

Project 2: Unit 2

Suman Paudel 33

2024-03-26

Contents

Part 1	1
Part 2	10
Part 3	84

Part 1

Task 1

Load all of the necessary packages need for task 1.

```
# Load the packages
library(foreign)
library(gt)
library(tidyverse)
library(magrittr)
library(readxl)
```

Load the Data using CSV module from base R

```
# load the data using Base R read.csv
data <- read.csv("covnep_252days.csv")
```

```
summary(data$totalCases)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         0         2     963   13376   19341   77816
```

Since we need value as 1 instead of zero We can achieve this using multiple ways like ifelse or pmax or subsetting

Using ifelse

```
# using ifelse
totalCases_ifelse <- ifelse(data$totalCases < 1, 1, data$totalCases)
summary(totalCases_ifelse)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##           1         2     963   13377   19341   77816
```

Using pmax

```
# using pmax
totalCases_pmax <- pmax(data$totalCases, 1)
summary(totalCases_pmax)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##           1         2     963   13377   19341   77816
```

Using subsetting

```
# subsetting
totalCases_subsetting <- data$totalCases
totalCases_subsetting[totalCases_subsetting < 1] <- 1
summary(totalCases_subsetting)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##           1         2     963   13377   19341   77816
```

Task 2

Read the .sav file using foreign library's read.spss function

For q01

```
# read the .sav file using read_sav function from haven
saq_data <- read.spss("SAQ8.sav",to.data.frame=TRUE)

# for q1
q01 <- saq_data$q01

# computer mathematical operations
datalevels_q01 <- levels(q01)
freq_q01 <- as.numeric(table(q01))
percent_q01 <- as.numeric(round(prop.table(freq_q01) * 100, 1))
valid_percent_q01 <- as.numeric(round(prop.table(freq_q01) * 100, 1))
cum_percent <- cumsum(percent_q01)

# Create data frame
data <- data.frame(
  Levels = datalevels_q01,
  Freq = freq_q01,
  Percent = percent_q01,
```

```

    Val_Percent = valid_percent_q01,
    Cum_Percent = cum_percent
  )

head(data)

```

Levels	Freq	Percent	Val_Percent	Cum_Percent
Strongly agree	270	10.5	10.5	10.5
Agree	1338	52.0	52.0	62.5
Neither	735	28.6	28.6	91.1
Disagree	187	7.3	7.3	98.4
Strongly disagree	41	1.6	1.6	100.0

```

# final version of calculated table for q01
data <- data %>% add_row(Levels = "Total", Freq = sum(data$Freq),
  Percent = sum(data$Percent),
  Val_Percent = sum(data$Val_Percent),
  Cum_Percent = NULL)

# aesthetics table using gt
data %>% gt(rowname_col = 'Levels') %>%
  tab_header(title = md("Statistics makes me cry")) %>%
  cols_label(Freq = "Frequency",
    Percent = "Percent",
    Val_Percent = "Valid Percent",
    Cum_Percent = "Cumulative Percent") %>%
  sub_missing(missing_text = "")

```

Statistics makes me cry

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly agree	270	10.5	10.5	10.5
Agree	1338	52.0	52.0	62.5
Neither	735	28.6	28.6	91.1
Disagree	187	7.3	7.3	98.4
Strongly disagree	41	1.6	1.6	100.0
Total	2571	100.0	100.0	

For q03

```

# extract q03

q03 <- saq_data$q03
datalevels_q03 <- levels(q03)
freq_q03 <- as.numeric(table(q03))
percent_q03 <- as.numeric(round(prop.table(freq_q03) * 100, 1))
valid_percent_q03 <- as.numeric(round(prop.table(freq_q03) * 100, 1))
cum_percent_q03 <- cumsum(percent_q03)

# convert the computed values into dataframe

```

```
data_q03 <- data.frame(
  Levels = datalevels_q03,
  Freq = freq_q03,
  Percent = percent_q03,
  Val_Percent = valid_percent_q03,
  Cum_Percent = cum_percent_q03
)
```

```
head(data_q03)
```

Levels	Freq	Percent	Val_Percent	Cum_Percent
Strongly agree	497	19.3	19.3	19.3
Agree	672	26.1	26.1	45.4
Neither	878	34.2	34.2	79.6
Disagree	448	17.4	17.4	97.0
Strongly disagree	76	3.0	3.0	100.0

```
# add row for total
data_q03 <- data_q03 %>% add_row(Levels = "Total",
  Freq = sum(data_q03$Freq),
  Percent = sum(data_q03$Percent),
  Val_Percent = sum(data_q03$Val_Percent),
  Cum_Percent = NULL)
```

```
# final version of calculated table
data_q03 %>% gt(rowname_col = 'Levels') %>%
  tab_header(title = md("Statistic makes me cry")) %>%
  cols_label(Freq = "Frequency",
    Percent = "Percent",
    Val_Percent = "Valid Percent",
    Cum_Percent = "Cumulative Percent") %>%
  sub_missing(missing_text = "")
```

Statistic makes me cry

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly agree	497	19.3	19.3	19.3
Agree	672	26.1	26.1	45.4
Neither	878	34.2	34.2	79.6
Disagree	448	17.4	17.4	97.0
Strongly disagree	76	3.0	3.0	100.0
Total	2571	100.0	100.0	

For q06

```
# extract q06
q06 <- saq_data$q06

# mathematical computation
datalevels_q06 <- levels(q06)
```

```

freq_q06 <- as.numeric(table(q06))
percent_q06 <- as.numeric(round(prop.table(freq_q06) * 100, 1))
valid_percent_q06 <- as.numeric(round(prop.table(freq_q06) * 100, 1))
cum_percent_q06 <- cumsum(percent_q06)

# convert into dataframe
data_q06 <- data.frame(
  Levels = datalevels_q06,
  Freq = freq_q06,
  Percent = percent_q06,
  Val_Percent = valid_percent_q06,
  Cum_Percent = cum_percent_q06
)

# add row for total
data_q06 <- data_q06 %>% add_row(Levels = "Total",
  Freq = sum(data_q06$Freq),
  Percent = sum(data_q06$Percent),
  Val_Percent = sum(data_q06$Val_Percent),
  Cum_Percent = NULL)

# final version of calculated table
data_q06 %>% gt(rownames_col = 'Levels') %>%
  tab_header(title = md("I have little experience of computer")) %>%
  cols_label(Freq = "Frequency",
    Percent = "Percent",
    Val_Percent = "Valid Percent",
    Cum_Percent = "Cumulative Percent") %>%
  sub_missing(missing_text = "")

```

I have little experience of computer

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly agree	702	27.3	27.3	27.3
Agree	1127	43.8	43.8	71.1
Neither	344	13.4	13.4	84.5
Disagree	252	9.8	9.8	94.3
Strongly disagree	146	5.7	5.7	100.0
Total	2571	100.0	100.0	

For q08

```

# for q08
q08 <- saq_data$q08

# mathematical computation
datalevels_q08 <- levels(q08)
freq_q08 <- as.numeric(table(q08))
percent_q08 <- as.numeric(round(prop.table(freq_q08) * 100, 2))
valid_percent_q08 <- as.numeric(round(prop.table(freq_q08) * 100, 2))
cum_percent_q08 <- cumsum(percent_q08)

```

```

# convert into dataframe
data_q08 <- data.frame(
  Levels = datalevels_q08,
  Freq = freq_q08,
  Percent = round(valid_percent_q08,1),
  Val_Percent = round(valid_percent_q08,1),
  Cum_Percent = round(cum_percent_q08,1)
)

# add row for total
data_q08 <- data_q08 %>% add_row(Levels = "Total",
  Freq = sum(data_q08$Freq),
  Percent = sum(data_q08$Percent),
  Val_Percent = sum(data_q08$Val_Percent),
  Cum_Percent = NULL)

# final version of calculated table
data_q08 %>% gt(rowname_col = 'Levels') %>%
  tab_header(title = md("I have never been good at mathematics")) %>%
  cols_label(Freq = "Frequency",
    Percent = "Percent",
    Val_Percent = "Valid Percent",
    Cum_Percent = "Cumulative Percent") %>%
  sub_missing(missing_text = "")

```

I have never been good at mathematics

	Frequency	Percent	Valid Percent	Cumulative Percent
Strongly agree	383	14.9	14.9	14.9
Agree	1487	57.8	57.8	72.7
Neither	482	18.8	18.8	91.5
Disagree	147	5.7	5.7	97.2
Strongly disagree	72	2.8	2.8	100.0
Total	2571	100.0	100.0	

Task 3

```

mr_drugs <- read_xlsx("MR_Drugs.xlsx")

inco <- mr_drugs %>% select(starts_with('inco'))

transform_inco <- mr_drugs %>% select(starts_with('inco')) %>%
  colSums() %>%
  enframe("income", "N") %>%
  mutate(Percent = round(N / sum(N) * 100, 1))

transform_inco

```

income	N	Percent
inco1	226	12.8
inco2	607	34.5
inco3	293	16.6
inco4	50	2.8
inco5	82	4.7
inco6	151	8.6
inco7	352	20.0

```
# get the frequencies of 0 and 1 and convert to dataframe
income_frequencies <- apply(inco, 2, table) %>%
  t() %>% as.data.frame()
income_frequencies
```

	0	1
inco1	746	226
inco2	365	607
inco3	679	293
inco4	922	50
inco5	890	82
inco6	821	151
inco7	620	352

```
transform_inco <- transform_inco %>%
  mutate(`Percent of Cases` =
    round(transform_inco$N / (transform_inco$N + income_frequencies[, 1]) * 100, 1))

transform_inco
```

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

Mathematical Computation

```
# final version of calculated table
final_inco <- transform_inco %>% add_row(
  income = "Total",
  N = sum(transform_inco$N),
  Percent = round(sum(transform_inco$Percent),2),
  "Percent of Cases" = round(sum(transform_inco$`Percent of Cases`),2),)

# converting into percentage
```

```
final_inco$Percent <- paste0(sprintf("%.1f", final_inco$Percent),"%")
final_inco$`Percent of Cases` <- paste0(sprintf("%.1f", final_inco$`Percent of Cases`),"%")
final_inco
```

income	N	Percent	Percent of Cases
inco1	226	12.8%	23.3%
inco2	607	34.5%	62.4%
inco3	293	16.6%	30.1%
inco4	50	2.8%	5.1%
inco5	82	4.7%	8.4%
inco6	151	8.6%	15.5%
inco7	352	20.0%	36.2%
Total	1761	100.0%	181.0%

Final Table using gt table

```
final_inco %>% gt(rowname_col = 'income') %>%
  tab_spanner(label='Response', columns = c('N', 'Percent')) %>%

  tab_header(title = md("$Income Frequencies")) %>%
  tab_footnote(footnote = "a. Dichotomy group tabulated at value 1",
    placement = c('auto')) %>% tab_options(footnotes.multiline = FALSE)
```

\$Income Frequencies

	Response		
	N	Percent	Percent of Cases
inco1	226	12.8%	23.3%
inco2	607	34.5%	62.4%
inco3	293	16.6%	30.1%
inco4	50	2.8%	5.1%
inco5	82	4.7%	8.4%
inco6	151	8.6%	15.5%
inco7	352	20.0%	36.2%
Total	1761	100.0%	181.0%

a. Dichotomy group tabulated at value 1

```
knitr::include_graphics('inco.png')
```


\$Income Frequencies			
	Response		
	N	Percent	Percent of Cases
inco1	226	12.8%	23.3%
inco2	607	34.5%	62.4%
inco3	293	16.6%	30.1%
inco4	50	2.8%	5.1%
inco5	82	4.7%	8.4%
inco6	151	8.6%	15.5%
inco7	352	20.0%	36.2%
Total	1761	100.0%	181.0%
a. Dichotomy group tabulated at value 1			

Part 2

Task 1

Load the necessary library needed for Part 2

```
library(jsonlite) #for working with json data
library(RSelenium) #for web scraping of dynamic table
library(rvest) #scraping the webpage into tibble or df
library(netstat) #for selenium driver
library(stringr) #string manipulation

data_1 = 'https://data.covid19india.org/v4/min/timeseries.min.json'
data_2 = 'https://data.covid19india.org/v4/min/data.min.json'
covid_data_1 <- jsonlite::fromJSON(data_1)
covid_data_2 <- jsonlite::fromJSON(data_2)

knitr::include_graphics('cov.png')
```

Name	Type	Value
covid_data_1	list [38]	List of length 38
AN	list [1]	List of length 1
dates	list [585]	List of length 585
2020-03-26	list [3]	List of length 3
delta	list [1]	List of length 1
confirmed	integer [1]	1
delta7	list [1]	List of length 1
confirmed	integer [1]	1
total	list [1]	List of length 1
confirmed	integer [1]	1

Sample of raw json data for first record

```
covid_1_parsed <-
  covid_data_1 %>% enframe() %>% unnest_wider(value) %>% unnest_wider(dates) %>%
  pivot_longer(cols = !name,
               names_to = 'date',
               values_to = "value") %>% unnest_wider(value)

knitr::include_graphics("covid.png")
```

	name	date	delta	delta7	total
1	AN	2020-03-26	list(confirmed = 1)	list(confirmed = 1)	list(confirmed = 1)
2	AN	2020-03-27	list(confirmed = 5)	list(confirmed = 6)	list(confirmed = 6)
3	AN	2020-03-28	list(confirmed = 3)	list(confirmed = 9)	list(confirmed = 9)
4	AN	2020-03-29	NULL	list(confirmed = 9)	list(confirmed = 9)
5	AN	2020-03-30	list(confirmed = 1)	list(confirmed = 10)	list(confirmed = 10)
6	AN	2020-03-31	NULL	list(confirmed = 10)	list(confirmed = 10)
7	AN	2020-04-01	NULL	list(confirmed = 10)	list(confirmed = 10)
8	AN	2020-04-02	NULL	list(confirmed = 9)	list(confirmed = 10)
9	AN	2020-04-03	NULL	list(confirmed = 4)	list(confirmed = 10)
10	AN	2020-04-04	NULL	list(confirmed = 1)	list(confirmed = 10)
11	AN	2020-04-05	NULL	list(confirmed = 1)	list(confirmed = 10)
12	AN	2020-04-06	NULL	NULL	list(confirmed = 10)
13	AN	2020-04-07	NULL	NULL	list(confirmed = 10)
14	AN	2020-04-08	list(confirmed = 1)	list(confirmed = 1)	list(confirmed = 11)
15	AN	2020-04-09	list(recovered = 10)	list(confirmed = 1, recovered = 10)	list(confirmed = 11, recovered = 10)
16	AN	2020-04-10	NULL	list(confirmed = 1, recovered = 10)	list(confirmed = 11, recovered = 10)

Showing 1 to 16 of 23,294 entries, 5 total columns

Sample parsed till dates

```
num_rows <- nrow(covid_1_parsed)
selected_rows <- sample(1:num_rows, 1000)
covid_1_parsed_subset <- covid_1_parsed[selected_rows, ]
```

```
knitr::include_graphics("covid2.png")
```

	name	date	delta	delta7	total
1	SK	2020-06-07	NULL	list(confirmed = 6, tested = 2080)	list(confirmed = 7, tested = 5005)
2	DL	2020-11-12	list(confirmed = 7053, deceased = 104, recovered = [...])	list(confirmed = 50375, deceased = 563, recovered [...])	list(confirmed = 467028, deceased = 7332, recovere [...])
3	JK	2020-11-26	list(confirmed = 487, deceased = 5, recovered = 47 [...])	list(confirmed = 3591, deceased = 50, recovered = [...])	list(confirmed = 108306, deceased = 1668, recovere [...])
4	WB	2020-08-05	list(confirmed = 2816, deceased = 61, recovered = [...])	list(confirmed = 18542, deceased = 356, recovered [...])	list(confirmed = 83800, deceased = 1846, recovered [...])
5	HP	2021-08-10	list(confirmed = 419, deceased = 2, other = -1, re [...])	list(confirmed = 2027, deceased = 14, other = -18, [...])	list(confirmed = 208616, deceased = 3521, other = [...])
6	WB	2021-10-13	list(confirmed = 771, deceased = 11, recovered = 7 [...])	list(confirmed = 5236, deceased = 72, recovered = [...])	list(confirmed = 1578482, deceased = 18935, recove [...])
7	DN	2020-08-29	list(confirmed = 15, other = 2, recovered = 29, te [...])	list(confirmed = 185, other = 7, recovered = 287, [...])	list(confirmed = 2308, deceased = 2, other = 29, r [...])
8	GJ	2020-10-08	list(confirmed = 1278, deceased = 10, recovered = [...])	list(confirmed = 9206, deceased = 78, recovered = [...])	list(confirmed = 147951, deceased = 3541, recovere [...])
9	MH	2020-09-15	list(confirmed = 20482, deceased = 515, other = 3, [...])	list(confirmed = 154084, deceased = 3002, other = [...])	list(confirmed = 1097856, deceased = 30409, other [...])
10	TR	2021-09-26	list(confirmed = 20, recovered = 37, tested = 3621 [...])	list(confirmed = 213, deceased = 3, recovered = 28 [...])	list(confirmed = 84050, deceased = 808, other = 63 [...])
11	KL	2020-07-12	list(confirmed = 435, deceased = 2, recovered = 13 [...])	list(confirmed = 2444, deceased = 6, recovered = 9 [...])	list(confirmed = 7874, deceased = 32, recovered = [...])
12	AP	2020-12-17	list(confirmed = 534, deceased = 2, recovered = 49 [...])	list(confirmed = 3353, deceased = 22, recovered = [...])	list(confirmed = 877348, deceased = 7069, recovere [...])
13	TR	2020-12-30	list(confirmed = 12, recovered = 14, tested = 1367 [...])	list(confirmed = 55, deceased = 2, recovered = 122 [...])	list(confirmed = 33255, deceased = 382, other = 23 [...])
14	TR	2021-09-12	list(confirmed = 44, recovered = 105, tested = 485 [...])	list(confirmed = 313, deceased = 2, recovered = 55 [...])	list(confirmed = 33255, deceased = 382, other = 23 [...])

Showing 1 to 15 of 1,000 entries, 5 total columns

```
covid_1_parsed_subset <- covid_1_parsed_subset %>%
  mutate(across(c(delta, delta7, total), ~ map(., ~ set_names( as_tibble(.x), paste0(cur_column(), "_",
  unnest_wider(c(delta, delta7, total))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(...)`.
```

Caused by warning:

```
## ! The `value` argument of `names<-()` must have the same length as `x` as of
## tibble 3.0.0.
```

```
covid_1_parsed_subset
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

namedated	deltadelta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta	delta
GJ 2020-12-14	1120	11	1389	55807A	NA	NA	NA	8635	87	10023	114825A	NA	NA	NA	22880	13822	2117087	25383	NA NA
KL 2020-04-22	11	NA	1	569	NA	NA	NA	50	NA	90	4346	NA	NA	NA	438	3	308	2082	NA NA NA
LD 2020-04-12	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA NA
CT 2020-07-30	256	1	285	8190	NA	NA	NA	2486	17	1534	4241	NA	NA	NA	8856	51	5921	3106	NA NA NA
MZ 2021-05-20	1192	NA	136	1622	NA	4255	67	1268	7	1202	17422	NA	1340	278	9444	30	7407	3545	NA 250240978
HR 2021-10-06	11	NA	21	2234	NA	37516	117326	1	90	14679	NA	49937	50567	750948	87576080	1254532	816986371	23833	
HR 2021-07-26	131	3	32	1626	NA	61935	4243201	17	258	16625	BA	24731	B82117	69770	62275943	107428	99880822	945037	
AS 2020-08-15	1057	7	1593	20805	NA	NA	NA	1784	42	12695	1118	NA	NA	NA	75559	82	53287	1705526	NA NA
ML 2021-02-18	-1	NA	24	NA	NA	453	137	15	NA	126	6170	NA	10032	607	13946	48	13743	5666	NA 21674607
AS 2020-12-11	127	1	149	21002	NA	NA	NA	927	13	972	15750	NA	NA	NA	21443	12002	09936	82574	NA NA
BR 2020-06-20	213	NA	269	5586	NA	NA	NA	1214	14	1681	31062	NA	NA	NA	7503	49	5367	1511	NA NA NA
KA 2020-12-01	1330	14	886	95607	NA	NA	NA	9802	97	10608	749535	A	NA	NA	88622	17792	25070	1197240	NA NA
LA 2020-04-17	NA	NA	NA	NA	NA	NA	NA	3	NA	3	917	NA	NA	NA	18	NA	14	917	NA NA NA
AR 2021-06-30	1286	4	207	4782	NA	12403	791	1941	12	1734	3381	NA	7235	14412	35857	172	32923	76589	NA 5092784530
OR 2020-04-13	-1	NA	6	308	NA	NA	NA	15	1	16	2775	NA	NA	NA	55	1	18	4170	NA NA NA
HP 2020-11-20	588	11	726	4022	1	NA	NA	3777	75	3608	234792	NA	NA	NA	32783	491	25432	179281	NA NA
LA 2020-12-11	1	1	17	480	NA	NA	NA	380	3	439	3918	NA	NA	NA	9112	123	8189	9959	NA NA NA

[illegible]

[illegible]

	deltacount	deltameddied	deltasected	deltarecovered	deltatotal	deltavaccinated	deltareduced7days	deltameddied7days	deltasected7days	deltarecovered7days	deltatotal7days	deltavaccinated7days	deltareduced10days	deltameddied10days	deltasected10days	deltarecovered10days	deltatotal10days	deltavaccinated10days
18	NA	33	12095	NA	73130	152289	167	1	303	203223	NA	735315	860646					
20	NA	37	1363	NA	4441	823	194	1	343	10849	NA	45971	5346					
248	NA	195	2759	NA	NA	NA	1405	7	1109	21442	NA	NA	NA					
449	2	623	79231	NA	7545	3139	4000	20	4447	732176	NA	246674	404371					
1153	12	607	53006	NA	NA	NA	5560	125	3953	203701	NA	NA	NA					
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA					
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA					
85	NA	336	NA	NA	NA	NA	558	2	612	4413	NA	NA	NA					
59	1	4	15010	-3	NA	NA	292	6	300	71886	19	11432	12300					
618	6	1178	38069	NA	305067	274947	7930	64	9772	369952	NA	878446	871242					
77	2	42	70248	NA	13832	24010	417	11	405	465879	NA	124603	199368					
203	4	185	7484	NA	NA	NA	1216	51	1056	46405	NA	NA	NA					
116	NA	127	41910	NA	NA	NA	826	6	1018	261759	NA	40222	NA					
305	3	469	5037	NA	3329	1938	2208	16	3003	38856	NA	17019	13593					
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA					
39687	156	20356	1057383	10	1825443	371119	205343	1063	130151	6371908	63	10990389	2184508					

delta7_confirmed	delta7_deceased	delta7_recovered	delta7_tested	delta7_other	delta7_vaccinated	delta7_vaccinated2
NA	NA	NA	NA	NA	NA	NA
272	1	98	2117	NA	NA	NA
NA	NA	7	4827	NA	NA	NA
2112	26	2917	17326	NA	NA	NA
3	NA	NA	NA	NA	NA	NA
41	1	13	4160	NA	NA	NA
1453	25	1693	56110	NA	NA	NA
5	NA	8	1786	NA	NA	NA
2443268	20140	1632826	11766876	1436	11064547	6625457
481	4	448	10189	NA	NA	NA
2619	14	3154	309799	NA	NA	NA
22	NA	26	NA	-2	70157	15341
570	8	1124	254383	NA	248474	242738
21792	377	21284	310555	NA	NA	NA
578	6	608	13003	NA	NA	NA
36	NA	NA	NA	NA	NA	NA
41407	728	41519	497862	NA	NA	NA
141036	910	130926	1126436	5	1187243	285630
36	NA	63	8369	NA	16347	1479
338	9	537	10149	NA	4814	517
6	NA	9	9474	NA	600	11539
159	13	153	183045	NA	318193	252461
295	73	276	292625	NA	775097	288873
26550	361	63258	715276	NA	1053614	167918
NA	NA	NA	NA	NA	NA	NA
49	NA	84	461896	NA	3418613	2834416
NA	NA	NA	NA	NA	NA	NA
8186	32	10075	223395	NA	NA	NA
148	3	144	1481	4	NA	NA

delta7_confirmed	delta7_deceased	delta7_recovered	delta7_tested	delta7_other	delta7_vaccinated	delta7_vaccinated2
9	1	12	2275	NA	NA	NA
12333	84	7531	127287	NA	NA	NA
141	9	216	4221	NA	3647	56
7549	114	10473	155760	NA	NA	NA
13400	64	15957	355263	NA	NA	NA
26	NA	43	NA	NA	NA	NA
1	NA	1	755	NA	NA	NA
-1	NA	18	2811	NA	NA	NA
1171	39	820	46450	NA	NA	NA
NA	NA	NA	158	NA	NA	NA
100932	714	90350	891672	9	608461	740794
6399	108	7888	760681	NA	1149863	254276
63	1	97	1078046	NA	3516623	3703748
148	6	133	10601	NA	669	371
1471	27	1965	11581	-3	NA	NA
16125	113	19222	287652	NA	NA	NA
4019	107	7821	137823	-1	65211	1564
1377	12	1864	27611	NA	NA	NA
185	5	156	220949	NA	234528	245524
4482	26	1782	18919	NA	51431	14593
3645	92	2572	71285	NA	NA	NA
2	NA	NA	NA	NA	4817	32006
1604	26	2072	24207	NA	17809	18559
98	NA	170	519240	NA	2121557	534709
202	3	405	45068	1	13536	71322
NA	NA	10	547	NA	NA	NA
6	NA	3	NA	NA	NA	NA
3495	73	3594	76403	NA	NA	NA
244551	2121	265543	11747611	65	36555035	16982822
454	11	638	15838	NA	7123	41300
4863	48	7225	246533	NA	NA	NA
42770	1026	48631	1373227	41	1392983	759560
938	5	638	20714	NA	NA	NA
1939	36	3356	59301	NA	NA	NA
382	12	417	13390	30	6466	21078
305	5	386	23654	NA	14190	NA
1249	16	1204	99735	NA	NA	NA
64488	1113	130058	1058998	NA	1615267	149506
18376	339	22611	419950	21	106654	NA
5802	53	3575	173680	NA	NA	NA
1804	29	3730	156031	NA	103909	NA
379336	5565	426160	1786039	181	944602	682831
4995	69	4943	212337	NA	3119129	551289
6364	40	1422	25474	NA	NA	NA
9891	136	12747	1065611	NA	NA	NA
146	NA	60	3718	NA	NA	NA
123	1	33	7967	NA	83499	10284
9	NA	32	NA	NA	26023	3082
12685	374	11595	160452	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
18	NA	NA	NA	NA	NA	NA
5286	64	4954	47624	NA	NA	NA

delta7_confirmed	delta7_deceased	delta7_recovered	delta7_tested	delta7_other	delta7_vaccinated1	delta7_vaccinated2
1195	13	1399	103757	NA	NA	NA
1544	4	1082	18072	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
627	56	749	19308	NA	NA	NA
1038	5	297	77963	3	NA	NA
4107	15	5552	57074	NA	5150	7806
NA	NA	NA	NA	NA	NA	NA
1023	3	841	23117	NA	NA	NA
983	12	1294	31547	NA	42722	37780
592	2	143	17124	NA	NA	NA
414	5	554	67627	NA	NA	NA
3461	49	3269	11356	NA	NA	NA
32	NA	40	951561	NA	1014591	1308099
53912	1382	54714	468663	NA	141058	68896
482763	2866	274214	7189639	81	13654971	1289551
NA	NA	5	2358	NA	5677	2700
1111	179	1668	1819201	NA	1647943	498480
2	NA	6	9733	NA	846	14079
2747	68	5633	23001	NA	34976	717
3032	22	1514	37349	NA	NA	NA
2183	11	2743	67284	NA	137953	34543
1019264	21807	1692613	18393985	392	16636005	1752783
304	5	332	440009	NA	547423	298155
21	NA	21	3754	NA	NA	NA
1705	48	1925	138532	NA	NA	NA

```
# total parsed
# for delta variants
covid_1_parsed_subset[789:885,] %>% select(starts_with('total'))
```

total_confirmed	total_deceased	total_recovered	total_tested	total_other	total_vaccinated1	total_vaccinated2
15116	154	14105	438825	NA	133700	46932
171052	3146	166813	1051410	NA	1046130	268339
83006	1361	74525	1527416	976	NA	NA
300153	1644	296740	9016741	NA	438673	155505
3366	2	3330	72410	29	NA	NA
400651	14463	239755	2021437	304	NA	NA
699	1	588	46119	5	4411	2052
2216812	26571	1932778	28546677	NA	7913888	2093048
93111	1562	87127	1881487	1243	NA	NA
42808	161	30500	663520	NA	651442	204046
755389	8221	726081	9020972	NA	4839906	990430
107229	1691	98867	1150256	NA	1171910	263362
33609	390	33072	648832	23	698444	87417
5014	62	4944	251774	NA	4045	182
5	NA	1	NA	NA	NA	NA
107921	955	104724	3946256	NA	NA	NA
51604	192	45418	778577	NA	665491	225976
97414	849	90385	2913412	NA	NA	NA

total_confirmed	total_deceased	total_recovered	total_tested	total_other	total_vaccinated1	total_vaccinated2
846480	12438	829850	16916170	NA	295338	14039
2638	34	1757	30973	NA	NA	NA
2962408	37517	2908622	45413942	28	35402071	12495692
30904734	410816	30054758	434058138	12260	306612781	74854865
196789	1753	112870	2299332	NA	NA	NA
7510	129	7365	426511	NA	180805	77305
255888	3355	202039	8045445	NA	2317478	533198
343609	5085	335462	9140410	NA	4380350	815133
167436	3073	162276	938454	NA	859902	130578
1911231	12919	1865956	22608072	NA	13322361	3354510
17316	325	12135	85906	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
100224	876	93874	3152647	NA	NA	NA
598334	16233	581055	11654465	NA	7251374	1552932
276004	1598	265048	24375830	1	3721925	511864
1793	27	261	56851	NA	NA	NA
57351	915	47129	783395	NA	451274	71208
198550	3375	190377	2159415	21	2151620	435973
338383	5000	325325	8544919	NA	3511232	715016
NA	NA	NA	NA	NA	NA	NA
91329	1430	82219	2153529	NA	NA	NA
4906793	28229	4797409	37300674	529	25194796	12850111
602375	16558	585566	15395263	NA	15939393	6234555
14145	33	11613	304526	NA	NA	NA
29561	376	29052	589389	NA	96396	46535
4991	62	4904	209437	NA	1032	NA
825064	10077	814778	26131219	NA	28113725	9054592
3	NA	NA	NA	NA	NA	NA
278207	4424	269361	12321047	NA	1992286	504411
83466	1448	81511	1143680	NA	1099252	622414
1692693	44248	1531277	9120515	625	NA	NA
5791413	97394	5486206	35774626	2839	18612983	4679276
403	1	4	18127	3	NA	NA
16617	169	14917	459023	NA	216713	58387
7385	13	5408	222429	NA	NA	NA
3471	9	2033	123269	18	NA	NA
2916	13	1762	156104	28	NA	NA
791750	10512	781090	13999557	NA	23316207	4552158
1587	80	152	22664	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA
34546	417	30325	365412	NA	122192	27795
588856	5655	576241	21586352	1347	14215451	3077839
62946	1033	52667	846789	NA	488761	71979
NA	NA	NA	NA	NA	NA	NA
187342	3040	165786	1888929	21	1985472	433542
24912	273	21401	418298	NA	NA	NA
263100	1553	261200	22899132	1	1227941	336166
5269292	78857	4654731	30351356	2410	15287399	4025544
323757	1860	319069	6378784	NA	NA	NA
790070	9019	780610	12655471	NA	18988628	3498980
1575577	18894	1549049	18471961	NA	46206046	17953133
129031	1983	125535	5830758	NA	537307	124190

delta_21_delta	delta_21_vaccinated	delta_21_deceased	delta_21_other	delta_21_confirmed	delta_21_recovered	delta_21_tested	delta_21_vaccinated	delta_21_deceased	delta_21_other	delta_21_confirmed	delta_21_recovered	delta_21_tested	delta_21_vaccinated	delta_21_deceased	delta_21_other
38915	1584	45202	95	NA	79	NA	529	611	572	25460833383	414843	3	NA		
10693917604	28163	292	11	345	NA	2432	2347	2339	724215623591	1373861	75	NA			
65158	2010	11914	7167	167	6439	NA	59521	53326	48406	506604109805	792534	3089	NA		
388	NA	2	11	NA	9	NA	26	58	28	2475	525	1532	NA	NA	
31	NA	4	NA	NA	NA	NA	NA	NA	NA	1625	69	796	NA	NA	
10804053161	31988	1172	20	1399	NA	13825	8117	15146	7735161611778	1282938	200	7			
1559	38	203	22	1	51	NA	530	256	460	13944	7871	41927	4	NA	
1689	705	4283	63	1	62	NA	734	439	652	12787	7665	71276	10	NA	
55368	5713	77062	16	NA	8	NA	63	105	74	393209148817	2034460	1	NA		
5945	10	51	579	2	610	NA	6770	4098	5685	42841	4302	11262	21	NA	
468	193	822	11	1	21	1	162	130	144	4606	4706	23628	6	13	
69675	21420	25651	488	4	450	NA	3381	3046	3230	473919731837	917236	24	NA		
34152	3321	4418	26	1	25	NA	159	192	162	221114200182	223256	8	NA		
2662	171	532	38	NA	45	NA	343	278	315	21095	8324	20073	NA	NA	
13454	9399	29672	2	NA	2	NA	31	27	26	95132	271500	864947	NA	NA	
389	NA	NA	21	1	8	NA	82	79	76	2847	782	14044	2	6	
25021	38781	26738	121	1	183	NA	1068	1189	1121	2610861072389961422	9	NA			
12370177325	106584	1009	19	1183	NA	8827	7407	8852	858304788134	1578082	97	NA			
3501	3	650	12	NA	2	NA	49	87	62	22126	4155	74642	NA	NA	
881379750410	933460	12907	251	13152	1	109755	96071	101753	843157758837804527536727	37					
1614583401	57556	6	NA	6	NA	74	63	58	11407302669213130828	1	NA				
9524	1065	12934	5	NA	9	NA	88	75	74	83377	25250	258381	1	12	
47417	209609	58881	914	15	913	NA	4193	6453	5953	30011438095971871612	86	NA			

```
# for delta7
covid_2_parsed %>% select(starts_with('delta7'))
```

delta7_confirmed	delta7_recovered	delta7_tested	delta7_vaccinated	delta7_vaccinated	delta7_deceased	delta7_other
3	5	8936	884	10640	NA	NA
2873	3590	254532	1223010	1887005	30	NA
66	97	4788	3312	23647	NA	NA
2056	2215	269097	274869	849889	24	NA
40	31	1378539	1286708	2144970	NA	NA
28	20	10726	3680	21641	NA	NA
205	103	147451	379374	604260	5	NA
267	239	395086	160323	269146	NA	NA
NA	2	NA	2802	14244	NA	NA
222	409	19026	8418	46494	6	NA
159	116	328489	335172	1660382	2	NA
1537	1154	64352	13244	234011	20	-1
95	87	148110	160777	368141	NA	NA
137	250	464579	299587	428313	3	NA
611	572	254608	33383	414843	3	NA
2347	2339	724215	623591	1373861	75	NA
53326	48406	506604	109805	792534	3089	NA
58	28	2475	525	1532	NA	NA
NA	NA	1625	69	796	NA	NA
8117	15146	773516	1611778	1282938	200	7
256	460	13944	7871	41927	4	NA
439	652	12787	7665	71276	10	NA

delta7_confirmed	delta7_recovered	delta7_tested	delta7_vaccinated	delta7_vaccinated	delta7_deceased	delta7_other
105	74	393209	148817	2034460	1	NA
4098	5685	42841	4302	11262	21	NA
130	144	4606	4706	23628	6	13
3046	3230	473919	731837	917236	24	NA
192	162	221114	200182	223256	8	NA
278	315	21095	8324	20073	NA	NA
27	26	95132	271500	864947	NA	NA
79	76	2847	782	14044	2	6
1189	1121	261086	1072389	961422	9	NA
7407	8852	858304	788134	1578082	97	NA
87	62	22126	4155	74642	NA	NA
96071	101753	8431577	15883780	24527530	3727	37
63	58	1140736	2266921	3130828	1	NA
75	74	83377	25250	258381	1	12
6453	5953	300114	3809597	1871612	86	NA

```
# for delta21_14
covid_2_parsed %>% select(starts_with('delta2'))
```

delta21_14_confirmed
9
3220
87
1499
30
23
124
195
4
409
149
958
83
78
529
2432
59521
26
NA
13825
530
734
63
6770
162
3381
159
343
31
82

delta21_14_confirmed
1068
8827
49
109755
74
88
4193

```
# for total
covid_2_parsed %>% select(starts_with('total'))
```

total_confirmed	total_deceased	total_recovered	total_tested	total_vaccinated1	total_vaccinated2	total_other
7651	129	7518	598033	294001	200157	NA
2066450	14373	2047722	29518787	32976969	20375181	NA
55155	280	54774	1185436	771875	534486	NA
610645	5997	600974	24712042	20172463	8068795	1347
726098	9661	716390	50531824	49874828	18346781	1
65351	820	64495	792851	926035	546981	NA
1006052	13577	992159	13709510	14851682	7343273	NA
1439870	25091	1414431	29427753	13055636	7425404	NA
10681	4	10644	72410	660753	370255	31
178108	3364	174392	1468399	1262568	911114	NA
826577	10089	816283	30928063	44735217	25972387	NA
224106	3738	218410	3685011	5713695	3443823	16
771252	10049	761068	13032504	17772376	8115463	NA
348764	5138	343518	15985878	14986646	5585648	NA
332249	4432	326915	16202346	9511073	5149471	NA
2988333	38082	2941578	50873103	42497761	22858384	29
4968657	31681	4857181	37886378	25306499	13658343	529
20962	208	20687	555568	208798	152280	NA
10365	51	10270	263541	55129	45951	44
6611078	140216	6450585	62667211	67198794	30975692	3619
83627	1450	81746	1151665	1103275	641819	NA
123731	1921	121102	1367673	1249436	719413	NA
792854	10524	782215	20294225	49911938	20838045	NA
121359	432	114612	1298444	711597	512029	NA
31842	685	29904	395416	709553	490663	1043
1041457	8386	1029147	21994343	25736641	11560912	NA
602401	16559	585591	15429415	15942714	6238973	NA
128013	1857	125726	1919060	733922	404355	NA
954429	8954	945443	14807752	42544909	20097635	NA
31979	396	31063	261343	521763	451509	325
671463	3956	663498	27569831	22498559	9772398	NA
2702623	36116	2655015	51159242	41279432	17619141	NA
84468	813	83466	1983127	2508477	1621329	63
34285612	458470	33661339	609201294	732371508	330752697	13197
1710158	22900	1687151	83635222	98178865	32681895	NA
343896	7400	330195	7781148	7478017	3898342	6150
1592908	19141	1565471	19228303	56192166	21559747	NA

```
# merge into single file
merged_df <- merge(covid_1_parsed_subset,
  covid_2_parsed,
  by.x = "name",
  by.y = "name",
  sort = T,
  all = F)

head(merged_df)
```

Task 2

```
# load the webdriver for firefox
rD <- rsDriver(browser="firefox",verbose = F, port = 14421L)
remDr <- rD[["client"]]
remDr$navigate("https://aqicn.org/forecast/kathmandu/")
aqi_html <- read_html(remDr$getPageSource() %>% unlist())

# scrape the needed table for data analysis
aqi_html %>% html_element(".forecast-body-table") %>%
  html_nodes("table") %>%
  html_table() ->
  forecast_table

# since forecast consist the list of dataframe
# extracted first value from the list which consists the required dataframe
aqi_table <- forecast_table %>% .[[1]]

knitr::include_graphics('aqi.png')
```


	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
1		Tuesday 26	Tuesday 26	Tuesday 26	Tuesday 26	Tuesday 26	Tuesday 26	Tuesday 26	Tuesday 26	NA	NA	Wednesday 27
2	hour	0	3	6	9	12	15	18	21	NA	NA	0
3	PM2.5	138138	138138	138137	137137	137137	137137	138138	138138	NA	NA	138138
4	PM10	5151	5151	5151	5046	4646	4646	4646	4646	NA	NA	5148
5	O3	44	44	113	3327	2823	2220	169	74	NA	NA	54
6	UVI									NA	NA	
7	Wind Speed (m/s)	2	2	2	1	3	3	2	1	NA	NA	2
8										NA	NA	
9	Temp.	13°	13°	17°	22°	22°	21°	16°	15°	NA	NA	15°
10	humidity									NA	NA	
11		6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	6:02 ~ 18:19	NA	NA	6:00 ~ 18:19

Showing 1 to 11 of 11 entries, 67 total columns

Data Wrangling the forecast Table AQI Kathmandu

- **Step 1:** Remove null columns that came while parsing
`aqi_table %>% select(-c('X10','X11','X20','X21','X30','X31','X40','X41','X50','X51','X60','X61'))`
- **Step 2:** Filter out rows with 'UVI' and 'humidity' in the 'X1' column where entire row is NULL
`aqi_table %>% filter(X1 != 'UVI')` and `aqi_table %>% filter(X1 != 'humidity')`
- **Step 3:** Replaced the value in the 'X1' column at index 9 with 'humidity'
`aqi_table %>% mutate(X1 = replace(X1, 9, "humidity"))`
- **Step 4:** Replaced the value in the 'X1' column at index 1 with 'Index'
`aqi_table %>% mutate(X1 = replace(X1, 1, "Index"))`
- **Step 5:** Filter out rows with empty values in the 'X1' column
`aqi_table %>% filter(X1 != '')`
- **Step 6:** Assigned the first row of the data frame as the column headers
`headers <- aqi_table[1,]`
`colnames(aqi_table) <- headers`
- **Step 7:** Remove the first row of the data frame
`aqi_table <- aqi_table[-1,]`
- **Step 8:** Converted the 'Index' column to row names
`aqi_table %>% column_to_rownames(var = 'Index')`

```
# extract first value from list
aqi_table <- forecast_table %>% .[[1]]

# delete null columns
aqi_table <- aqi_table %>%
  select(-c('X10','X11','X20','X21','X30','X31','X40','X41','X50','X51','X60','X61'))

# remove null row 'UVI'
aqi_table <- aqi_table %>% filter(X1 != 'UVI')

# since value of humidity interchange for now I have removed empty row.
aqi_table <- aqi_table %>% filter(X1 != 'humidity')

# now I have assigned the value at 1st column 9th row as 'humidity'
aqi_table <- aqi_table %>% mutate(X1 = replace(X1, 9, "humidity"))
```

[illegible]

Note Still some changes needs to be done on `hour`, `PM2.5`, `PM10` and `O3` as values are concatenated wrongly while parsing html.

[illegible]

[illegible]

```
knitr::include_graphics('aqi_final.png')
```

[illegible]

Final Version of Cleaned AQI Table

Part 3

Load the necessary library needed for Part 3

```
library(pdftools) # for working with pdf files
library(tm) # for text mining
library(wordcloud) # plotting word cloud
library(Rgraphviz) # plotting network like graph for word association
library(graph) # plotting network like graph for word association
library(ggplot2) # for bargraph
```

Load the pdf files and convert it to Corpus

```
# load the file path in list
files <- list.files(pattern = "pdf$")

# load the pdf files into list
pdf_files <- lapply(files, pdf_text)

# create a corpus from vector source i.e from list pdf_files
corpus <- Corpus(VectorSource(unlist(pdf_files)))
# copy the loaded corpus
corpus_copy <- corpus

# inspect first few texts of corpus
inspect(corpus[1:2])

## <<SimpleCorpus>>
## Metadata: corpus specific: 1, document level (indexed): 0
## Content: documents: 2
##
## [1] See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/312111111-Data-mining-in-education
## [2] Overview\n\n\nData mining in education\nCristobal Romero* and Sebastian Ventura\n\n\n27 doi: 10.1002/widm.1075\n\n\nINTRODUCTION
```

Text Mining Preprocessing

- Step 1: Convert all texts to lowercase
- Step 2: Remove punctuation
- Step 3: Remove numbers
- Step 4: Remove stop words or user defined stop words
- Step 5: Stem the corpus
- Step 6: Remove specific words which doesn't help the corpus
- Step 7: Create Term Document Matrix

```
# convert the all texts in lower
corpus <- tm_map(corpus, tolower)

## Warning in tm_map.SimpleCorpus(corpus, tolower): transformation drops documents

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

```
## <<SimpleCorpus>>
## Metadata: corpus specific: 1, document level (indexed): 0
## Content: documents: 2
##
## [1] see discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/312111111
## [2] overview\n\n\n\data mining in education\ncristobal romero* and sebastian ventura\n\n
27 doi: 10.1002/widm.1075\n\n\n\n\nintroduction
```

```
# remove punctuations
corpus <- tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents
```

```
# stem the corpus
corpus <- tm_map(corpus, stemDocument)
```

```
## Warning in tm_map.SimpleCorpus(corpus, stemDocument): transformation drops
## documents
```

```
remove <- function(x) gsub("values","value",x)
corpus <- tm_map(corpus, remove)
```

```
## Warning in tm_map.SimpleCorpus(corpus, remove): transformation drops documents
```

```
# create Term Document Matrix with word length 1 or many
tdm <- TermDocumentMatrix(corpus, control = list((wordLengths=c(1,Inf))))
```

Best way to create a Term Document Matrix with preprocessing

```
remove <- function(x) gsub("values","value",x)
corpus_copy <- tm_map(corpus_copy, remove)
```

```
## Warning in tm_map.SimpleCorpus(corpus_copy, remove): transformation drops
## documents
```

```
my_tdm <- TermDocumentMatrix(
  unlist(corpus_copy),
  control =
    list(
      removePunctuation = TRUE,
      stopwords = TRUE,
      tolower = TRUE,
      stemming = FALSE,
      removeNumbers = TRUE,
      bounds = list(global = c(3, Inf)),
      wordLengths = c(1,Inf),
      removeWords = (c("can","may","used")))
)
```

Most Frequent Terms

```
# finding frequency of words which is at least present 10 times
low_frequent_terms <- findFreqTerms(my_tdm, lowfreq = 10)
head(low_frequent_terms)
```

```
## [1] "article" "author" "authors" "content" "data" "discovery"
```

```
# finding frequency of words which is at max present 10 times
high_frequent_terms <- findFreqTerms(my_tdm, highfreq = 10)
head(high_frequent_terms)
```

```
## [1] "cordoba" "downloaded" "interdisciplinary"
## [4] "profile" "profiles" "publication"
```

Associated terms of the most frequent term

```
# associated terms for mining with correlation 0.3
findAssocs(my_tdm, "mining", 0.3)
```

```
## $mining
##      data      knowledge      databases      discovery      systems
##      0.55      0.54      0.49      0.45      0.45
##      database      kinds      patterns      user      mined
##      0.44      0.42      0.40      0.39      0.39
##      interactive      users      research      analysis      association
##      0.37      0.37      0.36      0.35      0.34
##      interestingness      erent      retrieval      rules      multimedia
##      0.33      0.32      0.31      0.31      0.31
##      challenges      techniques
##      0.30      0.30
```

```
# associated terms for mining with learning 0.3
findAssocs(my_tdm, "learning", 0.35)
```

```
## $learning
##      machine      intelligence      arti      cial      vol
##      0.74      0.56      0.52      0.50      0.43
##      shavlik      morgan      kaufmann      michalski      statistics
##      0.43      0.42      0.41      0.41      0.40
##      expert      mitchell      ijcai      international      learners
##      0.40      0.40      0.39      0.38      0.38
##      quinlan      decisiontree      bibliography      carbonell      kluwer
##      0.38      0.37      0.36      0.36      0.36
##      neter      mateo
##      0.35      0.35
```

```
# associated terms for mining with data 0.3
findAssocs(my_tdm, "data", 0.4)
```

```
## $data
##      mining      cleaning      integration      warehouse      warehouses
##      0.55      0.43      0.42      0.42      0.41
```

Top 10 words and their respective counts

```
# top 10 words and their respective counts
```

```
df <-  
  my_tdm %>%  
  as.matrix() %>%  
  rowSums() %>%  
  sort(decreasing = TRUE) %>%  
  head(10) %>%  
  enframe(name = "word", value = "counts")
```

df

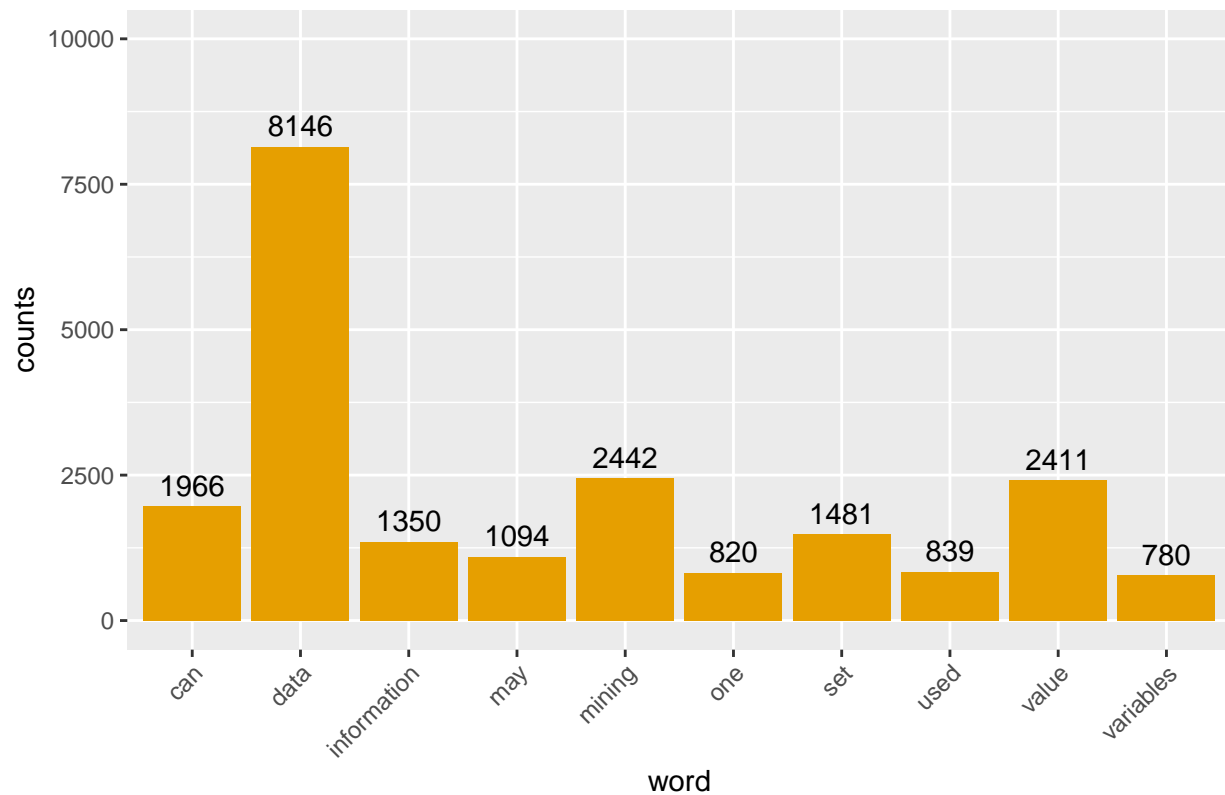
word	counts
data	8146
mining	2442
value	2411
can	1966
set	1481
information	1350
may	1094
used	839
one	820
variables	780

Bargraph of top 10 words and their respective counts

```
# using ggplot2  
bargraph <- ggplot(df, aes(word, counts)) +  
  geom_bar(stat = "identity", fill = "#E69F00") +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  
  labs(title = "Top 10 words by counts.") +  
  geom_text(aes(label = counts), vjust = -0.5) +  
  ylim(0, 10000L)
```

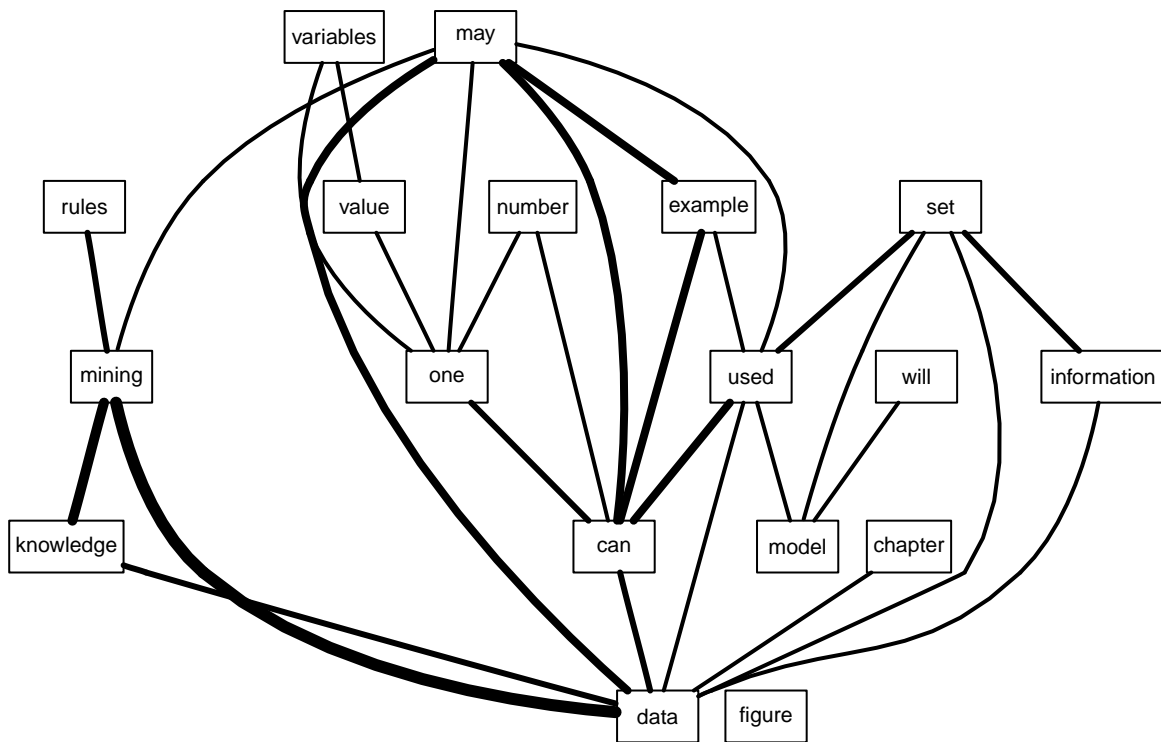
bargraph

Top 10 words by counts.



Correlation between top 650 frequent terms

```
top_650_frequent_tems <- findFreqTerms(my_tdm, lowfreq = 650)
plot(my_tdm, terms = top_650_frequent_tems, corThreshold = 0.2, weighting = T)
```

Topic Models

```
# topic models
library(topicmodels)
set.seed(123)

lda <- LDA(my_tdm, k=4)

# terms
head(terms(lda,3))

##      Topic 1      Topic 2      Topic 3      Topic 4
## [1,] "content173" "content781" "content15" "content945"
## [2,] "content171" "content686" "content16" "content878"
## [3,] "content473" "content711" "content669" "content791"

# topics
head(topics(lda))

## article  author authors content cordoba  data
##       3       3       3       1       3       2
```