STAT 4410/8416 Homework 4

Gerjol, Nicholas

Due on Nov 23, 2021

1. Exploring XML data; In this problem we will read the xml data. For this we will obtain a xml data called olive oils from the link http://www.ggobi.org/book/data/olive.xml. Please follow the directions in each step and provide your codes and output.

a. Parse the xml data from the above link and store in a object called `olive`. Answer the following questions using R code and type your answer:
    i. What is the name of the root of the xml file?

```r
library(XML)
myUrl <- "http://www.ggobi.org/book/data/olive.xml"
olive <- xmlParse(myUrl)
oliveroot <- xmlRoot(olive, skip= TRUE)
xmlName(oliveroot)
```

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

```
  ii. What is the count of data that is available under the root name?
```

```r
xmlSize(oliveroot)
```

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

```r
#root has 1 child: "data"
```

```
  iii. Extract the text value for the `description` of the XML
```

```r
xmlValue(oliveroot[[1]][[1]], trim=TRUE)
```

```
## [1] "This is XML created by GGobi"
```

b. Examine the actual file by going to the link above and answer the following questions using R code and type your answer:
    i. Identify the path of `real variables` in the xml tree

```r
library(xml2)
myxml <- read_xml("http://www.ggobi.org/book/data/olive.xml")
xml_path(xml_find_all(myxml, ".//realvariable"))
```

```
## [1] "/ggobidata/data/variables/realvariable[1]"
## [2] "/ggobidata/data/variables/realvariable[2]"
## [3] "/ggobidata/data/variables/realvariable[3]"
## [4] "/ggobidata/data/variables/realvariable[4]"
## [5] "/ggobidata/data/variables/realvariable[5]"
## [6] "/ggobidata/data/variables/realvariable[6]"
## [7] "/ggobidata/data/variables/realvariable[7]"
## [8] "/ggobidata/data/variables/realvariable[8]"
```

```
  ii. What is the `names` of real variables?
```

```
rvPath <- "//ggobidata/data/variables/realvariable"
rvnames <- xpathSApply(oliveroot, rvPath, xmlGetAttr, "name")
rvnames
```

```
## [1] "palmitic"    "palmitoleic" "stearic"     "oleic"       "linoleic"
## [6] "linolenic"   "arachidic"   "eicosenoic"
```

```
  iii. What is the count of the real variables?
```

```
length(rvnames)
```

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

```
  iv. Identify the path of `categorical variables` in the xml tree
```

```
xml_path(xml_find_all(myxml, ".//categoricalvariable"))
```

```
## [1] "/ggobidata/data/variables/categoricalvariable[1]"
## [2] "/ggobidata/data/variables/categoricalvariable[2]"
```

```
  v. What is the `names` of categorical variables?
```

```
cvPath <- "//ggobidata/data/variables/categoricalvariable"
cvnames <- xpathSApply(oliveroot, cvPath, xmlGetAttr, "name")
cvnames
```

```
## [1] "region" "area"
```

```
  vi. What is the count of the categorical variables?
```

```
length(cvnames)
```

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

```
  vii. How many levels does `categoricalvariable` with `name=area` have? Extract the text value for leve
```

```
varInfo <- oliveroot[[1]][[2]][[2]][[1]]
xmlSize(varInfo) #This is the number of levels
```

```
## [1] 9
```

```
xmlValue(oliveroot[[1]][[2]][[2]][[1]][[5]])
```

```
## [1] "Inland-Sardinia"
```

c. Notice the path for the data in xml file. Use that path to obtain the data and store the data in a data frame called `oliveDat`. Change the column names as you have obtained the column names. Display some data.

```
datPath <- "//ggobidata/data/records/record"
datValue <- xpathApply(olive, datPath, xmlValue)
datValue <- gsub('na ', 'na',datValue)
datValue <- strsplit(gsub('\\n','',datValue), split=" ")
oliveDat <- do.call(rbind.data.frame, datValue)
names(oliveDat) <- c(cvnames,rvnames)
head(oliveDat)
```

```
##   region area palmitic palmitoleic stearic oleic linoleic linolenic arachidic
## 1      1    1     1075          75     226  7823      672        na        60
## 2      1    1     1088          73     224  7709      781        31        61
## 3      1    1      911          54     246  8113      549        31        63
## 4      1    1      966          57      240  7952      619        50        78
## 5      1    1     1051          67      259  7771      672        50        80
## 6      1    1      911          49      268  7924      678        51        70
##   eicosenoic
## 1         29
## 2         29
## 3         29
## 4         35
## 5         46
## 6         44
```

d. Generate a plot of your choice to display any feature of `oliveDat` data. Notice that the column names are different fatty acids. The values are % of fatty acids found in the Italian olive oils coming from different regions and areas.

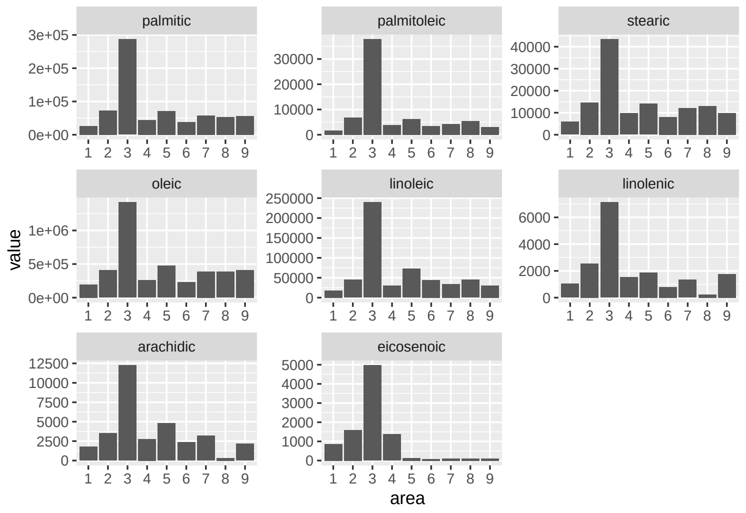
```
library(reshape2)
library(ggplot2)
oliveDat <- subset(oliveDat, select = -c(region))
head(oliveDat)
```

```
##   area palmitic palmitoleic stearic oleic linoleic linolenic arachidic
## 1    1     1075          75     226  7823      672        na        60
## 2    1     1088          73     224  7709      781        31        61
## 3    1      911          54     246  8113      549        31        63
## 4    1      966          57     240  7952      619        50        78
## 5    1     1051          67     259  7771      672        50        80
## 6    1      911          49     268  7924      678        51        70
##   eicosenoic
## 1         29
## 2         29
## 3         29
## 4         35
## 5         46
## 6         44
```

```
meltdat <- melt(oliveDat, "area")
meltdat$value <- as.numeric(meltdat$value)
head(meltdat)
```

```
##   area variable value
## 1    1 palmitic  1075
## 2    1 palmitic  1088
## 3    1 palmitic   911
## 4    1 palmitic   966
## 5    1 palmitic  1051
## 6    1 palmitic   911
```

```
p<-ggplot(data=meltdat, aes(x=area, y=value)) +
  geom_bar(stat="identity")
p + facet_wrap(~variable, scales = "free")
```



2. Working with date-time data; The object `myDate` contains the date and time. Based on this object answer the following questions using R code and type the answer after your findings.

```
myDate <- "2020-10-01 09:42:43"
```

a. Convert `myDate` into a date and time object with Portland, OR time zone. Display the result.

```
library(lubridate)
myDate <- ymd_hms(myDate)
myDate <- with_tz(myDate, tzone = "America/Los_Angeles")
myDate
```

```
## [1] "2020-10-01 02:42:43 PDT"
```

b. Write your codes so that it displays the week day and also the month of `myDate`.

```
wday(myDate, label=TRUE, abbr=FALSE)
```

```
## [1] Thursday
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday
```

```
month(myDate, label=TRUE, abbr=FALSE)
```

```
## [1] October
## 12 Levels: January < February < March < April < May < June < ... < December
```

c. What weekday and the month is it after exactly 100 years from `myDate`?

```
newdate <- myDate + years(100)
wday(newdate, label=TRUE, abbr=FALSE)
```

```
## [1] Tuesday
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday
```

```
month(newdate, label=TRUE, abbr=FALSE)
```

```
## [1] October
## 12 Levels: January < February < March < April < May < June < ... < December
```

d. Add two month with myDate and display the resulting date time. Explain why the time zone has changed even though you did not ask for time zone change.

```r
myDate + months(2)
```

```
## [1] "2020-12-01 02:42:43 PST"
```

```r
#The time zone changed to UTC because daylight savings time goes into effect
```

e. Suppose this homework is due on November 23, 2021 by 11.59PM. Compute and display how many seconds you got to complete this homework? Also compute the hours.

```r
duedate <- ymd_hms("2021-11-23 11:59:00")
difftime(duedate, myDate, units=c("secs"))
```

```
## Time difference of 36123377 secs
```

```r
difftime(duedate, myDate, units=c("hours"))
```

```
## Time difference of 10034.27 hours
```

```r
#use time zone in the second line
```

f. Suppose you are working with a Time-Series data. Where should the Time Value be? X-Axis or the Y-Axis? Explain your answer.

```r
#Time-series data should always have time as the x-axis represents the independent variable with the y-
```

g. How do you get the current date as set in the computer? Does the date belong to a Leap Year?

```r
 now()
```

```
## [1] "2021-11-21 15:24:19 CST"
```

```r
leap_year(now())
```

```
## [1] FALSE
```

h. For the years 2020 & 2021, count the number of weekends. Which year has the highest number of Weekends?

```r
year1startdate <- as.Date("2020-01-01")
year1enddate <- as.Date("2020-12-31")
year2startdate <- as.Date("2021-01-01")
year2enddate <- as.Date("2021-12-31")
sum(!weekdays(seq(year1startdate, year1enddate, "days")) %in% c("Saturday", "Sunday")) #2020
```

```
## [1] 262
```

```r
sum(!weekdays(seq(year2startdate, year2enddate, "days")) %in% c("Saturday", "Sunday")) #2021
```

```
## [1] 261
```

```r
#2020 has more weekends by a single day
```

i. What is the month(MM) and day(DD) on the 305th day of the current year?

```r
year <- floor_date(now(),unit="year")
newdate <- year + days(304)
month(newdate) #month
```

```
## [1] 11
```

```r
day(newdate) #day
```

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

j. The Date `2020-10-20` is formatted `YYYY-MM-DD`, format it into `MM-DD-YYYY`

```
date2 <- "2020-10-20"
trialdate <- ymd(date2)
format(trialdate,format="%m-%d-%Y")
```

```
## [1] "10-20-2020"
```

k. Find if the Date on the question above is the weekend or a weekday.

```
weekdays(trialdate) #Tuesday is a weekday
```

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

3. Creating HTML Page; In this problem we would like to create a basic HTML page. Please follow each of the steps below and finally submit your HTML file on Canvas. Please note that you don't need to answer these questions here in the .Rmd file.

a. Open a notepad or any plain text editor. Write down some basic HTML codes as shown in online (year 2014) Lecture 15, slide 6 and modify according to the following questions. Save the file as hw4.html and upload on Canvas as a separate file.
b. Write "What is data science?" in the first header tag, `<h1></h1>`
c. Hw1 solution contains the answer of what is data science. The answer has three paragraphs. Write the three paragraphs of text about data science in three different paragraph tags `<p></p>`. You can copy the text from hw1 solution.
d. Write "What we learnt from hw1" in second heading under tag `<h2></h2>`
e. Copy all the points we learnt in hw1 solution. List all the points under ordered list tag `<ol></ol>`. Notice that each item of the list should be inside list item tag `<li></li>`.
f. Now we want to make the text beautiful. For this we would write some CSS codes in between `<head></head>` tag under `<style></style>`. For this please refer to online (year 2014) lecture 15 slide 8. First change the fonts of the body tag to Helvetica Neue.
g. For the paragraph that contains the definition of data science, give an attribute `id='dfn'` and in CSS change the color of 'dfn' to white, background-color to olive and font to be bold.
h. For other paragraphs, give an attribute `class='cls'` and in CSS change the color of 'cls' to green.
i. Write CSS so that color of h1, h2 headers becomes `orange`.
j. (Optional and will not be graded) Write java Scripts codes so that onClick on `h1` header, it shows a message 'Its about data science'.

4. Walmart Sales Analysis Download and read the dataset `walmart_sales.csv` and `walmart_fuel_prices.csv`.

```
library(data.table)
wmsales <- fread("walmart_sales.csv")
wmfuel <- fread("walmart_fuel_prices.csv")
wmsales$Date <- ymd(wmsales$Date)
wmfuel$Date <- ymd(wmfuel$Date)
head(wmsales)
```

```
##    Store Dept       Date Weekly_Sales IsHoliday
## 1:     1    1 2010-02-05     24924.50     FALSE
## 2:     1    1 2010-02-12     46039.49      TRUE
## 3:     1    1 2010-02-19     41595.55     FALSE
## 4:     1    1 2010-02-26     19403.54     FALSE
## 5:     1    1 2010-03-05     21827.90     FALSE
## 6:     1    1 2010-03-12     21043.39     FALSE
```

```
head(wmfuel)
```

```
##    index Store       Date Temperature Fuel_Price IsHoliday
## 1:     1     1 2010-02-05       42.31      2.572     FALSE
## 2:     2     1 2010-02-12       38.51      2.548      TRUE
## 3:     3     1 2010-02-19       39.93      2.514     FALSE
## 4:     4     1 2010-02-26       46.63      2.561     FALSE
## 5:     5     1 2010-03-05       46.50      2.625     FALSE
## 6:     6     1 2010-03-12       57.79      2.667     FALSE
```

We will follow the following data description when working with the above 2 datasets:

- `index:` index is a default value of count
- `Store:` Store is represented in number ID(1,2,3,4,. . .)
- `Dept:` Dept is Department in each Store represented in number ID (1,2,3,4,. . .)
- `Date:` Date is in YYYY-MM-DD char format - *needs to be converted into Date data type*
- `Weekly_Sales:` Sales of a given Dept in a given Store for the Date
- `Temperature:` Average temperature on the Date at given Store region
- `Fuel_Price:` Cost of the Fuel on the given Date at a given Store
- `IsHoliday:` Is the given Date a holiday Week?

Answer all of the following questions below and support your answer showing the codes and a plot (if applicable):

a. For both datasets, breakdown the `Date` column and create additional new columns `Year`, `Month`, and `Day`. You should now have additional 3 new columns in your both dataset. Report only the column names for both the dataset.

```
wmsales$Year <- year(wmsales$Date)
wmsales$Month <- month(wmsales$Date)
wmsales$Day <- day(wmsales$Date)
wmfuel$Year <- year(wmfuel$Date)
wmfuel$Month <- month(wmfuel$Date)
wmfuel$Day <- day(wmfuel$Date)
colnames(wmsales)
```

```
## [1] "Store"        "Dept"         "Date"         "Weekly_Sales" "IsHoliday"
## [6] "Year"         "Month"        "Day"
```

```
colnames(wmfuel)
```

```
## [1] "index"       "Store"       "Date"        "Temperature" "Fuel_Price"
## [6] "IsHoliday"   "Year"        "Month"       "Day"
```

b. In `walmart_sales`: which `Month(s)` of `Year` have the highest `Weekly_Sales`? Report the Year, Month, Store, and Dept.

```
monthsales <- aggregate(Weekly_Sales ~ Year + Month + Store + Dept, wmsales, sum)
monthsales <- as.data.table(monthsales)
monthsales <- monthsales[order(-Weekly_Sales)]
head(monthsales, 1)
```

```
##    Year Month Store Dept Weekly_Sales
## 1: 2010    12    10   72      1216569
```

c. In `walmart_sales`: calculate the average monthly sales by Department for each Store. Which Store(s) has the highest average monthly sales on the department(s)? Report the Store, Department, Date.

```
avgsales <- aggregate(Weekly_Sales ~ Store + Dept, monthsales, mean)
avgsales <- as.data.table(avgsales)
avgsales <- avgsales[order(-Weekly_Sales)]
head(avgsales)
```

```
##    Store Dept Weekly_Sales
## 1:    14   92     790954.5
## 2:     2   92     714307.7
## 3:    20   92     713412.9
## 4:    13   92     702147.8
## 5:     4   92     690582.1
## 6:    20   95     652660.5
```

d. In `walmart_sales`: which month of year 2011 has the highest overall sales by Store? Name the holiday(Labor day, July 4th, Halloween, Thanksgiving, Christmas,… etcs) that falls on the month. After that do the same for 2012. Does the highest sales per month fall on the same holiday for both years? Report your findings for both year.

```
elevensales <- wmsales[ wmsales$"Year" %in% c(2011) ,]
elevensales<- aggregate(Weekly_Sales ~ Month + Store, elevensales, sum)
elevensales <- as.data.table(elevensales)
elevensales <- elevensales[order(-Weekly_Sales)]
head(elevensales,1) #This is Christmas
```

```
##    Month Store Weekly_Sales
## 1:    12    20     13206333
```

```
twelvesales <- wmsales[ wmsales$"Year" %in% c(2012) ,]
twelvesales<- aggregate(Weekly_Sales ~ Month + Store, twelvesales, sum)
twelvesales <- as.data.table(twelvesales)
twelvesales <- twelvesales[order(-Weekly_Sales)]
head(twelvesales,1) #Juneteeth or Fathers day?
```

```
##    Month Store Weekly_Sales
## 1:     6     4     10984472
```

e. In `walmart_sales`: report the lowest sales per month for the year 2011 for `IsHoliday == TRUE`. Name the holiday(Labor day, July 4th, Halloween, Thanksgiving, Christmas,… etcs) that falls on the month. Do the same for 2012 and report if the lowest sales are on the same month.

```
holidaysales <- wmsales[ wmsales$"IsHoliday" %in% c(TRUE) ,]
holidaysales <- holidaysales[holidaysales$"Year" %in% c(2011) ,]
holidaysales <- aggregate(Weekly_Sales ~ Month, holidaysales, sum)
holidaysales <- as.data.table(holidaysales)
holidaysales <- holidaysales[order(Weekly_Sales)]
head(holidaysales, 1) #Lowest is on christmas
```

```
##    Month Weekly_Sales
## 1:    12     46042461
```

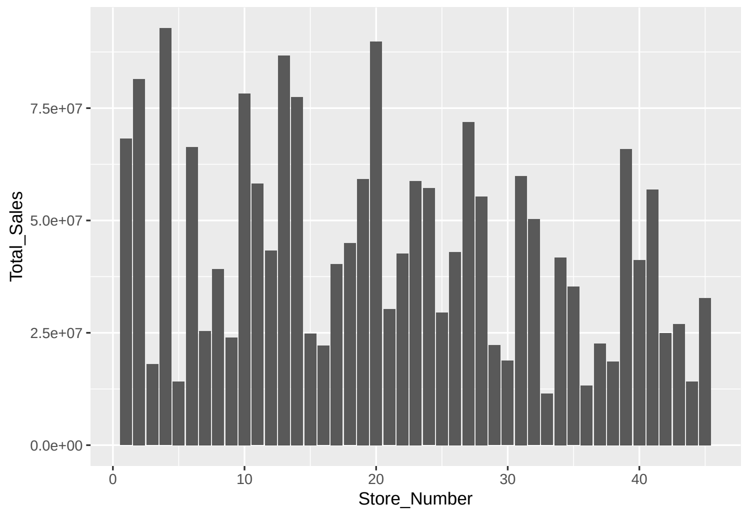
```
tholidaysales <- wmsales[ wmsales$"IsHoliday" %in% c(TRUE) ,]
tholidaysales <- tholidaysales[tholidaysales$"Year" %in% c(2012) ,]
tholidaysales <- aggregate(Weekly_Sales ~ Month, tholidaysales, sum)
tholidaysales <- as.data.table(tholidaysales)
tholidaysales <- tholidaysales[order(Weekly_Sales)]
head(tholidaysales, 1) #Lowest is labor day
```

```
##    Month Weekly_Sales
## 1:     9     48330059
```

f. In `walmart_sales`: We have 45 unique stores. Generate a nice plot on the total sales by store for the year 2012. Report the Store number.

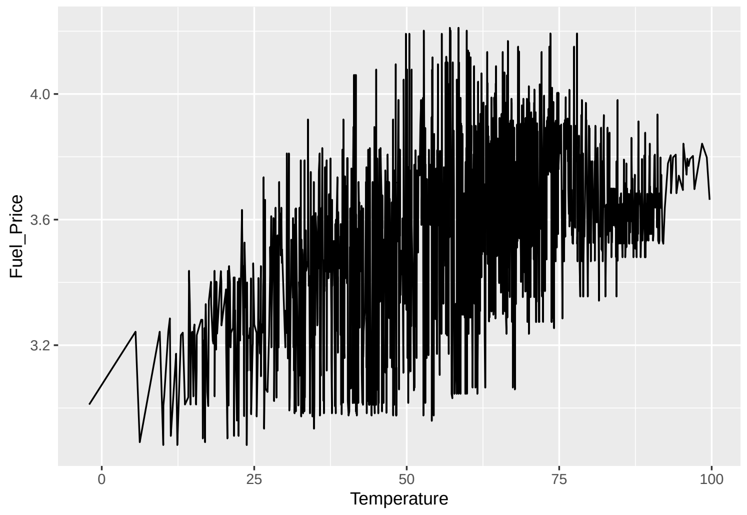
```
storesales <- wmsales[ wmsales$"Year" %in% c(2012) ,]
storesales<- aggregate(Weekly_Sales ~ Store, storesales, sum)
```

```r
names(storesales) <- c("Store_Number" , "Total_Sales")
p<-ggplot(data=storesales, aes(x=Store_Number, y=Total_Sales)) +
  geom_bar(stat="identity")
p
```



g. In `walmart_fuel_prices`: For the year 2011 do you think higher the `temperature` relates to higher `fuel price`? Support your answer with a nice plot.

```r
fueltemp <- wmfuel[ wmfuel$"Year" %in% c(2011) ,]
fueltemp <- aggregate(Fuel_Price ~ Temperature, fueltemp, mean)
p <- ggplot(fueltemp, aes(x=Temperature, y=Fuel_Price)) +
 geom_line()
p
```



```r
#the deviation of the data is quite large, but the overall trend is still positive correlation. I would
```

h. In `walmart_fuel_prices`: For the year 2010 which `Store` had the lowest Fuel Price? Report the `month` and `temperature`. On the same `month`, what was the highest fuel price for the store? Report the difference.

```
storefuel <-  wmfuel[ wmfuel$"Year" %in% c(2010) ,]
storefuel <- aggregate(Fuel_Price ~ Month + Day + Store, storefuel, sum)
storefuel <- as.data.table(storefuel)
storefuel <- storefuel[order(Fuel_Price)]
head(storefuel, 1)#Lowest fuel price store is store 36, 2.472 on February 19th
```

```
##    Month Day Store Fuel_Price
## 1:     2  19    36      2.472
```

```
lowprice <- storefuel[1,4]
storefuel <- storefuel[ storefuel$"Store" %in% c(36) ,]
storefuel <- storefuel[ storefuel$"Month" %in% c(2) ,]
storefuel <- storefuel[order(-Fuel_Price)]
head(storefuel, 1) #higest fuel price same month for store 36, 2.545 on February 5th
```

```
##    Month Day Store Fuel_Price
## 1:     2   5    36      2.545
```

```
highprice <- storefuel[1,4]
highprice - lowprice
```

```
##    Fuel_Price
## 1:      0.073
```

i. In `walmart_fuel_prices`: For the `IsHoliday == TRUE`, which month has the lowest `Fuel Price` for the year 2012? name the holiday(Labor day, July 4th, Halloween, Thanksgiving, Christmas,… etcs) that falls on the month. Also report month of the highest fuel price and name of the holiday.

```
storefuel <-  wmfuel[ wmfuel$"Year" %in% c(2012) ,]
storefuel <-  wmfuel[ wmfuel$"IsHoliday" %in% c(TRUE) ,]
storefuel <- aggregate(Fuel_Price ~ Month + Day + Store, storefuel, sum)
storefuel <- as.data.table(storefuel)
storefuel <- storefuel[order(Fuel_Price)]
head(storefuel, 1 ) #9/10 has the lowest fuel price at 2.513 this is around labor day
```

```
##    Month Day Store Fuel_Price
## 1:     9  10    36      2.513
```

```
storefuel <- storefuel[order(-Fuel_Price)]
head(storefuel, 1) #9/7 had the highest fuel price at 4.124 this is around labor day
```

```
##    Month Day Store Fuel_Price
## 1:     9   7    10      4.124
```

5. Optional for undergraduate but mandatory for graduate students Download the data from Github - click here

The link above contains a time-series data for COVID-19 confirmed cases in the US. Limit the data to only use `Nebraska State` and please answer the following questions:

a. What is the total confirmed cases in Nebraska as of October 30th 2020 as per the dataset?

```
library(dplyr)
fulldat <- fread("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c
nebraskadat <- fulldat[ fulldat$"Province_State" %in% c("Nebraska") ,]
temp <- select(nebraskadat, contains('10/31/20'))
temp[ , lapply(.SD, sum)]
```

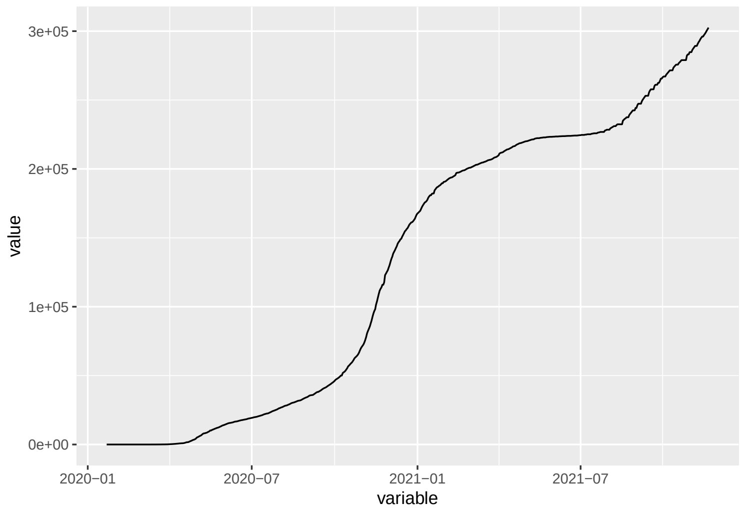
```
##    10/31/20
## 1:    70732
```

b. On what date has the highest confirmed cases? Demonstrate using a suitable graph for all the available data.

```
library(dplyr)
df = subset(nebraskadat, select = -c(UID, iso2, iso3, code3, FIPS, Admin2, Country_Region, Lat, Long_, 
test <- aggregate(. ~ Province_State, df, sum)
test
```

```
##   Province_State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1       Nebraska       0       0       0       0       0       0       0
##   1/29/20 1/30/20 1/31/20 2/1/20 2/2/20 2/3/20 2/4/20 2/5/20 2/6/20 2/7/20
## 1       0       0       0      0      0      0      0      0      0      0
##   2/8/20 2/9/20 2/10/20 2/11/20 2/12/20 2/13/20 2/14/20 2/15/20 2/16/20 2/17/20
## 1      0      0       0       0       0       0       0       0       0       0
##   2/18/20 2/19/20 2/20/20 2/21/20 2/22/20 2/23/20 2/24/20 2/25/20 2/26/20
## 1       0       0       0       0       0       0       0       0       0
##   2/27/20 2/28/20 2/29/20 3/1/20 3/2/20 3/3/20 3/4/20 3/5/20 3/6/20 3/7/20
## 1       0       0       0      0      0      0      0      0      1      1
##   3/8/20 3/9/20 3/10/20 3/11/20 3/12/20 3/13/20 3/14/20 3/15/20 3/16/20 3/17/20
## 1      1      3       3       5      10      14      18      17      18      23
##   3/18/20 3/19/20 3/20/20 3/21/20 3/22/20 3/23/20 3/24/20 3/25/20 3/26/20
## 1      24      29      37      38      51      51      66      71      74
##   3/27/20 3/28/20 3/29/20 3/30/20 3/31/20 4/1/20 4/2/20 4/3/20 4/4/20 4/5/20
## 1      82      96     108     145     172    210    246    279    321    364
##   4/6/20 4/7/20 4/8/20 4/9/20 4/10/20 4/11/20 4/12/20 4/13/20 4/14/20 4/15/20
## 1    417    447    519    568     635     699     791     814     897     901
##   4/16/20 4/17/20 4/18/20 4/19/20 4/20/20 4/21/20 4/22/20 4/23/20 4/24/20
## 1     952    1066    1249    1474    1648    1685    1813    2202    2424
##   4/25/20 4/26/20 4/27/20 4/28/20 4/29/20 4/30/20 5/1/20 5/2/20 5/3/20 5/4/20
## 1    2719    3030    3358    3517    3851    4281   5008   5317   5661   6037
##   5/5/20 5/6/20 5/7/20 5/8/20 5/9/20 5/10/20 5/11/20 5/12/20 5/13/20 5/14/20
## 1   6373   6689   7334   7818   8093    8171    8407    8532    8912    9260
##   5/15/20 5/16/20 5/17/20 5/18/20 5/19/20 5/20/20 5/21/20 5/22/20 5/23/20
## 1    9610   10220   10177   10625   10854   11122   11427   11662   11963
##   5/24/20 5/25/20 5/26/20 5/27/20 5/28/20 5/29/20 5/30/20 5/31/20 6/1/20 6/2/20
## 1   12134   12362   12619   12984   13249   13648   13905   14101  14345  14616
##   6/3/20 6/4/20 6/5/20 6/6/20 6/7/20 6/8/20 6/9/20 6/10/20 6/11/20 6/12/20
## 1  14885  15139  15406  15572  15664  15786  15918   16058   16315   16522
##   6/13/20 6/14/20 6/15/20 6/16/20 6/17/20 6/18/20 6/19/20 6/20/20 6/21/20
## 1   16640   16730   16851   17038   17231   17414   17588   17707   17812
##   6/22/20 6/23/20 6/24/20 6/25/20 6/26/20 6/27/20 6/28/20 6/29/20 6/30/20
## 1   17963   18099   18221   18346   18524   18775   18899   19042   19177
##   7/1/20 7/2/20 7/3/20 7/4/20 7/5/20 7/6/20 7/7/20 7/8/20 7/9/20 7/10/20
## 1  19310  19452  19660  19827  19929  20046  20201  20425  20623   20777
##   7/11/20 7/12/20 7/13/20 7/14/20 7/15/20 7/16/20 7/17/20 7/18/20 7/19/20
## 1   20998   21172   21399   21717   21979   22134   22361   22481   22583
##   7/20/20 7/21/20 7/22/20 7/23/20 7/24/20 7/25/20 7/26/20 7/27/20 7/28/20
## 1   22847   23190   23486   23818   24174   24395   24618   24899   25157
##   7/29/20 7/30/20 7/31/20 8/1/20 8/2/20 8/3/20 8/4/20 8/5/20 8/6/20 8/7/20
## 1   25422   25766   26211  26391  26702  26956  27178  27489  27821  28104
##   8/8/20 8/9/20 8/10/20 8/11/20 8/12/20 8/13/20 8/14/20 8/15/20 8/16/20 8/17/20
## 1  28245  28432   28696   29030   29244   29660   29988   30241   30372   30563
##   8/18/20 8/19/20 8/20/20 8/21/20 8/22/20 8/23/20 8/24/20 8/25/20 8/26/20
## 1   30825   31040   31348   31626   31780   31889   32047   32348   32727
##   8/27/20 8/28/20 8/29/20 8/30/20 8/31/20 9/1/20 9/2/20 9/3/20 9/4/20 9/5/20
## 1   33101   33436   33753   34046   34287  34574  34995  35469  35661  35805
##   9/6/20 9/7/20 9/8/20 9/9/20 9/10/20 9/11/20 9/12/20 9/13/20 9/14/20 9/15/20
## 1  35886  35975  36469  36917   37373   37851   38120   38188   38642   38970
##   9/16/20 9/17/20 9/18/20 9/19/20 9/20/20 9/21/20 9/22/20 9/23/20 9/24/20
## 1   39419   39921   40387   40797   41083   41388   41785   42278   42731
##   9/25/20 9/26/20 9/27/20 9/28/20 9/29/20 9/30/20 10/1/20 10/2/20 10/3/20
## 1   43162   43596   44063   44578   45044   45564   46185   46977   47403
##   10/4/20 10/5/20 10/6/20 10/7/20 10/8/20 10/9/20 10/10/20 10/11/20 10/12/20
## 1   47807   48259   48757   49396   50059   50059    51887    52382    52839
##   10/13/20 10/14/20 10/15/20 10/16/20 10/17/20 10/18/20 10/19/20 10/20/20
## 1    53543    54467    55428    56714    57332    58068    58817    59409
##   10/21/20 10/22/20 10/23/20 10/24/20 10/25/20 10/26/20 10/27/20 10/28/20
## 1    60308    61285    62510    63215    63797    64499    65376    66545
##   10/29/20 10/30/20 10/31/20 11/1/20 11/2/20 11/3/20 11/4/20 11/5/20 11/6/20
## 1    68150    69645    70732   71666   72620   74060   75888   78012   80693
##   11/7/20 11/8/20 11/9/20 11/10/20 11/11/20 11/12/20 11/13/20 11/14/20 11/15/20
## 1   82395   83969   85551    87733    89942    92553    94922    96834    98161
##   11/16/20 11/17/20 11/18/20 11/19/20 11/20/20 11/21/20 11/22/20 11/23/20
## 1   101601   103805   106617   109280   111661   113029   114061   115921
##   11/24/20 11/25/20 11/26/20 11/27/20 11/28/20 11/29/20 11/30/20 12/1/20
## 1   115921   117682   122952   124066   125323   126466   128407  130194
##   12/2/20 12/3/20 12/4/20 12/5/20 12/6/20 12/7/20 12/8/20 12/9/20 12/10/20
## 1  132530  134710  136325  138568  139834  141127  142595  143924   145774
##   12/11/20 12/12/20 12/13/20 12/14/20 12/15/20 12/16/20 12/17/20 12/18/20
## 1   146877   147688   148861   149344   150861   152103   153400   154745
##   12/19/20 12/20/20 12/21/20 12/22/20 12/23/20 12/24/20 12/25/20 12/26/20
## 1   155415   156382   157103   158324   159662   160357   161162   161337
##   12/27/20 12/28/20 12/29/20 12/30/20 12/31/20 1/1/21 1/2/21 1/3/21 1/4/21
## 1   161974   162849   163781   165297   166798 167716 168262 169000 169585
##   1/5/21 1/6/21 1/7/21 1/8/21 1/9/21 1/10/21 1/11/21 1/12/21 1/13/21 1/14/21
## 1 171033 172469 173591 174614 175620  176026  176670  177670  179199  180131
##   1/15/21 1/16/21 1/17/21 1/18/21 1/19/21 1/20/21 1/21/21 1/22/21 1/23/21
## 1  180910  180910  181978  182176  182176  184482  185346  186255  186854
##   1/24/21 1/25/21 1/26/21 1/27/21 1/28/21 1/29/21 1/30/21 1/31/21 2/1/21 2/2/21
## 1  187147  187793  188122  188784  189597  189597  190570  190713 190950 191437
##   2/3/21 2/4/21 2/5/21 2/6/21 2/7/21 2/8/21 2/9/21 2/10/21 2/11/21 2/12/21
## 1 192042 192549 193069 193421 193722 193826 194170  194632  195006  195485
##   2/13/21 2/14/21 2/15/21 2/16/21 2/17/21 2/18/21 2/19/21 2/20/21 2/21/21
## 1  197027  197236  197328  197447  197746  198042  198442  198751  198949
##   2/22/21 2/23/21 2/24/21 2/25/21 2/26/21 2/27/21 2/28/21 3/1/21 3/2/21 3/3/21
## 1  199045  199402  199782  200163  200447  200720  200882 200946 201346 201608
##   3/4/21 3/5/21 3/6/21 3/7/21 3/8/21 3/9/21 3/10/21 3/11/21 3/12/21 3/13/21
## 1 201973 202310 202653 203026 203027 203279  203587  203890  204162  204464
##   3/14/21 3/15/21 3/16/21 3/17/21 3/18/21 3/19/21 3/20/21 3/21/21 3/22/21
## 1  204638  204753  205103  205214  205539  205814  206246  206388  206571
##   3/23/21 3/24/21 3/25/21 3/26/21 3/27/21 3/28/21 3/29/21 3/30/21 3/31/21
## 1  206707  206939  207227  207667  208118  208424  208553  208912  209346
##   4/1/21 4/2/21 4/3/21 4/4/21 4/5/21 4/6/21 4/7/21 4/8/21 4/9/21 4/10/21
## 1 209896 211239 211570 211812 211958 212257 212785 213188 213574  214010
##   4/11/21 4/12/21 4/13/21 4/14/21 4/15/21 4/16/21 4/17/21 4/18/21 4/19/21
## 1  214207  214351  214682  215074  215383  215792  216297  216473  216613
##   4/20/21 4/21/21 4/22/21 4/23/21 4/24/21 4/25/21 4/26/21 4/27/21 4/28/21
## 1  217108   217596   217905   218197   218580   218732   218832   219090   219341
##   4/29/21 4/30/21 5/1/21 5/2/21 5/3/21 5/4/21 5/5/21 5/6/21 5/7/21 5/8/21
## 1  219559   219826 220032 220127 220225 220481 220720 220933 221153 221347
##   5/9/21 5/10/21 5/11/21 5/12/21 5/13/21 5/14/21 5/15/21 5/16/21 5/17/21
## 1 221434  221500  221911  222088  222247  222335  222335  222335  222512
##   5/18/21 5/19/21 5/20/21 5/21/21 5/22/21 5/23/21 5/24/21 5/25/21 5/26/21
## 1  222612   222676   222780   222884   222884   222884   223054   223126   223197
##   5/27/21 5/28/21 5/29/21 5/30/21 5/31/21 6/1/21 6/2/21 6/3/21 6/4/21 6/5/21
## 1  223243   223304   223304   223304   223368 223404 223434 223517 223558 223558
##   6/6/21 6/7/21 6/8/21 6/9/21 6/10/21 6/11/21 6/12/21 6/13/21 6/14/21 6/15/21
## 1 223558 223648 223685 223714  223749  223792  223792  223792  223847  223888
##   6/16/21 6/17/21 6/18/21 6/19/21 6/20/21 6/21/21 6/22/21 6/23/21 6/24/21
## 1  223931   223960   223986   223986   223986   224065   224103   224156   224206
##   6/25/21 6/26/21 6/27/21 6/28/21 6/29/21 6/30/21 7/1/21 7/2/21 7/3/21 7/4/21
## 1  224226   224226   224226   224330   224404   224488 224488 224682 224682 224682
##   7/5/21 7/6/21 7/7/21 7/8/21 7/9/21 7/10/21 7/11/21 7/12/21 7/13/21 7/14/21
## 1 224682 224873 224873 225069 225171  225171  225171  225171  225477  225600
##   7/15/21 7/16/21 7/17/21 7/18/21 7/19/21 7/20/21 7/21/21 7/22/21 7/23/21
## 1  225711   225861   225861   225861   225861   226307   226442   226606   226839
##   7/24/21 7/25/21 7/26/21 7/27/21 7/28/21 7/29/21 7/30/21 7/31/21 8/1/21 8/2/21
## 1   226839   226839   226839   226839   227848   228086   228450   228450 228450 228450
##   8/3/21 8/4/21 8/5/21 8/6/21 8/7/21 8/8/21 8/9/21 8/10/21 8/11/21 8/12/21
## 1 229443 229824 230236 230630 231069 231069 231069  231916  232399  232399
##   8/13/21 8/14/21 8/15/21 8/16/21 8/17/21 8/18/21 8/19/21 8/20/21 8/21/21
## 1  232399   232399   232399   232399   235075   235686   236346   236755   237492
##   8/22/21 8/23/21 8/24/21 8/25/21 8/26/21 8/27/21 8/28/21 8/29/21 8/30/21
## 1  237492   237492   239102   240028   240804   241671   242498   242498   242498
##   8/31/21 9/1/21 9/2/21 9/3/21 9/4/21 9/5/21 9/6/21 9/7/21 9/8/21 9/9/21
## 1  244254 244254 246218 247320 247320 247320 247320 249108 250264 251219
##   9/10/21 9/11/21 9/12/21 9/13/21 9/14/21 9/15/21 9/16/21 9/17/21 9/18/21
## 1  252159   253080   253080   253080   253080   255611   256906   257787   257787
##   9/19/21 9/20/21 9/21/21 9/22/21 9/23/21 9/24/21 9/25/21 9/26/21 9/27/21
## 1  257787   257787   260199   261080   261080   261080   262475   262475   263763
##   9/28/21 9/29/21 9/30/21 10/1/21 10/2/21 10/3/21 10/4/21 10/5/21 10/6/21
## 1  265516   265516   266449   267079   267079   267079   268381   269138   269942
##   10/7/21 10/8/21 10/9/21 10/10/21 10/11/21 10/12/21 10/13/21 10/14/21 10/15/21
## 1  270724  271550  271550   271550   271550   273416   274254   274993   275694
##   10/16/21 10/17/21 10/18/21 10/19/21 10/20/21 10/21/21 10/22/21 10/23/21
## 1   275694   275694   276817   277436   278174   278976   278976   278976
##   10/24/21 10/25/21 10/26/21 10/27/21 10/28/21 10/29/21 10/30/21 10/31/21
## 1   278976   278976   278976   282287   283153   283153   284766   284766
##   11/1/21 11/2/21 11/3/21 11/4/21 11/5/21 11/6/21 11/7/21 11/8/21 11/9/21
## 1  284766   286373   287345   288257   289282   289282   289282   290794   292032
##   11/10/21 11/11/21 11/12/21 11/13/21 11/14/21 11/15/21 11/16/21 11/17/21
## 1   292990   294247   295244   296106   296106   297214   298082   299149
##   11/18/21 11/19/21 11/20/21
## 1   300348   301436   302567
```

```
testmelt <- melt(test, id="Province_State")
testmelt$variable <- mdy(testmelt$variable)
p <- ggplot(testmelt, aes(x=variable, y=value)) +
 geom_line()
p
```



```
testmelt$variable[length(testmelt$variable)] #graph shows highest case is last entry
```

```
## [1] "2021-11-20"
```

c. Which County has the highest daily confirmed cases? Report both the County name and the date

```
trimmeddat <- subset(nebraskadat, select = -c(UID, iso2, iso3, code3, FIPS, Province_State, Country_Reg
countymelt <- melt(trimmeddat, id="Admin2")
countymelt <- as.data.table(countymelt)
sortdat <- countymelt[order(-value),]
names(sortdat) <- c("County", "date", "case_num")
sortdat[1]
```

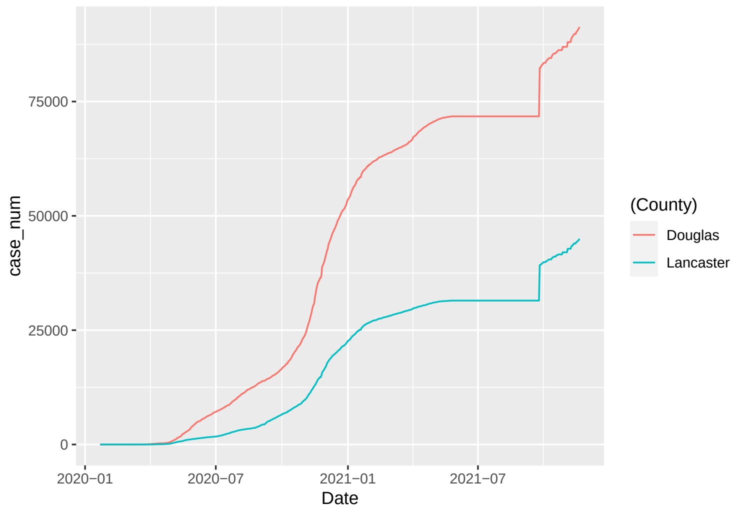
```
##     County     date case_num
## 1: Douglas 11/19/21    91168
```

d. Identify two countries that have top total confirmed cases. Generate a time series plot of daily confirm cases for these two countries.

```
df = subset(nebraskadat, select = -c(UID, iso2, iso3, code3, FIPS, Province_State, Country_Region, Lat,
df <- melt(df,id="Admin2")
test <- aggregate(. ~ Admin2, df, sum)
test <- as.data.table(test)
names(test) <- c("County", "datesum", "case_num_sum")
testsort <- test[order(-case_num_sum),]
testsort[1:2,] # This gives the top 2 confirmed cases
```

```
##       County datesum case_num_sum
## 1:   Douglas  224115     27483022
## 2: Lancaster  224115     12013132
```

```
df <- as.data.table(df)
names(df) <- c("County" , "Date", "case_num")
df$Date <- mdy(df$Date)
df <- df[ df$"County" %in% c("Douglas","Lancaster") ,]
p <- ggplot(df, aes(x=Date, y=case_num, color=(County))) +
 geom_line()
p
```



e. Show the total confirmed cases for all the locations in an interactive world map (hint: you may use `leaflet` package in R.

```
library(leaflet)

test$datesum <- NULL

tempdat = subset(nebraskadat, select = c(UID, iso2, iso3, code3, FIPS, Admin2, Province_State, Country_
names(test) <- c("Admin2", "case_num_sum")
merged <- merge(test, tempdat)
```

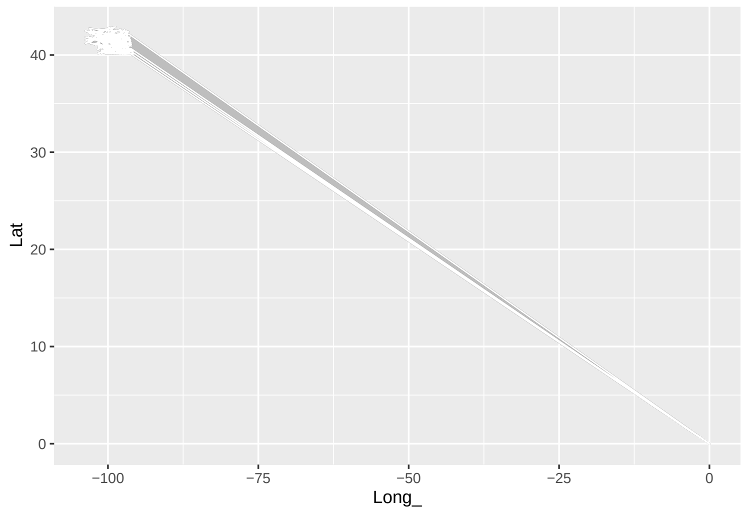
```
library(maps)
head(nebraskadat)
```

```
##         UID iso2 iso3 code3  FIPS   Admin2 Province_State Country_Region
## 1: 84031001   US  USA   840 31001    Adams       Nebraska             US
## 2: 84031003   US  USA   840 31003 Antelope       Nebraska             US
## 3: 84031005   US  USA   840 31005   Arthur       Nebraska             US
## 4: 84031007   US  USA   840 31007   Banner       Nebraska             US
## 5: 84031009   US  USA   840 31009   Blaine       Nebraska             US
## 6: 84031011   US  USA   840 31011    Boone       Nebraska             US
##         Lat      Long_           Combined_Key 1/22/20 1/23/20 1/24/20 1/25/20
## 1: 40.52449  -98.50118    Adams, Nebraska, US       0       0       0       0
## 2: 42.17696  -98.06663 Antelope, Nebraska, US       0       0       0       0
## 3: 41.56896 -101.69596   Arthur, Nebraska, US       0       0       0       0
## 4: 41.54634 -103.71143   Banner, Nebraska, US       0       0       0       0
## 5: 41.91312  -99.97678   Blaine, Nebraska, US       0       0       0       0
## 6: 41.70759  -98.06737    Boone, Nebraska, US       0       0       0       0
##    1/26/20 1/27/20 1/28/20 1/29/20 1/30/20 1/31/20 2/1/20 2/2/20 2/3/20 2/4/20
## 1:       0       0       0       0       0       0      0      0      0      0
## 2:       0       0       0       0       0       0      0      0      0      0
## 3:       0       0       0       0       0       0      0      0      0      0
## 4:       0       0       0       0       0       0      0      0      0      0
## 5:       0       0       0       0       0       0      0      0      0      0
## 6:       0       0       0       0       0       0      0      0      0      0
##    2/5/20 2/6/20 2/7/20 2/8/20 2/9/20 2/10/20 2/11/20 2/12/20 2/13/20 2/14/20
## 1:      0      0      0      0      0       0       0       0       0       0
## 2:      0      0      0      0      0       0       0       0       0       0
## 3:      0      0      0      0      0       0       0       0       0       0
## 4:      0      0      0      0      0       0       0       0       0       0
## 5:      0      0      0      0      0       0       0       0       0       0
## 6:      0      0      0      0      0       0       0       0       0       0
##    2/15/20 2/16/20 2/17/20 2/18/20 2/19/20 2/20/20 2/21/20 2/22/20 2/23/20
## 1:       0       0       0       0       0       0       0       0       0
## 2:       0       0       0       0       0       0       0       0       0
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       0       0       0       0       0       0       0       0       0
##    2/24/20 2/25/20 2/26/20 2/27/20 2/28/20 2/29/20 3/1/20 3/2/20 3/3/20 3/4/20
## 1:       0       0       0       0       0       0      0      0      0      0
## 2:       0       0       0       0       0       0      0      0      0      0
## 3:       0       0       0       0       0       0      0      0      0      0
## 4:       0       0       0       0       0       0      0      0      0      0
## 5:       0       0       0       0       0       0      0      0      0      0
## 6:       0       0       0       0       0       0      0      0      0      0
##    3/5/20 3/6/20 3/7/20 3/8/20 3/9/20 3/10/20 3/11/20 3/12/20 3/13/20 3/14/20
## 1:      0      0      0      0      0       0       0       0       0       0
## 2:      0      0      0      0      0       0       0       0       0       0
## 3:      0      0      0      0      0       0       0       0       0       0
## 4:      0      0      0      0      0       0       0       0       0       0
## 5:      0      0      0      0      0       0       0       0       0       0
## 6:      0      0      0      0      0       0       0       0       0       0
##    3/15/20 3/16/20 3/17/20 3/18/20 3/19/20 3/20/20 3/21/20 3/22/20 3/23/20
## 1:       0       0       0       0       1       1       1       1       1
## 2:       0       0       0       0       0       0       0       0       0
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       0       0       0       0       0       0       0       0       0
##    3/24/20 3/25/20 3/26/20 3/27/20 3/28/20 3/29/20 3/30/20 3/31/20 4/1/20
## 1:       1       1       1       1       2       2       3       3      4
## 2:       0       0       0       0       0       0       0       1      1
## 3:       0       0       0       0       0       0       0       0      0
## 4:       0       0       0       0       0       0       0       0      0
## 5:       0       0       0       0       0       0       0       0      0
## 6:       0       0       0       0       0       0       0       0      0
##    4/2/20 4/3/20 4/4/20 4/5/20 4/6/20 4/7/20 4/8/20 4/9/20 4/10/20 4/11/20
## 1:      6      6      8     11     13     15     27     28      38      45
## 2:      1      1      1      1      1      1      1      1       1       1
## 3:      0      0      0      0      0      0      0      0       0       0
## 4:      0      0      0      0      0      0      0      0       0       0
## 5:      0      0      0      0      0      0      0      0       0       0
## 6:      0      0      0      0      0      0      0      0       0       0
##    4/12/20 4/13/20 4/14/20 4/15/20 4/16/20 4/17/20 4/18/20 4/19/20 4/20/20
## 1:      54      54      62      62      65      64      85      98     100
## 2:       1       1       1       1       1       1       1       1       1
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       0       0       0       0       0       0       0       0       0
##    4/21/20 4/22/20 4/23/20 4/24/20 4/25/20 4/26/20 4/27/20 4/28/20 4/29/20
## 1:     101     103     111     120     125     129     141     143     147
## 2:       1       1       1       1       2       2       2       2       2
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       0       0       0       0       0       0       0       0       0
##    4/30/20 5/1/20 5/2/20 5/3/20 5/4/20 5/5/20 5/6/20 5/7/20 5/8/20 5/9/20
## 1:     164    187    202    203    204    204    212    216    218    223
## 2:       2      3      3      3      4      4      4      4      5      5
## 3:       0      0      0      0      0      0      0      0      0      0
## 4:       0      0      0      0      0      0      0      0      0      0
## 5:       0      0      0      0      0      0      0      0      0      0
## 6:       0      0      0      0      0      2      2      2      2      2
##    5/10/20 5/11/20 5/12/20 5/13/20 5/14/20 5/15/20 5/16/20 5/17/20 5/18/20
## 1:     223     227     228     236     238     246     251     252     252
## 2:       5       5       5       5       5       5       5       5       6
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       2       2       2       2       3       3       3       3       3
##    5/19/20 5/20/20 5/21/20 5/22/20 5/23/20 5/24/20 5/25/20 5/26/20 5/27/20
## 1:     255     256     258     258     264     265     265     265     264
## 2:       7       7       7       7       7       7       7       8       8
## 3:       0       0       0       0       0       0       0       0       0
## 4:       0       0       0       0       0       0       0       0       0
## 5:       0       0       0       0       0       0       0       0       0
## 6:       3       3       3       3       3       3       3       3       3
##    5/28/20 5/29/20 5/30/20 5/31/20 6/1/20 6/2/20 6/3/20 6/4/20 6/5/20 6/6/20
## 1:     265     268     270     270    272    272    273    275    275    275
## 2:       8       8       8       8      8      8      8      8      8      8
## 3:       0       0       0       0      0      0      0      0      0      0
## 4:       0       0       0       0      0      0      0      0      0      0
## 5:       0       0       0       0      0      0      0      0      0      0
## 6:       4       5       6       6      6      6      6      7      7      7
##    6/7/20 6/8/20 6/9/20 6/10/20 6/11/20 6/12/20 6/13/20 6/14/20 6/15/20 6/16/20
## 1:    275    276    277     277     277     278     278     278     278     278
## 2:      8      8      8       8       8       8       8       8       8       8
## 3:      0      0      0       0       0       0       0       0       0       0
## 4:      0      0      0       0       1       1       1       1       1       1
## 5:      0      0      0       0       0       0       0       0       0       0
## 6:      7      7      7       7       7       7       7       7       7       7
##    6/17/20 6/18/20 6/19/20 6/20/20 6/21/20 6/22/20 6/23/20 6/24/20 6/25/20
## 1:     278     278     280     281     282     282     282     283     285
## 2:       8       8       8       8       8       8       8       9       9
## 3:       0       0       0       0       0       0       0       0       0
## 4:       1       1       1       1       1       1       1       1       1
## 5:       0       0       0       0       0       0       0       0       0
## 6:       7       7       7       7       7       7       7       7       7
##    6/26/20 6/27/20 6/28/20 6/29/20 6/30/20 7/1/20 7/2/20 7/3/20 7/4/20 7/5/20
## 1:     287     287     288     292     294    296    296    298    299    299
## 2:       9       9       9       9       9      9      9      9      9      9
## 3:       0       0       0       0       0      0      0      0      0      0
## 4:       1       1       1       1       1      1      1      2      2      2
## 5:       0       0       0       0       0      0      0      0      0      0
## 6:        7        7        7        7         7         7         7         7         7         7
##    7/6/20 7/7/20 7/8/20 7/9/20 7/10/20 7/11/20 7/12/20 7/13/20 7/14/20 7/15/20
## 1:    301    302    304    309     309     312     312     314     315     315
## 2:      9      9      9     11      11      12      15      16      16      16
## 3:      0      0      0      0       0       0       0       0       0       0
## 4:      2      2      2      2       2       2       2       2       2       2
## 5:      0      0      0      0       0       0       0       0       0       0
## 6:      7      7      7      7       7       7       7       7       7       7
##    7/16/20 7/17/20 7/18/20 7/19/20 7/20/20 7/21/20 7/22/20 7/23/20 7/24/20
## 1:     315     316     316     316     320     320     320     320     323
## 2:      16      16      16      16      16      17      17      17      17
## 3:       0       0       0       0       0       0       0       0       0
## 4:       2       2       2       2       2       2       2       2       2
## 5:       0       0       0       0       0       0       0       0       0
## 6:       7       7       7       7       7       7       7       7       7
##    7/25/20 7/26/20 7/27/20 7/28/20 7/29/20 7/30/20 7/31/20 8/1/20 8/2/20 8/3/20
## 1:     324     326     328     329     331     336     339    340    341    344
## 2:      17      17      17      18      18      18      18     18     18     19
## 3:       0       0       0       0       0       0       0      0      0      0
## 4:       2       2       2       2       2       2       2      2      2      2
## 5:       0       0       0       0       0       0       0      0      0      0
## 6:       7       7       7       7       7       7       7      7      7      8
##    8/4/20 8/5/20 8/6/20 8/7/20 8/8/20 8/9/20 8/10/20 8/11/20 8/12/20 8/13/20
## 1:    346    346    350    353    353    353     358     361     362     363
## 2:     19     19     19     19     19     19      19      19      21      20
## 3:      0      0      1      1      1      1       1       1       1       1
## 4:      2      2      2      3      3      3       3       2       2       2
## 5:      0      0      0      0      0      0       0       0       0       0
## 6:      7      7      7      8      8      8       8       8       8      10
##    8/14/20 8/15/20 8/16/20 8/17/20 8/18/20 8/19/20 8/20/20 8/21/20 8/22/20
## 1:     364     364     365     370     370     375     379     381     381
## 2:      20      21      21      21      21      20      21      21      22
## 3:       1       1       1       1       1       1       1       1       1
## 4:       2       2       2       2       2       2       2       3       3
## 5:       0       0       0       0       0       0       0       0       0
## 6:      11      11      11      14      15      16      16      16      16
##    8/23/20 8/24/20 8/25/20 8/26/20 8/27/20 8/28/20 8/29/20 8/30/20 8/31/20
## 1:     381     384     387     392     396     399     401     403     410
## 2:      23      23      23      23      24      24      24      25      25
## 3:       1       1       1       1       1       1       1       1       1
## 4:       3       3       3       2       2       2       2       2       2
## 5:       0       0       0       0       0       0       0       0       0
## 6:      16      16      18      20      21      22      22      24      26
##    9/1/20 9/2/20 9/3/20 9/4/20 9/5/20 9/6/20 9/7/20 9/8/20 9/9/20 9/10/20
## 1:    416    419    428    429    429    430    430    436    441     448
## 2:     25     27     27     28     28     28     28     29     32      34
## 3:      1      1      1      1      1      1      1      1      1       1
## 4:      2      2      2      2      2      2      2      2      2       2
## 5:      0      0      0      0      0      0      0      0      0       0
## 6:     26     28     29     30     32     32     33     35     38      42
##    9/11/20 9/12/20 9/13/20 9/14/20 9/15/20 9/16/20 9/17/20 9/18/20 9/19/20
## 1:     453     458     458     469     480     482     491     499     506
## 2:      35      36      36      38      41      44      46      48      53
## 3:       1       1       1       1       1       1       1       1       1
## 4:        2       2       2       2       2       2       2       2       2
## 5:        0       0       0       0       0       0       0       0       0
## 6:       43      45      45      46      50      52      57      60      63
##     9/20/20 9/21/20 9/22/20 9/23/20 9/24/20 9/25/20 9/26/20 9/27/20 9/28/20
## 1:      508     510     514     520     534     542     543     548     558
## 2:       55      59      64      65      67      67      68      69      76
## 3:        1       1       1       1       1       1       1       1       1
## 4:        2       2       2       2       2       2       2       2       2
## 5:        0       0       0       0       0       1       1       1       1
## 6:       63      63      71      73      75      78      78      82      85
##     9/29/20 9/30/20 10/1/20 10/2/20 10/3/20 10/4/20 10/5/20 10/6/20 10/7/20
## 1:      570     574     580     586     596     596     607     613     617
## 2:       79      80      94      97      99     100     102     104     105
## 3:        1       1       1       1       1       1       1       1       1
## 4:        2       2       2       2       2       2       2       2       2
## 5:        1       1       1       1       1       1       1       1       1
## 6:       85      88      88      93      97      97      98      98     107
##     10/8/20 10/9/20 10/10/20 10/11/20 10/12/20 10/13/20 10/14/20 10/15/20
## 1:      625     625      649      650      655      662      681      691
## 2:      113     113      123      128      128      128      131      134
## 3:        1       1        2        3        3        3        3        3
## 4:        2       2        2        2        2        2        2        2
## 5:        1       1        1        1        2        3        3        3
## 6:      109     109      114      113      114      117      120      121
##     10/16/20 10/17/20 10/18/20 10/19/20 10/20/20 10/21/20 10/22/20 10/23/20
## 1:       698      704      709      733      754      760      778      803
## 2:       135      135      139      139      141      142      147      153
## 3:         3        3        3        3        3        3        3        3
## 4:         2        2        2        2        3        3        3        4
## 5:         3        3        3        3        2        2        2        2
## 6:       124      124      127      129      131      133      136      136
##     10/24/20 10/25/20 10/26/20 10/27/20 10/28/20 10/29/20 10/30/20 10/31/20
## 1:       808      814      839      861      878      896      912      913
## 2:       155      157      160      164      166      171      173      176
## 3:         4        4        4        4        4        4        5        5
## 4:         4        4        4        4        4        4        4        4
## 5:         2        2        2        2        2        2        3        3
## 6:       140      143      145      147      153      157      159      168
##     11/1/20 11/2/20 11/3/20 11/4/20 11/5/20 11/6/20 11/7/20 11/8/20 11/9/20
## 1:      922     932     955     979    1010    1031    1050    1066    1082
## 2:      182     183     187     191     203     204     210     216     222
## 3:        5       5       5       6       6       6       6       6       8
## 4:        4       4       4       4       4       5       5       5       5
## 5:        4       4       4       4       4       5       6       7       8
## 6:      170     171     177     180     186     197     203     210     211
##     11/10/20 11/11/20 11/12/20 11/13/20 11/14/20 11/15/20 11/16/20 11/17/20
## 1:      1105     1130     1164     1198     1223     1235     1246     1305
## 2:       229      233      238      244      250      252      262      263
## 3:         9       10       12       14       14       14       14       13
## 4:         5        5        6        6        7        7        9       10
## 5:         8        8       10       10       10       10       10       11
## 6:       217      223      225      233      238      241      249      253
##     11/18/20 11/19/20 11/20/20 11/21/20 11/22/20 11/23/20 11/24/20 11/25/20
## 1:      1336     1365     1449     1463     1480     1517     1517     1548
## 2:       281      290      291      297      301      306      306      306
## 3:        13       13       14       14       14       15       15       15
## 4:        11       12       13       13       13       13       13       13
## 5:        13       14       14       14       15       15       15       15
## 6:       271      272      278      281      285      287      287      294
##    11/26/20 11/27/20 11/28/20 11/29/20 11/30/20 12/1/20 12/2/20 12/3/20 12/4/20
## 1:     1586     1599     1609     1621     1658    1730    1780    1829    1872
## 2:      324      327      330      331      334     337     342     351     357
## 3:       16       16       16       16       16      16      16      16      16
## 4:       16       16       16       16       16      16      16      16      16
## 5:       15       15       15       15       15      15      15      15      15
## 6:      311      315      317      319      318     328     338     344     348
##    12/5/20 12/6/20 12/7/20 12/8/20 12/9/20 12/10/20 12/11/20 12/12/20 12/13/20
## 1:    1892    1903    1942    1975    1999     2032     2034     2047     2060
## 2:     359     361     365     367     373      376      376      380      383
## 3:      16      16      16      16      16       16       16       16       16
## 4:      16      17      17      17      17       18       20       20       20
## 5:      15      15      15      15      15       14       15       15       15
## 6:     354     359     361     367     367      373      381      380      382
##    12/14/20 12/15/20 12/16/20 12/17/20 12/18/20 12/19/20 12/20/20 12/21/20
## 1:     2075     2104     2120     2128     2139     2140     2151     2157
## 2:      382      386      385      391      407      408      412      415
## 3:       16       16       16       16       17       17       17       17
## 4:       20       21       23       23       23       23       23       23
## 5:       15       16       16       15       15       15       15       14
## 6:      381      384      385      390      393      393      394      395
##    12/22/20 12/23/20 12/24/20 12/25/20 12/26/20 12/27/20 12/28/20 12/29/20
## 1:     2177     2192     2197     2208     2209     2215     2219     2230
## 2:      416      419      421      421      422      422      424      424
## 3:       17       17       17       17       17       17       17       17
## 4:       24       25       25       25       25       25       25       26
## 5:       14       15       15       15       15       15       15       15
## 6:      400      404      405      409      410      411      410      412
##    12/30/20 12/31/20 1/1/21 1/2/21 1/3/21 1/4/21 1/5/21 1/6/21 1/7/21 1/8/21
## 1:     2271     2285   2297   2300   2306   2318   2341   2362   2389   2396
## 2:      427      427    427    427    429    432    435    437    438    437
## 3:       18       18     18     18     18     18     18     18     18     18
## 4:       26       26     26     26     26     26     26     26     26     26
## 5:       15       16     16     16     16     16     16     16     16     17
## 6:      416      418    421    422    424    424    431    440    439    442
##    1/9/21 1/10/21 1/11/21 1/12/21 1/13/21 1/14/21 1/15/21 1/16/21 1/17/21
## 1:   2405    2406    2423    2436    2451    2464    2479    2479    2507
## 2:    438     438     439     440     447     448     448     448     449
## 3:     18      18      18      18      18      18      18      18      18
## 4:     26      26      26      26      26      26      26      26      26
## 5:     18      17      17      18      18      18      18      18      18
## 6:    444     444     442     445     448     453     454     454     459
##    1/18/21 1/19/21 1/20/21 1/21/21 1/22/21 1/23/21 1/24/21 1/25/21 1/26/21
## 1:    2508    2508    2532    2537    2555    2563    2565    2570    2577
## 2:     451     451     453     453     454     454     456     458     459
## 3:      18      18      18      19      19      21      21      21      21
## 4:      26      26      26      27      27      27      27      27      27
## 5:      18      18      19      19      19      19      19      19      19
## 6:     459     459     464     466     466     467     470     470     470
##    1/27/21 1/28/21 1/29/21 1/30/21 1/31/21 2/1/21 2/2/21 2/3/21 2/4/21 2/5/21
## 1:    2594    2612    2612    2623    2623   2627   2636   2638   2641   2648
## 2:     458     461     461     461     461    461    462    462    461    461
## 3:      21      22      22      22      22     22     22     22     22     22
## 4:      27      28      28      28      28     28     30     30     30     30
## 5:      19      19      19      19      19     19     19     19     19     19
## 6:     473     472     472     474     474    474    474    475    476    477
##    2/6/21 2/7/21 2/8/21 2/9/21 2/10/21 2/11/21 2/12/21 2/13/21 2/14/21 2/15/21
## 1:   2649   2652   2652   2654    2660    2667    2669    2675    2678    2678
## 2:    461    461    461    462     461     461     462     463     464     465
## 3:     22     22     22     22      22      22      22      22      22      22
## 4:     30     30     30     31      31      31      31      31      31      31
## 5:     19     19     19     19      19      19      19      19      19      19
## 6:    479    479    480    479     482     482     482     498     498     498
##    2/16/21 2/17/21 2/18/21 2/19/21 2/20/21 2/21/21 2/22/21 2/23/21 2/24/21
## 1:    2679    2680    2683    2690    2689    2689    2690    2693    2698
## 2:     465     465     465     465     465     466     466     468     468
## 3:      22      22      22      22      22      22      22      22      22
## 4:      31      31      31      31      31      31      31      31      31
## 5:      19      19      19      19      19      19      19      19      19
## 6:     498     497     497     497     497     497     497     497     497
##    2/25/21 2/26/21 2/27/21 2/28/21 3/1/21 3/2/21 3/3/21 3/4/21 3/5/21 3/6/21
## 1:    2702    2703    2704    2711   2711   2714   2716   2720   2722   2722
## 2:     470     470     471     473    473    476    477    477    478    480
## 3:      22      22      22      22     22     22     22     22     22     22
## 4:      31      31      31      32     32     34     35     35     35     35
## 5:      19      19      19      19     19     19     19     19     20     20
## 6:     500     500     500     500    500    500    507    511    511    512
##    3/7/21 3/8/21 3/9/21 3/10/21 3/11/21 3/12/21 3/13/21 3/14/21 3/15/21 3/16/21
## 1:   2722   2723   2724    2724    2739    2758    2764    2766    2766    2764
## 2:    481    481    481     481     482     482     482     483     487     486
## 3:     22     22     22      22      22      22      22      22      22      22
## 4:     35     35     35      35      35      35      35      35      35      35
## 5:     20     20     20      20      20      20      20      20      20      20
## 6:    520    515    515     515     514     519     520     522     524     532
##    3/17/21 3/18/21 3/19/21 3/20/21 3/21/21 3/22/21 3/23/21 3/24/21 3/25/21
## 1:    2771    2771    2771    2775    2774    2775    2781    2783    2785
## 2:     486     485     485     485     485     485     485     485     486
## 3:      22      22      22      22      22      22      22      22      22
## 4:      35      35      35      35      35      35      35      35      35
## 5:      20      20      20      20      20      20      20      20      20
## 6:     526     529     533     542     544     544     537     537     538
##    3/26/21 3/27/21 3/28/21 3/29/21 3/30/21 3/31/21 4/1/21 4/2/21 4/3/21 4/4/21
## 1:    2787    2795    2794    2795    2796    2812   2824   2828   2828   2834
## 2:     486     487     487     487     487     492    495    495    497    500
## 3:      22      22      22      22      22      22     22     22     22     22
## 4:      35      35      35      35      35      35     35     35     35     35
## 5:      20      20      20      20      20      20     20     20     20     20
## 6:     548     543     552     554     555     559    555    565    567    567
##    4/5/21 4/6/21 4/7/21 4/8/21 4/9/21 4/10/21 4/11/21 4/12/21 4/13/21 4/14/21
## 1:   2834   2836   2842   2844   2848    2852    2852    2853    2857    2861
## 2:    500    500    502    505    507     510     511     513     513     514
## 3:     22     22     22     22     22      22      22      22      22      22
## 4:     35     35     35     35     35      35      35      35      35      35
## 5:     20     20     20     20     20      20      20      20      20      20
## 6:    568    568    569    572    574     576     576     576     576     576
##     4/15/21 4/16/21 4/17/21 4/18/21 4/19/21 4/20/21 4/21/21 4/22/21 4/23/21
## 1:    2859    2867    2870    2872    2876    2876    2879    2880    2882
## 2:     516     516     516     519     519     520     520     522     523
## 3:      22      22      22      22      22      22      22      22      22
## 4:      35      35      35      35      35      35      35      35      35
## 5:      20      20      20      20      20      20      20      20      20
## 6:     578     578     577     577     577     577     577     577     577
##     4/24/21 4/25/21 4/26/21 4/27/21 4/28/21 4/29/21 4/30/21 5/1/21 5/2/21 5/3/21
## 1:    2889    2891    2892    2897    2898    2898    2899   2905   2905   2907
## 2:     524     524     524     524     524     525     525    525    525    525
## 3:      22      22      22      22      22      22      22     22     22     22
## 4:      35      35      35      35      35      35      35     35     35     35
## 5:      20      20      20      20      20      20      20     20     20     20
## 6:     578     579     579     580     580     583     583    584    584    584
##     5/4/21 5/5/21 5/6/21 5/7/21 5/8/21 5/9/21 5/10/21 5/11/21 5/12/21 5/13/21
## 1:   2908   2913   2916   2925   2929   2930    2933    2959    2966    2970
## 2:    525    525    525    525    525    525     525     532     533     534
## 3:     22     22     22     22     22     22      22      22      22      22
## 4:     35     35     35     36     36     36      36      36      36      36
## 5:     20     21     21     21     21     21      21      21      21      21
## 6:    584    585    585    584    584    584     584     584     584     584
##     5/14/21 5/15/21 5/16/21 5/17/21 5/18/21 5/19/21 5/20/21 5/21/21 5/22/21
## 1:    2971    2971    2971    2972    2975    2981    2988    2990    2990
## 2:     534     534     534     535     535     535     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##     5/23/21 5/24/21 5/25/21 5/26/21 5/27/21 5/28/21 5/29/21 5/30/21 5/31/21
## 1:    2990    2992    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##     6/1/21 6/2/21 6/3/21 6/4/21 6/5/21 6/6/21 6/7/21 6/8/21 6/9/21 6/10/21
## 1:   2993   2993   2993   2993   2993   2993   2993   2993   2993    2993
## 2:    536    536    536    536    536    536    536    536    536     536
## 3:     22     22     22     22     22     22     22     22     22      22
## 4:     36     36     36     36     36     36     36     36     36      36
## 5:     21     21     21     21     21     21     21     21     21      21
## 6:    584    584    584    584    584    584    584    584    584     584
##     6/11/21 6/12/21 6/13/21 6/14/21 6/15/21 6/16/21 6/17/21 6/18/21 6/19/21
## 1:    2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##     6/20/21 6/21/21 6/22/21 6/23/21 6/24/21 6/25/21 6/26/21 6/27/21 6/28/21
## 1:    2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##    6/29/21 6/30/21 7/1/21 7/2/21 7/3/21 7/4/21 7/5/21 7/6/21 7/7/21 7/8/21
## 1:    2993    2993   2993   2993   2993   2993   2993   2993   2993   2993
## 2:     536     536    536    536    536    536    536    536    536    536
## 3:      22      22     22     22     22     22     22     22     22     22
## 4:      36      36     36     36     36     36     36     36     36     36
## 5:      21      21     21     21     21     21     21     21     21     21
## 6:     584     584    584    584    584    584    584    584    584    584
##    7/9/21 7/10/21 7/11/21 7/12/21 7/13/21 7/14/21 7/15/21 7/16/21 7/17/21
## 1:   2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:    536     536     536     536     536     536     536     536     536
## 3:     22      22      22      22      22      22      22      22      22
## 4:     36      36      36      36      36      36      36      36      36
## 5:     21      21      21      21      21      21      21      21      21
## 6:    584     584     584     584     584     584     584     584     584
##    7/18/21 7/19/21 7/20/21 7/21/21 7/22/21 7/23/21 7/24/21 7/25/21 7/26/21
## 1:    2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##    7/27/21 7/28/21 7/29/21 7/30/21 7/31/21 8/1/21 8/2/21 8/3/21 8/4/21 8/5/21
## 1:    2993    2993    2993    2993    2993   2993   2993   2993   2993   2993
## 2:     536     536     536     536     536    536    536    536    536    536
## 3:      22      22      22      22      22     22     22     22     22     22
## 4:      36      36      36      36      36     36     36     36     36     36
## 5:      21      21      21      21      21     21     21     21     21     21
## 6:     584     584     584     584     584    584    584    584    584    584
##    8/6/21 8/7/21 8/8/21 8/9/21 8/10/21 8/11/21 8/12/21 8/13/21 8/14/21 8/15/21
## 1:   2993   2993   2993   2993    2993    2993    2993    2993    2993    2993
## 2:    536    536    536    536     536     536     536     536     536     536
## 3:     22     22     22     22      22      22      22      22      22      22
## 4:     36     36     36     36      36      36      36      36      36      36
## 5:     21     21     21     21      21      21      21      21      21      21
## 6:    584    584    584    584     584     584     584     584     584     584
##    8/16/21 8/17/21 8/18/21 8/19/21 8/20/21 8/21/21 8/22/21 8/23/21 8/24/21
## 1:    2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##    8/25/21 8/26/21 8/27/21 8/28/21 8/29/21 8/30/21 8/31/21 9/1/21 9/2/21 9/3/21
## 1:    2993    2993    2993    2993    2993    2993    2993   2993   2993   2993
## 2:     536     536     536     536     536     536     536    536    536    536
## 3:      22      22      22      22      22      22      22     22     22     22
## 4:      36      36      36      36      36      36      36     36     36     36
## 5:      21      21      21      21      21      21      21     21     21     21
## 6:     584     584     584     584     584     584     584    584    584    584
##    9/4/21 9/5/21 9/6/21 9/7/21 9/8/21 9/9/21 9/10/21 9/11/21 9/12/21 9/13/21
## 1:   2993   2993   2993   2993   2993   2993    2993    2993    2993    2993
## 2:    536    536    536    536    536    536     536     536     536     536
## 3:     22     22     22     22     22     22      22      22      22      22
## 4:     36     36     36     36     36     36      36      36      36      36
## 5:     21     21     21     21     21     21      21      21      21      21
## 6:    584    584    584    584    584    584     584     584     584     584
##    9/14/21 9/15/21 9/16/21 9/17/21 9/18/21 9/19/21 9/20/21 9/21/21 9/22/21
## 1:    2993    2993    2993    2993    2993    2993    2993    2993    2993
## 2:     536     536     536     536     536     536     536     536     536
## 3:      22      22      22      22      22      22      22      22      22
## 4:      36      36      36      36      36      36      36      36      36
## 5:      21      21      21      21      21      21      21      21      21
## 6:     584     584     584     584     584     584     584     584     584
##    9/23/21 9/24/21 9/25/21 9/26/21 9/27/21 9/28/21 9/29/21 9/30/21 10/1/21
## 1:    2993    2993    3595    3595    3629    3658    3670    3705    3716
## 2:     536     536     624     624     634     636     636     640     645
## 3:      22      22      24      24      24      24      24      24      24
## 4:      36      36      43      43      43      45      44      44      45
## 5:      21      21      31      31      31      32      36      36      36
## 6:     584     584     656     656     663     664     664     667     666
##    10/2/21 10/3/21 10/4/21 10/5/21 10/6/21 10/7/21 10/8/21 10/9/21 10/10/21
## 1:    3716    3716    3737    3759    3773    3791    3799    3799     3799
## 2:     645     645     651     652     656     656     659     659      659
## 3:      24      24      24      24      24      24      24      24       24
## 4:      45      45      46      46      47      48      50      50       50
## 5:      36      36      36      37      38      38      38      38       38
## 6:     666     666     671     671     671     675     679     679      679
##    10/11/21 10/12/21 10/13/21 10/14/21 10/15/21 10/16/21 10/17/21 10/18/21
## 1:     3799     3820     3825     3829     3837     3837     3837     3851
## 2:      659      670      675      676      676      676      676      682
## 3:       24       24       24       24       24       24       24       24
## 4:       50       51       52       52       53       53       53       54
## 5:       38       38       38       38       38       38       38       38
## 6:      679      682      685      685      688      688      688      693
##    10/19/21 10/20/21 10/21/21 10/22/21 10/23/21 10/24/21 10/25/21 10/26/21
## 1:     3856     3875     3885     3885     3885     3885     3885     3885
## 2:      686      695      707      707      707      707      707      707
## 3:       24       25       25       25       25       25       25       25
## 4:       54       54       54       54       54       54       54       54
## 5:       38       38       38       38       38       38       38       38
## 6:      693      695      700      700      700      700      700      700
##    10/27/21 10/28/21 10/29/21 10/30/21 10/31/21 11/1/21 11/2/21 11/3/21 11/4/21
## 1:     3930     3930     3930     3930     3930    3930    3930    4004    4004
## 2:      716      716      716      716      716     716     716     734     734
## 3:       25       25       25       25       25      25      25      25      25
## 4:       58       58       58       58       58      58      58      60      60
## 5:       39       39       39       39       39      39      39      40      40
## 6:      708      708      708      708      708     708     708     730     730
##    11/5/21 11/6/21 11/7/21 11/8/21 11/9/21 11/10/21 11/11/21 11/12/21 11/13/21
## 1:    4004    4004    4004    4065    4094     4126     4141     4152     4152
## 2:     734     734     734     746     746      749      750      751      751
## 3:      25      25      25      25      25       25       25       25       25
## 4:      60      60      60      60      60       60       61       62       62
## 5:      40      40      40      40      40       40       41       41       41
## 6:      730      730      730      750      754       761       765       767       767
##    11/14/21 11/15/21 11/16/21 11/17/21 11/18/21 11/19/21 11/20/21
## 1:     4152     4177     4218     4239     4266     4292     4292
## 2:      751      754      755      758      761      763      763
## 3:       25       25       25       25       25       25       25
## 4:       62       63       63       63       65       66       66
## 5:       41       43       43       43       44       44       44
## 6:      767      777      778      782      785      787      787
```

```
states <- map_data("state")
ne_coords <- subset(states, region=="nebraska")
head(ne_coords)
```

```
##            long      lat group order   region subregion
## 8359 -104.0606 43.00621    29  8359 nebraska      <NA>
## 8360 -103.5106 42.99475    29  8360 nebraska      <NA>
## 8361 -103.0063 42.99475    29  8361 nebraska      <NA>
## 8362 -102.7944 42.99475    29  8362 nebraska      <NA>
## 8363 -102.0782 42.99475    29  8363 nebraska      <NA>
## 8364 -101.2302 42.99475    29  8364 nebraska      <NA>
```

```
counties <- map_data("county")
ne_county <- subset(counties, region=="nebraska")
nebraskadat$group <- ne_coords$group[1:95]
ne_map <- ggplot(data=merged, mapping = aes(x=Long_, y=Lat))  + geom_polygon(color="black", fill="gray"
ne_map + geom_polygon(data = merged, fill=NA, color="white")
```


Lat
Long_