Stats 102A - Homework 3

Zooey Nguyen

2/22/2021

```
source("105195172_stats102a_hw3.R")
load("wordlists.RData")
list2env(wordlists, envir=environment())
```

```
## <environment: R_GlobalEnv>
```

Warmup

Looks like all the words in x have "foo" somewhere in it but it could be anywhere from beginning to end.

```
print(regex_golf(Warmup$x, Warmup$y, pat_1))
```

```
## $score
## [1] 7
##
## $matched_x
##  [1] "afoot"   "catfoot" "dogfoot" "fanfoot" "foody"   "foolery" "foolish"
##  [8] "fooster" "footage" "foothot" "footle"  "footpad" "footway" "hotfoot"
## [15] "jawfoot" "mafoo"   "nonfood" "padfoot" "prefool" "sfoot"   "unfool"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "Atlas"   "Aymoro"  "Iberic"  "Mahran"  "Ormazd"  "Silipan" "altared"
##  [8] "chandoo" "crenel"  "crooked" "fardo"   "folksy"  "forest"  "hebamic"
## [15] "idgah"   "manlike" "marly"   "palazzi" "sixfold" "tarrock" "unfold"
```

Anchors

All the words in x have "ick" at the *end* of the word while none of them in y do. Some words in y have "ick" but not at the end.

```
print(regex_golf(Anchors$x, Anchors$y, pat_2))
```

```
## $score
## [1] 6
##
## $matched_x
##  [1] "Mick"         "Rick"         "allocochick"  "backtrick"    "bestick"
##  [6] "candlestick"  "counterprick" "heartsick"    "lampwick"     "lick"
## [11] "lungsick"     "potstick"     "quick"        "rampick"      "rebrick"
## [16] "relick"       "seasick"      "slick"        "tick"         "unsick"
## [21] "upstick"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "Kickapoo"      "Nickneven"     "Rickettsiales" "billsticker"
##  [5] "borickite"     "chickell"      "fickleness"    "finickily"
##  [9] "kilbrickenite" "lickpenny"     "mispickel"     "quickfoot"
## [13] "quickhatch"    "ricksha"       "rollicking"    "slapsticky"
## [17] "snickdrawing"  "sunstricken"   "tricklingly"   "unlicked"
## [21] "unnickeled"
```

Ranges

Looks like these all only have letters from a to f.

```
print(regex_golf(Ranges$x, Ranges$y, pat_3))
```

```
## $score
## [1] 7
##
## $matched_x
##  [1] "abac"   "accede" "adead"  "babe"   "bead"   "bebed"  "bedad"  "bedded"
##  [9] "bedead" "bedeaf" "caba"   "caffa"  "dace"   "dade"   "daff"   "dead"
## [17] "deed"   "deface" "faded"  "faff"   "feed"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "beam"   "buoy"   "canjac" "chymia" "corah"  "cupula" "griece" "hafter"
##  [9] "idic"   "lucy"   "martyr" "matron" "messrs" "mucose" "relose" "sonly"
## [17] "tegua"  "threap" "towned" "widish" "yite"
```

Backrefs

This one looks tricky. In the x all the words have some repetition of a substring with at least three letters. My capture group has to be greedy and enough to capture the second largest identical group.

```
print(regex_golf(Backrefs$x, Backrefs$y, pat_4))
```

```
## $score
## [1] 15
##
## $matched_x
##  [1] "allochirally"    "anticovenanting" "barbary"         "calelectrical"
##  [5] "entablement"     "ethanethiol"     "froufrou"        "furfuryl"
##  [9] "galagala"        "heavyheaded"     "linguatuline"    "mathematic"
## [13] "monoammonium"    "perpera"         "photophonic"     "purpuraceous"
## [17] "salpingonasal"   "testes"          "trisectrix"      "undergrounder"
## [21] "untaunted"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "anticker"        "corundum"        "crabcatcher"     "damnably"
##  [5] "foxtailed"       "galvanotactic"   "gummage"         "gurniad"
##  [9] "hypergoddess"    "kashga"          "nonimitative"    "parsonage"
## [13] "pouchlike"       "presumptuously"  "pylar"           "rachioparalysis"
## [17] "scherzando"      "swayed"          "unbridledness"   "unupbraidingly"
## [21] "wellside"
```

Abba

It looks like we need to ensure we DON'T match any words with the pattern abba in it, where a is one letter and b is another. Hmm.

```
print(regex_golf(Abba$x, Abba$y, pat_5))
```

```
## $score
## [1] 19
##
## $matched_x
##  [1] "acritan"          "aesthophysiology" "amphimictical"    "baruria"
##  [5] "calomorphic"      "disarmature"      "effusive"         "fluted"
##  [9] "fusoid"           "goblinize"        "nihilistic"       "noisefully"
## [13] "picrorhiza"       "postarytenoid"    "revolutionize"    "suprasphanoidal"
## [17] "suspenseful"      "tapachula"        "transmit"         "unversatile"
## [21] "vibetoite"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "abba"            "anallagmatic"    "bassarisk"       "chorioallantois"
##  [5] "coccomyces"      "commotive"       "engrammatic"     "glossoscopia"
##  [9] "hexacoralla"     "hippogriffin"    "inflammableness" "otto"
## [13] "overattached"    "saffarid"        "sarraceniaceae"  "scillipicrin"
## [17] "tlapallan"       "trillion"        "unclassably"     "unfitting"
## [21] "unsmelled"       "warrandice"
```

A man, a plan

Everything to match has to be a palindrome. So I think as a simpler check I can check just that we can match ab ba pattern at beginning/end of the word.

```
print(regex_golf(`A man, a plan`$x, `A man, a plan`$y, pat_6))
```

```
## $score
## [1] 13
##
## $matched_x
##  [1] "civic"   "deedeed" "degged"  "hallah"  "kakkak"  "kook"    "level"
##  [8] "murdrum" "noon"    "redder"  "repaper" "retter"  "reviver" "rotator"
## [15] "sexes"   "sooloos" "tebbet"  "tenet"   "terret"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "arrogatingly"  "camshach"      "cinnabar"      "defendress"
##  [5] "derivedly"     "gourmet"       "hamleteer"     "hydroaviation"
##  [9] "lophine"       "nonalcohol"    "outslink"      "pretest"
## [13] "psalterium"    "psorosperm"    "scrummage"     "sporous"
## [17] "springer"      "sunburn"       "teleoptile"    "unstuttering"
## [21] "womanways"
```

Prime

So there has to be a prime number of x's. Need to negative-match if anything is an exact multiple of two or more x's. So first check with negative lookahead that a group of two through however many x's does NOT perfectly fit into the string more than once. If we get that yes, it never worked for any length between two and however many, then we match the whole string.

```
print(regex_golf(Prime$x, Prime$y, pat_7))
```

```
## $score
## [1] 16
##
## $matched_x
##  [1] "xx"
##  [2] "xxx"
##  [3] "xxxxx"
##  [4] "xxxxxxx"
##  [5] "xxxxxxxxxxx"
##  [6] "xxxxxxxxxxxxx"
##  [7] "xxxxxxxxxxxxxxxxx"
##  [8] "xxxxxxxxxxxxxxxxxxx"
##  [9] "xxxxxxxxxxxxxxxxxxxxxxx"
## [10] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [11] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [12] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [13] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [14] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [15] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [16] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [17] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [18] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [19] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
## [20] "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "xxxx"                          "xxxxxx"
##  [3] "xxxxxxxx"                      "xxxxxxxxx"
##  [5] "xxxxxxxxxx"                    "xxxxxxxxxxx"
##  [7] "xxxxxxxxxxxxx"                 "xxxxxxxxxxxxxx"
##  [9] "xxxxxxxxxxxxxxx"               "xxxxxxxxxxxxxxxx"
## [11] "xxxxxxxxxxxxxxxxxx"            "xxxxxxxxxxxxxxxxxx"
## [13] "xxxxxxxxxxxxxxxxxxxx"          "xxxxxxxxxxxxxxxxxxxx"
## [15] "xxxxxxxxxxxxxxxxxxxxxx"        "xxxxxxxxxxxxxxxxxxxxxx"
## [17] "xxxxxxxxxxxxxxxxxxxxxxxx"      "xxxxxxxxxxxxxxxxxxxxxxxx"
## [19] "xxxxxxxxxxxxxxxxxxxxxxxxxx"    "xxxxxxxxxxxxxxxxxxxxxxxxxx"
```

oh my god it worked.

Four

Some letter is repeated four times with one character between each time.

```
print(regex_golf(Four$x, Four$y, pat_8))
```

```
## $score
## [1] 15
##
## $matched_x
##  [1] "Makaraka"          "Wasagara"          "degenerescence"
##  [4] "desilicification"  "elevener"          "hipponosological"
##  [7] "homoeomorphy"      "homologous"        "ileocolotomy"
## [10] "intervisibility"   "jararaca"          "locomotory"
## [13] "micropoikilitic"   "odontonosology"    "parhomologous"
## [16] "pogonotomy"        "promonopolist"     "protohomo"
## [19] "pseudoprimitivism" "tocororo"          "unintelligibility"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "Ludgate"          "Mitsukurinidae"   "Ternstroemiaceae" "arrhythmical"
##  [5] "bleater"          "energetics"       "inthrow"          "mecopterous"
##  [9] "multum"           "naphthalene"      "nullibicity"      "observancy"
## [13] "overpunishment"   "overregularly"    "overwilily"       "participator"
## [17] "predisable"       "reyield"          "rubeola"          "traitorlike"
## [21] "unregainable"
```

Order

All the letters are in descending alphabetical order. I really do not know how to do this in a pretty way so I'm brute forcing it with the letters... please don't judge me I am very tired.

```
print(regex_golf(Order$x, Order$y, pat_9))
```

```
## $score
## [1] 41
##
## $matched_x
##  [1] "access" "accloy" "adeem"  "aflow"  "aglow"  "beefin" "befist" "billot"
##  [9] "bossy"  "certy"  "chintz" "chips"  "chort"  "cloop"  "coost"  "demos"
## [17] "fitty"  "flory"  "flossy" "ghost"  "mopsy"
##
## $unmatched_x
## character(0)
##
## $matched_y
## character(0)
##
## $unmatched_y
##  [1] "analyse"   "balanism"  "baronet"   "biddable"  "griefless" "harebrain"
##  [7] "jestword"  "laicize"   "marvelry"  "oriole"    "pickietar" "preferee"
## [13] "primness"  "pulghere"  "rebirth"   "scupper"   "serigraph" "sororize"
## [19] "theowman"  "unfrayed"  "wagonman"
```

Alphabetical

Having a really difficult time figuring this one out and I thought it would be the easiest out of the last three, lol. Was not able to get a real solution in time but I kept my last test idea in just as a note that this is the one I attempted, but I matched everything instead of just matching the x-list.

```
print(regex_golf(Alphabetical$x, Alphabetical$y, pat_12))
```

```
## $score
## [1] 198
##
## $matched_x
##  [1] "aerate aerate arrest errant serene tanner testes"
##  [2] "aerate assent assent assert rester retest tenant"
##  [3] "aerate assert rearer renter resent serene teaser"
##  [4] "aerate easter easter tenant tester testes tsetse"
##  [5] "arrest arrest easter entree errant resent senate"
##  [6] "assent assess assets estate resent staree teaser"
##  [7] "assert astern renter rerent resent staree street"
##  [8] "assert enseat entree errata rennet teaser tsetse"
##  [9] "assert rennet renter reseat rester serene tenant"
## [10] "assess easter estate rennet rennet tenant testes"
## [11] "assess easter estate rerent resent retest snarer"
## [12] "assess renter renter searer seater snarer testes"
## [13] "astern enseat entree serene staree tartar tartar"
## [14] "astern rennet retest searer snarer tartar tester"
## [15] "enseat errata seater senate strata teaser tsetse"
## [16] "entree searer staree taster taster tenant testes"
## [17] "rerent rester tanner tartar teaser teaser testes"
##
## $unmatched_x
## character(0)
##
## $matched_y
##  [1] "aerate astern assess enseat senate street tsetse"
##  [2] "aerate rennet errant enseat rerent senate testes"
##  [3] "arrest assess assess assent astern searer testes"
##  [4] "assert assess errata enseat earner seater serene"
##  [5] "assert astern staree senate snarer tanner tester"
##  [6] "assert strata rerent rerent tanner testes tsetse"
##  [7] "assess easter entree rester reseat seater tartar"
##  [8] "astern assets rearer rearer assess rearer testes"
##  [9] "astern easter taster serene reseat taster tester"
## [10] "earner entree rerent reseat teaser strata staree"
## [11] "earner errant estate taster reseat estate taster"
## [12] "enseat astern arrest enseat searer seater tenant"
## [13] "errant errant senate renter rearer street tsetse"
## [14] "rennet rennet assent errant rester staree tester"
## [15] "rennet snarer senate retest tanner tartar tsetse"
## [16] "retest astern arrest tsetse strata senate tsetse"
## [17] "searer errant teaser staree assess teaser tsetse"
##
## $unmatched_y
## character(0)
```