STA 305 HW54

Jacky Guan

2023-11-24

Loading Required libraries

```
library(tidyverse)
library(openintro)
knitr::opts_chunk$set(warning = FALSE, message = FALSE, fig.align = "center")
```

```
setwd("Documents/SEM\ 3/STA\ 305/Homework/HW5")
```

1. Regular expressions

a) Use the words data set, find all the words that match the following pattern:

- are exactly four letters long

```
words_4 <- words %>%
  tibble() %>%
  filter(nchar(.) == 4)

print(words_4)
```

```
## # A tibble: 263 x 1
##    .
##    <chr>
##  1 able
##  2 also
##  3 area
##  4 away
##  5 baby
##  6 back
##  7 ball
##  8 bank
##  9 base
## 10 bear
## # i 253 more rows
```

- are either four or five letters long

```
words_45 <- words %>%
  tibble() %>%
  filter(nchar(.) == 4 | nchar(.) == 5)

print(words_45)
```

```
## # A tibble: 463 x 1
##    .
##    <chr>
##    1 able
##    2 about
##    3 admit
##    4 after
##    5 again
##    6 agent
##    7 agree
##    8 allow
##    9 along
##   10 also
## # i 453 more rows
```

- the second letter is “s” or “t”

```
s_t <- words %>%
  tibble() %>%
  filter(str_detect(words, "^.{1}[st]"))

print(s_t)
```

```
## # A tibble: 38 x 1
##      .
##      <chr>
##    1 as
##    2 ask
##    3 associate
##    4 assume
##    5 at
##    6 attend
##    7 especial
##    8 issue
##    9 it
##   10 item
## # i 28 more rows
```

- contains the pattern like “oxx” where “o” is one letter and “x” is another letter

```
oxx <- words %>%
  tibble() %>%
  filter(str_detect(words, "o(.)\\1"))

print(oxx)
```

```
## # A tibble: 28 x 1
##      .
##      <chr>
##    1 across
##    2 bottle
##    3 bottom
##    4 coffee
##    5 colleague
##    6 collect
##    7 college
##    8 comment
##    9 commit
##   10 committee
## # i 18 more rows
```

- contains “a”, “e” and “o” at the same time

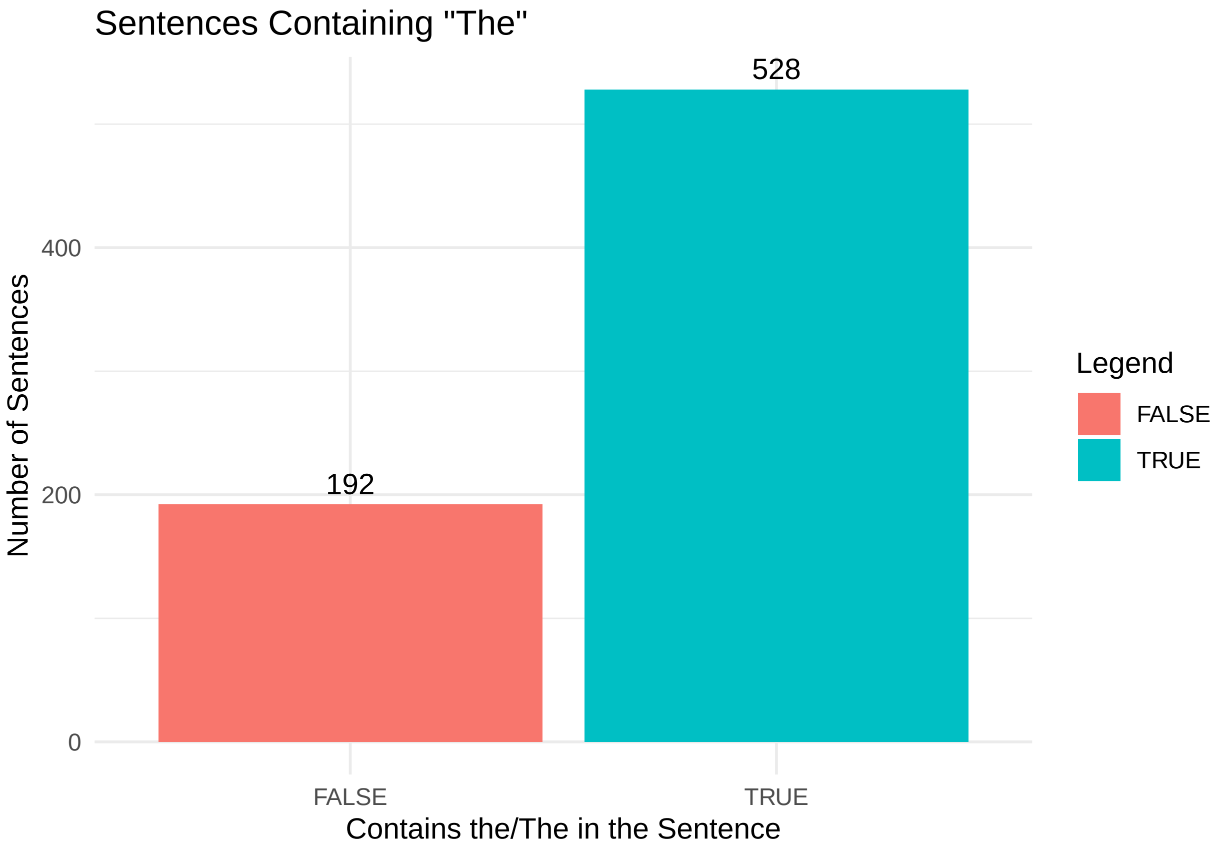
```r
aeo <- words %>%
  tibble() %>%
  filter(str_detect(words, "^(?=.*a)(?=.*e)(?=.*o).*"))

print(aeo)
```

```
## # A tibble: 14 x 1
##    .
##    <chr>
##  1 absolute
##  2 afternoon
##  3 another
##  4 appropriate
##  5 associate
##  6 colleague
##  7 compare
##  8 encourage
##  9 operate
## 10 organize
## 11 probable
## 12 programme
## 13 reason
## 14 relation
```

b) Use the sentences data set, make the following plot

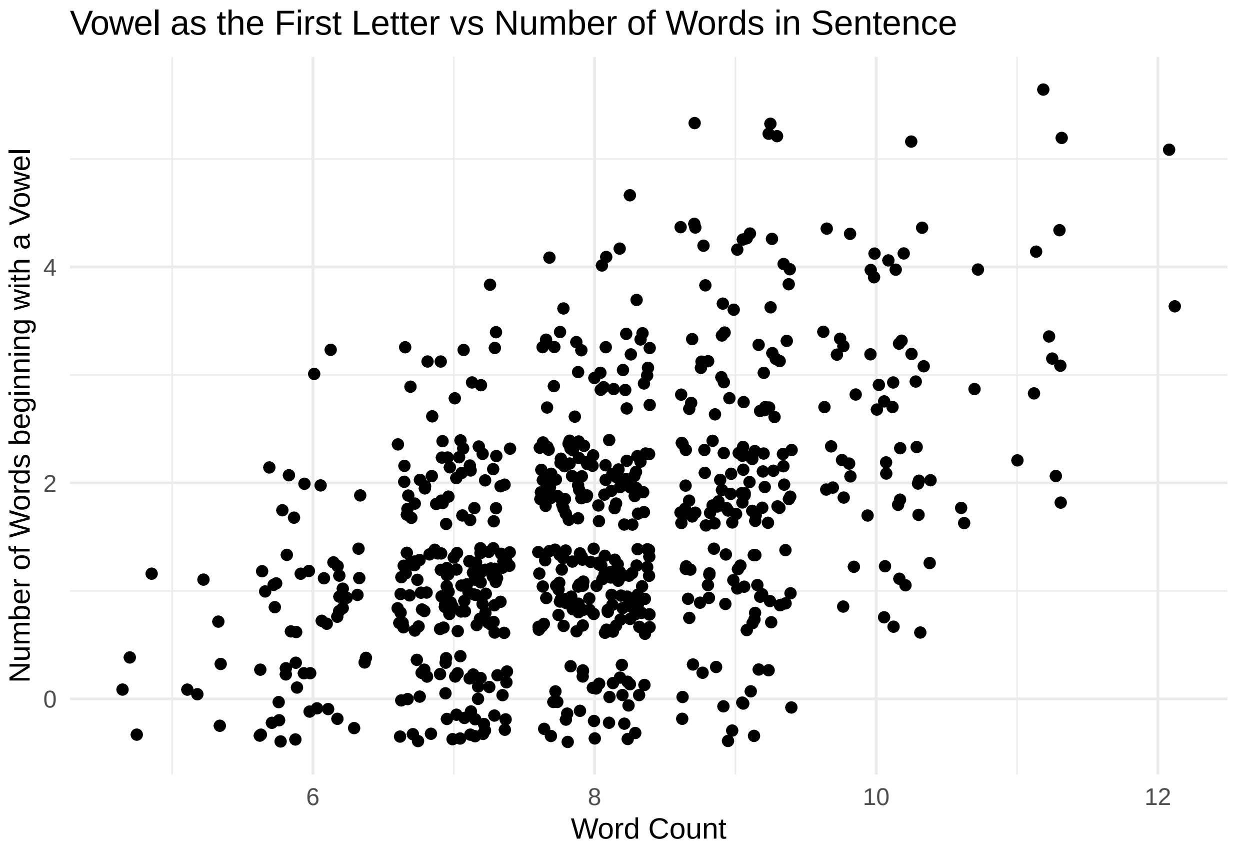
- a bar plot counting sentences with and without “the” (or “The”).

```r
the <- sentences %>%
  str_detect("(the|The)") %>%
  tibble() %>%
  rename(Contains_The = ".")

ggplot(the) +
  geom_bar(aes(Contains_The, fill = Contains_The)) +
  theme_minimal() +
  labs(x = "Contains the/The in the Sentence",
       y = "Number of Sentences",
       title = "Sentences Containing \"The\"",
       fill = "Legend") +
  geom_text(
    aes(x = factor(Contains_The), label = after_stat(count)),
    stat = "count",
    vjust = -0.5,
    color = "black",
    size = 4
  )
```



- a scatterplot with x being the average length of words in a sentence, and y being the number of words starting with “a” or “e” or “i” or “o” or “u” in the sentence.

```
sentence_summary <- sentences %>%
  tibble() %>%
  rename(sentence = ".") %>%
  mutate(word_count = str_count(sentence, " ") + 1,
         vowel_start = str_count(sentence, "\\b[aeiouAEIOU]")) %>%
  ggplot() +
  geom_point(aes(word_count, vowel_start), position = "jitter") +
  theme_minimal() +
  labs(x = "Word Count",
       y = "Number of Words beginning with a Vowel",
       title = "Vowel as the First Letter vs Number of Words in Sentence")

sentence_summary
```



c) Application

i) Download the Oxford English Dictionary as a ".txt" file from https://canvas.feitian.edu/files/9699/ download?download_frd=1 Done

```
oxford <- tibble(read.delim("Oxford_English_Dictionary.txt"))
```

ii) Read it into RStudio with read_lines() function (check how to use it by yourself)

iii) Turn the dictionary into a tibble and remove all blank lines

```
oxford <- oxford %>%
  rename(definition = A) %>%
  mutate(word = str_extract(definition, "\\b\\w+")) %>%
  select(word, definition)
```

iv) Use regular expression to extract all words for each item in a separate column named "words"

2. Factors

```
bank <- tibble(read.csv("BankChurners.csv"))
```

a) Use the BankChurners.csv to answer the following questions:

- Which features can be regarded as a factor?

Factor features include Attrition_Flag, Gender, Dependent_count, Education_Level, Marital_Status, Income_Category, and Card_Category.

- Which features can be regarded as an ordered factor (ordinal)?

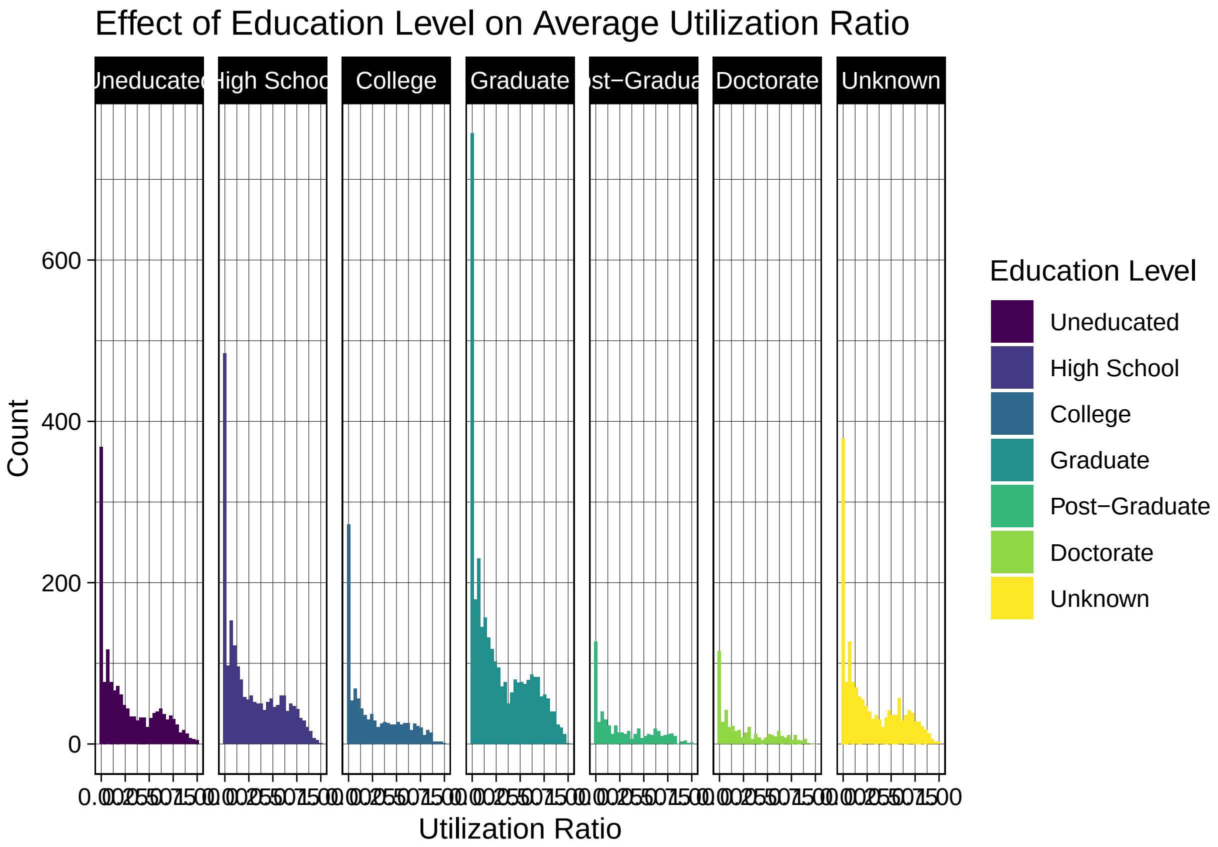
Of the aforementioned factors, Education_Level, Income_Category, Dependent_count, and Card_Category can be considered ordered factors.

- Read BankChurners.csv into RStudio, then change the columns that you answered above into factors or ordered factors.

```
bank <- bank %>%
  mutate(across(c(Attrition_Flag, Gender, Marital_Status), as.factor)) %>%
  mutate(Dependent_count = factor(Dependent_count, ordered = TRUE,
                                  levels = c("0", "1", "2", "3",
                                             "4", "5", "6"))) %>%
  mutate(Education_Level = factor(Education_Level, ordered = TRUE,
                                  levels = c("Uneducated", "High School",
                                             "College", "Graduate",
                                             "Post-Graduate", "Doctorate",
                                             "Unknown"))) %>%
  mutate(Income_Category = factor(Income_Category, ordered = TRUE,
                                  levels = c("Less than $40K", "$40K - $60K",
                                             "$60K - $80K", "$80K - $120K",
                                             "$120K +"))) %>%
  mutate(Card_Category = factor(Card_Category, ordered = TRUE,
                                levels = c("Blue", "Silver", "Gold")))
```

- Visualize the effect of education level on Average Utilization Ratio

```
ggplot(bank) +
  facet_grid(~ Education_Level) +
  geom_histogram(aes(Avg_Utilization_Ratio, fill = Education_Level)) +
  theme_linedraw() +
  labs(x = "Utilization Ratio",
       y = "Count",
       title = "Effect of Education Level on Average Utilization Ratio",
       fill = "Education Level")
```



b) Use the gss_cat data set

- What are the levels of marital variable?

The levels of marital status include the following:

```
levels(gss_cat$marital)
```

```
## [1] "No answer"     "Never married" "Separated"     "Divorced"
## [5] "Widowed"       "Married"
```

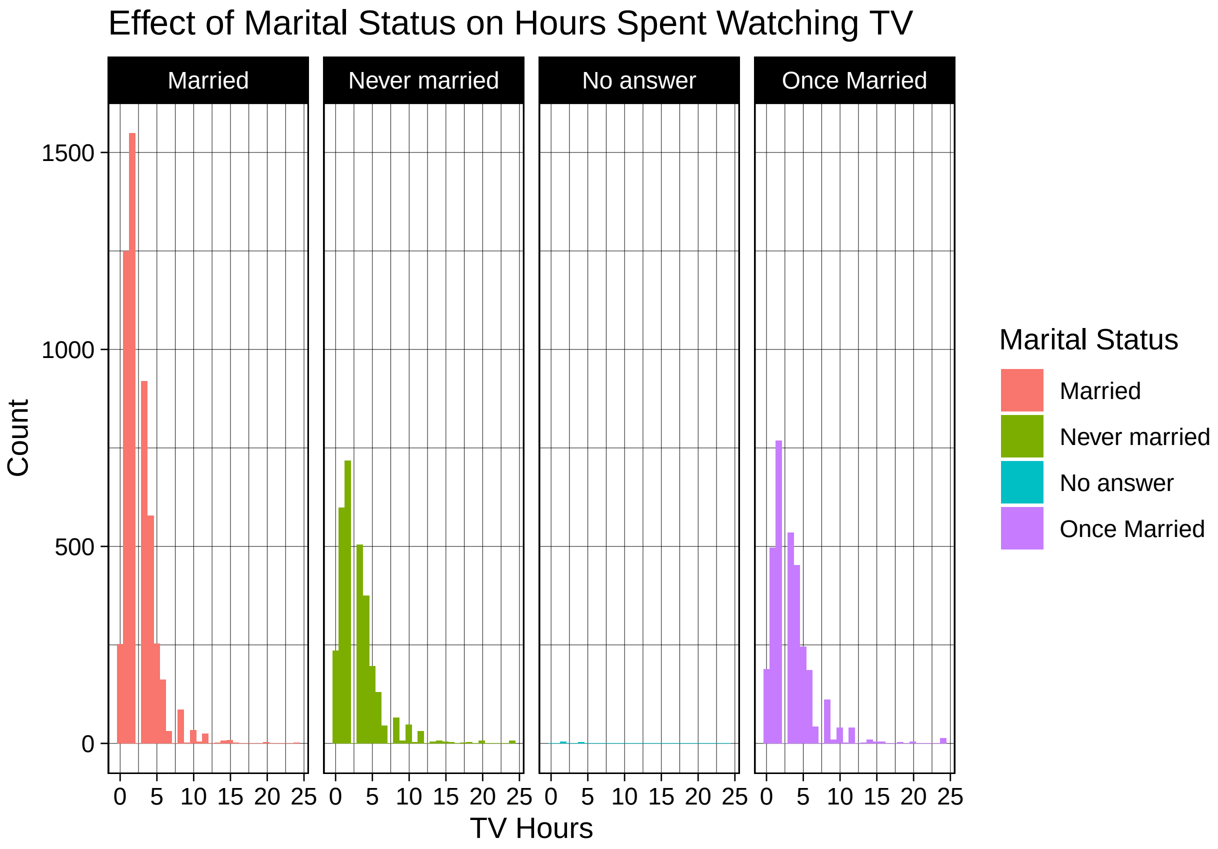
- Combine “Separated”, “Divorced”, “Widowed” into a new category “Once Married”

```
gss_cat <- gss_cat %>%
  mutate(marital = case_when(
    marital == "Separated" ~ "Once Married",
    marital == "Divorced" ~ "Once Married",
    marital == "Widowed" ~ "Once Married",
    TRUE ~ marital
  ))
```

- Use the new levels, explore whether there is an effect of martial status on tvhours.

It seems that there is a plausible correlation with being once married and the increse in time spent in front of a tv. For people once married, the median time spent in front of a tv seems to be greater compared to unmarried and married individuals.

```
ggplot(gss_cat) +
  facet_grid(~ marital) +
  geom_histogram(aes(tvhours, fill = marital)) +
  theme_linedraw() +
  labs(x = "TV Hours",
       y = "Count",
       title = "Effect of Marital Status on Hours Spent Watching TV",
       fill = "Marital Status")
```


Effect of Marital Status on Hours Spent Watching TV
Married
Never married
No answer
Once Married
Count
TV Hours
Marital Status
Married
Never married
No answer
Once Married