lab5

Credit: materials adapted from Patrick Chester, with some examples taken from Ken Benoit's NYU Dept. of Politics short course Fall 2014

1 Setting up

```r
# Clear Global Environment
rm(list = ls())
setwd("/Users/Lingyi/TAD/lab/Text-as-Data-Lab-Spr2018/W5_02_27_18")

# Libraries
library(quanteda)
```

```
## quanteda version 1.0.0

## Using 3 of 4 threads for parallel computing

##
## Attaching package: 'quanteda'

## The following object is masked from 'package:utils':
##
##     View
```

```r
library(quanteda.corpora)
```

2 Loading data: conservative manifestos

```r
# read in the files
filenames <- list.files(path = "conservative_manifestos", full.names=TRUE)
cons_manifestos <- lapply(filenames, readLines)
cons_manifestos <- unlist(lapply(cons_manifestos, function(x) paste(x, collapse = " ")))
# because readLines returns a vector with each elements = lines

# get the date docvar from the filename
dates <- unlist(regmatches(unlist(filenames), gregexpr("[[:digit:]]+", unlist(filenames))))

# Construct dataframe
manifestos_df <- data.frame(
  year = dates, text = cons_manifestos, stringsAsFactors = FALSE)
```

3 Regular Expressions

```r
# Examples

words <- c("Washington Post", "NYT", "Wall Street Journal",
           "Peer-2-Peer", "Red State", "Cheese", "222", ",")
```

```r
# Exploring by character type
grep("\\w", words, value = T)
```

```
## [1] "Washington Post"     "NYT"                 "Wall Street Journal"
## [4] "Peer-2-Peer"         "Red State"           "Cheese"
## [7] "222"
```

```r
# Elements that have alphanumeric characters
grep("\\w{7}", words, value = T)
```

```
## [1] "Washington Post"     "Wall Street Journal"
```

```r
# Elements that have words that are at least 7 characters long
grep("\\d", words, value = T)
```

```
## [1] "Peer-2-Peer" "222"
```

```r
# Elements that contain numbers
grep("\\W", words, value = T)
```

```
## [1] "Washington Post"     "Wall Street Journal" "Peer-2-Peer"
## [4] "Red State"           ","
```

```r
# Elements that contain nonword characters (Including white space)

# note that  grep returns the full element that matched the pattern

words2 <- c("voting", "votes", "devoted!", "vote?", "ddd.")

grep("\\!$", words2)
```

```
## [1] 3
```

```r
grepl("\\.$", words2)
```

```
## [1] FALSE FALSE FALSE FALSE  TRUE
```

```r
grep("^vot", words2)
```

```
## [1] 1 2 4
```

```r
# Returns the index of matching items in the vector
grep("^vot", words2, value = T)
```

```
## [1] "voting" "votes"  "vote?"
```

```r
# Returns the elements of the vector that matched the pattern
grepl("^vot", words2)
```

```
## [1]  TRUE  TRUE FALSE  TRUE FALSE
```

```r
# Returns a logical vector indicating whether or
# not the component containes the expression

# you can use the indices to select elements from the original vector that you want
words2[grepl("^vot", words2)]
```

```
## [1] "voting" "votes"  "vote?"
```

```r
presidents <- c("Roosevelt-33", "Roosevelt-37", "Obama-2003")
```

```
# Use gsub to replace patterns with a string
gsub("(\\w)-(\\d{2})", "\\1-19\\2", presidents)
```

```
## [1] "Roosevelt-1933" "Roosevelt-1937" "Obama-192003"
```

```
# Parentheses can identify components that can later be referenced by \\1 - \\9

gsub("(\\w)-(\\d{2})$", "\\1-19\\2", presidents)
```

```
## [1] "Roosevelt-1933" "Roosevelt-1937" "Obama-2003"
```

```
# We want to use the $ to indicate that the pattern should come
# at the end of the word, to avoid the mismatch in Obama-192003

# Note that regex expressions in R are similar to those in other languages but there are some key diffe

# Resources:
# https://rstudio-pubs-static.s3.amazonaws.com/74603_76cd14d5983f47408fdf0b323550b846.html
# http://r4ds.had.co.nz/strings.html#matching-patterns-with-regular-expressions
```

4 Selecting Features from DFM using Regular Expressions

```
# Using simple texts

testText <- paste0("The quick brown fox named Seamus jumps over the lazy dog also",
                   " named Seamus, with the newspaper from a a boy named Seamus,",
                   " in his mouth.")

# keep only words ending in "s"
print(dfm(testText, select = "s$", valuetype = "regex"))
```

```
## Document-feature matrix of: 1 document, 3 features (0% sparse).
## 1 x 3 sparse Matrix of class "dfm"
##        features
## docs    seamus jumps his
##   text1      3     1   1
```

```
testTweets <- c("2 + 2 = 4 #1984",
                "I thought you said the park? Why are we at the vet? #QuestionsFromPets",
                "Holy freeway #flooding Batman! #californiastorms taking their toll.")

# keep only hashtags i.e. expressions starting with a pound sign
print(dfm(testTweets, select="^#", valuetype = "regex"))
```

```
## Document-feature matrix of: 3 documents, 4 features (66.7% sparse).
## 3 x 4 sparse Matrix of class "dfm"
##        features
## docs    #1984 #questionsfrompets #flooding #californiastorms
##   text1     1                  0         0                 0
##   text2     0                  1         0                 0
##   text3     0                  0         1                 1
```

```
# Selecting features from a corpus

data("data_corpus_irishbudget2010")
```

```
irishbudgets_dfm <- dfm(data_corpus_irishbudget2010, select=c("tax|budg|^auster"),
             valuetype = "regex")
# valuetype = "regex" ensures that the select input
# will be interpreted as a regular expression

# You can pass a list of words to the "select" parameter in dfm,
#but using regular expressions can enable you to get all variants of a word
View(irishbudgets_dfm)
```

5 Dictionaries

```
# Here, dictionary = list of words, not the data structure.
# There are no dictionaries in R :( :( :(

mytexts <- c("The new law included a capital gains tax, and an inheritance tax.",
             "New York City has raised a taxes: an income tax and a sales tax.")

mydict <- c("tax", "income", "capital", "gains", "inheritance")

print(dfm(mytexts, select = mydict))
```

```
## Document-feature matrix of: 2 documents, 5 features (40% sparse).
## 2 x 5 sparse Matrix of class "dfm"
##        features
## docs    capital gains tax inheritance income
##   text1       1     1   2           1      0
##   text2       0     0   2           0      1
```

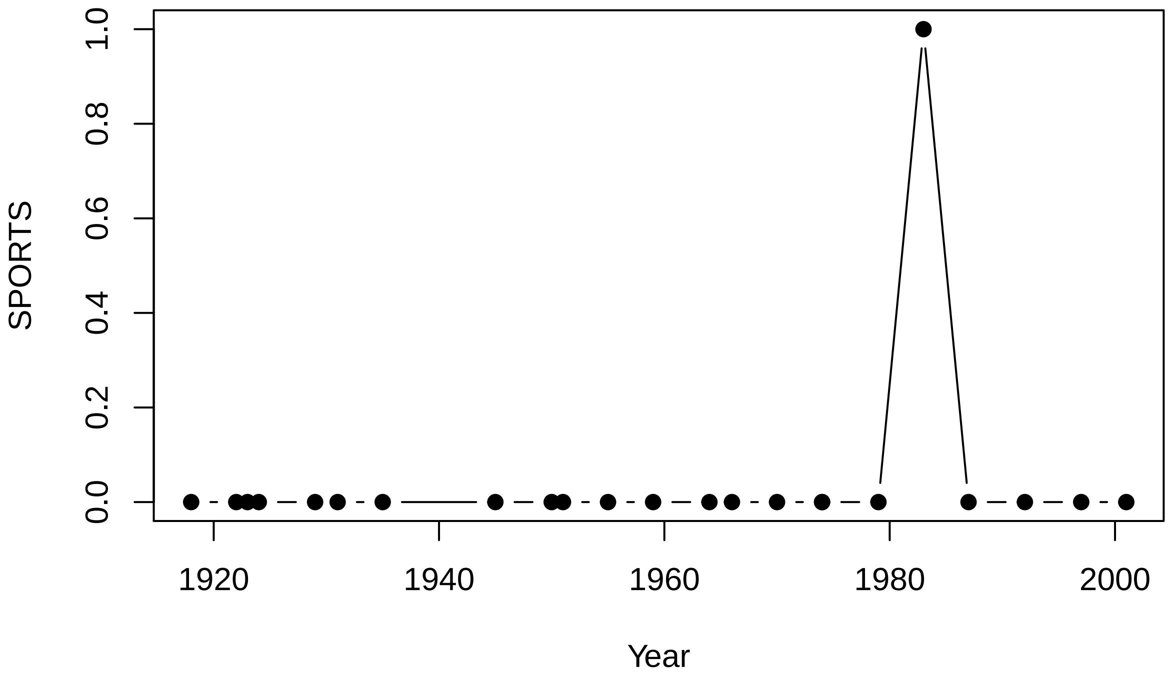
```
# Example: Laver Garry dictionary
lgdict <- dictionary(file = "LaverGarry.cat", format = "wordstat")

# What's in this thing?
lgdict
```

```
## Dictionary object with 9 primary key entries and 2 nested levels.
## - [CULTURE]:
##   - people, war_in_iraq, civil_war
##   - [CULTURE-HIGH]:
##     - art, artistic, dance, galler*, museum*, music*, opera*, theatre*
##   - [CULTURE-POPULAR]:
##     - media
##   - [SPORT]:
##     - angler*
## - [ECONOMY]:
##   - [+STATE+]:
##     - accommodation, age, ambulance, assist, benefit, care, carer*, child*, class, classes, clinics,
##   - [=STATE=]:
##     - accountant, accounting, accounts, advert*, airline*, airport*, audit*, bank*, bargaining, brea
##   - [-STATE-]:
##     - assets, autonomy, barrier*, bid, bidders, bidding, burden*, charit*, choice*, compet*, confide
## - [ENVIRONMENT]:
##   - [CON ENVIRONMENT]:
##     - produc*
##   - [PRO ENVIRONMENT]:
##     - car, catalytic, chemical*, chimney*, clean*, congestion, cyclist*, deplet*, ecolog*, emission*
## - [GROUPS]:
##   - [ETHNIC]:
##     - asian*, buddhist*, ethnic*, race, raci*
##   - [WOMEN]:
##     - girls, woman, women
## - [INSTITUTIONS]:
##   - [CONSERVATIVE]:
##     - authority, continu*, disrupt*, inspect*, jurisdiction*, legitimate, manag*, moratorium, rul*, s
##   - [NEUTRAL]:
##     - administr*, advis*, agenc*, amalgamat*, appoint*, assembly, chair*, commission*, committee*, co
##   - [RADICAL]:
##     - abolition, accountable, answerable, consult*, corrupt*, democratic*, elect*, implement*, modern
## - [LAW_AND_ORDER]:
##   - [LAW-CONSERVATIVE]:
##     - assaults, bail, burglar*, constab*, convict*, court, courts, custod*, dealing, delinquen*, dete
##   - [LAW-LIBERAL]:
##     - harassment, non-custodial
## - [RURAL]:
##   - agricultur*, badgers, bird*, countryside, farm*, feed, fish*, forest*, hens, horse*, landscape*,
## - [URBAN]:
##   - town*
## - [VALUES]:
##   - [CONSERVATIVE]:
##     - defend, defended, defending, discipline, glories, glorious, grammar, heritage, histor*, honour
##   - [LIBERAL]:
##     - cruel*, discriminat*, human*, injustice*, innocent, inter_racial, minorit*, repressi*, rights,
```

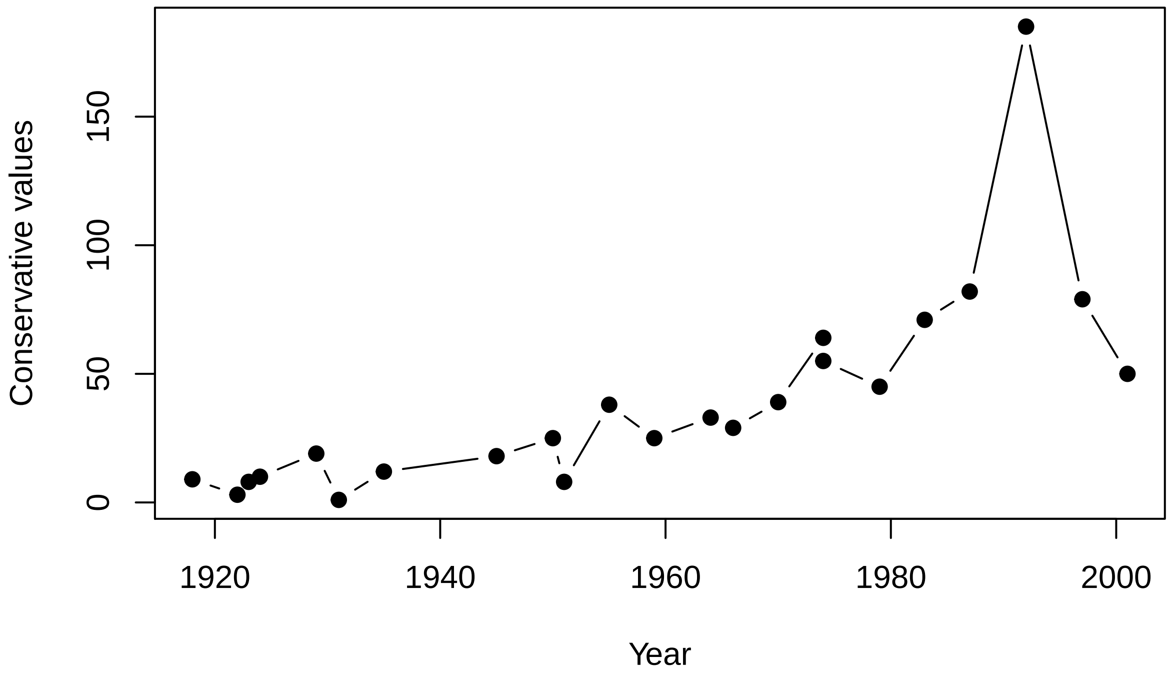
```r
# Run the conservative manifestos through this dictionary
manifestos_lg <- dfm(manifestos_df$text, dictionary = lgdict)

featnames(manifestos_lg)
```

```
##  [1] "CULTURE"                          "CULTURE.CULTURE-HIGH"
##  [3] "CULTURE.CULTURE-POPULAR"          "CULTURE.SPORT"
##  [5] "ECONOMY.+STATE+"                  "ECONOMY.=STATE="
##  [7] "ECONOMY.-STATE-"                  "ENVIRONMENT.CON ENVIRONMENT"
##  [9] "ENVIRONMENT.PRO ENVIRONMENT"      "GROUPS.ETHNIC"
## [11] "GROUPS.WOMEN"                     "INSTITUTIONS.CONSERVATIVE"
## [13] "INSTITUTIONS.NEUTRAL"             "INSTITUTIONS.RADICAL"
## [15] "LAW_AND_ORDER.LAW-CONSERVATIVE" "LAW_AND_ORDER.LAW-LIBERAL"
## [17] "RURAL"                            "URBAN"
## [19] "VALUES.CONSERVATIVE"              "VALUES.LIBERAL"
```

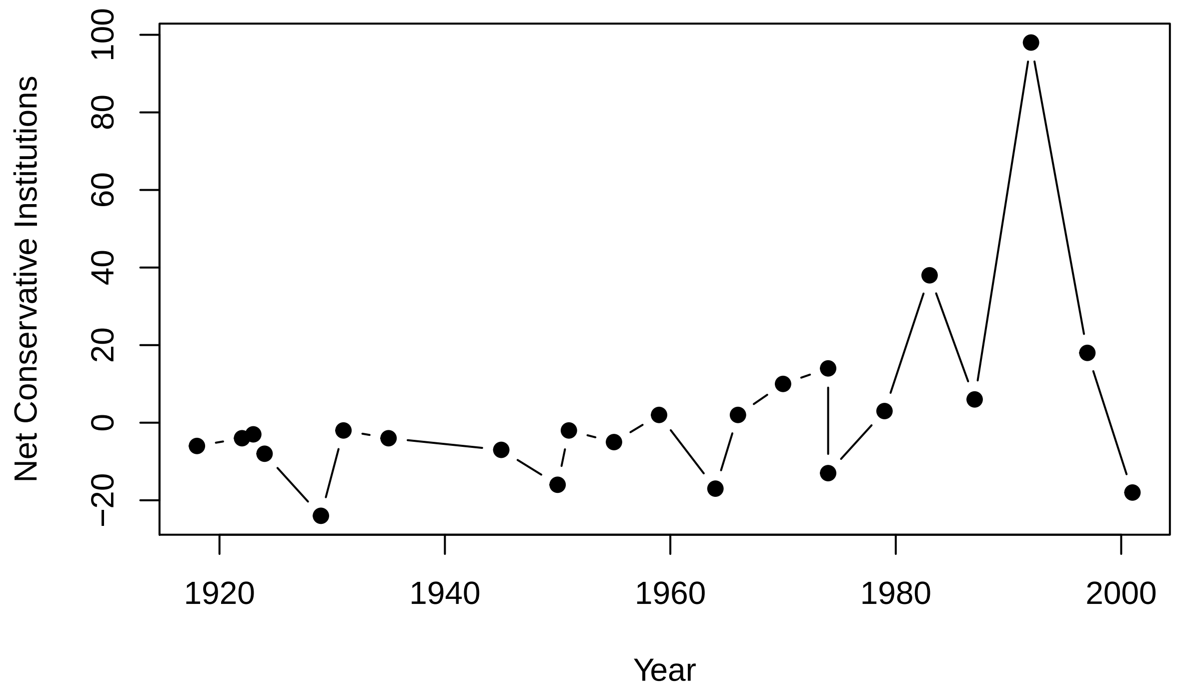
```r
# plot it

plot(manifestos_df$year,
     manifestos_lg[,"CULTURE.SPORT"],
     xlab="Year", ylab="SPORTS", type="b", pch=19)
```



```
plot(manifestos_df$year,
     manifestos_lg[,"VALUES.CONSERVATIVE"],
     xlab="Year", ylab="Conservative values", type="b", pch=19)
```



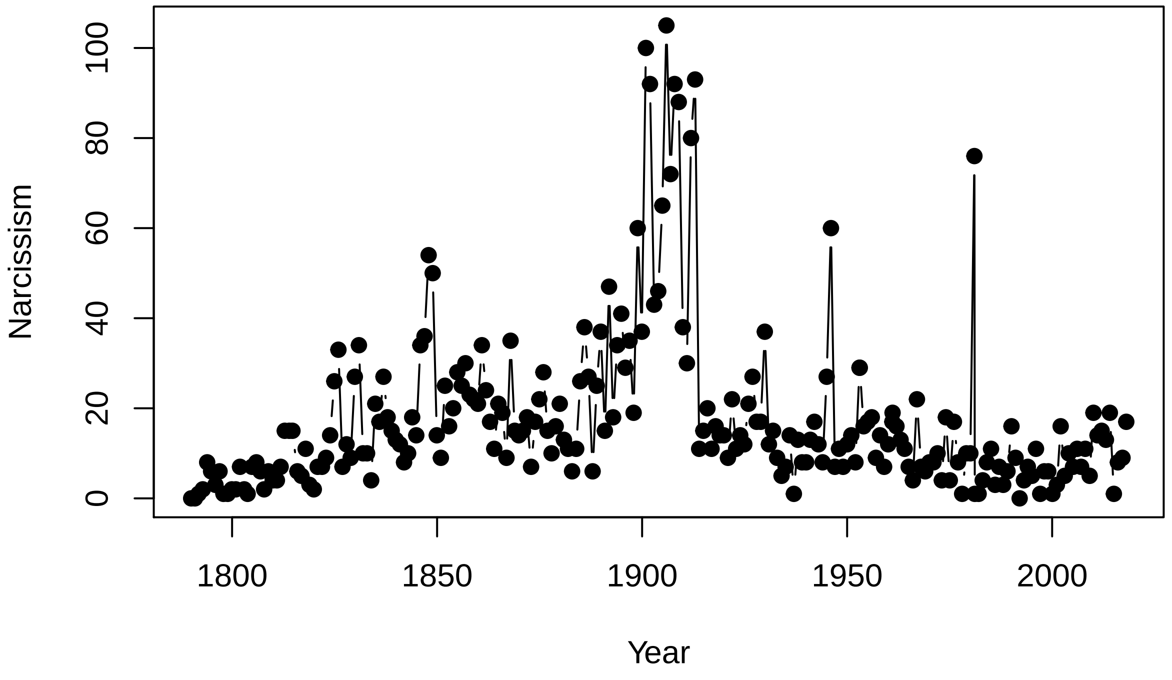
```
plot(manifestos_df$year,
     manifestos_lg[
       ,"INSTITUTIONS.CONSERVATIVE"] - manifestos_lg[
       ,"INSTITUTIONS.RADICAL"],
     xlab="Year", ylab="Net Conservative Institutions", type="b", pch=19)
```



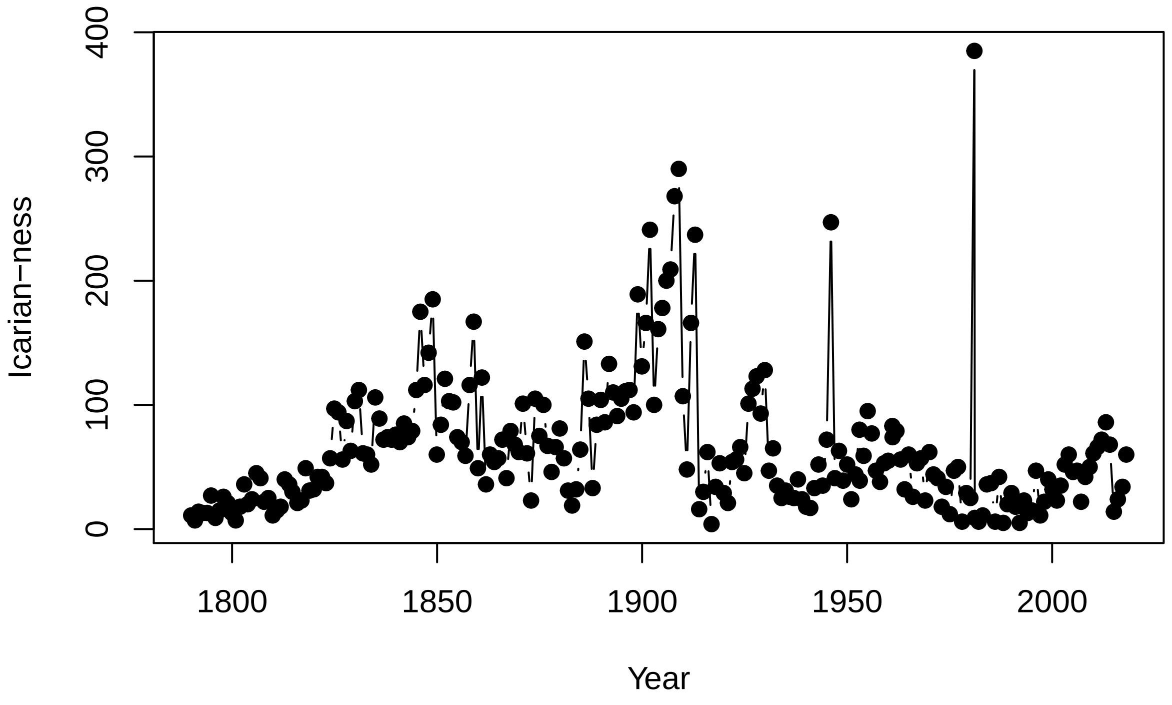
```r
# RID Dictionary--Regressive Imagery Dictionary

rid_dict <- dictionary(file = "RID.cat", format = "wordstat")

data("data_corpus_sotu")

sotus_texts <- texts(data_corpus_sotu)

# Get the docvars from the corpus object
year <- (data_corpus_sotu$documents$Date)
pres <- (data_corpus_sotu$documents$President)

sotu_rid_dfm <- dfm(data_corpus_sotu, dictionary = rid_dict)

# Look at the categories
featnames(sotu_rid_dfm)
```

```
##  [1] "PRIMARY.NEED.ORALITY"
##  [2] "PRIMARY.NEED.ANALITY"
##  [3] "PRIMARY.NEED.SEX"
##  [4] "PRIMARY.SENSATION.TOUCH"
##  [5] "PRIMARY.SENSATION.TASTE"
##  [6] "PRIMARY.SENSATION.ODOR"
##  [7] "PRIMARY.SENSATION.GEN_SENSATION"
##  [8] "PRIMARY.SENSATION.SOUND"
##  [9] "PRIMARY.SENSATION.VISION"
## [10] "PRIMARY.SENSATION.COLD"
## [11] "PRIMARY.SENSATION.HARD"
## [12] "PRIMARY.SENSATION.SOFT"
## [13] "PRIMARY.DEFENSIVE_SYMBOL.PASSIVITY"
## [14] "PRIMARY.DEFENSIVE_SYMBOL.VOYAGE"
## [15] "PRIMARY.DEFENSIVE_SYMBOL.RANDOM MOVEMENT"
## [16] "PRIMARY.DEFENSIVE_SYMBOL.DIFFUSION"
## [17] "PRIMARY.DEFENSIVE_SYMBOL.CHAOS"
## [18] "PRIMARY.REGR_KNOL.UNKNOW"
## [19] "PRIMARY.REGR_KNOL.TIMELESSNES"
## [20] "PRIMARY.REGR_KNOL.COUNSCIOUS"
## [21] "PRIMARY.REGR_KNOL.BRINK-PASSAGE"
## [22] "PRIMARY.REGR_KNOL.NARCISSISM"
## [23] "PRIMARY.REGR_KNOL.CONCRETENESS"
## [24] "PRIMARY.ICARIAN_IM.ASCEND"
## [25] "PRIMARY.ICARIAN_IM.HEIGHT"
## [26] "PRIMARY.ICARIAN_IM.DESCENT"
## [27] "PRIMARY.ICARIAN_IM.DEPTH"
## [28] "PRIMARY.ICARIAN_IM.FIRE"
## [29] "PRIMARY.ICARIAN_IM.WATER"
## [30] "SECONDARY.ABSTRACT_TOUGHT"
## [31] "SECONDARY.SOCIAL_BEHAVIOR"
## [32] "SECONDARY.INSTRU_BEHAVIOR"
## [33] "SECONDARY.RESTRAINT"
## [34] "SECONDARY.ORDER"
## [35] "SECONDARY.TEMPORAL_REPERE"
## [36] "SECONDARY.MORAL_IMPERATIVE"
## [37] "EMOTIONS.POSITIVE_AFFECT"
## [38] "EMOTIONS.ANXIETY"
## [39] "EMOTIONS.SADNESS"
## [40] "EMOTIONS.AFFECTION"
## [41] "EMOTIONS.AGGRESSION"
## [42] "EMOTIONS.EXPRESSIVE_BEH"
## [43] "EMOTIONS.GLORY"
```

```r
# Inspect the results graphically
plot(year,
     sotu_rid_dfm[,"PRIMARY.REGR_KNOL.NARCISSISM"],
     xlab="Year", ylab="Narcissism", type="b", pch=19)
```



```r
plot(year,
     sotu_rid_dfm[,"PRIMARY.ICARIAN_IM.FIRE"] + sotu_rid_dfm[
       ,"PRIMARY.ICARIAN_IM.ASCEND"] +sotu_rid_dfm[
       ,"PRIMARY.ICARIAN_IM.DESCENT"] + sotu_rid_dfm[
       ,"PRIMARY.ICARIAN_IM.DEPTH"] + sotu_rid_dfm[
       ,"PRIMARY.ICARIAN_IM.HEIGHT"] + sotu_rid_dfm[
       ,"PRIMARY.ICARIAN_IM.WATER"],
     xlab="Year", ylab="Icarian-ness", type="b", pch=19)
```


Icarian−ness
Year