## inco5  inco5  82     4.7              8.4
## inco6  inco6 151     8.6             15.5
## inco7  inco7 352    20.0             36.2
```

```
row.names(income_frequencies) <- NULL
total <- c(
  "Total",
  sum(as.numeric(income_frequencies$N)),
  sum(as.numeric(income_frequencies$Percent)),
  sum(as.numeric(income_frequencies$Percent_of_cases))
)
income_frequencies <- rbind(income_frequencies, total)
income_frequencies
```

```
##   levels    N Percent Percent_of_cases
## 1  inco1  226    12.8             23.3
## 2  inco2  607    34.5             62.4
## 3  inco3  293    16.6             30.1
## 4  inco4   50     2.8              5.1
## 5  inco5   82     4.7              8.4
## 6  inco6  151     8.6             15.5
## 7  inco7  352      20             36.2
## 8  Total 1761     100              181
```

In the R code, we used the readxl library to read data from an Excel file (MR_Drugs.xlsx) into a dataframe called MR_Drugs. We then used the head() function to view the first six rows of the dataset. Next, we extracted specific income-related columns (inco1 to inco7) into a new dataframe, drugs_data, to focus on relevant data. Using the colSums() function, we calculated the total occurrences for each income category. To better understand the distribution, we normalized these counts into percentages relative to the total sum, both within income categories and across the dataset. To organize our findings, we created a dataframe, income_frequencies, which included income levels, total counts, percentages, and percentages of cases. Finally, we added a total row to summarize the overall counts and percentages across all income levels.

Import SAQ.sav file into R studio as SAQ data frame and create given tables and interpret each frequency table carefully

```
library(haven)
suppressWarnings(library(summarytools))

SAQ8 <- read_sav("SAQ8.sav")
head(SAQ8)
```

```
## # A tibble: 6 x 8
##   q01                 q02         q03     q04     q05     q06     q07     q08    
##   <dbl+lbl>           <dbl+lbl>   <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l>
## 1 2 [Agree]           1 [Strongl~ 4 [Dis~ 2 [Agr~ 2 [Agr~ 2 [Agr~ 3 [Nei~ 1 [Str~
## 2 1 [Strongly agree]  1 [Strongl~ 4 [Dis~ 3 [Nei~ 2 [Agr~ 2 [Agr~ 2 [Agr~ 2 [Agr~
## 3 2 [Agree]           3 [Neither] 2 [Agr~ 2 [Agr~ 4 [Dis~ 1 [Str~ 2 [Agr~ 2 [Agr~
## 4 3 [Neither]         1 [Strongl~ 1 [Str~ 4 [Dis~ 3 [Nei~ 3 [Nei~ 4 [Dis~ 2 [Agr~
## 5 2 [Agree]           1 [Strongl~ 3 [Nei~ 2 [Agr~ 2 [Agr~ 3 [Nei~ 3 [Nei~ 2 [Agr~
## 6 2 [Agree]           1 [Strongl~ 3 [Nei~ 2 [Agr~ 4 [Dis~ 4 [Dis~ 4 [Dis~ 2 [Agr~
```

```
# check the structure of the data frame
str(SAQ8)
```

```
## tibble [2,571 x 8] (S3: tbl_df/tbl/data.frame)
##  $ q01: dbl+lbl [1:2571] 2, 1, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 1, 2,...
##    ..@ label      : chr "Statistics makes me cry"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:6] 1 2 3 4 5 9
##    .. ..- attr(*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q02: dbl+lbl [1:2571] 1, 1, 3, 1, 1, 1, 3, 2, 3, 4, 1, 1, 1, 2, 2, 1, 2, 2,...
##    ..@ label      : chr "My friends will think I'm stupid for not being able to cope with SPSS"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q03: dbl+lbl [1:2571] 4, 4, 2, 1, 3, 3, 3, 3, 1, 4, 5, 3, 3, 1, 3, 2, 5, 3,...
##    ..@ label      : chr "Standard deviations excite me"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q04: dbl+lbl [1:2571] 2, 3, 2, 4, 2, 2, 2, 2, 4, 3, 2, 3, 4, 2, 4, 2, 2, 3,...
##    ..@ label      : chr "I dream that Pearson is attacking me with correlation coefficients"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:6] 1 2 3 4 5 9
##    .. ..- attr(*, "names")= chr [1:6] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q05: dbl+lbl [1:2571] 2, 2, 4, 3, 2, 4, 2, 2, 5, 2, 2, 4, 3, 2, 2, 2, 1, 3,...
##    ..@ label      : chr "I don't understand statistics"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q06: dbl+lbl [1:2571] 2, 2, 1, 3, 3, 4, 2, 2, 3, 1, 1, 3, 2, 2, 2, 2, 1, 4,...
##    ..@ label      : chr "I have little experience of computers"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q07: dbl+lbl [1:2571] 3, 2, 2, 4, 3, 4, 2, 2, 5, 2, 2, 3, 3, 3, 3, 2, 1, 3,...
##    ..@ label      : chr "All computers hate me"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
##  $ q08: dbl+lbl [1:2571] 1, 2, 2, 2, 2, 2, 2, 2, 5, 2, 2, 1, 3, 2, 2, 2, 1, 2,...
##    ..@ label      : chr "I have never been good at mathematics"
##    ..@ format.spss: chr "F1.0"
##    ..@ labels     : Named num [1:5] 1 2 3 4 5
##    .. ..- attr(*, "names")= chr [1:5] "Strongly agree" "Agree" "Neither" "Disagree" ...
```

```
# find the frequency of each column
freq(SAQ8$q01, cumul = TRUE, round.digits = 1)
```

```
## Frequencies
## SAQ8$q01
## Label: Statistics makes me cry
## Type: Numeric (labelled)
##
##                               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## --------------------------- ------ --------- -------------- --------- --------------
##          Strongly agree [1]    270      10.5           10.5      10.5           10.5
##                   Agree [2]   1338      52.0           62.5      52.0           62.5
##                 Neither [3]    735      28.6           91.1      28.6           91.1
##                Disagree [4]    187       7.3           98.4       7.3           98.4
##       Strongly disagree [5]     41       1.6          100.0       1.6          100.0
##            Not answered [9]      0       0.0          100.0       0.0          100.0
##                        <NA>      0                                0.0          100.0
##                       Total   2571     100.0          100.0     100.0          100.0
```

```
freq(SAQ8$q02, cumul = TRUE, round.digits = 1)
```

```
## Frequencies
## SAQ8$q02
## Label: My friends will think I’m stupid for not being able to cope with SPSS
## Type: Numeric (labelled)
##
##                               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## --------------------------- ------ --------- -------------- --------- --------------
##          Strongly agree [1]   1436      55.9           55.9      55.9           55.9
##                   Agree [2]    808      31.4           87.3      31.4           87.3
##                 Neither [3]    206       8.0           95.3       8.0           95.3
##                Disagree [4]    101       3.9           99.2       3.9           99.2
##       Strongly disagree [5]     20       0.8          100.0       0.8          100.0
##                        <NA>      0                                0.0          100.0
##                       Total   2571     100.0          100.0     100.0          100.0
```

```
freq(SAQ8$q03, cumul = TRUE, round.digits = 1)
```

```
## Frequencies
## SAQ8$q03
## Label: Standard deviations excite me
## Type: Numeric (labelled)
##
##                               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## --------------------------- ------ --------- -------------- --------- --------------
##           Strongly agree [1]    497      19.3           19.3      19.3           19.3
##                    Agree [2]    672      26.1           45.5      26.1           45.5
##                  Neither [3]    878      34.2           79.6      34.2           79.6
##                 Disagree [4]    448      17.4           97.0      17.4           97.0
##        Strongly disagree [5]     76       3.0          100.0       3.0          100.0
##                         <NA>      0                                0.0          100.0
##                        Total   2571     100.0          100.0     100.0          100.0
```

Here, we use the ‘haven’ and ‘summarytools’ libraries to analyze data from a SPSS (.sav) file named “SAQ8.sav”. After loading the data into a dataframe called “SAQ8”, the head() function displays the first six(default) rows of the dataset. The str() function provides additional information about the dataframe’s structure. The freq() function is then employed multiple times to generate frequency tables for different variables within the dataset. Each frequency table displays the count and percentage of responses for a specific question or variable, along with cumulative percentages as cumul = TRUE. The round.digits argument controls the precision of the percentages displayed in the frequency tables.

Text mining and word cloud generation

```
old_directory <- getwd()

suppressWarnings({
    library(pdftools)
    library(tm)
    library(magrittr)
    library(Rgraphviz)
    library(wordcloud)
})
```

```
## Using poppler version 22.02.0

## Loading required package: NLP

## Loading required package: graph

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:NLP':
##
##     annotation

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: grid
```

```
## Loading required package: RColorBrewer
```

```
files <- list.files(pattern = "\\.pdf$", full.names = TRUE)
length(files)
```

```
## [1] 10
```

```
pdf_files <- lapply(files, pdf_text)
corpus <- Corpus(VectorSource(unlist(pdf_files)))
inspect(corpus[1])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
## [1] 60                                   JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL.
```

```
corpus_copy <- corpus
suppressWarnings({
  corpus <- tm_map(corpus, tolower)
})

inspect(corpus[1:2])
```

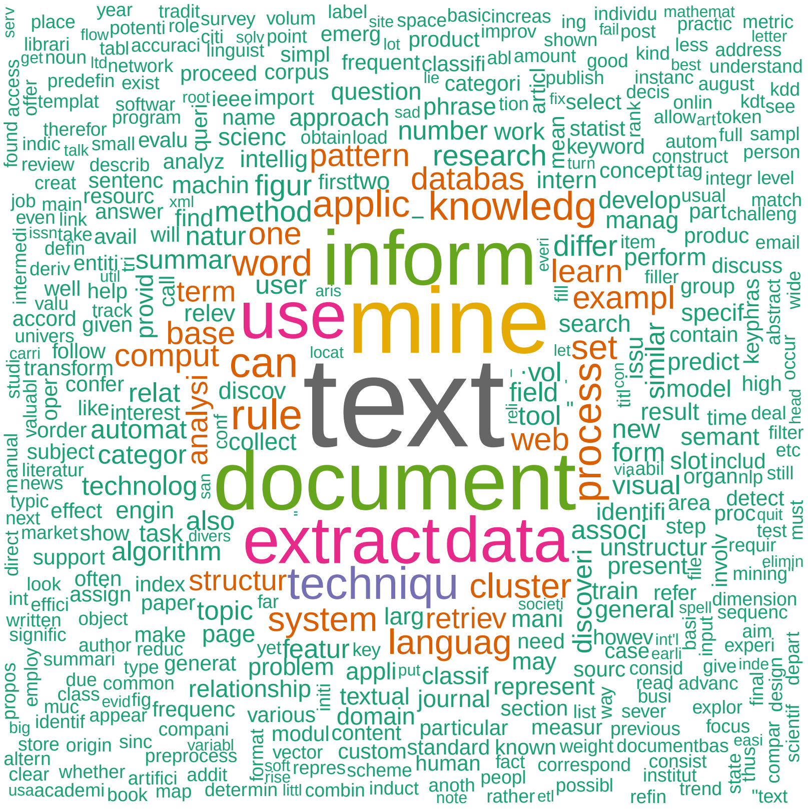
```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 2
##
## [1] 60                                   journal of emerging technologies in web intelligence, vol.
```

```
suppressWarnings({
    corpus <- tm_map(corpus, removePunctuation)
    corpus <- tm_map(corpus, removeWords, stopwords("en"))
    corpus <- tm_map(corpus, stripWhitespace)
    corpus <- tm_map(corpus, removeNumbers)
    corpus <- tm_map(corpus, stemDocument)
})

corpus_copy <- corpus
```

```r
tdm <- TermDocumentMatrix(corpus, control = list(wordLengths = c(3, Inf)))
low_frequent_terms <- findFreqTerms(tdm, lowfreq = 25)

head(low_frequent_terms)
```

```
## [1] "allow"   "also"    "analysi" "anoth"   "answer"  "appli"
```

```r
suppressWarnings({
    mat <- as.matrix(tdm)
    freq <- mat %>% rowSums() %>%
    sort(decreasing = TRUE)
    wordcloud(
      words = names(freq),
      freq = freq,
      min.freq = 5,
      random.order = FALSE,
      colors = brewer.pal(8, "Dark2")
    )
})
```



The word cloud visually represents the frequency of words related to text mining, information retrieval, and natural language processing. The most prominent words (largest in size) indicate their higher frequency and importance in the data set.

`text`, `document`, `mine`, `extract`, `inform`, `use`, `data` are the most frequent words and suggest that the primary topic revolves around text mining, document processing, and extracting useful information from text data.

`analysis`, `rule`, `process`, `knowledge`, `language`, `pattern`, `retrieval`, `classification` are the significant related terms. These words indicate key techniques and objectives in text mining, such as analyzing textual patterns, retrieving relevant information, and applying rules or classifications.

`sentiment`, `vector`, `model`, `corpus`, `journal`, `approach`, `visual`, `automatic`, `structure` are less frequent but relevant terms. These words hint at advanced techniques in text analysis, such as sentiment analysis, vector space modeling, and structured information extraction.