CUNY DATA607_Wk3_Herold_Regex

Automated Data Collection with R (p. 217) 8.3

Using Stringr package.

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.1
```

```
raw.data <- "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-6542Rev.Timothy Lovejoy555 8
904Ned Flanders636-555-3226Simpson,Homer5553642Dr. Julius Hibbert"

name <- unlist(str_extract_all(raw.data, "[[:alpha:]., ]{2,}"))

# Add spaces after periods and commas to later help with extracting
name <- str_replace(name, pattern = "\\.", replacement = ". ")
name <- str_replace(name, pattern = ",", replacement = ", ")

name
```

```
## [1] "Moe Szyslak"            "Burns,  C.  Montgomery"
## [3] "Rev. Timothy Lovejoy"   "Ned Flanders"
## [5] "Simpson, Homer"         "Dr.  Julius Hibbert"
```

```
df <- data.frame(name = name, stringsAsFactors = F)
df
```

```
##                      name
## 1            Moe Szyslak
## 2 Burns,  C.  Montgomery
## 3   Rev. Timothy Lovejoy
## 4           Ned Flanders
## 5         Simpson, Homer
## 6    Dr.  Julius Hibbert
```

Determining if the name has a title.

```
## Removing the presumed titles from the names
df$temp.name <- str_remove(name, "[[:alpha:]]{2,}\\.")

## Check for periods after 2+ letters to signal titles
has.title <- str_detect(name, "[[:alpha:]]{2,}\\.")

## Add has.title column to dataframe
df <- data.frame(df, has.title = has.title)

df
```

```
##                     name              temp.name has.title
## 1            Moe Szyslak            Moe Szyslak     FALSE
## 2 Burns,  C.  Montgomery Burns,  C.  Montgomery     FALSE
## 3   Rev. Timothy Lovejoy        Timothy Lovejoy      TRUE
## 4           Ned Flanders           Ned Flanders     FALSE
## 5         Simpson, Homer         Simpson, Homer     FALSE
## 6    Dr.  Julius Hibbert         Julius Hibbert      TRUE
```

To separate the first and last names, we need to detect if there are any commas, which would change the regular order of First, then Last, name.

```
df$has.comma <- NULL

## Check for commas for last names first
df$has.comma <- str_detect(df$temp.name, ",")

df
```

```
##                     name              temp.name has.title has.comma
## 1            Moe Szyslak            Moe Szyslak     FALSE     FALSE
## 2 Burns,  C.  Montgomery Burns,  C.  Montgomery     FALSE      TRUE
## 3   Rev. Timothy Lovejoy        Timothy Lovejoy      TRUE     FALSE
## 4           Ned Flanders           Ned Flanders     FALSE     FALSE
## 5         Simpson, Homer         Simpson, Homer     FALSE      TRUE
## 6    Dr.  Julius Hibbert         Julius Hibbert      TRUE     FALSE
```

Now, we extract out the parts of temp.name, filling into the first_name and last_name fields depending on the Boolean of whether or not there was a comma in the name. Last name first.

```
df$last_name <- NULL
df$last_name[df$has.comma == TRUE] <- unlist(str_extract_all(df$temp.name[df$has.comma == TRUE],
"^[[:alpha:]]{2,}"))
df$last_name[df$has.comma == FALSE] <- unlist(str_extract_all(df$temp.name[df$has.comma == FALSE],
 "[[:alpha:]]{2,}$"))
df$last_name
```

```
## [1] "Szyslak"  "Burns"    "Lovejoy"  "Flanders" "Simpson"  "Hibbert"
```

Then first names, after they have been padded.

```
## Need to remove padding of temp.names first
df$temp.name <- str_trim(df$temp.name, side = "both")

df$first_name <- NULL
df$first_name[df$has.comma == TRUE] <- unlist(str_extract_all(df$temp.name[df$has.comma == TRUE],
"[[:alpha:][.][:blank:]]{2,}$"))
df$first_name[df$has.comma == FALSE] <- unlist(str_extract_all(df$temp.name[df$has.comma == FALSE
], "^[[:alpha:]]{2,}"))

##  I recognize that I did not generalize in the period issue in the first name, coding for C. Mon
tgomery in this problem. I think of the problems that "St." and "Jr." must cause.
df2 <- data.frame(df$name,df$first_name,df$last_name,df$has.title)
df2
```

```
##                  df.name    df.first_name df.last_name df.has.title
## 1            Moe Szyslak              Moe      Szyslak        FALSE
## 2 Burns,  C.  Montgomery   C.  Montgomery        Burns        FALSE
## 3   Rev. Timothy Lovejoy          Timothy      Lovejoy         TRUE
## 4           Ned Flanders              Ned     Flanders        FALSE
## 5         Simpson, Homer            Homer      Simpson        FALSE
## 6    Dr.  Julius Hibbert           Julius      Hibbert         TRUE
```

Trimming the first names again, then detecting for spaces to indicate two names.

```
df$first_name <- unlist(str_trim(df$first_name, side = "both"))
df2$is.twonames <- unlist(str_detect(df$first_name, " "))
df2
```

```
##                  df.name    df.first_name df.last_name df.has.title
## 1            Moe Szyslak              Moe      Szyslak        FALSE
## 2 Burns,  C.  Montgomery   C.  Montgomery        Burns        FALSE
## 3   Rev. Timothy Lovejoy          Timothy      Lovejoy         TRUE
## 4           Ned Flanders              Ned     Flanders        FALSE
## 5         Simpson, Homer            Homer      Simpson        FALSE
## 6    Dr.  Julius Hibbert           Julius      Hibbert         TRUE
##   is.twonames
## 1       FALSE
## 2        TRUE
## 3       FALSE
## 4       FALSE
## 5       FALSE
## 6       FALSE
```

Another problem would be determining if, given three names, a second name should be part of the first or last.

Automated Data Collection with R (p. 217) 8.4

[0-9]+\$

```
rawdata1 <- c("999$", "2222$", "333")
unlist(str_extract_all(rawdata1, "[0-9]+\\$"))
```

```
## [1] "999$"  "2222$"
```

\b[a-z]{1,4}\b

```
rawdata2 <- c("man","bird","Way")
unlist(str_extract_all(rawdata2, "\\b[a-z]{1,4}\\b"))
```

```
## [1] "man"  "bird"
```

.*?\.txt$

```
rawdata3 <- c(".txt","wow.dog.txt", "tree.look.txt2")
unlist(str_extract_all(rawdata3, ".*?\\.txt$"))
```

```
## [1] ".txt"        "wow.dog.txt"
```

\d{2}/\d{2}/\d{4}

```
rawdata4 <- c("22/09/1976","65/33/9999", "653.33/8888")
unlist(str_extract_all(rawdata4, "\\d{2}/\\d{2}/\\d{4}"))
```

```
## [1] "22/09/1976" "65/33/9999"
```

<(.+?)>.+?</\1>

```
rawdata5 <- c("<d> </d>","<meta> weep </meta>","giant")
unlist(str_extract_all(rawdata5, "<(.+?)>.+?</\\1>"))
```

```
## [1] "<d> </d>"            "<meta> weep </meta>"
```