STA 445 Assignment 3

Kaylin McLiverty

2023-10-25

Chapter 11

Exercise 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.

a) This regular expression matches: *the expression is detecting any 'a' characters within the strings of the vector.*

```
strings <- c('testing','chasing','that','feeling')
data.frame( string = strings ) %>%
mutate( result = str_detect(string, 'a') )
```

```
##    string result
## 1 testing  FALSE
## 2 chasing   TRUE
## 3    that   TRUE
## 4 feeling  FALSE
```

b) This regular expression matches: *the expression is detecting any 'ab' characters (in that particular order) within the strings of the vector.*

```
strings <- c('absolute', 'ballet', 'scales', 'atob')
data.frame( string = strings ) %>%
mutate( result = str_detect(string, 'ab') )
```

```
##     string result
## 1 absolute   TRUE
## 2   ballet  FALSE
## 3   scales  FALSE
## 4     atob  FALSE
```

c) This regular expression matches: *this expression is detecting any 'a' or 'b' character within the strings of the vector. The order does not matter, as long as there is at least one 'a' or 'b'.*

```
strings <- c('absolute', 'ballet', 'scales', 'blink', 'learn', 'txt')
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '[ab]') )
```

```
##     string result
## 1 absolute   TRUE
## 2   ballet   TRUE
## 3   scales   TRUE
## 4    blink   TRUE
## 5    learn   TRUE
## 6      txt  FALSE
```

d) This regular expression matches: *the expression detects if there is an 'a' or 'b' character at the start of the string within the vector of strings.*

```r
strings <- c('absolute', 'ballet', 'scales', 'blink', 'learn', 'txt', 'rib', 1)
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##     string result
## 1 absolute   TRUE
## 2   ballet   TRUE
## 3   scales  FALSE
## 4    blink   TRUE
## 5    learn  FALSE
## 6      txt  FALSE
## 7      rib  FALSE
## 8        1  FALSE
```

e) This regular expression matches: *This expression detects if a string begins with any number, followed by a blank space, and then followed by an 'a' or 'A' character. If the previous conditions are not satisfied, then a FALSE result will show.*

```r
strings <- c('1 aAaAAAAa', '2 A quick fox', '1 backwards', '4a' )
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##          string result
## 1    1 aAaAAAAa   TRUE
## 2 2 A quick fox   TRUE
## 3   1 backwards  FALSE
## 4            4a  FALSE
```

f) This regular expression matches: *This expression detects if a string begins with any number, followed by a blank space or not, and then followed by an 'a' or 'A' character. This is similar to part e, but now, the expression will accept (a number, space, 'a' or 'A' combination) OR (a number, 'a' or 'A' combination).*

```r
strings <- c('1 aAaAAAAa', '2 A quick fox', '1 backwards', '4a', 'A', '9A')
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##          string result
## 1    1 aAaAAAAa   TRUE
## 2 2 A quick fox   TRUE
## 3   1 backwards  FALSE
## 4            4a   TRUE
## 5             A  FALSE
## 6            9A   TRUE
```

g) This regular expression matches: *This expression detects if there is any character with zero or more repetitions.*

```r
strings <- c('i     have an     idea     ', '1', '     1 2     3', ' y', 'y   ','g','')
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '.*') )
```

```
##                         string result
## 1 i     have an     idea        TRUE
## 2                           1    TRUE
## 3                   1 2     3    TRUE
## 4                           y    TRUE
## 5                        y       TRUE
## 6                           g    TRUE
## 7                                TRUE
```

h) This regular expression matches: *This expression detects strings that contain any of the two same alphanumeric characters immediately followed by 'bar'.*

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

```
##    string result
## 1   ttbar   TRUE
## 2   --bar  FALSE
## 3 nnmmbar  FALSE
```

i) This regular expression matches: *This expression detects strings that contain any of the two same alphanumeric characters immediately followed by 'bar' OR the expression will take the any number of characters as long if it ends with 'foo.bar'.*

```
strings <- c('nnbar', 'foooo.bar', '*-foo.bar', 'foo.bar', 'nn.bar',
             'abdcfoo.bar', '1234foo.bar','--.bar')
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##        string result
## 1       nnbar   TRUE
## 2   foooo.bar  FALSE
## 3   *-foo.bar   TRUE
## 4     foo.bar   TRUE
## 5      nn.bar  FALSE
## 6 abdcfoo.bar   TRUE
## 7 1234foo.bar   TRUE
## 8      --.bar  FALSE
```

Exercise 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.

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

file.names1 <- str_replace_all(file.names, pattern='_', replacement='\\.')

file.names1
```

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

```
file.names2 <- str_split_fixed(file.names1, pattern='\\.', n=6)
file.names2
```

```
##      [,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"
```

```
Year = str_sub(file.names2[ ,4], start=1, end=4)
Year
```

```
## [1] "2012" "2012" "2012"
```

```
Month = str_sub(file.names2[ ,4], start=5, end=6)
Month
```

```
## [1] "06" "06" "07"
```

```
Day = str_sub(file.names2[ ,4], start=7, end=8)
Day
```

```
## [1] "21" "22" "02"
```

```
Hour = str_sub(file.names2[ ,5], start=1, end=2)
Hour
```

```
## [1] "21" "05" "02"
```

```
Minute = str_sub(file.names2[ ,5], start=3, end=4)
Minute
```

```
## [1] "34" "01" "35"
```

```
Second = str_sub(file.names2[ ,5], start=5, end=6)
Second
```

```
## [1] "22" "48" "01"
```

```
file.names.final = data.frame(
  Site = file.names2[,1],
  Plot = file.names2[,2],
  Camera = file.names2[,3],
  Year = Year,
  Month = Month,
  Day = Day,
  Hour = Hour,
  Minute = Minute,
  Second = Second
)
file.names.final
```

```
##   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
```

Exercise 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.'

Gettysburg <- str_replace_all(Gettysburg, pattern='(\\W+\\s)',
                         replacement= ' ')
Gettysburg <- str_remove_all(Gettysburg, pattern='(\\.)')
Gettysburg <- str_remove_all(Gettysburg, pattern='(\\-)')

Gettysburg <- str_split(Gettysburg, pattern=' ')
Gettysburg
```

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

```
sum(nchar(Gettysburg[[1]]))/271
```

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

Chapter 12

Exercise 1

Convert the following to date or date/time objects.

```
a) September 13, 2010.
b) Sept 13, 2010.
c) Sep 13, 2010.
d) S 13, 2010. Comment on the month abbreviation needs.
e) 07-Dec-1941.
f) 1-5-1998. Comment on why you might be wrong.
g) 21-5-1998. Comment on why you know you are correct.
h) 2020-May-5 10:30 am
i) 2020-May-5 10:30 am PDT (ex Seattle)
j) 2020-May-5 10:30 am AST (ex Puerto Rico)
```

```r
# Exercise 1 part a
a <- mdy('September 13, 2010')
a
```

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

```r
# Exercise 1 part b
b <- mdy('Sept 13, 2010')
```

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

```r
b
```

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

Part b fails because 'Sept' is not one of the known abbreviations within lubridate package.

```r
#Exercise 1 part c
c <- mdy('Sep 13, 2010')
c
```

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

```r
#Exercise 1 part d
d <- mdy('S 13, 2010')
```

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

```r
d
```

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

Part d fails because 's' is not one of the known abbreviations within lubridate package.

```r
#Exercise 1 part e
e <- dmy('07-Dec-1941')
e
```

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

```r
#Exercise 1 part f
f <- dmy('1-5-1998')
f
```

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

The date given in part f is ambiguous and thus, we don't know if the '1' or '5' is the day of the month or the month itself. I chose '1' to represent the day and chose '5' to represent the month.

```r
#Exercise 1 part g
g <- dmy('21-5-1998')
g
```

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

I know that I am correct in part g because the number '21' is not ambiguous and represents the day of the month, leaving the '5' to represent the month.

```r
#Exercise 1 part h
h <- ymd_hm('2020-May-5 10:30 am')
h
```

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

```r
#Exercise 1 part i
i <- ymd_hm('2020-May-5 10:30 am PDT', tz='America/Los_Angeles')
i
```

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

#Exercise 1 part f

```r
#Exercise 1 part j
j <- ymd_hm('2020-May-5 10:30 am AST', tz='America/Puerto_Rico')
j
```

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

Exercise 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.
b) Calculate your current age (in years). Hint: Check your age is calculated correctly if
your birthday was yesterday and if it were tomorrow!
c) Using your result in part (b), calculate the date of your next birthday.
d) The number of _days_ until your next birthday.
e) The number of _months_ and _days_ until your next birthday.
```

```r
#Exercise 2 part a
dob <- mdy('November 2, 2001')
dob
```

```
## [1] "2001-11-02"
```

```r
#The date of my 64th birthday
dob + years(64)
```

```
## [1] "2065-11-02"
```

```r
#Exercise 2 part b
#My current age
today <- mdy('October 25, 2023')
n <- as.period(dob%--%today)
year(n)
```

```
## [1] 21
```

```r
#Exercise 2 part c
#Date of my next birthday
myBday <- update(dob, year=2023)
myBday
```

```
## [1] "2023-11-02"
```

```r
#Exercise 2 part d
#Days until my birthday
as.period(today%--%myBday, unit='days')
```

```
## [1] "8d 0H 0M 0S"
```

```r
#Exercise 2 part e
#Months and days until my birthday
as.period(today%--%myBday, unit='months')
```

```
## [1] "8d 0H 0M 0S"
```

Exercise 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?

```
#Phone call time in Arizona
aztime <- mdy_h('May 8, 2015, 3pm', tz='US/Arizona')
aztime
```

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

```
#Phone call time in Auckland, NZ
with_tz(aztime, tzone='Pacific/Auckland')
```

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

Exercise 5

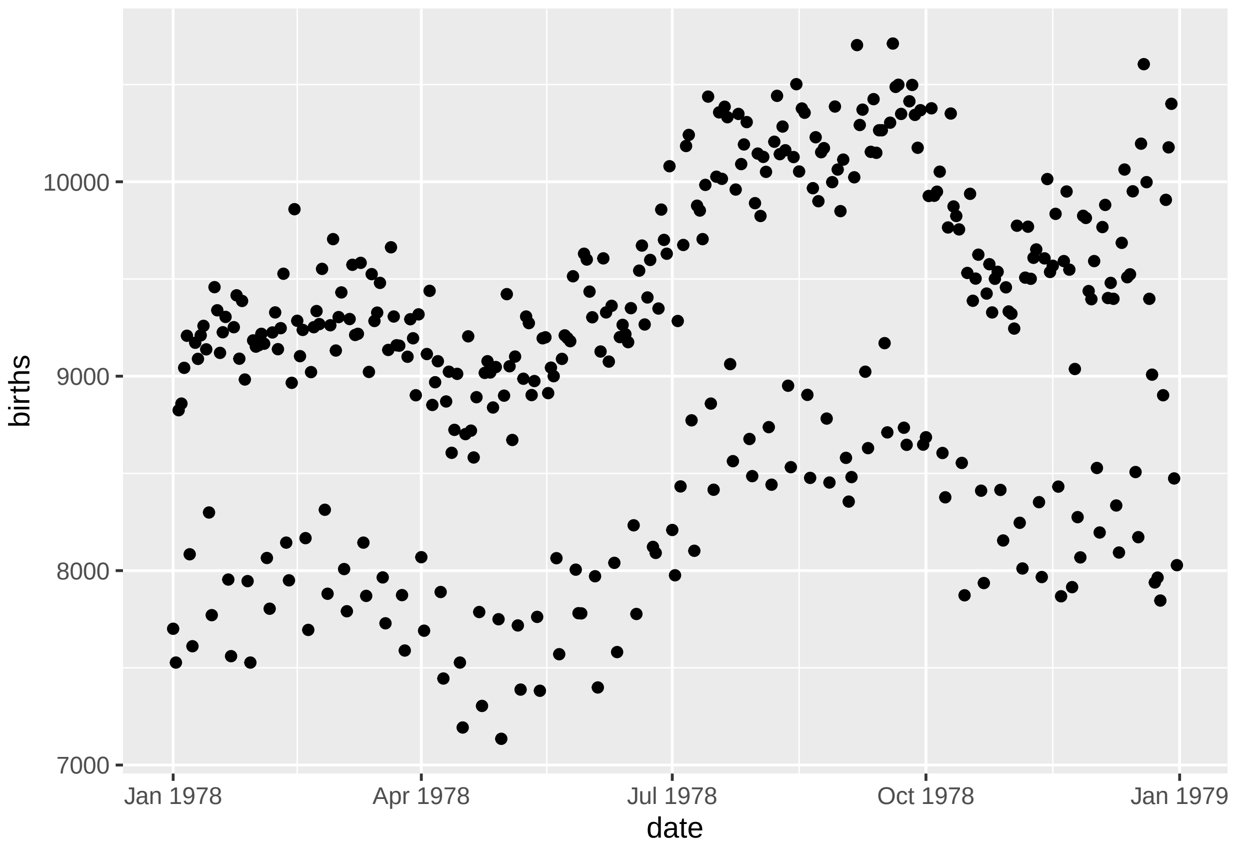
It turns out there is some interesting periodicity regarding the number of births on particular days of the year.

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

```
#Exercise 5 part a
#installed the mosaicData package and loaded the Births78 data set
data('Births78', package='mosaicData')
Births <- data.frame(Births78[,c(1,2)])
head(Births)
```

```
##         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
```

```
#Exercise 5 part b
ggplot(data=Births, aes(x=date, y=births))+
  geom_point()
```

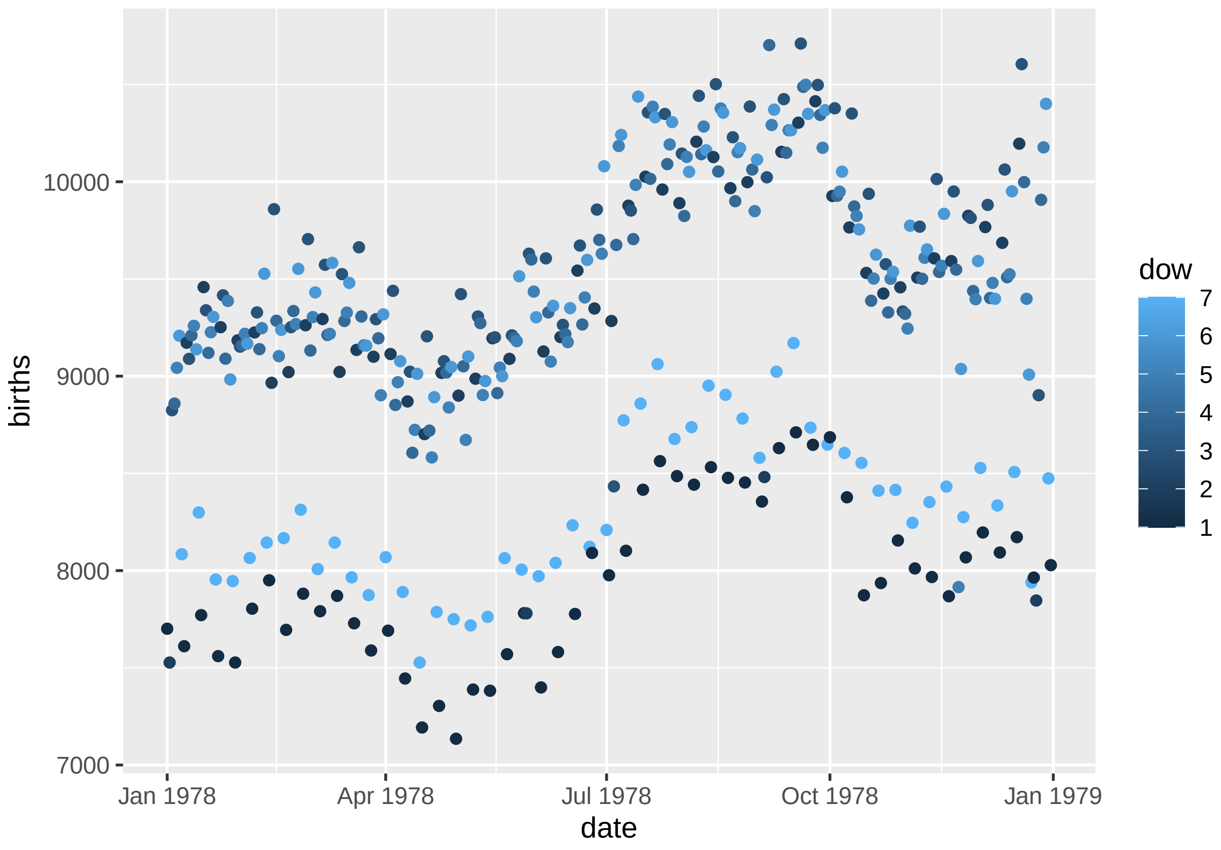


The zig zag shape of the graph definitely stands out to me. I think we might have this trend because the number of births fluctuate depending on the day of the month.

```
#Exercise 5 part c
#Finding out what day of the week the births occured.
Births <- dplyr::mutate(Births, dow=wday(date))
head(Births)
```

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

```
#Exercise 5 part d
ggplot(data=Births, aes(x=date, y=births))+
  geom_point(aes(color=dow))
```



It appears that a lower number of births occurred on Sundays and Saturdays in 1978!