STA_445_HW3

Bianca L.

2023-10-17

```
library(tidyverse)
library(stringr)
library(lubridate)
library(mosaicData)
```

Problem 1: Chapter 11 Problem 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: any string with at least one lower case a.

```
      strings <- c("poop", "ahole", "a", "Angelique Smells like doo doo",
                   "Calliope is a dork!", "Aaah Real Monsters")
      data.frame( string = strings ) %>%
      mutate( result = str_detect(string, 'a') )
```

```
##                          string result
## 1                          poop  FALSE
## 2                         ahole   TRUE
## 3                             a   TRUE
## 4 Angelique Smells like doo doo  FALSE
## 5           Calliope is a dork!   TRUE
## 6            Aaah Real Monsters   TRUE
```

b. This regular expression matches: The expression must have exactly "ab" in it in that order.

```
      strings <- c("angelique stinks", "bianca rules", "abra abra cadbra I wanna reach out and grab ya
      data.frame( string = strings ) %>%
        mutate( result = str_detect(string, 'ab') )
```

```
##                                       string result
## 1                             angelique stinks  FALSE
## 2                                bianca rules  FALSE
## 3 abra abra cadbra I wanna reach out and grab ya   TRUE
## 4                                      banana  FALSE
## 5                                   jkjkabjkjk   TRUE
```

c. This regular expression matches: If there is either an "a" or "b" in the expression.

```
      strings <- c("anglique is a loser", "bianca rules, nathaniel drools",  "Nacho Flay died this mo
      data.frame( string = strings ) %>%
        mutate( result = str_detect(string, '[ab]') )
```

d. This regular expression matches: The string must BEGIN with either an "a" or a "b".

```
strings <- c("ABBA", "acha", "abba", "biabiabiabia!", "pysduck")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

e. This regular expression matches: Any number of digits followed by a white space followed by either "a" or "A".

```
strings <- c("365 days a year", "785 apples", "6apples", "6 Apples", "apples")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

This regular expression matches: Any number of digits followed by zero or more white spaces, followed by either "a" or "A".

```
strings <- c("16 Candles", "16 apples", "16apples", "16 Apples")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

g. This regular expression matches: This matches with zero or more characters. Even white space and nothing at all is a character. This one is impossible to get a FALSE.

```
strings <- c("", "grumble", "997 tofu", " ")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '.*') )
```

h. This regular expression matches: Matches anything that begins with two digits or two letters followed by the word bar.

```
strings <- c("12bar", "  bar", "123bar", "aabar", "**bar")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '^\\w{2}bar') )
```

i. This regular expression matches: Any string that begins with two digits or two letters followed by the word bar OR contains the string "foo" followed by a period, followed by "bar".

```
strings <- c("aabar", "12bar", "foo.bar", "footbar", "foo bar")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

Problem 2: Chapter 11 Problem 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.

```
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:

```
data.frame(str_split_fixed(file.names, pattern='[_, .]', n=6)) %>%
  mutate(Site = X1,
         Plot = X2,
         Camera = X3,
         Year = str_sub(X4, start=1, end=4),
         Month = str_sub(X4, start = 5, end = 6),
```

```
         Day = str_sub(X4, start = 7, end = 8),
         Hour = str_sub(X5, start = 1, end=2),
         Minute = str_sub(X5, start =3, end=4),
         Second = str_sub(X5, start=5, end=6)) %>%
  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
```

Poblem 3: Chapter 11 Problem 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*).

OKAY! Let's do this!

```
Gettysburg1 <- str_replace_all(Gettysburg[[1]], pattern="[-\\.,\n]", replacement='')

Gettysburg2 <- str_replace_all(Gettysburg1[[1]], pattern="  ", replacement = " ")

getty.list <- str_split(Gettysburg2[[1]], pattern=" ")

mean(str_length(getty.list[[1]]))
```

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

```
getty.list
```

```
## [[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"
```

Problem 4: Chapter 12 Problem 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.

```
str_replace("Sept 13, 2010", "t", "") %>%
mdy()
```

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

mdy is okay with the abbreviation Sep but not Sept

c) Sep 13, 2010.

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

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

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

```
str_replace("S 13, 2010", "S", "Sep") %>%
mdy()
```

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

S is not an acceptable abbreviation of September. It needs to be replaced with Sep.

e) 07-Dec-1941.

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

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

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

```
mdy("1-5-1998")
```

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

Since both 1 and 5 are acceptable values of for the month, it is impossible to tell if the first number is the month or the day. Same with the second number.

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

```
dmy("21-5-1998")
```

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

There is no 21st month. Hence the first number must be the day and the second value must be the month.

h) 2020-May-5 10:30 am

```
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)

```
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)

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

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

Problem 5: Chapter 12 Problem 2

Using just your date of birth (ex Sep 7, 1998) and today's date calculate the following:

a) Calculate the date of your 64th birthday.

```
mdy("Sep 25, 1983") + dyears(64)
```

```
## [1] "2047-09-25"
```

b) Calculate your current age (in years).

```
(age <- year(as.period(mdy_hm("Sep 25, 1983 6:05 pm", tz="US/Arizona") %--% today(), )))
```

```
## [1] 40
```

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

```
(next.b.day <- mdy("Sep 25, 1983") +dyears(age+1))
```

```
## [1] "2024-09-24 06:00:00 UTC"
```

d) The number of *days* until your next birthday.

```
as.period(today()%--%next.b.day  , unit="days")
```

```
## [1] "334d 6H 0M 0S"
```

e) The number of *months* and *days* until your next birthday. as.period(today()%–%next.b.day)

```
as.period(today()%--%next.b.day)
```

```
## [1] "10m 29d 6H 0M 0S"
```

Problem 6: Chapter 12 Problem 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?

```
mdy_hm("May 8, 2015 3:00 pm", tz="US/Arizona") %>%
  with_tz(tzone="NZ")
```

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

Problem 7: Chapter 12 Problem 5

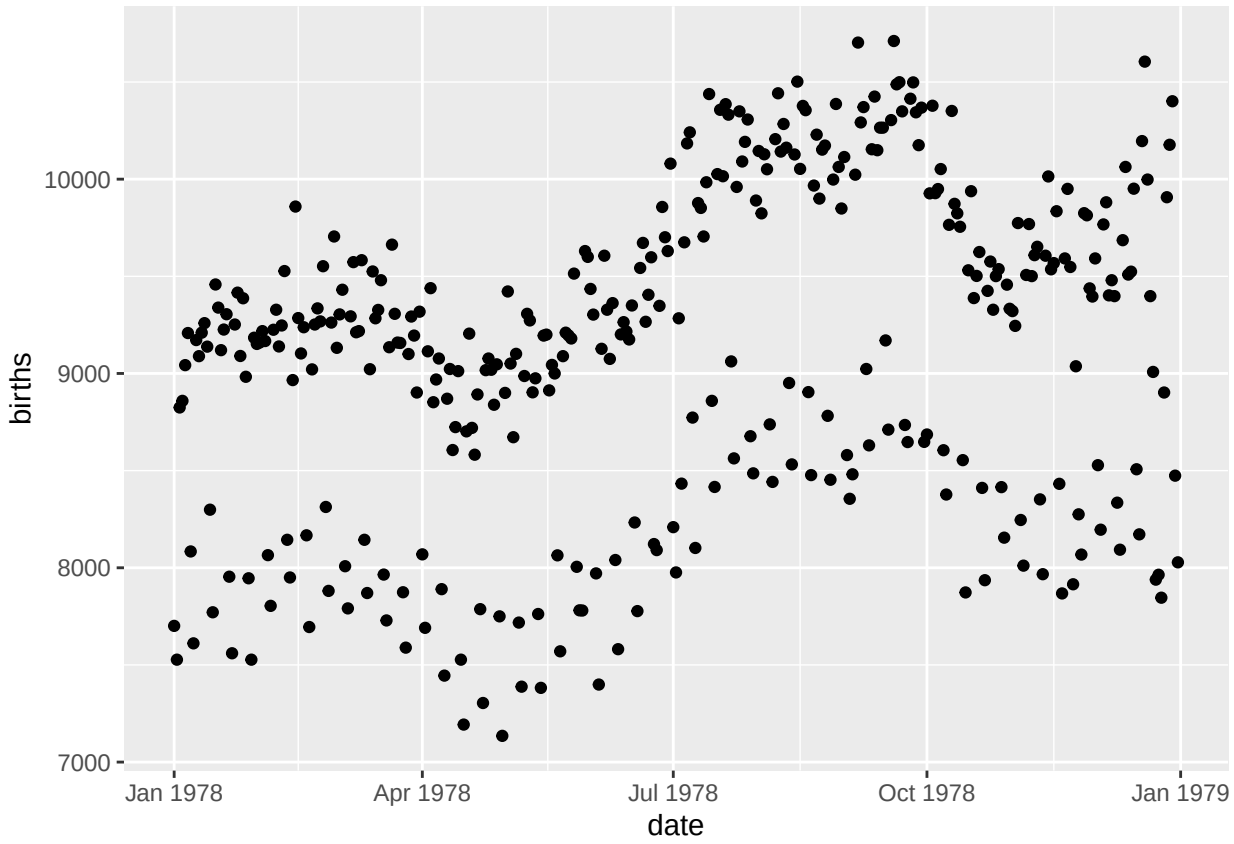
It turns out there is some interesting periodicity regarding the number of births on particular days of the year.

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`.

```
bday.dat <- Births78 %>% select(date, 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?

```
ggplot(data=bday.dat) +
  geom_point(aes(x=date, y=births))
```



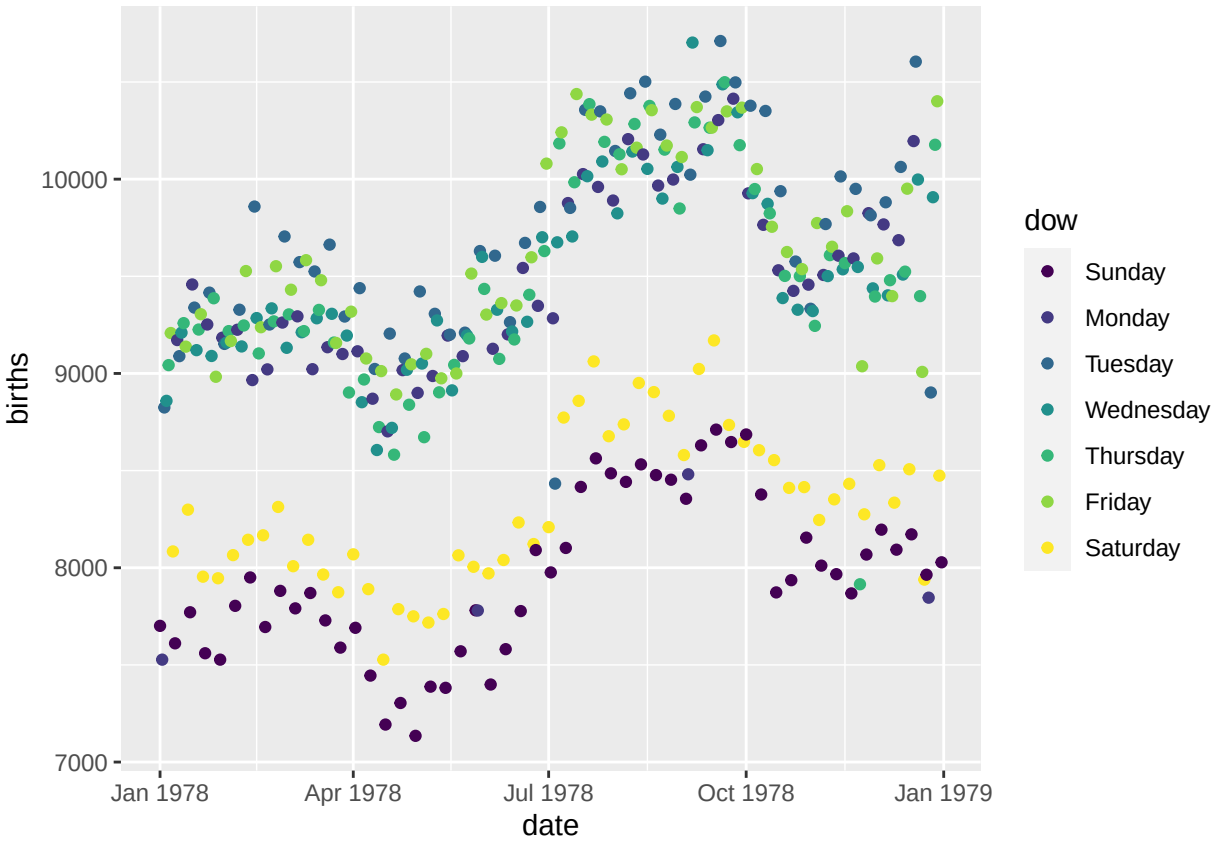
People tend to enjoy making babies when it is cold out. There is a peak in births around August which means that people make babies around the holiday time when they are given time off and can participate in recreational activities. There is also days of the weeks where more babies are born than other days of the week. That is why we see "two" curves

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.

```
bday.dat2 <- bday.dat %>% mutate(dow=wday(date, label=TRUE, abbr=FALSE))
```

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

```
ggplot(data=bday.dat2) +
  geom_point(aes(x=date, y=births, color=dow))
```



Here we can see that more babies are born on weekdays then weekends. This is because doctors force women to have babies during the week so they don't have to work on the weekends. For example, my little sister was born on the day after Christmas because the dumb doctor was going on vacation. I think she was not ready to come out yet. That is why she is the way she is.