

# HW\_3\_StringProcessing.r

*todddesktop*

*Sun Sep 18 21:10:31 2016*

```
initializeNames <- function()
{
  raw.data <- "555-1239Moe Szyslak(636) 555-0113Burns, C. Montgomery555-6542Rev. Timothy Lovejoy555 89
  library(stringr)
  return (unlist(str_extract_all(raw.data, "[[:alpha:]]., ]{2,}"))))
}

getFirstLastNames <-function(names)
{
  str_c(getFirstNames(name), " ", getLastNames(name))
}

getLastNames <- function(names)
{
  lastNames <- vector(mode = "character", length = 6)
  for(i in 1:length(names))
  {
    if (!is.na(str_locate(names[i], ",")[1])) ##names reversed
    {
      lastNames[i] <- getNameAtFront(names[i])
    }
    else
    {
      lastNames[i] <- getNameAtEnd(names[i])
    }
  }
  return(lastNames)
}

getNameAtFront <-function(names)
{
  return(str_extract(names, "[[:alpha:]]+"))
}

getNameAtEnd <-function(names)
{
  return(str_extract(names, "[[:alpha:]]+.$"))
}

##takes a vector in of full names, and returns na, or the first name for those full names
##where first name is at end of string (e.g, after a "," and/or "." for middle name)
getFirstNames <- function(names)
{
  firstFromFront <- getNameAtFront(names)
  firstFromLast <- getNameAtEnd(names)
```

```

    #each list has alternating NA in equivalent indexes, so just take "min" which will return non na va
    #and plop them into new vector
    str_trim(pmin(firstFromFront, firstFromLast, na.rm=TRUE))
}

getFirstNameAtFront <- function(names)
{
    return(str_extract(name, "[[:alpha:]]+ "))
}

getFirstNameAtEnd <-function(names)
{
    firstNames <- str_extract(names, "[[:alpha:]] .+") # extracts comma and all text after that (if e
    #replace comma and space(s) with nada, or common spaces, alpha, and period and spaces (e.g., middle
    firstNames <- str_replace(firstNames, " *[:alpha:]]*\\. *|, ", "")
##    firstNames <- str_replace(firstNames, ", +", "")
    return(firstNames)
}

isTitleInName <- function(names)
{
    return(!is.na(str_extract(names, "^[:alpha:]]+\\.\\.")))
}

isMiddleInitialInName <- function(names)
{
    return(!is.na(str_extract(names, " [[:alpha:]]\\.\\.")))
}

```