Assignment2_445

Michael Rivera

2023-10-17

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(lubridate)
```

Question 1

Convert the following to date or date/time objects.

a. September 13, 2010.

```
mdy('September 13, 2010')
```

```
## [1] "2010-09-13"
```

b. Sept 13, 2010.

```
make_date(year = 2010, month = 9, day = 13)
```

```
## [1] "2010-09-13"
```

c. Sep 13, 2010.

```
mdy('Sep 13, 2010')
```

```
## [1] "2010-09-13"
```

d. S 13, 2010. Comment on the month abbreviation needs.

```
make_date(year = 2010, month = 9, day = 13)
```

```
## [1] "2010-09-13"
```

```
# it appears that you need the month as a number, a 3 character abbreviation, or the full month written
```

e. 07-Dec-1941.

```
dmy('07-Dec-1941')
```

```
## [1] "1941-12-07"
```

f. 1-5-1998. Comment on why you might be wrong.

```r
dmy('1-5-1998')
```

```
## [1] "1998-05-01"
```

```r
mdy('1-5-1998')
```

```
## [1] "1998-01-05"
```

```r
# due to the format
# there is some ambiguity
# on if 1 and 5 refer to
# the 1st or 5th of January/May
```

g. 21-5-1998. Comment on why you know you are correct.

```r
dmy('21-5-1998')
```

```
## [1] "1998-05-21"
```

```r
# I know I chose the correct format here
# because there is no doubt about
# which number refers to the day/month/year
# because months only go up to 12 and years
# are the only 4 digit numbers
```

h. 2020-May-5 10:30 am

```r
ymd_hm('2020-May-5 10:30 am')
```

```
## [1] "2020-05-05 10:30:00 UTC"
```

i. 2020-May-5 10:30 am PDT (ex Seattle)

```r
ymd_hm('2020-May-5 10:30 am', tz = 'US/Pacific')
```

```
## [1] "2020-05-05 10:30:00 PDT"
```

j. 2020-May-5 10:30 am AST (ex Puerto Rico)

```r
ymd_hm('2020-May-5 10:30 am', tz = 'America/Puerto_Rico')
```

```
## [1] "2020-05-05 10:30:00 AST"
```

Question 2

Using just your date of birth (ex Sep 7, 1998) and today's date calculate the following Write your code in a manner that the code will work on any date after you were born.:

a. Calculate the date of your 64th birthday.

```r
bday <- dmy('27 June, 1997')
bday
```

```
## [1] "1997-06-27"
```

```r
bday + years(64)
```

```
## [1] "2061-06-27"
```

b. Calculate your current age (in years). Hint: Check your age is calculated correctly if your birthday was yesterday and if it were tomorrow!

```
bday
```

```
## [1] "1997-06-27"
```

```
timenow <- lubridate::now()
timenow
```

```
## [1] "2023-10-24 15:24:06 MST"
```

```
age = interval(bday, timenow)

agenow <- as.period(age)
agenow
```

```
## [1] "26y 3m 27d 22H 24M 6.14233303070068S"
```

c. Using your result in part (b), calculate the date of your next birthday.

```
#nextbday2 <- dmy('27 June 2024')
#nextbday2

nextbday <- update(bday, year=2024)
nextbday
```

```
## [1] "2024-06-27"
```

d. The number of days until your next birthday.

```
daysto <- interval(timenow, nextbday)

daystonextbday <- as.period(daysto, unit = 'days')

day(daystonextbday)
```

```
## [1] 246
```

e. The number of months and days until your next birthday.

```
MDtonextbday <- as.period(daysto)
month(MDtonextbday)
```

```
## [1] 8
```

```
day(MDtonextbday)
```

```
## [1] 2
```

Question 3

Suppose you have arranged for a phone call to be at 3 pm on May 8, 2015 at Arizona time. However, the recipient will be in Auckland, NZ. What time will it be there?

```
mycall <- dmy_hm('8 May, 2015 3:00 pm', tz = 'US/Arizona')
mycall
```

```
## [1] "2015-05-08 15:00:00 MST"
```

```
mycall %>%
  with_tz(., tzone = 'NZ')
```

```
## [1] "2015-05-09 10:00:00 NZST"
```

Question 5

It turns out there is some interesting periodicity regarding the number of births on particular days of the year.

a. Using the mosaicData package, load the data set Births78 which records the number of children born on each day in the United States in 1978.

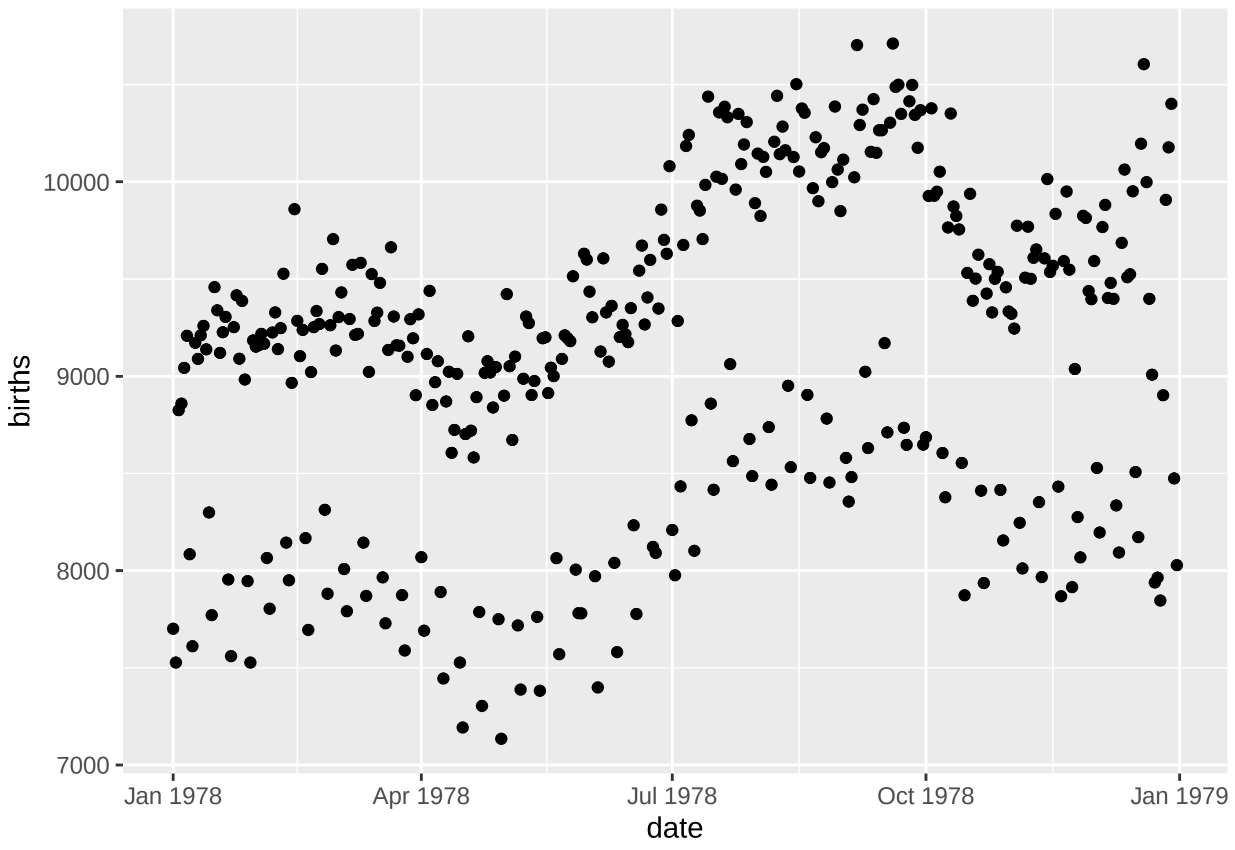
Because this problem is intended to show how to calculate the information using the date, remove all the columns except date and births.

```r
library(mosaicData)
data(Births78)
my78 <- Births78[c(1,2)]
head(my78)
```

```
##         date births
## 1 1978-01-01   7701
## 2 1978-01-02   7527
## 3 1978-01-03   8825
## 4 1978-01-04   8859
## 5 1978-01-05   9043
## 6 1978-01-06   9208
```

b. Graph the number of births vs the date with date on the x-axis. What stands out to you? Why do you think we have this trend?

```r
ggplot(data = my78, aes(x = date, y = births)) +
  geom_point()
```



```r
# There seems to be periods where births are higher
# rather than a consistently spread out distribution
# This might be due to people preferring to give
# birth on certain days of the week like the weekend
```

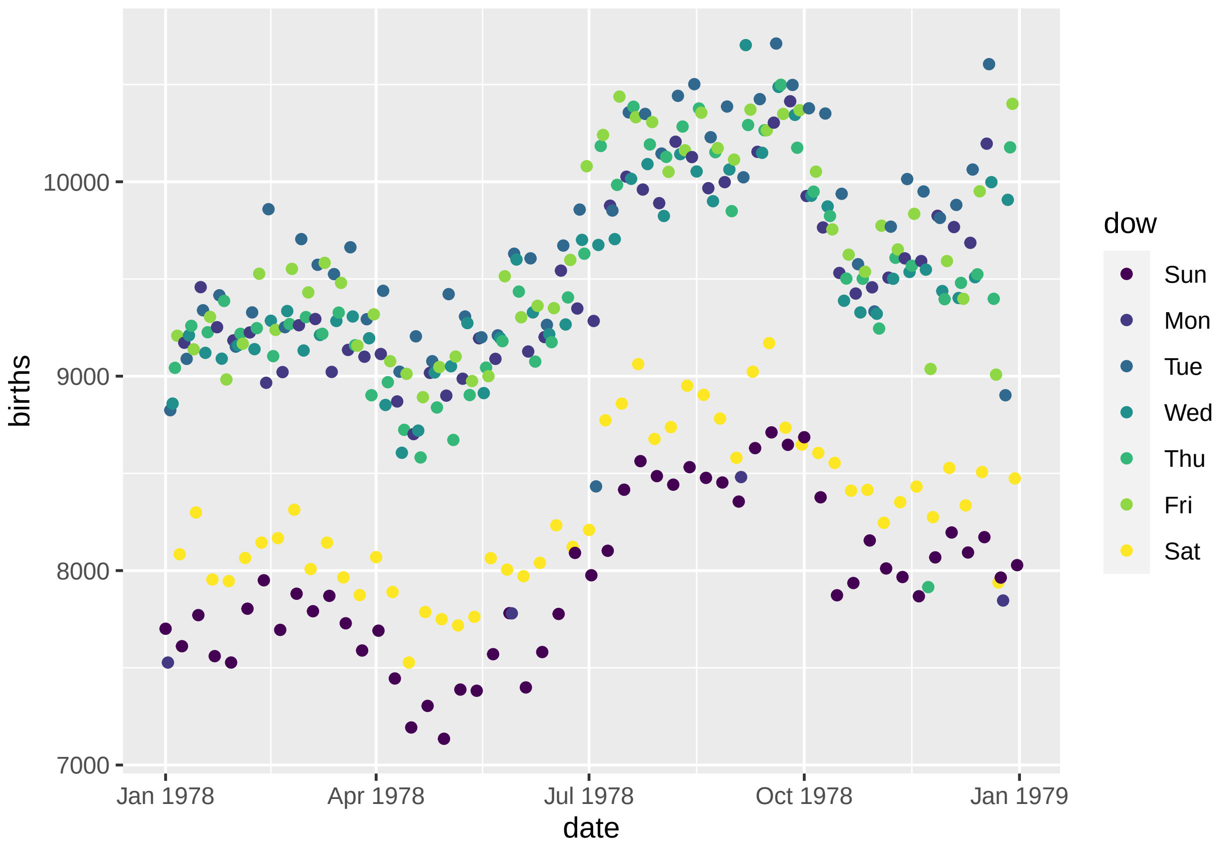
c. To test your assumption, we need to figure out the what day of the week each observation is. Use dplyr::mutate to add a new column named dow that is the day of the week (Monday, Tuesday, etc). This calculation will involve some function in the lubridate package and the date column.

```r
my78 <- my78 %>%
  mutate(
    dow = (wday(my78$date, label = TRUE))
  )
head(my78)
```

```
##         date births dow
## 1 1978-01-01   7701 Sun
## 2 1978-01-02   7527 Mon
## 3 1978-01-03   8825 Tue
## 4 1978-01-04   8859 Wed
## 5 1978-01-05   9043 Thu
## 6 1978-01-06   9208 Fri
```

d.Plot the data with the point color being determined by the day of the week variable.

```r
ggplot(data = my78, aes(x = date, y = births)) +
  geom_point(aes(color=dow))
```



```r
library(tidyverse)
library(stringr)
```

Question 1

For the following regular expression, explain in words what it matches on. Then add test strings to demonstrate that it in fact does match on the pattern you claim it does.

Make sure that your test set of strings has several examples that match as well as several that do not.

If you copy the Rmarkdown code for these exercises directly from my source pages, make sure to remove the eval=FALSE from the R-chunk headers.

a. This regular expression matches: Insert your answer here…

strings that contain the character 'a', but, crucially, not 'A', which is why 'Anne' returns a FALSE, while 'Michael' returns a TRUE

```r
strings <- c('Anne', 'Michael', 'Evan', 'Zeke')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'a') )
```

```
##    string result
## 1    Anne  FALSE
## 2 Michael   TRUE
## 3    Evan   TRUE
## 4    Zeke  FALSE
```

b. This regular expression matches: Insert your answer here… This expression detects if your string contains 'ab' anywhere in the string, but, crucially, this still does not include capitalization, such as with ‘Abby’.

```r
strings <- c('Abby', 'Gabby', 'jab', 'ACAB')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##   string result
## 1   Abby  FALSE
## 2  Gabby   TRUE
## 3    jab   TRUE
## 4   ACAB  FALSE
```

c) This regular expression matches: *Insert your answer here...* This expression searches for if the strings contain either ‘a’ or ‘b’ as a character, even if only one is present. This still does not include capitalization ‘AB’.

```r
        strings <- c('AB', 'ab', 'Jab', 'Abby', 'Gabby')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '[ab]') )
```

```
##   string result
## 1     AB  FALSE
## 2     ab   TRUE
## 3    Jab   TRUE
## 4   Abby   TRUE
## 5  Gabby   TRUE
```

d) This regular expression matches: *Insert your answer here...* This expression is looking for the characters ‘ab’ in a string, but only at the beginning of a string as indicated by the ^.

```r
        strings <- c('ab', 'Gabby', 'Abby', 'JaB', 'abbyyy')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^[ab]') )
```

```
##   string result
## 1     ab   TRUE
## 2  Gabby  FALSE
## 3   Abby  FALSE
## 4    JaB  FALSE
## 5 abbyyy   TRUE
```

e. This expression detects within the strings if there are any digits that are separated by white space from the characters ‘a’ or ‘A’

```r
strings <- c('13 aA', 'a 123', '123 a', 'r123A', 'r123 A')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##   string result
## 1  13 aA   TRUE
## 2  a 123  FALSE
## 3  123 a   TRUE
## 4  r123A  FALSE
## 5 r123 A   TRUE
```

f. This expression detects if there are any digits which are followed by zero or more white space containing the characters ‘a’ or ‘A’

```
        strings <- c('13 aA', 'a 123', '123 a', 'r123A', 'r123 A', 'a123 a', 'a123')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##   string result
## 1  13 aA   TRUE
## 2  a 123  FALSE
## 3  123 a   TRUE
## 4  r123A   TRUE
## 5 r123 A   TRUE
## 6 a123 a   TRUE
## 7   a123  FALSE
```

g.This expression detects if any character of any length exists. this includes ' '.

```
        strings <- c('', 'a', '123', 'eeee', 'WEWEWE')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '.*') )
```

```
##   string result
## 1          TRUE
## 2      a   TRUE
## 3    123   TRUE
## 4   eeee   TRUE
## 5 WEWEWE   TRUE
```

h. this expression detects if the beginning of a given string contains two alphanumeric characters prior to the characters 'bar'.

```
        strings <- c('1', '11bar', '11 bar', 'd11bar', '11bard')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##   string result
## 1      1  FALSE
## 2  11bar   TRUE
## 3 11 bar  FALSE
## 4 d11bar  FALSE
## 5 11bard   TRUE
```

i.this expression detects if a given string contains 'foo' followed by a '.' character, followed by 'bar'

```
        strings <- c('foo.bar', 'foobar', '11bar')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##    string result
## 1 foo.bar   TRUE
## 2  foobar  FALSE
## 3   11bar   TRUE
```

Question 2

The following file names were used in a camera trap study.

S number represents the site, P is the plot within a site, C is the camera number within the plot,

the first string of numbers is the

YearMonthDay

and the second string of numbers is the

HourMinuteSecond.

```
file.names <- c( 'S123.P2.C10_20120621_213422.jpg',
                 'S10.P1.C1_20120622_050148.jpg',
                 'S187.P2.C2_20120702_023501.jpg')
```

Produce a data frame with columns corresponding to the site, plot, camera, year, month, day, hour, minute, and second for these three file names. So we want to produce code that will create the data frame:

```
camframe <- c('site', 'plot', 'camera', 'year', 'month', 'day', 'hour', 'minute', 'second')

filenames2 <- str_replace_all(file.names, pattern = '_', replacement = '\\.')

filenames2
```

```
## [1] "S123.P2.C10.20120621.213422.jpg" "S10.P1.C1.20120622.050148.jpg"
## [3] "S187.P2.C2.20120702.023501.jpg"
```

```
filenames3 <- str_split_fixed(filenames2, pattern = '\\.', 6)


Year <- str_sub(filenames3[,4], start = 1, end = 4)
month <- str_sub(filenames3[,4], start = 5, end = 6)
day <- str_sub(filenames3[,4], start = 7, end = 8)

hour <- str_sub(filenames3[,5], start = 1, end = 2)
minute <- str_sub(filenames3[,5], start = 3, end = 4)
second <- str_sub(filenames3[,5], start = 5, end = 6)


ffile <- cbind(filenames3, Year, month, day, hour, minute, second)
ffile
```

```
##                                                  Year   month day  hour minute
## [1,] "S123" "P2" "C10" "20120621" "213422" "jpg" "2012" "06"  "21" "21" "34"
## [2,] "S10"  "P1" "C1"  "20120622" "050148" "jpg" "2012" "06"  "22" "05" "01"
## [3,] "S187" "P2" "C2"  "20120702" "023501" "jpg" "2012" "07"  "02" "02" "35"
##      second
## [1,] "22"
## [2,] "48"
## [3,] "01"
```

```
ffileDF <- data.frame(ffile)

colnames(ffileDF) <- c('site', 'plot', 'camera', 'ymd', 'hms', 'filetype', 'year', 'month', 'day', 'hou

ffileDF %>%
   select('site', 'plot', 'camera', 'year', 'month', 'day', 'hour', 'minute', 'second')
```

```
##   site plot camera year month day hour minute second
## 1 S123   P2    C10 2012    06  21   21     34     22
## 2  S10   P1     C1 2012    06  22   05     01     48
## 3 S187   P2     C2 2012    07  02   02     35     01
```

Question 3

The full text from Lincoln's Gettysburg Address is given below.

Calculate the mean word length Note: consider 'battle-field' as one word with 11 letters).

```
Gettysburg <- 'Four score and seven years ago our fathers brought forth on this continent, a new nation
```

```
Gettysburg2 <- str_replace_all(Gettysburg, pattern = ',', replacement = '')

Gettysburg2 <- str_replace_all(Gettysburg2, pattern = '-', replacement = '')

Gettysburg2 <- str_replace_all(Gettysburg2, pattern = '\\.', replacement = '')

get_split <- scan(text = Gettysburg2, what = ' ')

totalchar <- 0
words <- 0
for (i in 1:length(get_split)){
  word_len <- str_length(get_split[i])
  totalchar <- totalchar + word_len
  words <- words + 1
}

meanchar <- totalchar/words
meanchar
```

```
## [1] 4.239852
```

```
totalchar
```

```
## [1] 1149
```

```
get_split
```

```
##   [1] "Four"        "score"       "and"         "seven"       "years"
##   [6] "ago"         "our"         "fathers"     "brought"     "forth"
##  [11] "on"          "this"        "continent"   "a"           "new"
##  [16] "nation"      "conceived"   "in"          "Liberty"     "and"
##  [21] "dedicated"   "to"          "the"         "proposition" "that"
##  [26] "all"         "men"         "are"         "created"     "equal"
##  [31] "Now"         "we"          "are"         "engaged"     "in"
##  [36] "a"           "great"       "civil"       "war"         "testing"
##  [41] "whether"     "that"        "nation"      "or"          "any"
##  [46] "nation"      "so"          "conceived"   "and"         "so"
##  [51] "dedicated"   "can"         "long"        "endure"      "We"
##  [56] "are"         "met"         "on"          "a"           "great"
##  [61] "battlefield" "of"          "that"        "war"         "We"
##  [66] "have"        "come"        "to"          "dedicate"    "a"
##  [71] "portion"     "of"          "that"        "field"       "as"
##  [76] "a"           "final"       "resting"     "place"       "for"
##  [81] "those"       "who"         "here"        "gave"        "their"
##  [86] "lives"       "that"        "that"        "nation"      "might"
##  [91] "live"        "It"          "is"          "altogether"  "fitting"
##  [96] "and"         "proper"      "that"        "we"          "should"
## [101] "do"          "this"        "But"         "in"          "a"
## [106] "larger"      "sense"       "we"          "can"         "not"
## [111] "dedicate"    "we"          "can"         "not"         "consecrate"
## [116] "we"           "can"          "not"          "hallow"       "this"
## [121] "ground"       "The"          "brave"        "men"          "living"
## [126] "and"          "dead"         "who"          "struggled"    "here"
## [131] "have"         "consecrated"  "it"           "far"          "above"
## [136] "our"          "poor"         "power"        "to"           "add"
## [141] "or"           "detract"      "The"          "world"        "will"
## [146] "little"       "note"         "nor"          "long"         "remember"
## [151] "what"         "we"           "say"          "here"         "but"
## [156] "it"           "can"          "never"        "forget"       "what"
## [161] "they"         "did"          "here"         "It"           "is"
## [166] "for"          "us"           "the"          "living"       "rather"
## [171] "to"           "be"           "dedicated"    "here"         "to"
## [176] "the"          "unfinished"   "work"         "which"        "they"
## [181] "who"          "fought"       "here"         "have"         "thus"
## [186] "far"          "so"           "nobly"        "advanced"     "It"
## [191] "is"           "rather"       "for"          "us"           "to"
## [196] "be"           "here"         "dedicated"    "to"           "the"
## [201] "great"        "task"         "remaining"    "before"       "us"
## [206] "that"         "from"         "these"        "honored"      "dead"
## [211] "we"           "take"         "increased"    "devotion"     "to"
## [216] "that"         "cause"        "for"          "which"        "they"
## [221] "gave"         "the"          "last"         "full"         "measure"
## [226] "of"           "devotion"     "that"         "we"           "here"
## [231] "highly"       "resolve"      "that"         "these"        "dead"
## [236] "shall"        "not"          "have"         "died"         "in"
## [241] "vain"         "that"         "this"         "nation"       "under"
## [246] "God"          "shall"        "have"         "a"            "new"
## [251] "birth"        "of"           "freedom"      "and"          "that"
## [256] "government"   "of"           "the"          "people"       "by"
## [261] "the"          "people"       "for"          "the"          "people"
## [266] "shall"        "not"          "perish"       "from"         "the"
## [271] "earth"
```

```
#Gettysburg2
```