FDS-A2: Data Wrangling

Suman Paudel

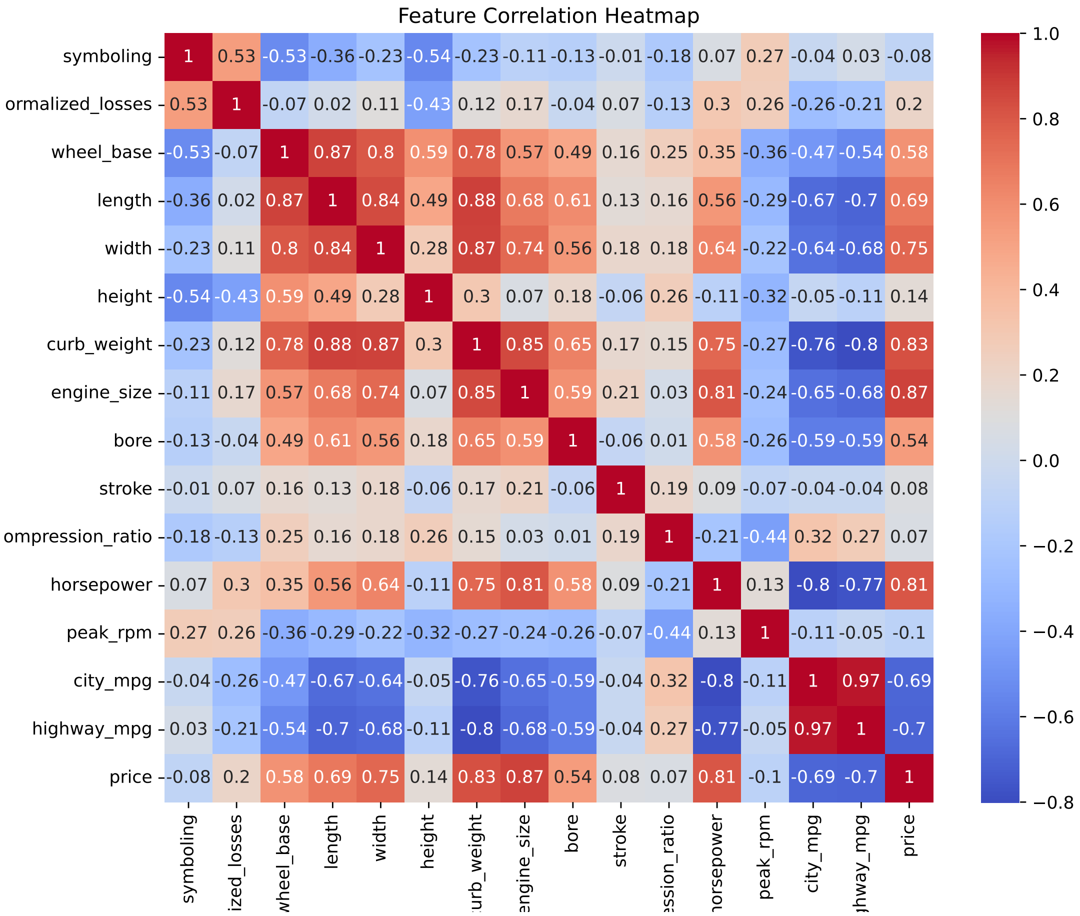
This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(tidyverse)

auto_data <- read.csv("automobile.data", header = FALSE, na.strings = "?")


# change the names of columns of the auto_data dataframe

colnames(auto_data) <- c("symboling", "normalized_losses", "make",
                         "fuel_type", "aspiration", "num_doors",
                         "body_style", "drive_wheels", "engine_location",
                         "wheel_base", "length", "width",
                         "height", "curb_weight", "engine_type",
                         "num_cylinders", "engine_size", "fuel_system",
                         "bore", "stroke", "compression_ratio",
                         "horsepower", "peak_rpm", "city_mpg",
                         "highway_mpg", "price")

auto_data<-auto_data %>% select_if(is.numeric)
# Save the transformed data to a CSV file
write.csv(auto_data, "transformed_data.csv", row.names = FALSE)
```

Heat Map

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# # Load the transformed data from converted from R
transformed_data = pd.read_csv("transformed_data.csv")
corr_matrix = transformed_data.corr()
corr_matrix = round(corr_matrix,2)

# Plot the correlation
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", square=True)
plt.title("Feature Correlation Heatmap")
```



Task 3 Refer the Opinion published on Himalayan times on Dec 19, 2023 and perform a text preprocessing and generate word cloud.

```
import requests
from bs4 import BeautifulSoup
url = 'https://thehimalayantimes.com/opinion/navigating-nepals-digital-frontier-\
understanding-cybersecurity-in-the-digital-age-ensuring-data-safety-and-the-role-of-ai'

x = requests.get(url)
soup = BeautifulSoup(x.content, 'html.parser')
post_content = soup.find('div', {'class': 'post-content'})
paragraphs = post_content.find_all('p')
final_list = ''
for paragraph in range(0, len(paragraphs)-2):
    final_list += (paragraphs[paragraph].text)

with open('himalayan_times.txt','w+') as file:
    file.write(final_list)
```

```
## 5513
```

```r
library(tm)
library(Rgraphviz)
library(wordcloud)

text_document <- readLines('himalayan_times.txt')
corpus <- Corpus(VectorSource(text_document))
```

Text Preprocessing:

- Remove Punctutaion
- Remove Stop Words
- Stemming
- Convert to Lower
- Remove any Numbers
- Any customer remove words

```r
my_stopwords <- c("can","may","used")
corpus <- tm_map(corpus, removeWords, my_stopwords)
my_tdm <- TermDocumentMatrix(
  corpus,
  control =
    list(
      removePunctuation = TRUE,
      stopwords = TRUE,
      tolower = TRUE,
      stemming = FALSE,
      removeNumbers = TRUE,
      bounds = list(global = c(1, Inf)),
      wordLenghts = c(1,Inf),
      removeWords = (c("can","may","used")))
)
```

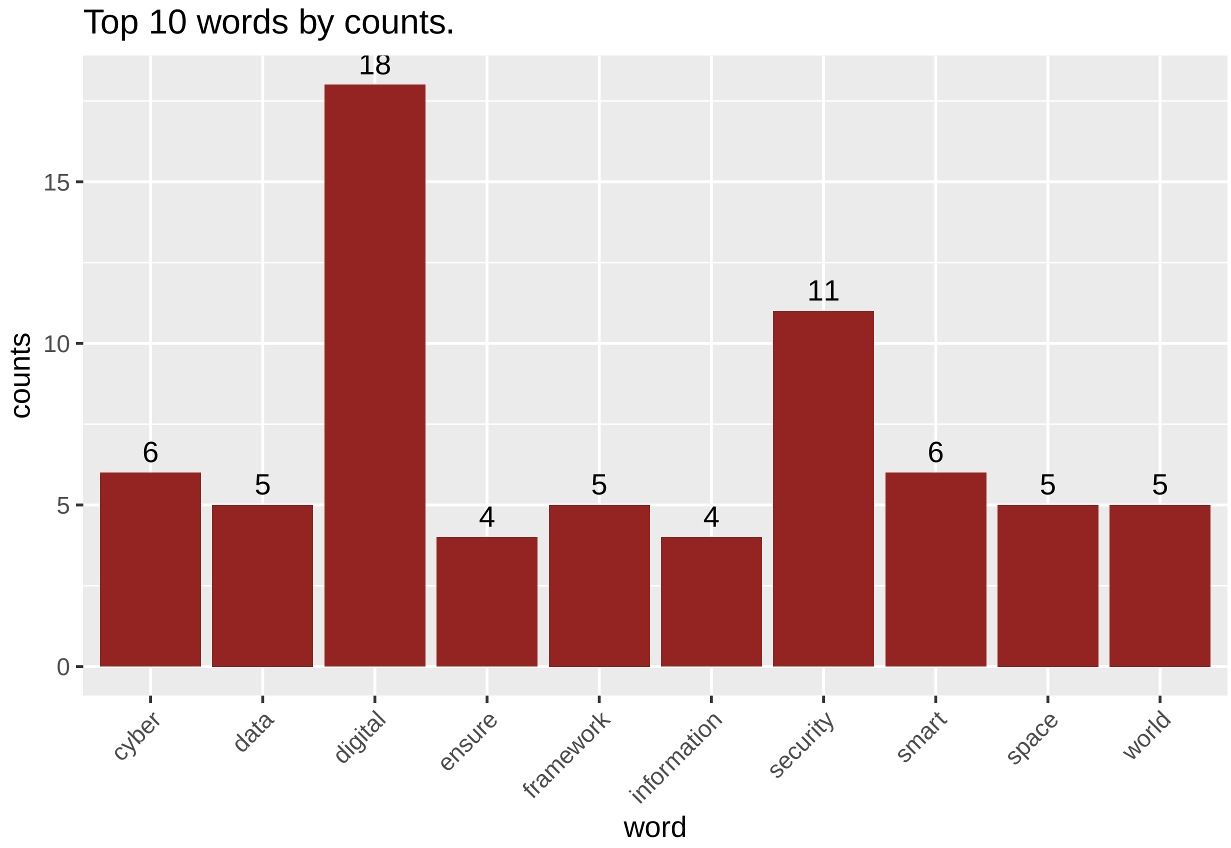
```r
# find the frequent_terms in the corpus
frequent_terms <- findFreqTerms(my_tdm)
head(frequent_terms,20)
```

```
##  [1] "ability"       "access"        "accessed"      "achieve"
##  [5] "additionally"  "adoption"      "advances"      "advent"
##  [9] "ais"           "aithe"         "aligned"       "alikebuilding"
## [13] "allocated"     "allocating"    "allowing"      "along"
## [17] "already"       "also"          "always"        "amounts"
```

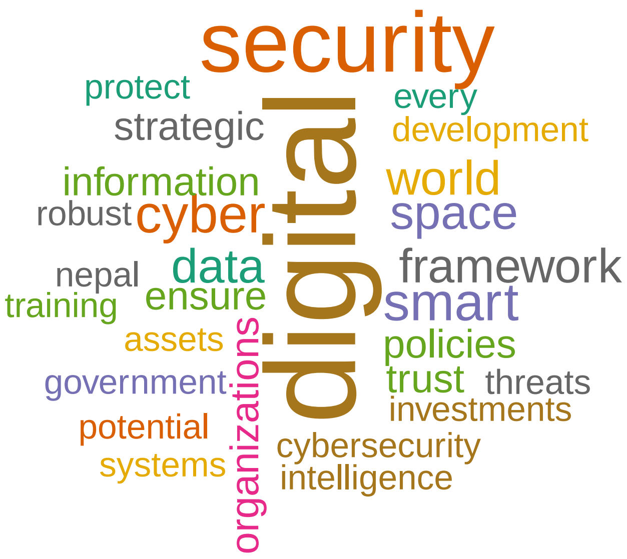
```r
mat <- as.matrix(my_tdm)
freq <- mat %>% rowSums() %>% sort(decreasing = T)

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

```
## # A tibble: 6 x 2
##   word      counts
##   <chr>      <dbl>
## 1 digital       18
## 2 security      11
## 3 cyber          6
## 4 smart          6
## 5 data           5
## 6 framework      5
```

```
# top 10 words and counts using bargraph
library(ggplot2)
ggplot(df, aes(word, counts)) +
  geom_bar(stat = "identity", fill = "#932421") +
  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)
```



```
# plot word cloud
wordcloud(
  words = names(freq),
  freq = freq,
  random.order = FALSE,
  colors = brewer.pal(8, "Dark2"),
  scale = c(4, 0.5),
  random.color = TRUE,
)
```


security
protect
every
strategic
development
digital
information
world
robust
cyber
space
nepal
data
framework
training
ensure
smart
assets
policies
organizations
government
trust
threats
investments
potential
cybersecurity
systems
intelligence