How Severe Weather Events Affect Health and the Economy

Reproducible Research Course Project #2

Kevin Bitinsky

2020-05-10

Synopisis

This report investigates data from the National Oceanic and Atmospheric Administrationv(NOAA) Storm Database to identify the type of storms that cause the greatest impact to both population health and economic damage.

The data required some processing in order to structure and analyze it. However, according to the data that was accesssed, and this analysis, the worst impacts are:

- **Tornados -** which inflicted nearly 100,000 injuries and fatalities, combined.
- **Floods -** which caused over $150 billion in damages.

The following document will describe the extract, tranformation and analysis of the data in order to reach these conclusions.

Background

The Storm Data may be found at the course website Coursera Reproducible Research: storm data

For more information available from the National Centers for Environmental Information:

- National Weather Service Storm Data Documentation
- National Climatic Data Center Storm Events FAQ

Data Processing

```
library(tidyverse)
library(stringdist)
```

Load the data

- Download the data, if it doesn't already exist within the local ./data subfolder.
- Read it in using the base read.csv() function. Note that the data is read in directly from the compressed bz2 format.

```
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
ifelse(!dir.exists(file.path("data")),
       dir.create(file.path("data")), "Folder already exists")
```

```
## [1] "Folder already exists"
```

```
destfile <- "./data/StormData.csv.bz2"
if (!file.exists(destfile)) {
  download.file(url ,destfile,method="curl")
}
data <- read.csv(bzfile("./data/StormData.csv.bz2"),
                 header = TRUE, stringsAsFactors = FALSE)
str(data) # examine the structure of the data
```

```
## 'data.frame':    902297 obs. of  37 variables:
##  $ STATE__   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ BGN_DATE  : chr  "4/18/1950 0:00:00" "4/18/1950 0:00:00" "2/20/1951 0:00:00" "6/8/1951 0:00:00" .
##  $ BGN_TIME  : chr  "0130" "0145" "1600" "0900" ...
##  $ TIME_ZONE : chr  "CST" "CST" "CST" "CST" ...
##  $ COUNTY    : num  97 3 57 89 43 77 9 123 125 57 ...
##  $ COUNTYNAME: chr  "MOBILE" "BALDWIN" "FAYETTE" "MADISON" ...
##  $ STATE     : chr  "AL" "AL" "AL" "AL" ...
##  $ EVTYPE    : chr  "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
##  $ BGN_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ BGN_AZI   : chr  "" "" "" "" ...
##  $ BGN_LOCATI: chr  "" "" "" "" ...
##  $ END_DATE  : chr  "" "" "" "" ...
##  $ END_TIME  : chr  "" "" "" "" ...
##  $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ COUNTYENDN: logi  NA NA NA NA NA NA ...
##  $ END_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ END_AZI   : chr  "" "" "" "" ...
##  $ END_LOCATI: chr  "" "" "" "" ...
##  $ LENGTH    : num  14 2 0.1 0 0 1.5 1.5 0 3.3 2.3 ...
##  $ WIDTH     : num  100 150 123 100 150 177 33 33 100 100 ...
##  $ F         : int  3 2 2 2 2 2 2 1 3 3 ...
##  $ MAG       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
##  $ PROPDMG   : num  25 2.5 25 2.5 2.5 2.5 2.5 2.5 25 25 ...
##  $ PROPDMGEXP: chr  "K" "K" "K" "K" ...
##  $ CROPDMG   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CROPDMGEXP: chr  "" "" "" "" ...
##  $ WFO       : chr  "" "" "" "" ...
##  $ STATEOFFIC: chr  "" "" "" "" ...
##  $ ZONENAMES : chr  "" "" "" "" ...
```

```
## $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
## $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
## $ LATITUDE_E: num  3051 0 0 0 0 ...
## $ LONGITUDE_: num  8806 0 0 0 0 ...
## $ REMARKS   : chr  "" "" "" "" ...
## $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
```

Transform the data

From the related documentation, it appears that the only the following variables are useful for this study:

- **EVTYPE**, the event type
- **FATALITIES**, the number of fatalities
- **INJURIES**, the number of injuries
- **PROPDMG**, the amount of property damage
- **PROPDMGEXP**, the exponent of the property damage
- **CROPDMG**, the amount of crop damage
- **CROPDMGEXP**, the exponent of the crop damage

```r
#subset data for only relevent columns
data<- data[c("EVTYPE","FATALITIES","INJURIES","PROPDMG","PROPDMGEXP",
              "CROPDMG","CROPDMGEXP")]
```

According to the National Weather Service Storm Data Documentation, (*Section2.1.1 - Storm Data Event Table*), there are 48 event types (EVTYPE).

However, the count of the unique values for EVTYPE reveal that there are actually 985 distinct event types in the data. Inspection of the data reveals that the there is case-sensitivity, spelling errors, and other information that was added to the EVTYPE.

The package stringdist::amatch() was used in an attempt to find the approximate match. Note that the parameters of amatch() used were selected simply following suggestions in Stack Overflow String Matching. Other parameters were attempted but not much effort spent in trying to optimize this process.

```r
print("number of event types before processing:")
```

```
## [1] "number of event types before processing:"
```

```r
length(unique(data$EVTYPE))
```

```
## [1] 985
```

```r
#convert EVTYPE to lower case to help improve matching and reduce redundancy
data<- data %>% mutate(EVTYPE = tolower(EVTYPE))

# EVTYPES copied from 2.1.1 Storm Data Event Table
```

```
#     (again, convert to lower case to improve matching)
event_names <- tolower(c("Astronomical Low Tide", "Avalanche", "Blizzard",
                 "Coastal Flood", "Cold/Wind Chill", "Debris Flow", "Dense Fog",
                 "Dense Smoke", "Drought", "Dust Devil", "Dust Storm",
                 "Excessive Heat", "Extreme Cold/Wind Chill", "Flash Flood",
                 "Flood", "Frost/Freeze", "Funnel Cloud", "Freezing Fog",
                 "Hail", "Heat", "Heavy Rain", "Heavy Snow", "High Surf",
                 "High Wind", "Hurricane (Typhoon)", "Ice Storm",
                 "Lake-Effect Snow", "Lakeshore Flood ", "Lightning",
                 "Marine Hail", "Marine High Wind", "Marine Strong Wind",
                 "Marine Thunderstorm Wind", "Rip Current", "Seiche", "Sleet",
                 "Storm Surge/Tide", "Strong Wind", "Thunderstorm Wind",
                 "Tornado", "Tropical Depression", "Tropical Storm",
                 "Tsunami", "Volcanic Ash", "Waterspout", "Wildfire",
                 "Winter Storm", "Winter Weather"))

data$EVTYPE <- data$EVTYPE <- fct_explicit_na(factor(
          amatch(data$EVTYPE, table = event_names, method='osa',maxDist=4),
          levels = 1:48, labels = event_names),
          na_level = "unknown")

print("number of event types after processing:")
```

```
## [1] "number of event types after processing:"
```

```
length(unique(data$EVTYPE))
```

```
## [1] 48
```

Process the Population Health data

It is assumed that the public health factors that are of interest would be the total number of inujuries, and the total fatalities. Summarize the data to find each total by the Event Type. This is done by using *dplyr*. The most harmful events are what is of interest, therefore only the top ten are kept. However, it would be easy to keep all 48, if it were required for future studies. *tidyr::gather()* is then used to create a tidy dataset.

```
#Subset for factors affecting population health
populationHealth <- data %>%
      group_by(EVTYPE) %>%
      summarize(injuries = sum(INJURIES), fatalities = sum(FATALITIES))

# Distill the Top 10 data, for both injuries and fatalities.

# add the factors together
populationHealth$sum <- populationHealth$injuries + populationHealth$fatalities
#order by sum
populationHealth <- populationHealth[order(-populationHealth$sum),]
#remove sum column (extraneous information)
populationHealth <- populationHealth[-4]
#select the top ten
populationHealth <- populationHealth[1:10,]
```

```r
#gather the data to convert to long format
populationHealth <- gather(populationHealth, "fatalities", "injuries",
                           key = "incident", value = "cases")
populationHealth <- populationHealth[order(-populationHealth$cases),]
head(populationHealth)
```

```
## # A tibble: 6 x 3
##   EVTYPE         incident   cases
##   <fct>          <chr>      <dbl>
## 1 tornado        injuries   91364
## 2 high wind      injuries    8397
## 3 flood          injuries    7909
## 4 excessive heat injuries    6527
## 5 tornado        fatalities  5633
## 6 lightning      injuries    5232
```

Process the Economic Damage data

As per the information sheet describing the data, the *xDMG* and *xDMGEXP* variables together indicate the economic cost. xDMG is the value and xDMGEXP is the exponent, or multiplier. However, the xDMGEXP is not given in numeric values, but in acronyms.

> Estimates should be rounded to three significant digits, followed by an alphabetical character signifying the magnitude of the number, i.e., 1.55B for $1,550,000,000. Alphabetical characters used to signify magnitude include “K” for thousands, “M” for millions, and “B” for billions.
>
> — National Weather Service Storm Data Documentation - Section2.7 - STORM DATA PREPARATION

Examine the possible values within the xDMGEXP variables:

```r
unique(data$PROPDMGEXP)
```

```
##  [1] "K" "M" ""  "B" "m" "+" "0" "5" "6" "?" "4" "2" "3" "h" "7" "H" "-" "1" "8"
```

```r
unique(data$CROPDMGEXP)
```

```
## [1] ""  "M" "K" "m" "B" "?" "0" "k" "2"
```

As can be seen above, there are many entries that include other characters besides those described within the documentation (“K”, “M”, or “B”). Most of the other characters are numeric. Therefore, my assumption is that the character is always representative of the scientific notation exponent.

xDMGEXP	meaning	numeric	value
0	base	0	10^0
1		1	10^1
H, h, 2	hecto	2	10^2
K, k, 3	kilo	3	10^3
4		4	10^4
5		5	10^5

xDMGEXP	meaning	numeric	value
M, m, 6	mega	6	10^6
7		7	10^7
8		8	10^8
B, b, 9	giga	9	10^9

There are other characters, but for now conider them to mean it is a multiplier of 1.

AS with Population Health, the data is summarize to find each total by the Event Type. Only the ten most harmful events are kept and then the data is converted to a tidy dataset.

```
#Convert multiplier to numeric
data$PROPDMGEXP <- as.numeric(recode(data$PROPDMGEXP,
                        '0'           ='1e+0',
                        '1'           ='1e+01',
                        'H'           ='1e+02',
                        'h'           ='1e+02',
                        '2'           ='1e+02',
                        'K'           ='1e+03',
                        'k'           ='1e+03',
                        '3'           ='1e+03',
                        '4'           ='1e+04',
                        '5'           ='1e+05',
                        'M'           ='1e+06',
                        'm'           ='1e+06',
                        '6'           ='1e+06',
                        '7'           ='1e+07',
                        '8'           ='1e+08',
                        'B'           ='1e+09',
                        'b'           ='1e+09',
                        '9'           ='1e+09',
                        .default      ='1'))

data$CROPDMGEXP <- as.numeric(recode(data$CROPDMGEXP,
                        '0'           ='1e+0',
                        '1'           ='1e+01',
                        'H'           ='1e+02',
                        'h'           ='1e+02',
                        '2'           ='1e+02',
                        'K'           ='1e+03',
                        'k'           ='1e+03',
                        '3'           ='1e+03',
                        '4'           ='1e+04',
                        '5'           ='1e+05',
                        'M'           ='1e+06',
                        'm'           ='1e+06',
                        '6'           ='1e+06',
                        '7'           ='1e+07',
                        '8'           ='1e+08',
                        'B'           ='1e+09',
                        'b'           ='1e+09',
                        '9'           ='1e+09',
                        .default      ='1'))
```

```
#Mulitply value by the exponent
data$PropDmgTotl <- data$PROPDMG * data$PROPDMGEXP
data$CropDmgTotl <- data$CROPDMG * data$CROPDMGEXP

#Subset for factors affecting economic data
econDamage <- data %>%
      group_by(EVTYPE) %>%
      summarize(property = sum(PropDmgTotl), crop = sum(CropDmgTotl))

# Distill the Top 10 data, for both property and crop damage

# add the factors together
econDamage$sum <- econDamage$property + econDamage$crop
#order by sum
econDamage <- econDamage[order(-econDamage$sum),]
#remove sum
econDamage <- econDamage[1:3]
#select the top ten
econDamage <- econDamage[1:10,]


#gather the data to convert to long format
econDamage <- gather(econDamage, "property", "crop", key = "incident", value = "value")
econDamage <- econDamage[order(-econDamage$value),]
head(econDamage)
```

```
## # A tibble: 6 x 3
##   EVTYPE             incident         value
##   <fct>              <chr>            <dbl>
## 1 flood              property 144810802357
## 2 unknown            property  74498066629.
## 3 hurricane (typhoon) property  69305840000
## 4 tornado            property  56952254426.
## 5 flash flood        property  17139746082.
## 6 hail               property  15741123563.
```

Results

Population Health

Question: Across the United States, which types of events are most harmful with respect to population health?

Assume that there is no weighting for fatalities vs injuries. I.e., it is the *total* number of all incidents affecting health that are important. Therefore, as per the following chart, the event that causes the largest impact to population health are **tornados** with **91346 total incidents**.

```
ggplot(data=populationHealth,
       aes(x=reorder(EVTYPE, -cases), y = cases, fill = incident)) +
  geom_bar(stat="identity") +
```

```
  theme(axis.text.x = element_text(angle=90)) +
  ggtitle("Population Health Inflicted by Different Types of Storms",
    subtitle = "For both Fatalites and Injuries") +
  xlab("Storm Type") +
  ylab("Number of Incidents")
```

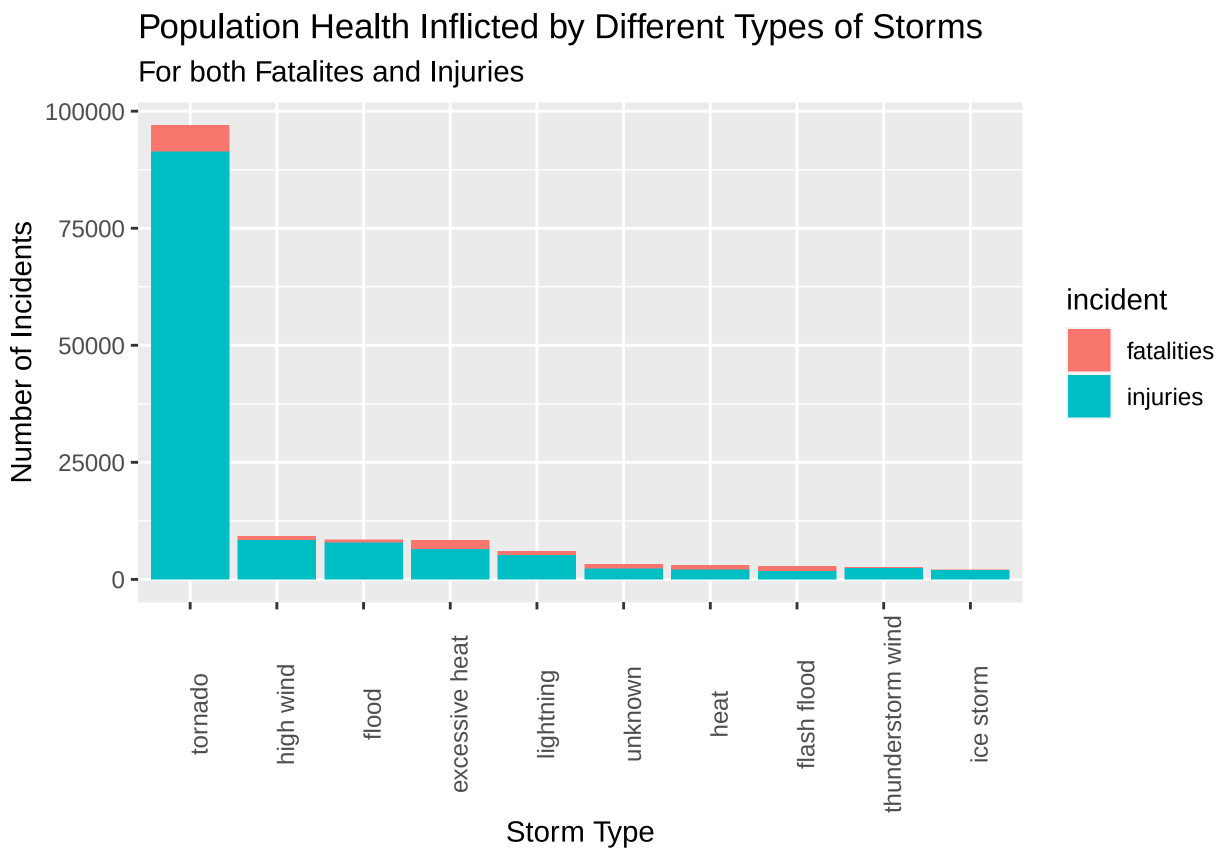

Figure 1: Fig 1. How Severe Weather Events Affect Population Health.

Economic Damage

Question: Across the United States, which types of events have the greatest economic consequences?

As indicated in the following chart, the event that causes the largest economic impact are **floods** with over **144 billion dollars in total damages**.

```
ggplot(data=econDamage,
       aes(x=reorder(EVTYPE, -value), y = value/1e+9, fill = incident)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle=90)) +
  ggtitle("Economic Damage Inflicted by Different Types of Storms",
    subtitle = "For Damages to Property and Crops") +
  xlab("Storm Type") +
  ylab("Cost of Damage (Billion $)")
```

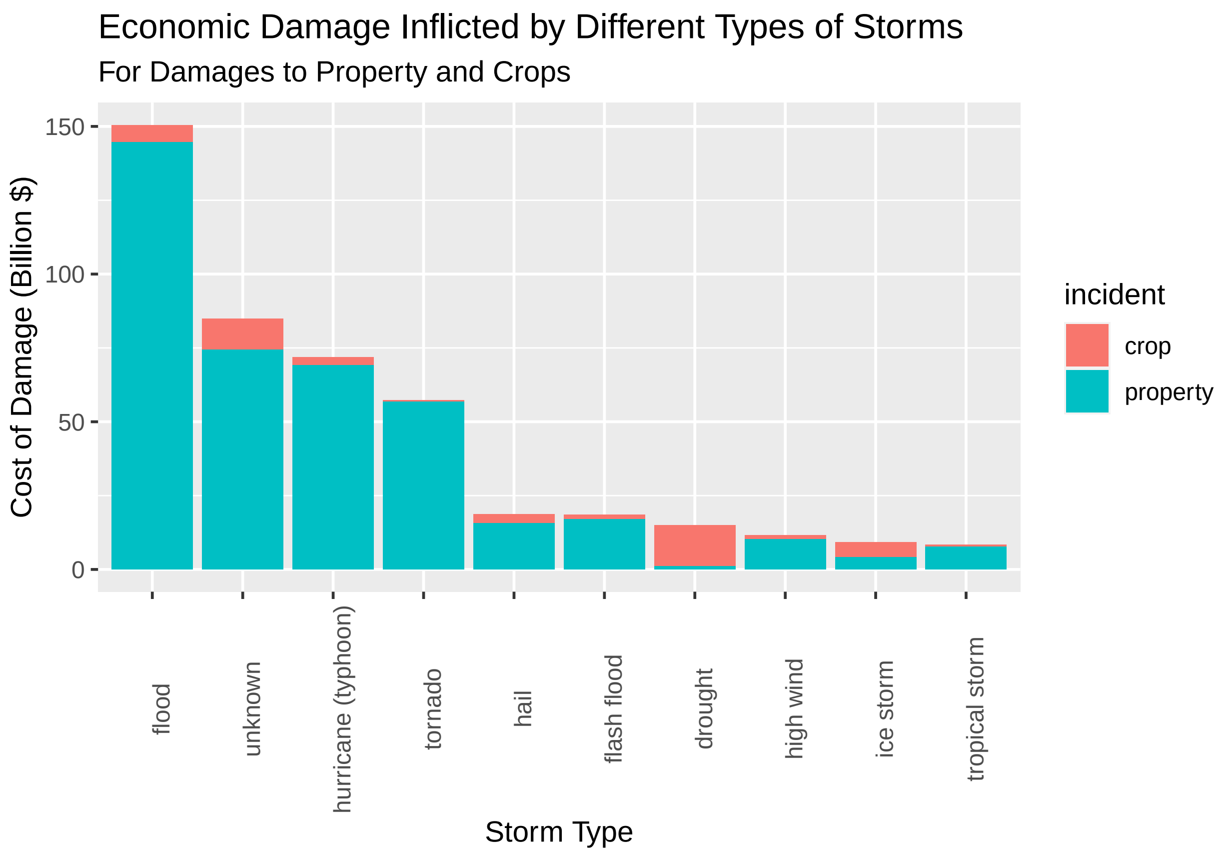

Figure 2: Fig 2. How Severe Weather Events Affect Economic Health.

Note About Results

- This report relies heavily upon both the event type (EVTYPE) being accurate and the costs being entered correctly. Both fields are questionable and further refinement would need to be made to be able to draw reliable conclusions.
- The EVTYPE entries are not in the format dictated by the specification. The specification states that event name should be the one that most accurately describes the meteorological event. However, from the data it appears that multiple events were included in single entires. Therefore further work needs to be completed in order to determine which is the appropriate event to use (or divide up those data points).
- Most entries do not follow the specified format, often employing abbreviations. A mapping could be created to line up abbreviations with their appropriate event name. However, according to the specification there are multiple similar events; such as Cold/Wind Chill vs Extreme Cold/Wind Chill which further complicates how to determine which was the intended EVTYPE.
- The specification identifies how to utilize the cost exponent field, which does not appear to have been followed. Assumptions were made on how to deal with this but those assumptions introduce error into the analysis.
- The data examined is from 1950 to 2011; in the earlier years of this dataset there are generally fewer events recorded. However, no attempts were made to compensate for the fewer recorded events.

Environment

The following is the environment in which this was run:

```r
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.1252  LC_CTYPE=English_Canada.1252
## [3] LC_MONETARY=English_Canada.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tufte_0.6          stringdist_0.9.5.5 forcats_0.5.0      stringr_1.4.0
##  [5] dplyr_0.8.5        purrr_0.3.4        readr_1.3.1        tidyr_1.0.2
##  [9] tibble_3.0.1       ggplot2_3.3.0      tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.0.0 xfun_0.13        haven_2.2.0      lattice_0.20-38
##  [5] colorspace_1.4-1 vctrs_0.2.4      generics_0.0.2   htmltools_0.4.0
##  [9] yaml_2.2.1       utf8_1.1.4       rlang_0.4.5      pillar_1.4.3
## [13] glue_1.4.0       withr_2.2.0      DBI_1.1.0        dbplyr_1.4.3
## [17] modelr_0.1.7     readxl_1.3.1     lifecycle_0.2.0  munsell_0.5.0
## [21] gtable_0.3.0     cellranger_1.1.0 rvest_0.3.5      codetools_0.2-16
## [25] evaluate_0.14    labeling_0.3     knitr_1.28       parallel_3.6.3
## [29] fansi_0.4.1      highr_0.8        broom_0.5.6      Rcpp_1.0.4.6
## [33] scales_1.1.0     backports_1.1.6  jsonlite_1.6.1   farver_2.0.3
## [37] fs_1.4.1         hms_0.5.3        digest_0.6.25    stringi_1.4.6
## [41] grid_3.6.3       cli_2.0.2        tools_3.6.3      magrittr_1.5
## [45] crayon_1.3.4     pkgconfig_2.0.3  ellipsis_0.3.0   xml2_1.3.2
## [49] reprex_0.3.0     lubridate_1.7.8  assertthat_0.2.1 rmarkdown_2.1
## [53] httr_1.4.1       rstudioapi_0.11  R6_2.4.1         nlme_3.1-144
## [57] compiler_3.6.3
```

```r
packageVersion("tidyverse")
```

```
## [1] '1.3.0'
```

```r
packageVersion("stringdist")
```

```
## [1] '0.9.5.5'
```