Funding and Publication of Research on Gun Violence and Other Leading Causes of Death

David E. Stark, MD, MS; Nigam H. Shah, MBBS, PhD

January 3, 2017

Online Supplement

This R Markdown document contains detailed methods and annotated code associated with Stark DE, Shah NH. Funding and Publication of Research on Gun Violence and Other Leading Causes of Death. *JAMA*. 2017;317(1) and enables fully reproducible research.

The script `gun_violence_annotated_code.Rmd` should be run within the same working directory as the accompanying file, `Compressed Mortality, 2004-2014.txt`. Internet access is required, as the code will retrieve data from the MEDLINE and Federal RePORTER databases. Running this `.Rmd` script in RStudio will take approximately 4-6 hours to complete on a standard machine.

---

load required libraries

```
# load libraries
require(RCurl)
require(XML)
require(ggplot2)
require(scales)
library(pander)
require(dplyr)
require(RColorBrewer)
require(ggrepel)
require(SPARQL)
require(gridExtra)
```

CDC Mortality Rates

CDC mortality statistics were accessed from 2004 to 2014 (the most recent year available). Results were grouped by ‘Injury Mechanism & All Other Leading Causes’ and sorted by mortality rate. 13 nonspecific causes of death were excluded (see below) and the top 30 causes of death were retained for further analysis.

Nonspecific causes of death excluded from analysis:

```
All other diseases (Residual); Symptoms, signs and abnormal clinical and laboratory
findings, not elsewhere classified; Other diseases of respiratory system; Other diseases
of the circulatory system; In situ neoplasms, benign neoplasms and neoplasms of uncertain
or unknown behavior; Certain conditions originating in the perinatal period; Congenital
malformations, deformations and chromosomal abnormalities; Unspecified Injury; Other and
unspecified infectious and parasitic diseases and their sequelae; Other disorders of
circulatory system; Complications of medical and surgical care; Other specified,
classifiable Injury; Other specified, not elsewhere classified Injury
```

CDC-derived causes of death were manually mapped to their corresponding Medical Subject Heading (MeSH) term(s). Ambiguous mappings were resolved by inspecting ICD-10 codes associated with a particular cause of death.

The downloaded CDC file `Compressed Mortality, 2004-2014.txt` was annotated with 4 additional columns prior to importing for analysis:

- `Remove` Flag indicating nonspecific causes of death for removal
- `MeSH.Terms` Mapped term(s) corresponding to CDC-derived cause of death
- `MeSH.IDs` Corresponding MeSH Unique ID(s)
- `Abbreviation` Abbreviated term used for plots

Code to import CDC mortality data

```r
# import CDC mortality data with manually mapped MeSH terms
mortality <- read.delim("Compressed Mortality, 2004-2014.txt",
    stringsAsFactors = FALSE)

# remove nondescript causes of death
mortality <- filter(mortality, Remove == FALSE)

# Create 'Cause' column defining injury versus non-injury
for (row in 1:nrow(mortality)) {
    if (substr(mortality$Injury.Mechanism...All.Other.Leading.Causes[row],
        start = 1, stop = 10) == "Non-Injury") {
        mortality$Cause[row] <- "Non-Injury"
    } else {
        mortality$Cause[row] <- "Injury"
    }
}

# convert multiple MeSH queries into list, remove
# leading/trailing whitespace, convert to upper
convertList <- function(terms) {
    as.list(toupper(trimws(strsplit(terms, ";")[[1]])))
}
mortality$MeSH.Terms <- sapply(mortality$MeSH.Terms, convertList)
mortality$MeSH.IDs <- sapply(mortality$MeSH.IDs, convertList)
```

MEDLINE Publication Volume

For each cause of death, MEDLINE was queried for the total number of publications between 2004 and 2015 indexed with the corresponding MeSH term(s) including descendant terms (terms subsumed under a parent term within the MeSH hierarchy).

This was performed using the MEDLINE E-utilities API and the code below.

Code to import MEDLINE publication data

```r
# Return total number of articles for each set of MeSH
# queries

# Generate PubMed query
mortality$PubMed.Query <- sapply(mortality$MeSH.Terms, paste,
    "[mesh]", sep = "", collapse = " OR ")

getPubmedTrend <- function(query, minYear = 2004, maxYear = 2015) {
    # Retreives PubMed trend (counts results by year).  Args:
    # query: <string> Search query minYear: <int> minimum year to
    # return maxYear: <int> maximum year to return Returns: A
    # table containing year, count

    # PubMed EUtils URL for retrieving search results counts
    pubmed <- paste("https://eutils.ncbi.nlm.nih.gov/entrez/eutils/",
        "esearch.fcgi?db=pubmed;rettype=count;term=", sep = "")
    # encode query as a URL
    query <- URLencode(query)
    curl <- getCurlHandle()
    output <- data.frame(Year = minYear:maxYear, Count = 0)
    # retrieve counts for each year in range
    for (i in minYear:maxYear) {
        query_year <- paste(query, "+AND+", i, "%5Bppdat%5D",
            sep = "")
        result <- getURL(paste(pubmed, query_year, sep = ""),
            curl = curl)
        result <- xmlTreeParse(result, asText = TRUE)
        count <- as.numeric(xmlValue(result[["doc"]][["eSearchResult"]][["Count"]]))
        output$Count[output$Year == i] <- count
    }

    return(output)
}

# For each cause of death, run PubMed query and sum total
# results over 2004-2015
mortality$Publications <- sapply(mortality$PubMed.Query, function(x) colSums(getPubmedTrend(x,
    2004, 2015))[2])
```

Federal RePORTER Funding Data

Research funding data from 2004 to 2015 (all years available) was accessed from Federal RePORTER, a database of projects funded by U.S. federal agencies. Projects are indexed using the computerized Research, Condition, and Disease Categorization system derived in part from MeSH. For each cause of death, Federal RePORTER was queried for the total funding awarded to projects containing corresponding MeSH terms, including descendant terms.

Code to import Federal RePORTER funding data

```
# Import Federal ExPORTER data

FedReporter <- NULL
for (year in 2004:2015) {
    temp <- tempfile()
    download.file(paste("https://federalreporter.nih.gov/FileDownload/",
        "DownloadFile?fileToDownload=FedRePORTER_PRJ_C_FY", year,
        ".zip", sep = ""), temp)
    data <- read.csv(unz(temp, paste("FedRePORTER_PRJ_C_FY",
        year, ".csv", sep = "")), stringsAsFactors = FALSE, header = FALSE,
        skip = 1)
    unlink(temp)
    FedReporter <- rbind(FedReporter, data)
}
colnames(FedReporter) <- c("SM_Application_ID", "Project_Terms",
    "Project_Title", "Department", "Agency", "IC_Center", "Project_Number",
    "Project_Start_Date", "Project_End_Date", "Contact_PI_Project_Leader",
    "Other_PIs", "Congressional_District", "DUNS_Number", "Organization_Name",
    "Organization_City", "Organization_State", "Organization_Zip",
    "Organization_Country", "Budget_Start_Date", "Budget_End_Date",
    "CFDA_Code", "FY", "FY_Total_Cost", "FY_Total_Cost_Sub_Projects")

# convert project terms to list and all caps
FedReporter$Project_Terms <- sapply(FedReporter$Project_Terms,
    convertList)
# convert funding NAs to zeros
FedReporter$FY_Total_Cost[is.na(FedReporter$FY_Total_Cost)] <- 0
```

MeSH term expansion

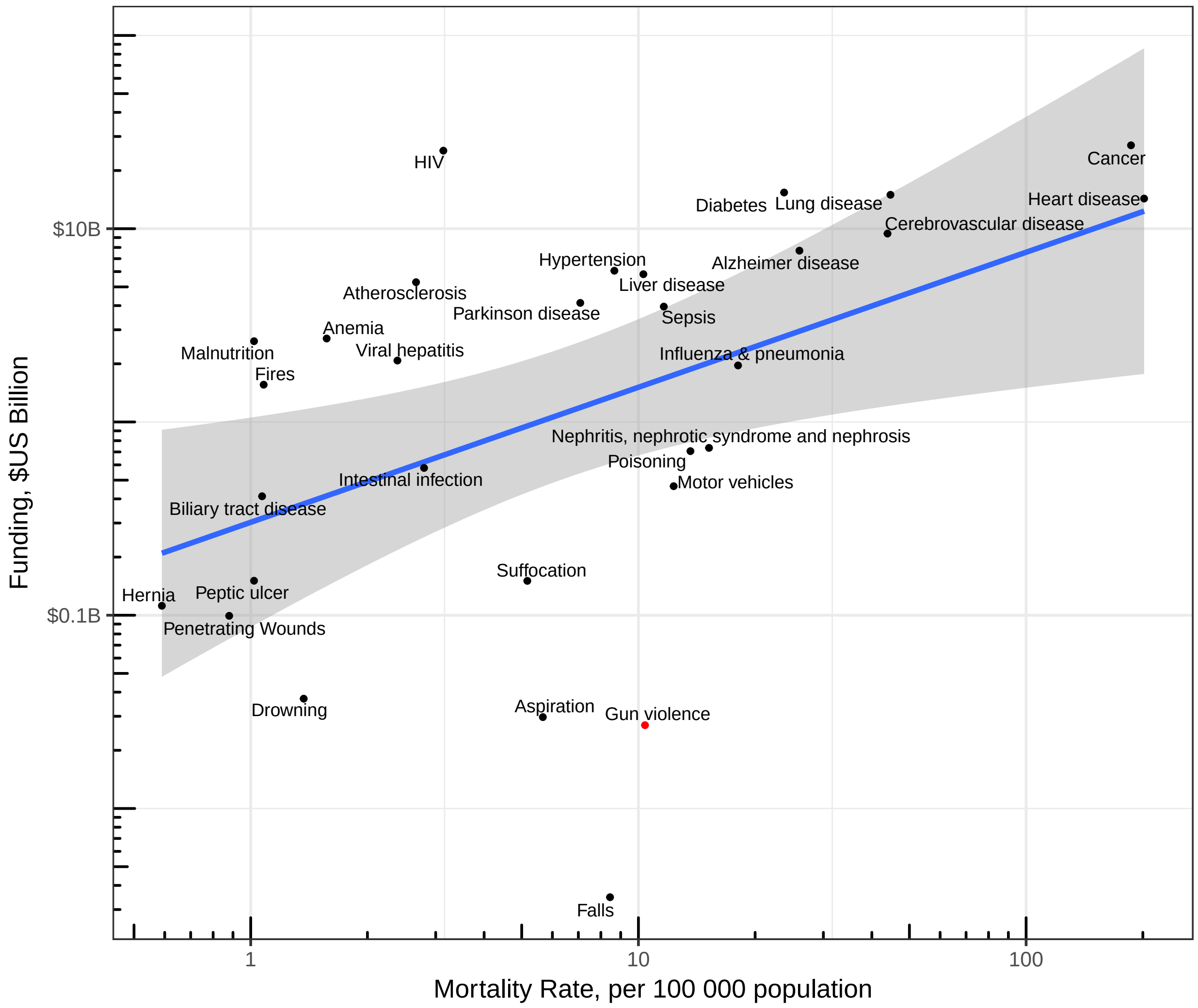
In order to ensure complete coverage of search terms, MeSH terms were expanded to include all descendant terms (MEDLINE does this automatically in its queries but Federal RePORTER does not.) The MeSH SPARQL endpoint was used to perform MeSH term expansion.

```
# For each MeSH query (or set of MeSH queries) return list of
# descendant queries

stripExtra <- function(term) {
    substr(term, 2, nchar(term) - 4)
}

getChildren <- function(term) {
    endpoint <- "https://id.nlm.nih.gov/mesh/sparql"

    # Query to retrieve all synonym terms of the input term
    query <- paste("PREFIX mesh: <http://id.nlm.nih.gov/mesh/>
PREFIX mesh2015: <http://id.nlm.nih.gov/mesh/2015/>
                  PREFIX mesh2016: <http://id.nlm.nih.gov/mesh/2016/>
                  PREFIX meshv: <http://id.nlm.nih.gov/mesh/vocab#>
                  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

                  SELECT  DISTINCT ?labelA
                  FROM <http://id.nlm.nih.gov/mesh>
                  WHERE {{

                  mesh:",
        term, " rdfs:label ?labelA .
                  } UNION
                  {
                  mesh:",
        term, " meshv:treeNumber ?treeNum .
                  ?childTreeNum meshv:parentTreeNumber+ ?treeNum .
                  ?descriptorA meshv:treeNumber ?childTreeNum .
                  ?descriptorA rdfs:label ?labelA .
                  }}",
        sep = "")
    df <- SPARQL(endpoint, query, extra = "format=HTML&inference=TRUE")$results
    colnames(df) <- NULL
    return(toupper(sapply(df, stripExtra)))
}

for (row in 1:nrow(mortality)) {
    mortality$MeSH.Children[row] <- paste(unlist(sapply(mortality$MeSH.IDs[[row]],
        getChildren)), collapse = ";")
}

mortality$MeSH.Children <- sapply(mortality$MeSH.Children, convertList)

# function to invert strings with commas
invertCommas <- function(term) {
    s1 <- (strsplit(term, ", "))[[1]]  # split at commas
    output <- paste(rev(s1), collapse = " ")  # reverse and collapse
    return(output)
}

for (causeNum in 1:length(mortality$MeSH.Children)) {
    for (termNum in 1:length(mortality$MeSH.Children[[causeNum]])) {
        mortality$MeSH.Children[[causeNum]][[termNum]] <- invertCommas(mortality$MeSH.Children[[causeNu
    }
}

# For each bundled set of children queries, search Federal
# RePORTER and return total funding

mortality$Total.Funding <- 0
mortality$Total.Projects <- 0
for (row in 1:nrow(mortality)) {
    # FedReporter match for 'Fires' is 'FIRE - DISASTERS'
    # FedReporter match for 'ALZHEIMER DISEASE' is 'ALZHEIMER'S
    # DISEASE'
    terms <- mortality$MeSH.Children[[row]]
    # append new terms
    if (row == which(mortality$Abbreviation == "Alzheimer disease")) {
        terms[length(terms) + 1] <- "ALZHEIMER'S DISEASE"
    }
    if (row == which(mortality$Abbreviation == "Fires")) {
        terms[length(terms) + 1] <- "FIRE - DISASTERS"
    }
    matchTerm <- vapply(FedReporter$Project_Terms, function(x) {
        ifelse(length(intersect(terms, unlist(x)) != 0), TRUE,
            FALSE)
    }, TRUE, USE.NAMES = FALSE)
    mortality$Total.Funding[row] <- sum(FedReporter$FY_Total_Cost[matchTerm])
    mortality$Total.Projects[row] <- sum(matchTerm)
}

data <- select(mortality, Cause, Abbreviation, Crude.Rate, Publications,
    Total.Funding, Total.Projects)

# Sort by mortality rate and filter top 30 causes of death
# for inclusion
data <- slice(arrange(data, desc(Crude.Rate)), 1:30)

write.csv(x = data, file = "gun_violence_data_frame.csv")
```

Results

Code to plot funding and publication volume for leading causes of death

```
# formatting for funding axis labels
funding_format <- function(x) {
    return(paste("$", x, "B", sep = ""))
}

# plot log(mortality) x log(funding)
Fig1A_funding <- ggplot(data, aes(x = Crude.Rate, y = (Total.Funding/1e+09))) +
    stat_smooth(method = "lm") + geom_point(size = 0.75, color = as.numeric(data$Abbreviation ==
    "Gun violence") + 1) + scale_color_brewer(type = "qual",
    palette = "Set1") + geom_text_repel(aes(label = Abbreviation),
    size = 2.4, segment.size = 0.5, box.padding = unit(0.1, "lines")) +
    scale_y_log10(labels = funding_format) + scale_x_log10() +
    annotation_logticks() + theme_bw(base_size = 10) + labs(y = "Funding, $US Billion") +
    labs(x = "Mortality Rate, per 100 000 population")
Fig1A_funding
```



```
# plot log(mortality) x log(publications)
Fig1B_publications <- ggplot(data, aes(x = Crude.Rate, y = (Publications/1000))) +
    stat_smooth(method = "lm") + geom_point(size = 0.75, color = as.numeric(data$Abbreviation ==
    "Gun violence") + 1) + scale_color_brewer(type = "qual",
    palette = "Set1") + geom_text_repel(aes(label = Abbreviation),
    size = 2.4, segment.size = 0.5, box.padding = unit(0.1, "lines")) +
    scale_y_log10() + scale_x_log10() + annotation_logticks() +
    theme_bw(base_size = 10) + labs(y = "Publications, in Thousands") +
    labs(x = "Mortality Rate, per 100 000 population")
Fig1B_publications
```

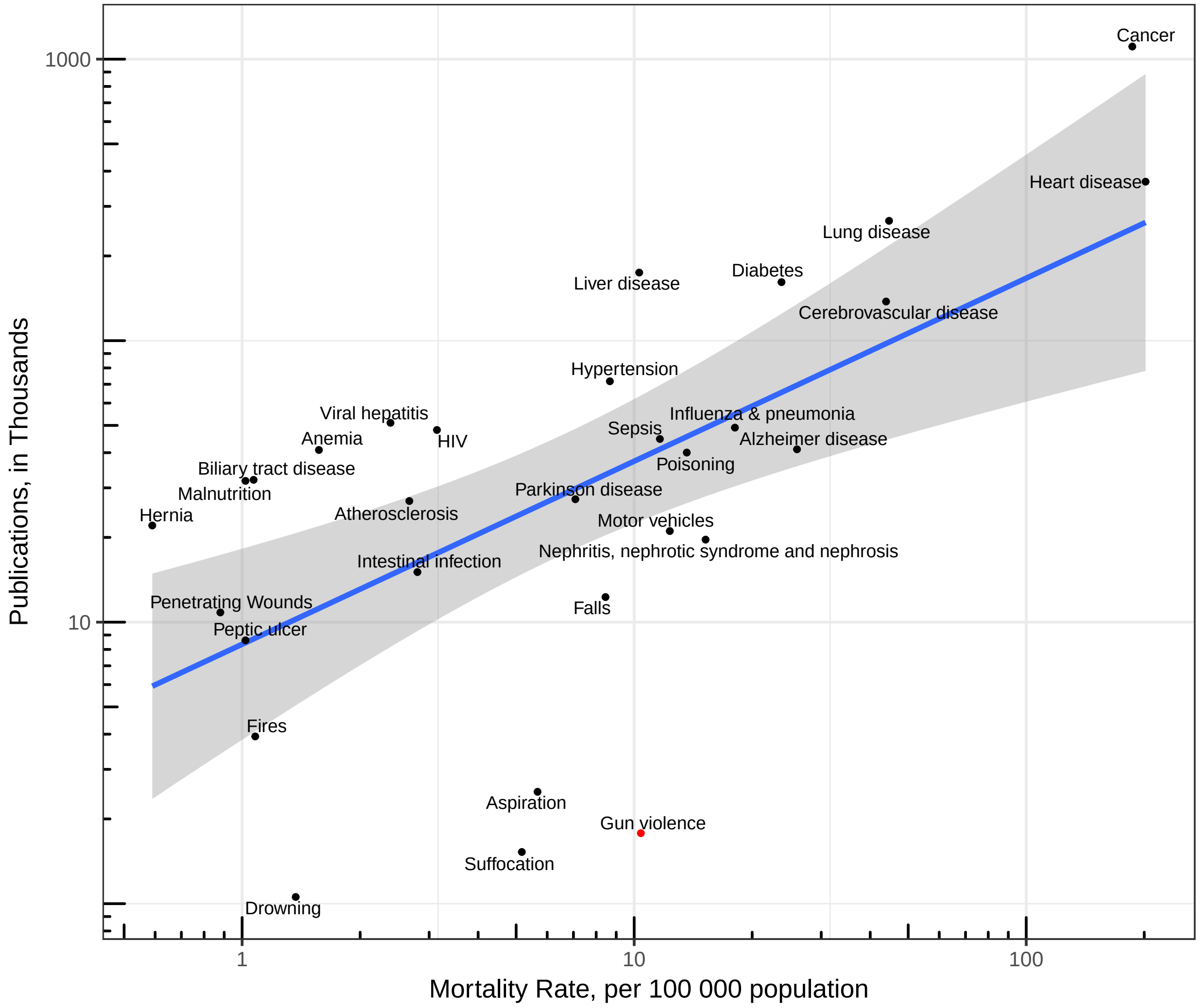


Figure 1. Mortality Rate vs Funding and Publication Volume for 30 Leading Causes of Death in the United States
HIV indicates human immunodeficiency virus. Shaded areas indicate 95% CIs. Plotting is on a log-log scale. Funding represents the total funding awarded over the years 2004 to 2015. Dollar amounts have not been correct for the year in which they were reported.

To determine how research funding and publication volume correlated with mortality, two linear regression analyses were performed using mortality rate as a predictor, and funding or publication count as outcomes. The predictor and outcomes were log-transformed and studentized residuals (residual divided by estimated standard error) were calculated to determine the extent to which a given cause of death was an outlier in terms of research funding or publication volume.

Code to calculate predicted funding, publication volume, and studentized residuals

```r
# Regress mortality rate on publications, calculate predicted
# values and residuals

lm.fit <- lm(log(Publications) ~ log(Crude.Rate), data = data)
data$Publications.Predicted <- predict(lm.fit)
```

```r
data$Publications.Residuals <- rstudent(lm.fit)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = log(Publications) ~ log(Crude.Rate), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0680 -0.6618  0.2085  1.0048  1.5239
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9.0295     0.3815  23.669  < 2e-16 ***
## log(Crude.Rate)   0.6503     0.1583   4.107 0.000315 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.309 on 28 degrees of freedom
## Multiple R-squared:  0.376,  Adjusted R-squared:  0.3537
## F-statistic: 16.87 on 1 and 28 DF,  p-value: 0.0003148
```

```r
# Regress mortality rate on funding, calculate predicted
# values and residuals

lm.fit <- lm(log(Total.Funding) ~ log(Crude.Rate), data = data)
data$Funding.Predicted <- predict(lm.fit)
data$Funding.Residuals <- rstudent(lm.fit)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = log(Total.Funding) ~ log(Crude.Rate), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9579 -1.0057  0.5303  1.3246  3.6281
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19.5280     0.6093  32.051   <2e-16 ***
## log(Crude.Rate)   0.6986     0.2528   2.763     0.01 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.09 on 28 degrees of freedom
## Multiple R-squared:  0.2143, Adjusted R-squared:  0.1862
## F-statistic: 7.635 on 1 and 28 DF,  p-value: 0.01
```

Table 1: Publication Residuals.

Cause of Death	Publications	Predicted Publications	Residual
Heart disease	367,021	262,987	0.28
Cancer	1,108,824	250,009	1.29
Lung disease	266,455	98,741	0.79

Cause of Death	Publications	Predicted Publications	Residual
Cerebrovascular disease	137,724	97,631	0.27
Alzheimer disease	41,112	69,472	-0.41
Diabetes	161,406	65,486	0.7
Influenza & pneumonia	49,092	54,808	-0.08
Nephritis, nephrotic syndrome and nephrosis	19,643	48,998	-0.71
Poisoning	40,032	45,604	-0.1
Motor vehicles	21,068	42,746	-0.54
Sepsis	44,742	41,152	0.06
Gun violence	1,780	38,267	-2.63
Liver disease	174,556	38,027	1.2
Hypertension	71,727	33,997	0.57
Falls	12,278	33,434	-0.77
Parkinson disease	27,334	29,801	-0.07
Aspiration	2,497	25,794	-1.9
Suffocation	1,526	24,291	-2.31
HIV	48,165	17,564	0.78
Intestinal infection	15,062	16,302	-0.06
Atherosclerosis	26,933	15,806	0.41
Viral hepatitis	51,067	14,708	0.97
Anemia	40,892	11,191	1.02
Drowning	1,056	10,242	-1.88
Fires	3,928	8,774	-0.63
Biliary tract disease	32,029	8,721	1.04
Malnutrition	31,760	8,454	1.06
Peptic ulcer	8,620	8,454	0.02
Penetrating Wounds	10,828	7,680	0.27
Hernia	22,050	5,922	1.07

Table 2: Funding Residuals.

Cause of Death	Funding	Predicted Funding	Residual
Heart disease	$ 14,321,858,796	$ 12,325,926,912	0.08
Cancer	$ 27,016,486,769	$ 11,673,635,026	0.44
Lung disease	$ 14,981,871,266	$ 4,302,601,404	0.62
Cerebrovascular disease	$ 9,437,338,094	$ 4,250,661,487	0.39
Alzheimer disease	$ 7,696,955,032	$ 2,949,067,366	0.47
Diabetes	$ 15,391,928,292	$ 2,767,677,507	0.84
Influenza & pneumonia	$ 1,961,448,140	$ 2,285,885,854	-0.07
Nephritis, nephrotic syndrome and nephrosis	$ 734,164,560	$ 2,026,634,939	-0.49
Poisoning	$ 706,501,911	$ 1,876,177,527	-0.47
Motor vehicles	$ 465,306,940	$ 1,750,179,433	-0.64
Sepsis	$ 3,953,262,529	$ 1,680,151,972	0.41
Gun violence	$ 26,985,174	$ 1,553,931,197	-2.09
Liver disease	$ 5,816,305,613	$ 1,543,477,074	0.64
Hypertension	$ 6,061,443,473	$ 1,368,449,332	0.72
Falls	$ 3,474,852	$ 1,344,095,579	-3.41
Parkinson disease	$ 4,131,670,585	$ 1,187,839,267	0.6
Aspiration	$ 29,703,780	$ 1,017,121,968	-1.79

Cause of Death	Funding	Predicted Funding	Residual
Suffocation	$ 150,602,842	$ 953,591,822	-0.9
HIV	$ 25,335,507,032	$ 673,074,126	1.85
Intestinal infection	$ 578,223,980	$ 621,283,907	-0.03
Atherosclerosis	$ 5,287,280,680	$ 600,987,308	1.07
Viral hepatitis	$ 2,078,698,410	$ 556,225,691	0.64
Anemia	$ 2,703,174,544	$ 414,714,470	0.92
Drowning	$ 36,997,030	$ 377,054,299	-1.16
Fires	$ 1,559,007,376	$ 319,327,182	0.79
Biliary tract disease	$ 412,671,814	$ 317,258,576	0.13
Malnutrition	$ 2,617,357,039	$ 306,826,570	1.07
Peptic ulcer	$ 150,842,111	$ 306,826,570	-0.35
Penetrating Wounds	$ 99,295,436	$ 276,756,268	-0.51
Hernia	$ 111,951,630	$ 209,310,456	-0.31

Code for residuals plot

```r
# plot funding residuals x publication residuals
fig2 <- ggplot(data, aes(x = Publications.Residuals, y = Funding.Residuals)) +
    geom_hline(yintercept = 0, size = 0.5, color = "gray") +
    geom_vline(xintercept = 0, size = 0.5, color = "gray") +
    geom_point(size = 0.75, color = as.numeric(data$Abbreviation ==
        "Gun violence") + 1) + scale_color_brewer(type = "qual",
    palette = "Set1") + geom_text_repel(aes(label = Abbreviation),
    size = 2.4, segment.size = 0.5, box.padding = unit(0.1, "lines")) +
    theme_bw(base_size = 10) + labs(y = "Funding, Studentized Residuals") +
    labs(x = "Publications, Studentized Residuals")
fig2
```

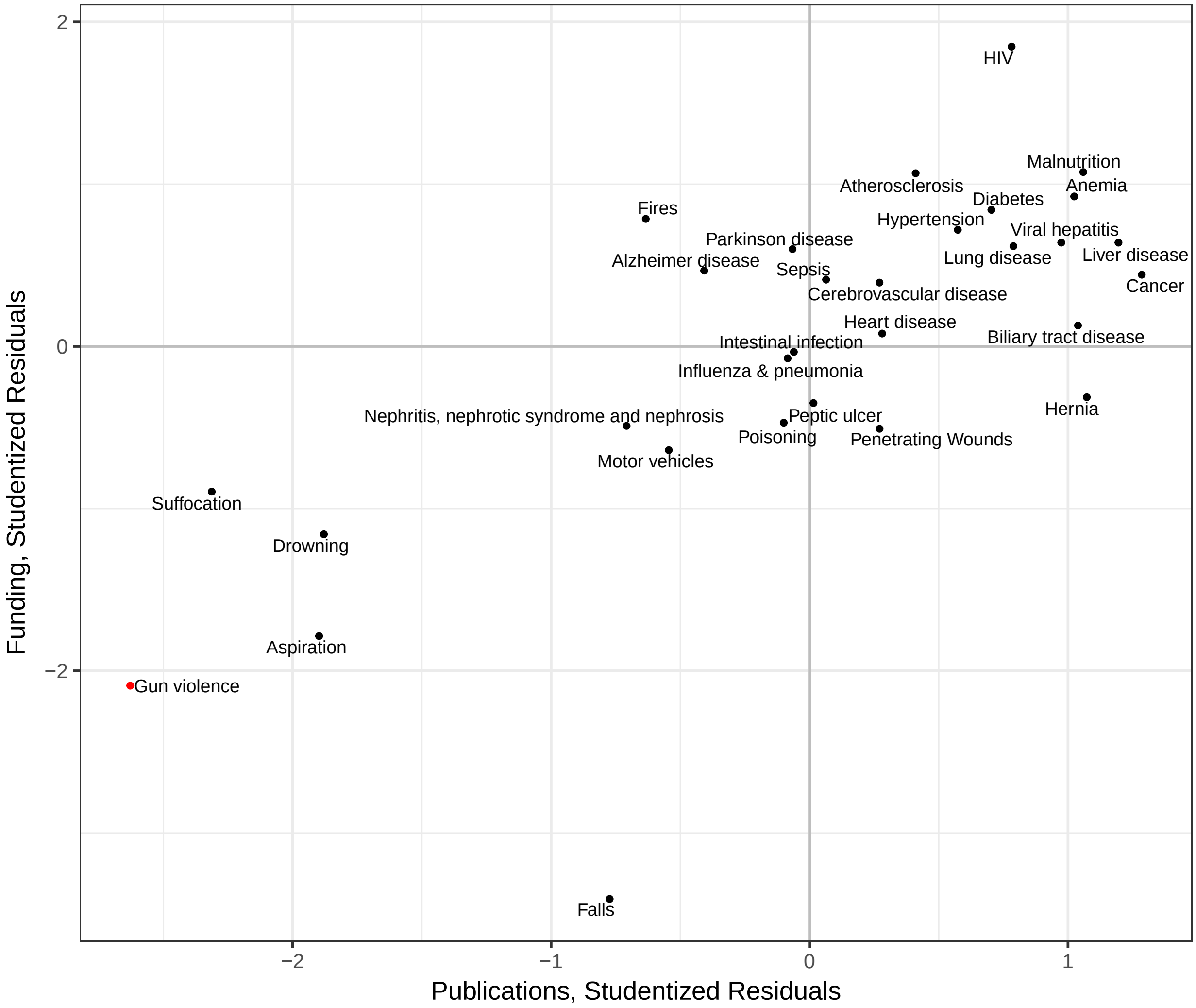


Figure 2. Studentized Residual Predicted vs Observed Funding and Publication Volumes for 30 Leading Causes of Death in the United States
HIV indicates human immunodeficiency virus. Mortality rate was used to predict funding and research volume. Studentized residuals (residual divided by estimated standard error) were calculated to give a standardized estimate of predicted vs observed funding and publication volume. The 4 quadrants represent observed funding greater than predicted, observed publication volume less than predicted (upper-left); observed funding and publication volume greater than predicted (upper-right); observed funding less than predicted, observed publication volume greater than predicted (lower-right); observed funding and publication volume less than predicted (lower-left).

“‘