Exercises 9

Brady Lamson

2022-04-14

Exercise 1

```
paste("Today's date is", date())
```

```
## [1] "Today's date is Thu Apr 14 13:03:41 2022"
```

```
paste("X", 1:5, sep = "")
```

```
## [1] "X1" "X2" "X3" "X4" "X5"
```

Exercise 2

```
# A

first <- "Louis"
middle <- "Daniel"
last <- "Armstrong"

paste(first, middle, last)
```

```
## [1] "Louis Daniel Armstrong"
```

```
# B

first <- c("Louis", "John", "Miles", "Ella")
middle <- c("Daniel", "William", "Dewey", "Jane")
last <- c("Armstrong", "Coltrane", "Davis", "Fitzgerald")

paste(first, middle, last)
```

```
## [1] "Louis Daniel Armstrong" "John William Coltrane"  "Miles Dewey Davis"
## [4] "Ella Jane Fitzgerald"
```

```
# C

full <- "Sarah Lois Vaughan"

strsplit(full, split = " ")
```

```
## [[1]]
## [1] "Sarah"   "Lois"    "Vaughan"
```

```
# D

full <- c("Sarah Lois Vaughan", "Thelonious Sphere Monk", "Chet Henry Baker",
"Wynton Learson Marsalis")

strsplit(full, split = " ")
```

```
## [[1]]
## [1] "Sarah"   "Lois"    "Vaughan"
##
## [[2]]
## [1] "Thelonious" "Sphere"     "Monk"
##
## [[3]]
## [1] "Chet"  "Henry" "Baker"
##
## [[4]]
## [1] "Wynton"   "Learson"  "Marsalis"
```

Exercise 3

```
quote <- "Aunt Petunia was horse-faced and bony; Dudley was blond, pink, and
porky. Harry, on the other hand, was small and skinny, with brilliant green
eyes and jet-black hair that was always untidy. He wore round glasses, and on
his forehead was a thin, lightning-shaped scar."

quote <-
    quote %>%
    gsub(pattern = "\n", replacement = " ") %>%
    gsub(pattern = "[\\.,;]", replacement = "")

quote
```

```
## [1] "Aunt Petunia was horse-faced and bony Dudley was blond pink and porky Harry on the other hand wa
```

```
gregexpr(pattern = "-", text = quote)
```

```
## [[1]]
## [1]  23 149 246
## attr(,"match.length")
## [1] 1 1 1
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

The three values returned represent the indices where dashes appear. That is, the 23rd character, the 149th character and the 246th character.

```
regexpr(pattern = "-", text = quote)
```

```
## [1] 23
## attr(,"match.length")
## [1] 1
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

This represents the index of the first instance of a dash appearing in the character string.

Exercise 4

```
# A

badges <-
    "Badges? We ain't got no badges. We don't need no badges. I don't have to
show you any stinking badges!" %>%
    gsub(pattern = "\n", replacement = " ") %>%
    tolower()

badges
```

```
## [1] "badges? we ain't got no badges. we don't need no badges. i don't have to show you any stinking b
```

```
# B

badges <-
    badges %>%
    gsub(pattern = "!", replacement = "")

badges
```

```
## [1] "badges? we ain't got no badges. we don't need no badges. i don't have to show you any stinking b
```

```
# C

badges <-
    badges %>%
    gsub(pattern = "[\\?,\\.]", replacement = "")

badges
```

```
## [1] "badges we ain't got no badges we don't need no badges i don't have to show you any stinking badg
```

```
# D

badges <-
    badges %>%
    strsplit(split = " ")

badges
```

```
## [[1]]
##  [1] "badges"   "we"       "ain't"    "got"      "no"       "badges"
##  [7] "we"       "don't"    "need"     "no"       "badges"   "i"
## [13] "don't"    "have"     "to"       "show"     "you"      "any"
## [19] "stinking" "badges"
```

```
# E

badges <-
    badges %>%
    unlist()
```

```
badges %>%
    grep(pattern = "badges")
```

```
## [1]  1  6 11 20
```

```
badges %>%
    grepl(pattern = "badges")
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

```
# F

badges %>%
    nchar()
```

```
##  [1] 6 2 5 3 2 6 2 5 4 2 6 1 5 4 2 4 3 3 8 6
```

Exercise 5

```
# A / B

quote <-
    "I have a dream that one day this nation will rise up and live out
the true meaning of its creed, 'We hold these truths to be self-evident,
that all men are created equal.' I have a dream that one day on the
red hills of Georgia, sons of former slaves and the sons of former
slave owners will be able to sit down together at the table of
brotherhood. I have a dream that one day even the state of Mississippi,
a state sweltering with the heat of injustice, sweltering with the
heat of oppression, will be transformed into an oasis of freedom and
justice. I have a dream that my four little children will one day live
in a nation where they will not be judged by the color of their skin
but by the content of their character." %>%
    gsub(pattern = "\n", replacement = " ") %>%
    gsub(pattern = "[\\.,']", replacement = " ") %>%
    strsplit(split = " ") %>%
    unlist()
```

```
q_nchars <- nchar(quote)
```

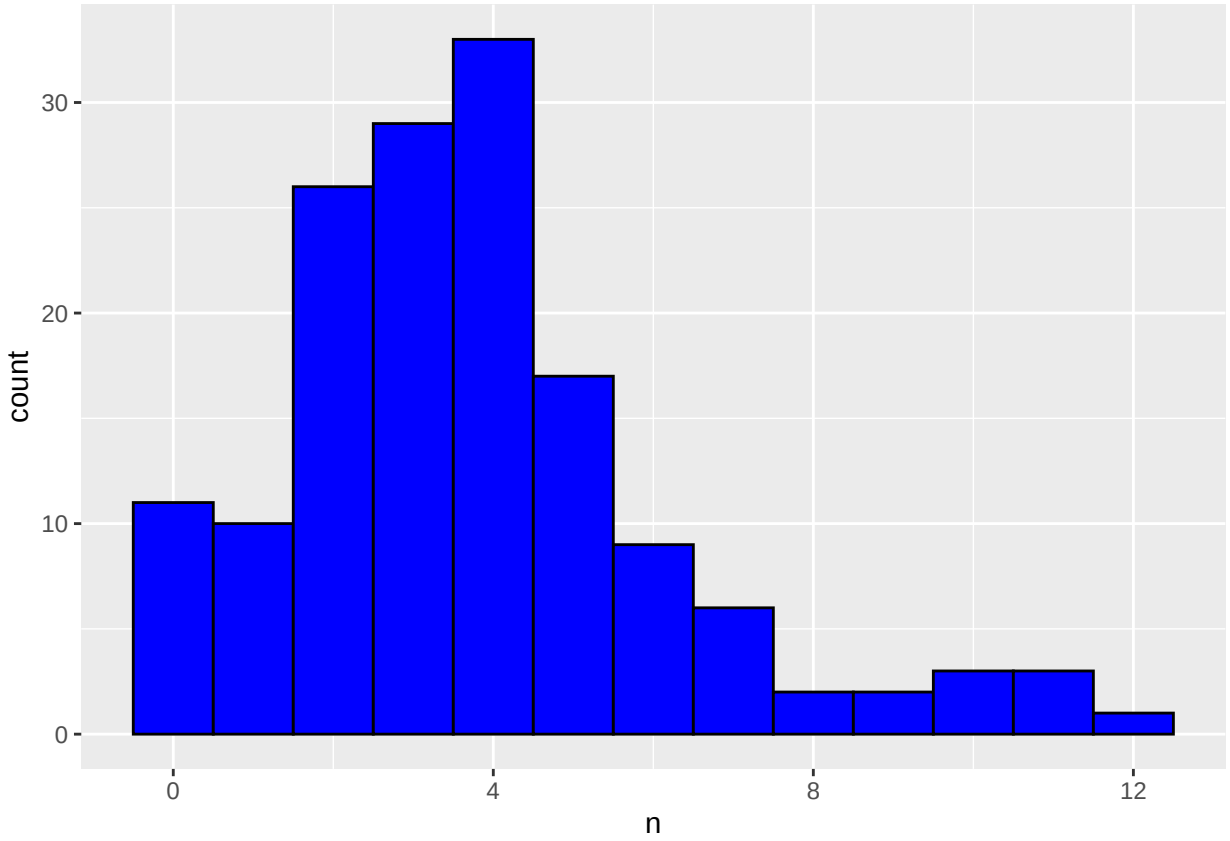
```
# C

mean(q_nchars)
```

```
## [1] 3.756579
```

```
# D

ggplot(data = data.frame(n = q_nchars)) +
    geom_histogram(mapping = aes(x = n), fill = "blue", color = "black", binwidth = 1)
```



This histogram is absolutely right-skewed.

Exercise 6

```
quote <- "I have a dream that one day this nation will rise up and live out
the true meaning of its creed, 'We hold these truths to be self-evident,
that all men are created equal.' I have a dream that one day on the
red hills of Georgia, sons of former slaves and the sons of former
slave owners will be able to sit down together at the table of
brotherhood. I have a dream that one day even the state of Mississippi,
a state sweltering with the heat of injustice, sweltering with the
heat of oppression, will be transformed into an oasis of freedom and
justice. I have a dream that my four little children will one day live
in a nation where they will not be judged by the color of their skin
but by the content of their character." %>%
    gsub(pattern = "\n", replacement = " ") %>%
    gsub(pattern = "[\\.,']", replacement = "")
```

```
#A

quote %>%
    gregexpr(pattern = "freedom")
```

```
## [[1]]
## [1] 524
## attr(,"match.length")
## [1] 7
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

The starting character position of "freedom" is index 534.

```
# B

quote.list <- strsplit(quote, split = " ")
quote.vec <- unlist(quote.list)    # Could also use quote.vec <- quote.list[[1]]
quote.vec
```

```
##   [1] "I"            "have"         "a"            "dream"        "that"
##   [6] "one"          "day"          "this"         "nation"       "will"
##  [11] "rise"         "up"           "and"          "live"         "out"
##  [16] "the"          "true"         "meaning"      "of"           "its"
##  [21] "creed"        "We"           "hold"         "these"        "truths"
##  [26] "to"           "be"           "self-evident" "that"         "all"
##  [31] "men"          "are"          "created"      "equal"        "I"
##  [36] "have"         "a"            "dream"        "that"         "one"
##  [41] "day"          "on"           "the"          "red"          "hills"
##  [46] "of"           "Georgia"      "sons"         "of"           "former"
##  [51] "slaves"       "and"          "the"          "sons"         "of"
##  [56] "former"       "slave"        "owners"       "will"         "be"
##  [61] "able"         "to"           "sit"          "down"         "together"
##  [66] "at"           "the"          "table"        "of"           "brotherhood"
##  [71] "I"            "have"         "a"            "dream"        "that"
##  [76] "one"          "day"          "even"         "the"          "state"
##  [81] "of"           "Mississippi"  "a"            "state"        "sweltering"
##  [86] "with"         "the"          "heat"         "of"           "injustice"
```

```
##  [91] "sweltering"   "with"         "the"          "heat"         "of"
##  [96] "oppression"   "will"         "be"           "transformed"  "into"
## [101] "an"           "oasis"        "of"           "freedom"      "and"
## [106] "justice"      "I"            "have"         "a"            "dream"
## [111] "that"         "my"           "four"         "little"       "children"
## [116] "will"         "one"          "day"          "live"         "in"
## [121] "a"            "nation"       "where"        "they"         "will"
## [126] "not"          "be"           "judged"       "by"           "the"
## [131] "color"        "of"           "their"        "skin"         "but"
## [136] "by"           "the"          "content"      "of"           "their"
## [141] "character"
```

This returns everything separated by spaces, which gives us every single individual word as a separate string in a vector.

```
# C

grep(pattern = "the", x = quote.vec)
```

```
##  [1]  16  24  43  53  65  67  70  79  87  93 124 130 133 137 140
```

```
which(quote.vec == "the")
```

```
## [1]  16  43  53  67  79  87  93 130 137
```

`grep()` simply looks for the letters "the" in each string, the word itself doesn't need to be "the" exactly. The `which()` command is different though, it only returns the indices for the word "the".

```
# D

grep(pattern = "ing$", x = quote.vec, value = TRUE)
```

```
## [1] "meaning"    "sweltering" "sweltering"
```

This checks for strings ending in "ing" specifically.

```
grep(pattern = "^th", x = quote.vec, value = TRUE)
```

```
##  [1] "that"  "this"  "the"   "these" "that"  "that"  "the"   "the"   "the"
## [10] "that"  "the"   "the"   "the"   "that"  "they"  "the"   "their" "the"
## [19] "their"
```

This is the `$` equivalent for the beginning of a string. So this finds all strings that start with "th".

Exercise 7

```
county1 <-
    c("De Witt County", "Lac qui Parle County", "Lewis and Clark County",
      "St John the Baptist Parish")

county2 <-
    c("De Witt County", "Lac qui Parle County", "Lewis and Clark County",
      "St. John the Baptist Parish")

county3 <-
    c("De Witt", "Lac Qui Parle", "Lewis & Clark", "St. John the Baptist")
```

```
# A and B

# Extra whitespace added to "county" to capture the whitespace before it.
county1_new <-
    county1 %>%
    gsub(pattern = " County", replacement = "") %>%
    gsub(pattern = " Parish", replacement = "")
county1_new
```

```
## [1] "De Witt"             "Lac qui Parle"       "Lewis and Clark"
## [4] "St John the Baptist"
```

```
county2_new <-
    county2 %>%
    gsub(pattern = " County", replacement = "") %>%
    gsub(pattern = " Parish", replacement = "")
county2_new
```

```
## [1] "De Witt"              "Lac qui Parle"        "Lewis and Clark"
## [4] "St. John the Baptist"
```

```
# C
substr(x = county1, start = 1, stop = nchar(county1) - 7)
```

```
## [1] "De Witt"             "Lac qui Parle"       "Lewis and Clark"
## [4] "St John the Baptist"
```

```
substr(x = county2, start = 1, stop = nchar(county2) - 7)
```

```
## [1] "De Witt"              "Lac qui Parle"        "Lewis and Clark"
## [4] "St. John the Baptist"
```

```
# D

gsub(pattern = " County| Parish", replacement = "", x = county1)
```

```
## [1] "De Witt"             "Lac qui Parle"       "Lewis and Clark"
## [4] "St John the Baptist"
```

```
gsub(pattern = " County| Parish", replacement = "", x = county2)
```

```
## [1] "De Witt"              "Lac qui Parle"        "Lewis and Clark"
## [4] "St. John the Baptist"
```

```
# E
```

```
gsub(pattern = " County|Parish", replacement = "", x = county1)
```

```
## [1] "De Witt"              "Lac qui Parle"        "Lewis and Clark"
## [4] "St John the Baptist "
```

```
gsub(pattern = " County|Parish", replacement = "", x = county2)
```

```
## [1] "De Witt"               "Lac qui Parle"         "Lewis and Clark"
## [4] "St. John the Baptist "
```

This fails to capture the whitespace before Parish.

```
# F

gsub(pattern = ".", replacement = "", x = county2)
```

```
## [1] "" "" "" ""
```

```
gsub(pattern = ".", replacement = "", x = county3)
```

```
## [1] "" "" "" ""
```

The period just means "anything" essentially, if we want to remove periods we need to either escape the character or set `fixed = TRUE`

```
# G

gsub(pattern = "\\.", replacement = "", x = county2)
```

```
## [1] "De Witt County"             "Lac qui Parle County"
## [3] "Lewis and Clark County"     "St John the Baptist Parish"
```

```
gsub(pattern = "\\.", replacement = "", x = county3)
```

```
## [1] "De Witt"             "Lac Qui Parle"       "Lewis & Clark"
## [4] "St John the Baptist"
```

```
# I

gsub(pattern = ".", replacement = "", x = county2, fixed = TRUE)
```

```
## [1] "De Witt County"             "Lac qui Parle County"
## [3] "Lewis and Clark County"     "St John the Baptist Parish"
```

```
gsub(pattern = ".", replacement = "", x = county3, fixed = TRUE)
```

```
## [1] "De Witt"             "Lac Qui Parle"       "Lewis & Clark"
## [4] "St John the Baptist"
```

Exercise 8

```
my.string <- "<p> This is a short paragraph.</p>"

gregexpr(pattern = "<.*>", text = my.string)
```

```
## [[1]]
## [1] 1
## attr(,"match.length")
## [1] 34
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

This is capturing the pattern "<.>", essentially, the backslash in the second instance of the pattern takes the place of the p, and since theres another character inside of those brackets it doesn't consider this a pattern.

The above is actually wrong, based on what I learned in the lecture it seems that this returns the entire string. Since the `.*` allows for any number of characters, it just seems everything on the outside. *** ###

Exercise 9

```
# sotu.wrd.vec <- scan('state_of_the_union.txt',
#                      what = "",
#                      blank.lines.skip = TRUE)

sotu.wrd.vec <-
  readr::read_file("state_of_the_union.txt") %>%
  stringr::str_squish() %>%
  stringr::str_split(pattern = " ")
```

```
# This creates a single "character" string (one-element vector) containing all speeches.
sotu.string <- paste(sotu.wrd.vec, collapse = " ")

# This splits sotu.string into separate speeches (demarcated by ***):
sotu.list <- strsplit(sotu.string, split = "\\*\\*\\*")

# This converts the one-element sotu.list to a vector:
sotu.spch.vec <- unlist(sotu.list)

# Could also use sotu.spch.vec <- sotu.list[[1]]

# This removes the empty first element:
sotu.spch.vec <- sotu.spch.vec[-1]
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
# Create a "VectorSource" class object:
sotu.vecsrc <- VectorSource(sotu.spch.vec)

# Convert to a "Corpus" class object:
sotu.corp <- VCorpus(sotu.vecsrc)
```

```
# This cleans the speeches:
sotu.corp <-
    sotu.corp %>%
    tm_map(FUN = stripWhitespace) %>%
    tm_map(FUN = removeNumbers) %>%
    tm_map(FUN = removePunctuation) %>%
    tm_map(FUN = content_transformer(tolower)) %>%
    tm_map(FUN = removeWords, stopwords("english"))
```

```
dtm <- DocumentTermMatrix(sotu.corp)
```

```
dtm_sums <-
    dtm %>%
    as.matrix() %>%
    apply(MARGIN = 2, sum)
```

```r
subset(dtm_sums, dtm_sums > 3000)
```

```
##        can   congress    country government      great       made        may
##       4226       4849       3261       6517       3177       3029       3290
##       must        now     people     states     united       upon       will
##       3145       3017       3820       6272       4651       3744       9158
##       year
##       3533
```

```r
# B

findAssocs(dtm, terms = "peace", corlimit = .5) %>%
    head()
```

```
## $peace
##           war      military          army        forces contributions
##          0.59          0.55          0.54          0.54          0.51
##       nations
##          0.50
```