# Assignment_3

Evan Z-G

2023-10-17

## Chapter 11

### 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: All strings that contain the character 'a'.

```
strings <- c('Evan', 'River', 'Angelica', 'Conner')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'a') )
```

```
##     string result
## 1     Evan   TRUE
## 2    River  FALSE
## 3 Angelica   TRUE
## 4   Conner  FALSE
```

#### b. This regular expression matches: Strings that contain the character sequence 'ab'

```
strings <- c('Lab', 'able', 'fan', 'bob')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##   string result
## 1    Lab   TRUE
## 2   able   TRUE
## 3    fan  FALSE
## 4    bob  FALSE
```

#### c. This regular expression matches: Strings that contain the either the character 'a', and/or the character 'b'

```
strings <- c('Lab', 'able', 'fan', 'bob')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##   string result
## 1    Lab   TRUE
## 2   able   TRUE
## 3    fan   TRUE
## 4    bob   TRUE
```

### d. This regular expression matches: Strings that have the character ‘a’ or the character ‘b’ at the beginning of the string.

```
strings <- c('Lab', 'able', 'fan', 'bob')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##   string result
## 1    Lab  FALSE
## 2   able   TRUE
## 3    fan  FALSE
## 4    bob   TRUE
```

### e. This regular expression matches: A string composed of any number of digits, followed by one white space, followed by either ‘a’ or ‘A’.

```
strings <- c('3 a', '44  a', '45 A', 'a 5')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##   string result
## 1    3 a   TRUE
## 2  44  a  FALSE
## 3   45 A   TRUE
## 4    a 5  FALSE
```

### f. This regular expression matches: A string composed of any number of digits, followed by an optional number of white spaces, followed by the character ‘a’ or ‘A’.

```
strings <- c('3 a', '44  a', '45A', 'a 5')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##   string result
## 1    3 a   TRUE
## 2  44  a   TRUE
## 3    45A   TRUE
## 4    a 5  FALSE
```

### g. This regular expression matches: A string containing any number of instances of the character ‘.’, including 0 instances

```
strings <- c('Evan', 'E.van', 'Ev..an')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '.*') )
```

```
##   string result
## 1   Evan   TRUE
## 2  E.van   TRUE
## 3 Ev..an   TRUE
```

**h. This regular expression matches: A string beginning with any two alphanumeric characters, followed by the sequence ‘bar’**

```
strings <- c('abar', 'aabar', '15bar', 'ajrhvfbvfjhbbar', 'askjhwvbak')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##            string result
## 1            abar  FALSE
## 2           aabar   TRUE
## 3           15bar   TRUE
## 4 ajrhvfbvfjhbbar  FALSE
## 5      askjhwvbak  FALSE
```

**i. This regular expression matches: A string containing either the sequence ‘foo.bar’ or two alphanumeric characters followed by the sequence ‘bar’**

```
strings <- c('foo.bar', 'foobar', '11bar', 'afoo.bar')
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
## 4 afoo.bar   TRUE
```

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

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

```
## [1] "S123.P2.C10_20120621_213422.jpg" "S10.P1.C1_20120622_050148.jpg"
## [3] "S187.P2.C2_20120702_023501.jpg"
```

```r
string_frame <- c()
for (x in 1:length(file.names)) {
  row1 <- c(file.names[x])

  row1 <- str_replace_all(row1, pattern='_', replacement='\\.')



  my_row <- str_split_fixed(row1, '\\.', 6)

  string_frame = rbind(string_frame,my_row)
}


print(string_frame)
```

```
##      [,1]   [,2] [,3]  [,4]       [,5]     [,6] 
## [1,] "S123" "P2" "C10" "20120621" "213422" "jpg"
## [2,] "S10"  "P1" "C1"  "20120622" "050148" "jpg"
## [3,] "S187" "P2" "C2"  "20120702" "023501" "jpg"
```

```r
year <- str_sub(string_frame[,4], start=1, end=4)
month <- str_sub(string_frame[,4], start=5, end=6)
day <- str_sub(string_frame[,4], start=7, end=8)
hour <- str_sub(string_frame[,5], start=1, end=2)
minute <- str_sub(string_frame[,5], start=3, end=4)
second <- str_sub(string_frame[,5], start=5, end=6)

finalfile <- cbind(string_frame, year, month, day, hour, minute, second)


reallyfinal <- data.frame(finalfile)

colnames(reallyfinal) <- c("site", "plot", "time1", "time2", "filetype", "camera", "year", "month", "da

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

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

## 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).

```r
Gettysburg <- 'Four score and seven years ago our fathers brought forth on this
continent, a new nation, conceived in Liberty, and dedicated to the proposition
that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any
```

```
nation so conceived and so dedicated, can long endure. We are met on a great
battle-field of that war. We have come to dedicate a portion of that field, as
a final resting place for those who here gave their lives that that nation might
live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate -- we can not consecrate -- we can
not hallow -- this ground. The brave men, living and dead, who struggled here,
have consecrated it, far above our poor power to add or detract. The world will
little note, nor long remember what we say here, but it can never forget what
they did here. It is for us the living, rather, to be dedicated here to the
unfinished work which they who fought here have thus far so nobly advanced. It
is rather for us to be here dedicated to the great task remaining before us --
that from these honored dead we take increased devotion to that cause for which
they gave the last full measure of devotion -- that we here highly resolve that
these dead shall not have died in vain -- that this nation, under God, shall
have a new birth of freedom -- and that government of the people, by the people,
for the people, shall not perish from the earth.'


Gettysburg1 <- str_replace_all(Gettysburg, "\\,", " ")
Gettysburg2 <- str_replace_all(Gettysburg1, "\\--", " ")
Gettysburg2 <- str_replace_all(Gettysburg1, "\\-", "")
Gettysburg3 <- str_replace_all(Gettysburg2, "\\.", " ")

# word_list <- str_split(Gettysburg, pattern = " ")
word_list <- scan(text = Gettysburg3, what = " ")
print(word_list)
```

```
##   [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"
```

```r
sum <- 0
tally <- 0
length(word_list)
```

```
## [1] 271
```

```r
word_list[1]
```

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

```r
for (i in 1:length(word_list)){

  word_len <- str_length(word_list[i])

  sum <- sum + word_len

  tally <- tally + 1

}

sum
```

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

```r
tally
```

```
## [1] 271
```

```
mean_word_len <- sum/tally
mean_word_len
```

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

# Chapter 12

## 1. Convert the following to date or date/time objects.

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

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

```
mdy('Sept 13, 2010') # The month needs to be in the correct abbrevation
```

```
## Warning: All formats failed to parse. No formats found.
```

```
## [1] NA
```

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

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

```
mdy('S 13, 2010') # The month needs to be in the correct abbreviation
```

```
## Warning: All formats failed to parse. No formats found.
```

```
## [1] NA
```

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

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

```
dmy('1-5-1998') # There is no way to know if this is day-month or month-day
```

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

```
dmy('21-5-1998') # It is obvious here because there is no 21st month
```

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

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

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

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

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

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

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

## 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.:

```
# Calculate the date of your 64th birthday.
birthdate <- ymd("1982 April 24")
```

```
bd64 <- birthdate + years(64)
bd64
```

```
## [1] "2046-04-24"
```

```
# Calculate your current age (in years). Hint: Check your age is calculated correctly if your birthday
now <- lubridate::today()
agenow <- interval(birthdate, now)
readable <- as.period(agenow)
year(readable)
```

```
## [1] 41
```

```
# Using your result in part (b), calculate the date of your next birthday.
til_next <- years(42) - readable
nextbirth <- now + til_next
nextbirth
```

```
## [1] "2024-04-24"
```

```
# The number of days until your next birthday.
daystil <- interval(now, nextbirth)
daystil2 <- as.period(daystil, unit="days")
day(daystil2)
```

```
## [1] 183
```

```
# The number of months and days until your next birthday.
daystil <- interval(now, nextbirth)
daystil2 <- as.period(daystil)
month(daystil2)
```

```
## [1] 6
```

```
day(daystil2)
```

```
## [1] 0
```

### 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?

```
appointment <- mdy_hm('5-8-2015 3:00 PM', tz='US/Arizona') %>% with_tz(tzone = 'Pacific/Auckland')
appointment
```

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

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

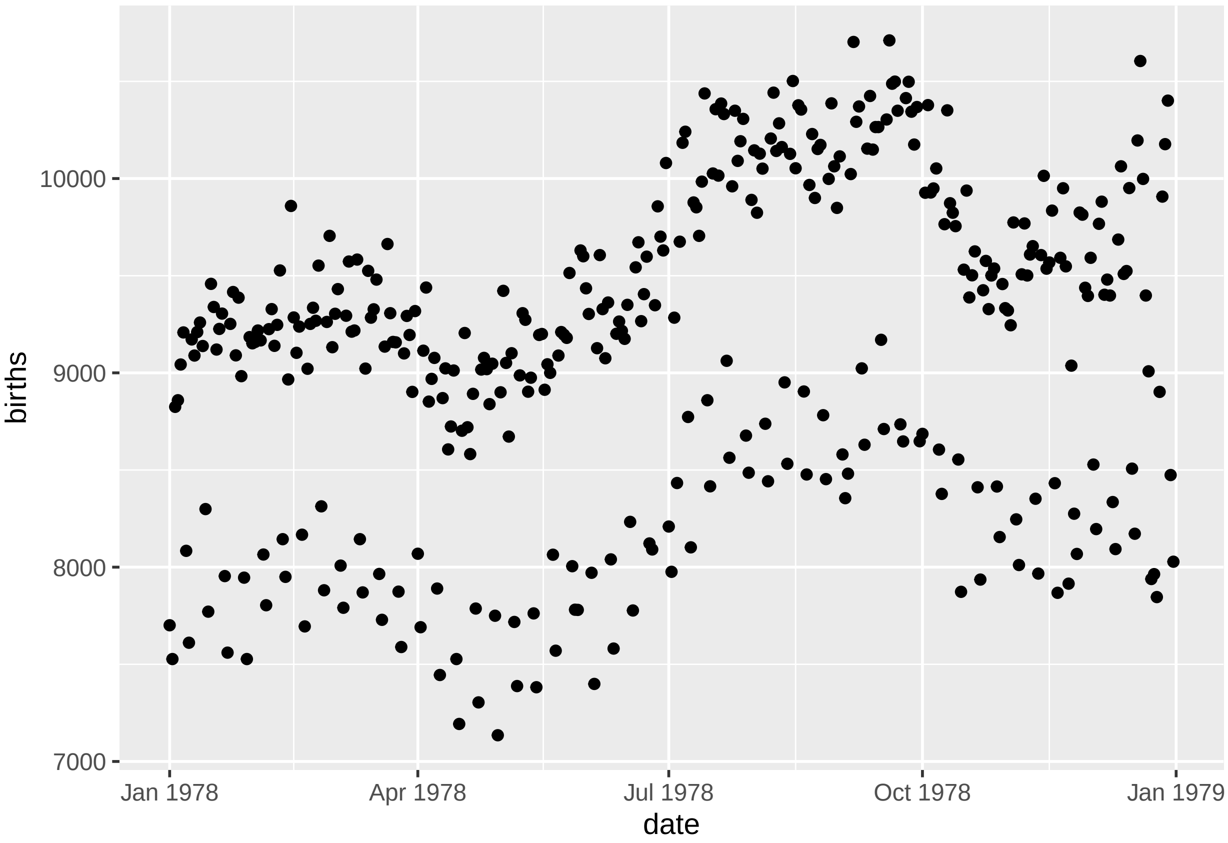
Using the mosaicData package, load the data sed Graph the number of births vs the date with date on the x-axis. What stands out to you? Why do you think we have this trend? To test your assumption, we need to figure out the what day of the week each observation is. Use dplyr::mutate to add a new column 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. Plot the data with the point color being determined by the day of the week variable.

```
# a. Using the mosaicData package, load the data set Births78 which records the number of children born

library(mosaicData)
frame1 <- Births78[c(1,2)]
head(frame1)
```

```
##         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
ggplot(data=frame1, aes(x=date,y=births)) +
  geom_point()
```

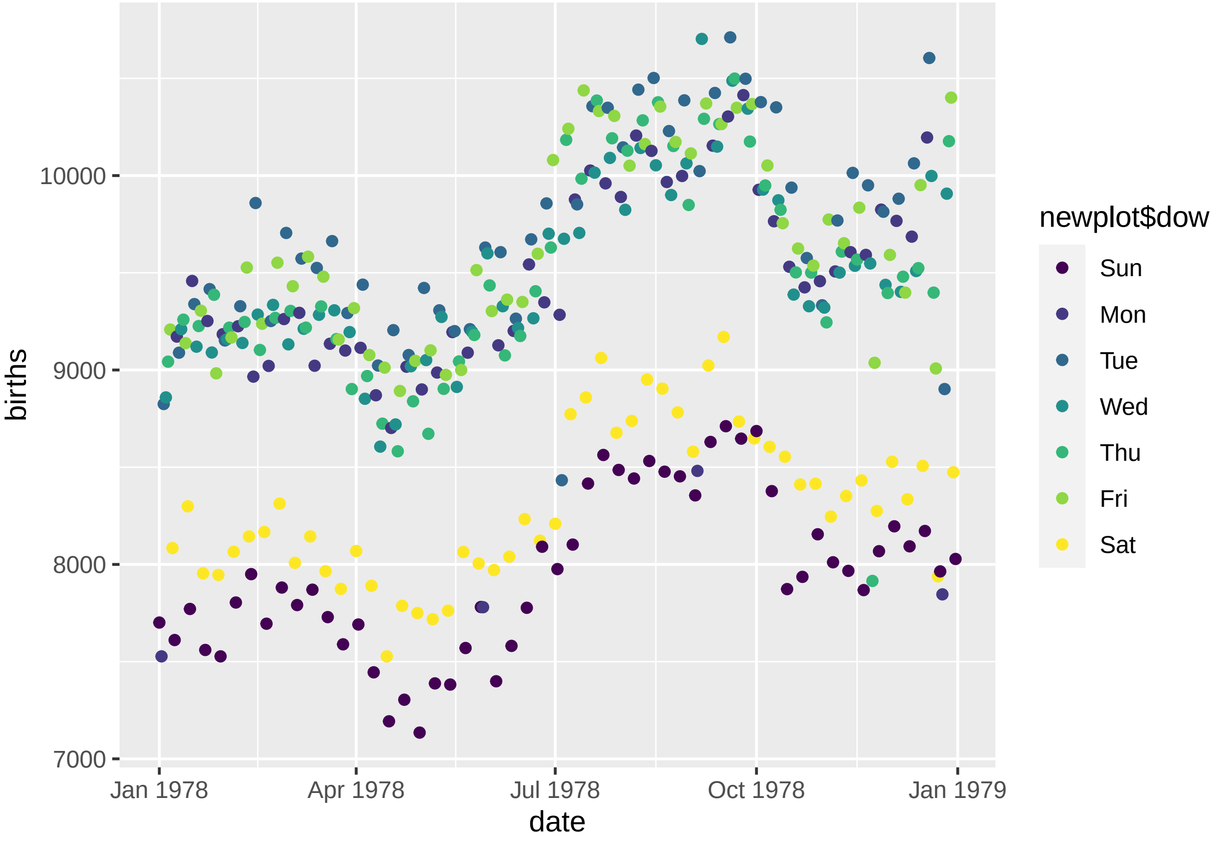


```
# c. To test your assumption, we need to figure out the what day of the week each observation is. Use d
newplot <- frame1 %>% mutate(dow = wday(frame1$date, label=TRUE))
head(newplot)
```

```
##         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.
ggplot(data=newplot, aes(x=date, y=births)) +
  geom_point(aes(color=newplot$dow))
```

```
## Warning: Use of `newplot$dow` is discouraged.
## i Use `dow` instead.
```


births
Jan 1978
Apr 1978
Jul 1978
Oct 1978
Jan 1979
date
newplot$dow
● Sun
● Mon
● Tue
● Wed
● Thu
● Fri
● Sat