Practice2

Kishore Hari

2022-11-24

This is an example for cleaning data. Download the text file at this link: https://michaelgastner.com/DAVisR_data/homicides.txt. If you open it in a text editor like wordpad, you will see that there is a consistent structure. However, the delimiters are not consistent. Therefore, your task is to convert the text data into a dataframe. You can read the file using the function .

```
x <- read_lines("https://michaelgastner.com/DAVisR_data/homicides.txt")
```

a. *What is the class of x?*

```
class(x)
```

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

b. *What are the possible delimiters that you see in the data?*

Possible delimiters: comma, "<", ">", "<dd>", "</dd>", "<dl>", "</dl>". Note that a delimiter doesn't have to be a single character.

```
write_lines(x[1], file = stdout())
```

```
## 39.311024, -76.674227, iconHomicideShooting, 'p2', '<dl><dt>Leon
Nelson</dt><dd class="address">3400 Clifton Ave.<br />Baltimore, MD
21216</dd><dd>black male, 17 years old</dd><dd>Found on January 1,
2007</dd><dd>Victim died at Shock Trauma</dd><dd>Cause: shooting</dd></dl>'
```

c. *How many columns can the data be divided into, given that each column must only have one category of information?*

Minimum 5. I count 12 in total. But there can be more?

d. *Can you use comma as a delimiter to split the data? If you do so, will there by any inconsistencies?*

There are three ways to do this: using read_csv, read.csv with quote argument and splitting the strings based on commas, like we discussed in class. Let us look at the first two methods:

```
# Method 1: read_csv
df <- read_csv("https://michaelgastner.com/DAVisR_data/homicides.txt",
    col_names = F)
```

```
## Warning: One or more parsing issues, see ‘problems()‘ for details

## Rows: 1249 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): X3, X4, X5, X6, X7, X8
## dbl (2): X1, X2
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
ncol(df)
```

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

```
head(df, n = 10)
```

```
## # A tibble: 10 x 8
##       X1    X2 X3                       X4    X5              X6    X7    X8
##    <dbl> <dbl> <chr>                    <chr> <chr>           <chr> <chr> <chr>
##  1  39.3 -76.7 iconHomicideShooting     'p2'  "'<dl><dt>Leon ~ MD 2~ 17 y~ 2007~
##  2  39.3 -76.7 iconHomicideShooting     'p3'  "'<dl><dt>Eddie~ MD 2~ 26 y~ 2007~
##  3  39.3 -76.6 iconHomicideBluntForce   'p4'  "'<dl><dt>Nelse~ MD 2~ 44 y~ 2007~
##  4  39.4 -76.6 iconHomicideAsphyxiation 'p5'  "'<dl><dt>Thoma~ MD 2~ 21 y~ 2007~
##  5  39.2 -76.6 iconHomicideBluntForce   'p6'  "'<dl><dt>Edwar~ MD 2~ 61 y~ 2007~
##  6  39.4 -76.6 iconHomicideShooting     'p7'  "'<dl><dt>Micha~ MD 2~ 46 y~ 2007~
##  7  39.3 -76.6 iconHomicideShooting     'p8'  "'<dl><dt>Ray A~ MD 2~ 27 y~ 2007~
##  8  39.3 -76.6 iconHomicideShooting     'p9'  "'<dl><dt>Yule ~ MD 2~ 21 y~ 2007~
##  9  39.3 -76.6 iconHomicideShooting     'p10' "'<dl><dt>Marcu~ MD 2~ 16 y~ 2007~
## 10  39.3 -76.6 iconHomicideShooting     'p11' "'<dl><dt>Rodne~ MD 2~ 21 y~ 2007~
```

```
nrow(df)
```

```
## [1] 1249
```

```
tail(df, n = 10)
```

```
## # A tibble: 10 x 8
##       X1    X2 X3                         X4      X5           X6    X7    X8
##    <dbl> <dbl> <chr>                      <chr>   <chr>        <chr> <chr> <chr>
##  1  39.3 -76.7 icon_homicide_shooting     'p1232' "'<dl><dt><~ MD 2~ "201~ <NA>
##  2  39.3 -76.6 icon_homicide_shooting     'p1231' "'<dl><dt><~ MD 2~ "201~ <NA>
##  3  39.3 -76.7 icon_homicide_asphyxiation 'p1230' "'<dl><dt><~ MD 2~ "201~ then~
##  4  39.3 -76.7 icon_homicide_shooting     'p1229' "'<dl><dt><~ MD 2~ "201~ <NA>
##  5  39.3 -76.7 icon_homicide_shooting     'p1228' "'<dl><dt><~ MD 2~ "201~ foun~
##  6  39.2 -76.6 icon_homicide_shooting     'p1226' "'<dl><dt><~ MD 2~ "201~ but ~
##  7  39.3 -76.6 icon_homicide_shooting     'p1225' "'<dl><dt><~ MD 2~ "201~ <NA>
##  8  39.3 -76.6 icon_homicide_stabbing     'p1224' "'<dl><dt><~ MD 2~ "201~ <NA>
##  9  39.3 -76.7 icon_homicide_shooting     'p1223' "'<dl><dt><~ MD 2~ "201~ <NA>
## 10  39.3 -76.7 icon_homicide_bluntforce   'p1227' "'<dl><dt><~ MD 2~ "201~ 2006~
```

```r
# Method 2: read.csv with quote argument
df2 <- read.csv("https://michaelgastner.com/DAVisR_data/homicides.txt",
    header = F, quote = "'")
ncol(df2)
```

```
## [1] 5
```

```r
head(df2, n = 10)
```

```
##           V1          V2                         V3   V4
## 1  39.311024  -76.674227      iconHomicideShooting   p2
## 2  39.312641  -76.698948      iconHomicideShooting   p3
## 3  39.309781  -76.649882    iconHomicideBluntForce   p4
## 4  39.363925  -76.598772 iconHomicideAsphyxiation   p5
## 5  39.238928  -76.602718    iconHomicideBluntForce   p6
## 6  39.352676  -76.607979      iconHomicideShooting   p7
## 7  39.310999  -76.622023      iconHomicideShooting   p8
## 8  39.311103  -76.584475      iconHomicideShooting   p9
## 9  39.348101  -76.564960      iconHomicideShooting  p10
## 10 39.315050  -76.568647      iconHomicideShooting  p11
##
## 1        <dl><dt>Leon Nelson</dt><dd class="address">3400 Clifton Ave.<br />Baltimore, MD 21216</dd>
## 2             <dl><dt>Eddie Golf</dt><dd class="address">4900 Challedon Road<br />Baltimore, MD 212
## 3   <dl><dt>Nelsene Burnette</dt><dd class="address">2000 West North Ave<br />Baltimore, MD 21217</d
## 4   <dl><dt>Thomas MacKenney</dt><dd class="address">5900 Northwood Drive<br />Baltimore, MD 21212</
## 5            <dl><dt>Edward Canupp</dt><dd class="address">500 Maude Ave.<br />Baltimore, MD 21225<
## 6           <dl><dt>Michael Cunningham</dt><dd class="address">5200 Ready Ave.<br />Baltimore, MD 2
## 7              <dl><dt>Ray Alston</dt><dd class="address">300 West North Ave.<br />Baltimore, MD 21
## 8       <dl><dt>Yule Henderson</dt><dd class="address">1800 North Montford Ave.<br />Baltimore, MD 2
## 9         <dl><dt>Marcus McDowell</dt><dd class="address">5100 Harford Road<br />Baltimore, MD 21214
## 10          <dl><dt>Rodney Gardner</dt><dd class="address">3100 Ravenwood Road<br />Baltimore, MD 2
```

```r
nrow(df2)
```

```
## [1] 1012
```

```r
tail(df2, n = 10)
```

```
##
## 1003
## 1004  2012</dd><dd>Victim died at Maryland Shock Trauma Center</dd><dd>Cause: Shooting</dd><dd class=
## 1005
## 1006
## 1007
## 1008
## 1009
## 1010
## 1011
## 1012
##
## 1003  2012</dd><dd>Victim died at Johns Hopkins Hospital</dd><dd>Cause: Shooting</dd><dd class="popu
```

```
## 1004
## 1005
## 1006
## 1007
## 1008
## 1009
## 1010
## 1011
## 1012
##
## 1003
## 1004  2012</dd><dd>Victim died at Sinai Hospital</dd><dd>Cause: Shooting</dd></dl>\n39.28846940000, -
## 1005
## 1006
## 1007
## 1008
## 1009
## 1010
## 1011
## 1012
##
## 1003  2012</dd><dd>Victim died at Scene</dd><dd>Cause: Shooting</dd><dd class="popup-note"><p>Found s
## 1004
## 1005
## 1006
## 1007
## 1008
## 1009
## 1010
## 1011
## 1012
##
## 1003
## 1004                                                                                2012</dd><dd
## 1005
## 1006
## 1007                                                                  <dl><dt><a href="http://essentia
## 1008                 <dl><dt><a href="http://essentials.baltimoresun.com/micro_sun/homicides/victim/
## 1009
## 1010                          <dl><dt><a href="http://essentials.baltimoresun.com/micro_sun/homicide
## 1011
## 1012  <dl><dt><a href="http://essentials.baltimoresun.com/micro_sun/homicides/victim/1227/joseph-curt
```

Notice the difference between the two dataframes. The number of columns being different is expected, since the last column in quotes has multiple commas. But the number of rows is also different. In such cases, one should start by looking at the two data frames, trying to identify the differences. In this case I found that the head (first 10 rows) look similar, but the tail (last 10 rows) look different.

e. *Explore the "quote" argument in read_csv. See that it helps reduce the inconsistencies found in the previous question.*
f. *Write a code to split each row in 12 parts: Longitude, Lattitude, Category_of_homicide, case_number, Victim_name, Address, Victim_description, Gender, Victim_age, Date_of_homicide, Place_of_death, Cause_of_death. Hint: Use the function* `str_split`.
g. *Using the code above, convert the text file into a dataframe of 12 columns.*

h. *Using a barplot of the gender column, find out which gender has the highest number of homicides? Similarly find out, which cause of death is most common? Which month had the highest number of homicides? (Note that you would need to split the date column into day, month and year to do this. Use the* `separate` *function in tidyr package)*