Managing aspatial data in R

Taylor Saunders

2024-03-13

Read in WHO csv data

Use setwd command to set the working directory

```
setwd("C:/Users/taylo/OneDrive/Desktop/GIS_470/Module_0")
```

```
my_first_df <- read.csv("WHO.csv", head=TRUE)
```

Exploring the dataframe

```
head(my_first_df)
```

```
##               country region LDC literacy   GNI totpop pcturban pctpov TFR
## 1             Albania      5   0     98.7  6000   3172       46    1.5 2.1
## 2             Algeria      1   0     69.9  5940  33351       64      . 2.4
## 3              Angola      1   1     67.4  3890  16557       54      . 6.5
## 4 Antigua and Barbuda      2   0        . 15130     84       37      . 2.2
## 5       Arab Emirates      4   0     88.5 31190   4248       77      . 2.3
## 6           Argentina      2   0     97.2 11670  39134       90    6.6 2.3
##   totexphlth totmort f_mort m_mort IMR f_LE m_LE comm_YLL chron_YLL  EPI ACSAT
## 1        358     137    103    170  15   73   69       17        63   84    91
## 2        188     135    122    149  33   72   70       50        30   77    92
## 3         71     493    447    539 154   43   40       84         8 39.5    31
## 4        652     151    115    187  10   75   70       21        69    .     .
## 5        673      78     62     84   8   80   77       12        59   64    98
## 6       1665     124     86    162  14   78   72       18        66 81.8    91
##   WATSUP AGSUB WATST
## 1     96   6.2     0
## 2     85  55.9  24.5
## 3     53     0   5.5
## 4      .     .     .
## 5    100     0  41.6
## 6     96  13.7  24.1
```

```
str(my_first_df)
```

```
## 'data.frame':    180 obs. of  23 variables:
##  $ country   : chr  "Albania" "Algeria" "Angola" "Antigua and Barbuda" ...
##  $ region    : int  5 1 1 2 4 2 5 3 3 5 ...
##  $ LDC       : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ literacy  : chr  "98.7" "69.9" "67.4" "." ...
##  $ GNI       : int  6000 5940 3890 15130 31190 11670 4950 33940 36040 5430 ...
##  $ totpop    : int  3172 33351 16557 84 4248 39134 3010 20530 8327 8406 ...
##  $ pcturban  : int  46 64 54 37 77 90 64 88 66 52 ...
```

```
##  $ pctpov    : chr  "1.5" "." "." "." ...
##  $ TFR       : num  2.1 2.4 6.5 2.2 2.3 2.3 1.3 1.8 1.4 1.7 ...
##  $ totexphlth: int  358 188 71 652 673 1665 272 3122 3545 218 ...
##  $ totmort   : int  137 135 493 151 78 124 184 65 79 188 ...
##  $ f_mort    : int  103 122 447 115 62 86 115 47 51 138 ...
##  $ m_mort    : int  170 149 539 187 84 162 262 82 105 241 ...
##  $ IMR       : int  15 33 154 10 8 14 21 5 4 73 ...
##  $ f_LE      : int  73 72 43 75 80 78 72 84 83 66 ...
##  $ m_LE      : int  69 70 40 70 77 72 65 79 77 62 ...
##  $ comm_YLL  : chr  "17" "50" "84" "21" ...
##  $ chron_YLL : chr  "63" "30" "8" "69" ...
##  $ EPI       : chr  "84" "77" "39.5" "." ...
##  $ ACSAT     : chr  "91" "92" "31" "." ...
##  $ WATSUP    : chr  "96" "85" "53" "." ...
##  $ AGSUB     : chr  "6.2" "55.9" "0" "." ...
##  $ WATST     : chr  "0" "24.5" "5.5" "." ...
```

```
tail(my_first_df)
```

```
##       country region LDC literacy   GNI totpop pcturban pctpov TFR totexphlth
## 175   Vanuatu      6   1     75.5  3480    221       24       . 3.9        139
## 176 Venezuela      2   0       93 10970  27191       94   18.5 2.6        396
## 177  Viet Nam      6   0     90.3  2310  86206       27       . 2.2        264
## 178     Yemen      4   1     54.1  2090  21732       28       . 5.6         82
## 179    Zambia      1   1       68  1140  11696       35   63.8 5.3         62
## 180  Zimbabwe      1   1     89.5   507  13228       36       . 3.3        147
##     totmort f_mort m_mort IMR f_LE m_LE comm_YLL chron_YLL  EPI ACSAT WATSUP
## 175     187    166    207  30   70   67       39        51    .     .      .
## 176     142     95    187  18   78   71       24        45   80    68     83
## 177     155    116    194  15   75   69       40        44 73.9    61     85
## 178     250    217    282  75   62   59       61        28 49.7    43     67
## 179     617    597    644 102   43   42       92         6 55.1    55     58
## 180     751    755    755  55   43   44       90         7 69.3    53     81
##     AGSUB WATST
## 175     .     .
## 176   0.9   9.7
## 177  11.8     3
## 178  17.3  55.9
## 179   0.1   0.1
## 180   0.3  20.4
```

String values/vectors: contain words, letters, or any other character type Numerical values/vectors: hold different forms of numbers

Creating a string value:

```
name_value <- "Taylor"
```

Creating a numeric value:

```
age_value <- 21
```

```
print(name_value)
```

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

```
print(age_value)
```

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

```
print(age_value - 10)
```

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

```
print(age_value/2)
```

```
## [1] 10.5
```

Vectors

Vector: a column of numeric or string values

Creating a string vector:

```
#think of c as combine
name_vector <- c("Dylan", "Sarah", "Daniel")
print(name_vector)
```

```
## [1] "Dylan"  "Sarah"  "Daniel"
```

```
favorite_food <- c("Pasta", "Cheese", "Pizza")
```

Creating a numeric vector:

```
age_vector <- c(30, 15, 22)
print(age_vector)
```

```
## [1] 30 15 22
```

Data Frames

one variable, 3 observation data frame

```
df1 <- data.frame(name_vector,age_vector, favorite_food)
```

add a column/variable

```
df1$age_15 <- age_vector + 15
```

```
head(df1)
```

```
##   name_vector age_vector favorite_food age_15
## 1       Dylan         30         Pasta     45
## 2       Sarah         15        Cheese     30
## 3      Daniel         22         Pizza     37
```

Exercise: if my age is 21, how old will I be in 2025?

```
my_age <- 21
print(my_age + (2050 - 2024))
```

```
## [1] 47
```

Replacing DF values

Replace Sarah with Tom

```
name_vector[2] <- "Tom"
print(name_vector)
```

```
## [1] "Dylan"  "Tom"    "Daniel"
```

NEW DF

```
name_vector2 <- c("Dylan", "Sarah", "Lisa")
country_origin <- c("Ireland", "USA", "Mexico")
df2 <- data.frame(name_vector2, country_origin)
```

Merging

x is the larger data frame, typically y is the attributes we want to add into our primary df (x) by = unique characteristic used to join the two DF

```
#basic merge
df_merge <- merge(df1, df2, by.x = "name_vector", by.y = "name_vector2")
```

```
head(df_merge)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1       Dylan         30         Pasta     45        Ireland
## 2       Sarah         15        Cheese     30            USA
```

Only Dylan and Sarah are in both vectors used to merge

Solution:

```
# all = TRUE
#keep all observations
df4 <- merge(df1, df2, by.x = "name_vector", by.y = "name_vector2", all = TRUE)
```

```
head(df_merge)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1       Dylan         30         Pasta     45        Ireland
## 2       Sarah         15        Cheese     30            USA
```

```
#keep row structure of df1 and only add the columns that match in df2
df_merge <- merge(df1, df2, by.x = "name_vector", by.y = "name_vector2", all.x = TRUE)
```

```
head(df_merge)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1      Daniel         22         Pizza     37           <NA>
## 2       Dylan         30         Pasta     45        Ireland
## 3       Sarah         15        Cheese     30            USA
```

```
df_merge <- merge(df1, df2, by.x = "name_vector", by.y = "name_vector2", all.y = TRUE)

head(df_merge)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1       Dylan         30         Pasta     45        Ireland
## 2        Lisa         NA          <NA>     NA         Mexico
## 3       Sarah         15        Cheese     30            USA
```

Saving as CSV

```
write.csv(df_merge, "C:/Users/taylo/OneDrive/Desktop/GIS_470/Module_0/df_merge.csv")
```

reading in a csv

```
df_merge_read <- read.csv("C:/Users/taylo/OneDrive/Desktop/GIS_470/Module_0/df_merge.csv", head=T)
```

Descriptive Statistics

```
head(df4)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1      Daniel         22         Pizza     37           <NA>
## 2       Dylan         30         Pasta     45        Ireland
## 3        Lisa         NA          <NA>     NA         Mexico
## 4       Sarah         15        Cheese     30            USA
```

```
str(df4)
```

```
## 'data.frame':    4 obs. of  5 variables:
##  $ name_vector   : chr  "Daniel" "Dylan" "Lisa" "Sarah"
##  $ age_vector    : num  22 30 NA 15
##  $ favorite_food : chr  "Pizza" "Pasta" NA "Cheese"
##  $ age_15        : num  37 45 NA 30
##  $ country_origin: chr  NA "Ireland" "Mexico" "USA"
```

```
table(df4$name_vector)
```

```
##
## Daniel  Dylan   Lisa  Sarah
##      1      1      1      1
```

duplicating the df

```
#doubles
df4_dup <- rbind(df4, df4)
table(df4_dup$name_vector)
```

```
##
## Daniel  Dylan   Lisa  Sarah
##      2      2      2      2
```

Basic Statistics

```
print(df4_dup)
```

```
##   name_vector age_vector favorite_food age_15 country_origin
## 1      Daniel         22         Pizza     37           <NA>
## 2       Dylan         30         Pasta     45        Ireland
## 3        Lisa         NA          <NA>     NA         Mexico
## 4       Sarah         15        Cheese     30            USA
## 5      Daniel         22         Pizza     37           <NA>
## 6       Dylan         30         Pasta     45        Ireland
## 7        Lisa         NA          <NA>     NA         Mexico
## 8       Sarah         15        Cheese     30            USA
```

```
mean(df4_dup$age_vector)
```

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

We have missing values for Lisa Solution:

```r
mean(df4_dup$age_vector, na.rm=TRUE)
```

```
## [1] 22.33333
```

```r
max(df4_dup$age_vector, na.rm=TRUE)
```

```
## [1] 30
```

```r
min(df4_dup$age_vector, na.rm=TRUE)
```

```
## [1] 15
```

```r
#standard deviation
sd(df4_dup$age_vector, na.rm=TRUE)
```

```
## [1] 6.713171
```

```r
summary(df4_dup$age_vector)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   15.00   16.75   22.00   22.33   28.00   30.00       2
```

```r
#correlation
cor.test(df4_dup$age_vector, df4_dup$age_15)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df4_dup$age_vector and df4_dup$age_15
## t = Inf, df = 4, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  1 1
## sample estimates:
## cor
##   1
```