Funding and publication of research on gun violence and other leading causes of death

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

August 18, 2016

CDC Mortality Rates

CDC mortality statistics (http://wonder.cdc.gov/cmf-icd10.html) 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 non-descript causes of death were excluded (see below) and the top 30 causes of death were retained for further analysis.

Non-descript causes of death excluded from analysis:

```
All other diseases (Residual); Symptoms, signs and abnormal clinical and laboratory findin
gs, not elsewhere classified; Other diseases of respiratory system; Other diseases of the
 circulatory system; In situ neoplasms, benign neoplasms and neoplasms of uncertain or unk
nown behavior; Certain conditions originating in the perinatal period; Congenital malforma
tions, deformations and chromosomal abnormalities; Unspecified Injury; Other and unspecifi
ed infectious and parasitic diseases and their sequelae; Other disorders of circulatory sy
stem; Complications of medical and surgical care; Other specified, classifiable Injury; Ot
her specified, not elsewhere classified Injury
```

CDC-derived causes of death were manually mapped to their corresponding Medical Subject Heading (MeSH) term(s) (http://www.ncbi.nlm.nih.gov/mesh). Ambiguous mappings were resolved by inspecting ICD-10 codes associated with a particular cause of death (http://wonder.cdc.gov/wonder/help/cmf.html#Injury%20Mechanism%20&%20All%20Other%20Leading%20Causes).

The downloaded CDC file (Compressed Mortality, 2004-2014.txt) was annotated with 4 additional columns prior to importing for analysis:
* Remove: Flag indicating non-descript 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

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

# remove overly broad/vague 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
convertList <- function(terms) {
  as.list(toupper(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 (http://www.ncbi.nlm.nih.gov/books/NBK25501/) API and the code below.

```
# 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 <- 'http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&rettype=c
ount&term='
  # encode query as a URL
  query <- URLencode(query)
  curl <- getCurlHandle()
  output <- data.frame(NULL)
  # 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 <- rbind(output, data.frame('Year' = i, 'Count' = count))
  }

  return(output)
}

# For each mechanism run PubMed query and sum total results over 2004-2015
mortality$Publications <- NA
x = 1
for(query in mortality$PubMed.Query) {
  mortality$Publications[x] <- colSums(getPubmedTrend(query, 2004, 2015))[2]
  x=x+1
}
```

Federal RePORTER Funding Data

Research funding data from 2004 to 2015 (all years available) was accessed from Federal RePORTER (https://federalreporter.nih.gov/projects/switchQueryForm?mode=Advanced), a database of projects funded by U.S. federal agencies. Projects are indexed using the computerized Research, Condition, and Disease Categorization (RCDC) 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.

```
# Import Federal ExPORTER data

# Code to download zip csv files from web
temp <- tempfile()
download.file("https://federalreporter.nih.gov/FileDownload/DownloadFile?fileToDownload=Fe
dRePORTER_PRJ_C_FY2004.zip",temp)
data <- read.csv(unz(temp, "FedRePORTER_PRJ_C_FY2004.csv"), stringsAsFactors = FALSE)
unlink(temp)

FedReporter <- NA
for(year in 2004:2015) {
  temp <- tempfile()
  download.file(paste('https://federalreporter.nih.gov/FileDownload/DownloadFile?fileToDow
nload=FedRePORTER_PRJ_C_FY',year, '.zip', sep = ''),temp)
  data <- read.csv(unz(temp, paste('FedRePORTER_PRJ_C_FY',year, '.csv', sep = '')), string
sAsFactors = FALSE, header = FALSE, skip = 1)
  unlink(temp)
  FedReporter <- rbind(FedReporter, data)
}
colnames(FedReporter) <- c('SM_Application_ID','Project_Terms','Project_Title','Departmen
t','Agency','IC_Center','Project_Number','Project_Start_Date','Project_End_Date','Contact_
PI_Project_Leader','Other_PIs','Congressional_District','DUNS_Number','Organization_Nam
e','Organization_City','Organization_State','Organization_Zip','Organization_Country','Bud
get_Start_Date','Budget_End_Date','CFDA_Code','FY','FY_Total_Cost','FY_Total_Cost_Sub_Proj
ects')

# 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.) We used the MeSH SPARQL endpoint (https://hhs.github.io/meshrdf/sparql-and-uri-requests.html) 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]], getChildr
en)), 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
[[causeNum]][[termNum]])
  }
}
```

```
# For each bundled set of children queries, search Federal RePORTER and return total fundi
ng

getFunding <- function(terms) {
  funding = 0
  projects = 0
  row = 1
  for (row in 1:nrow(FedReporter)) {
    if (length(intersect(terms, FedReporter$Project_Terms[row][[1]])) != 0) {
      funding = funding + FedReporter$FY_Total_Cost[row]
      projects = projects + 1
    }
  }
  return(c(funding, projects))
}

mortality$Total.Funding <- NA
mortality$Total.Projects <- NA
for (row in 1:nrow(mortality)) {
  result <- getFunding(mortality$MeSH.Children[[row]])
  mortality$Total.Funding[row] <- result[1]
  mortality$Total.Projects[row] <- result[2]
}
```

The above code runs over several hours. Therefore, we write the output to file to facilitate easier loading for subsequent analyses.

```
# write to file
write.csv(x = select(mortality, c(-MeSH.Terms, -MeSH.IDs, -MeSH.Children)), file = 'gun_vi
olence_results_8_18_2.csv')
```

```
# load file from above ETL process
mortality <- read.csv("~/Dropbox/gun violence/gun_violence_results_8_18_2.csv", stringsAsF
actors=FALSE)
data <- select(mortality, Cause, Abbreviation, Crude.Rate, Publications, Total.Funding, To
tal.Projects)

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

Results

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

# plot log(mortality) x log(publications)
pubs <- ggplot(data, aes(x = Crude.Rate, y = (Publications/1000))) + stat_smooth(method = "l
 + geom_point(size = 0.75, color = as.numeric(data$Abbreviation=='Gun Violence')+1) + scal
e_color_brewer(type = 'qual', palette = 'Set1') + geom_text_repel(aes(label = Abbreviation),
 size = 3, segment.size = 0, box.padding = unit(0.1, "lines"))  + scale_y_log10() + scale_
x_log10() + annotation_logticks() + theme_bw(base_size = 10) + labs(y = "Publications (tho
usands)") + labs(x = "Morality Rate (per 100,000 population)") + theme(aspect.ratio=1)

# plot log(mortality) x log(funding)
funding <- ggplot(data, aes(x = Crude.Rate, y = (Total.Funding/1000000000))) + stat_smooth(m
thod = "lm") + geom_point(size = 0.75, color = as.numeric(data$Abbreviation=='Gun Violenc
e')+1) + scale_color_brewer(type = 'qual', palette = 'Set1') + geom_text_repel(aes(label =
 Abbreviation), size = 3, segment.size = 0, box.padding = unit(0.1, "lines"))  + scale_y_l
og10(labels = funding_format) + scale_x_log10() + annotation_logticks() + theme_bw(base_si
ze = 10) + labs(y = "Funding") + labs(x = "Morality Rate (per 100,000 population)") + them
e(aspect.ratio=1)

# Combine funding and publication panels in one plot
# Code to set equal panel widths
gPubs <- ggplotGrob(pubs)
gFunding <- ggplotGrob(funding)
gPubs$widths <- gFunding$widths
gPubs$heights <- gFunding$heights
grid.arrange(gFunding, gPubs, ncol = 2)
```

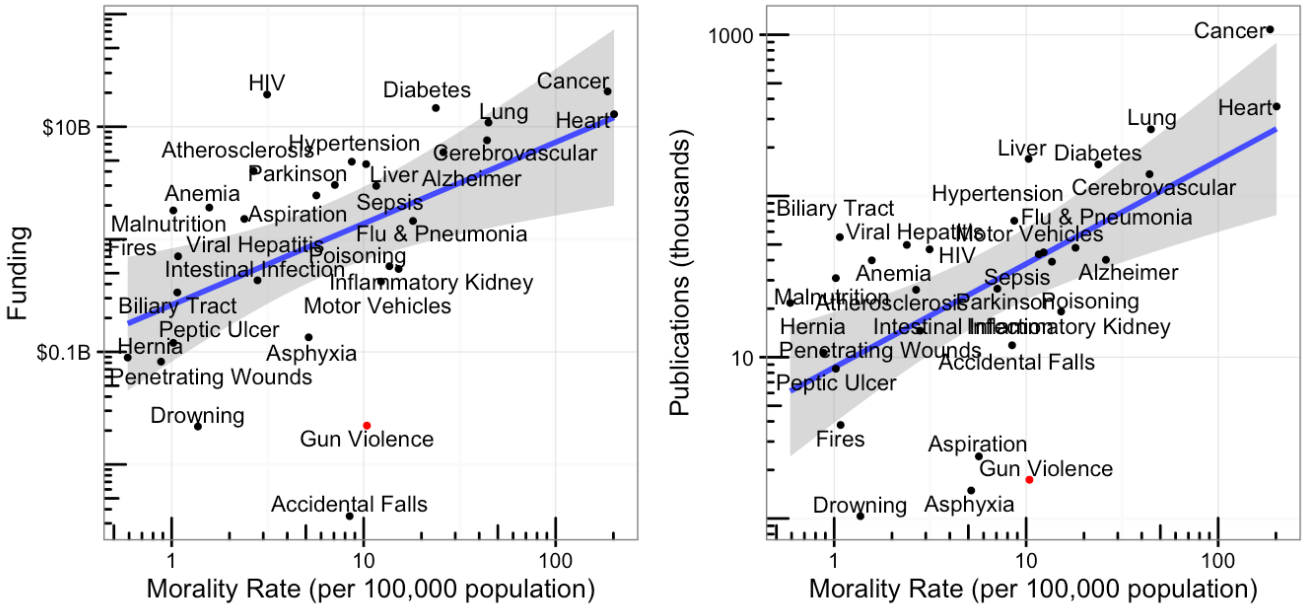

Figure 1: Funding and publication volume for leading causes of death.
Mortality rate versus funding (left) and publication volume (right) is plotted for 30 leading causes of death. Plotting is on a log-log scale with line of best fit included.

Correlation between funding versus publication volume

```
# plot log(publications) x log(funding)
pubs_fund <- ggplot(data, aes(x = (Publications/1000), y = (Total.Funding/1000000000))) +
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_repe
l(aes(label = Abbreviation), size = 3, segment.size = 0, box.padding = unit(0.1, "lines"))
  + scale_y_log10(labels = dollar_format(suffix = 'B')) + scale_x_log10() + annotation_log
ticks() + theme_bw() + labs(y = "Funding") + labs(x = "Publications (thousands)") +
theme(aspect.ratio=1)

# Combine funding and publication panels in one plot
# Code to set equal panel widths
gPubs_fund <- ggplotGrob(pubs_fund)

gPubs_fund$heights <- gFunding$heights
grid.arrange(gPubs_fund)
```

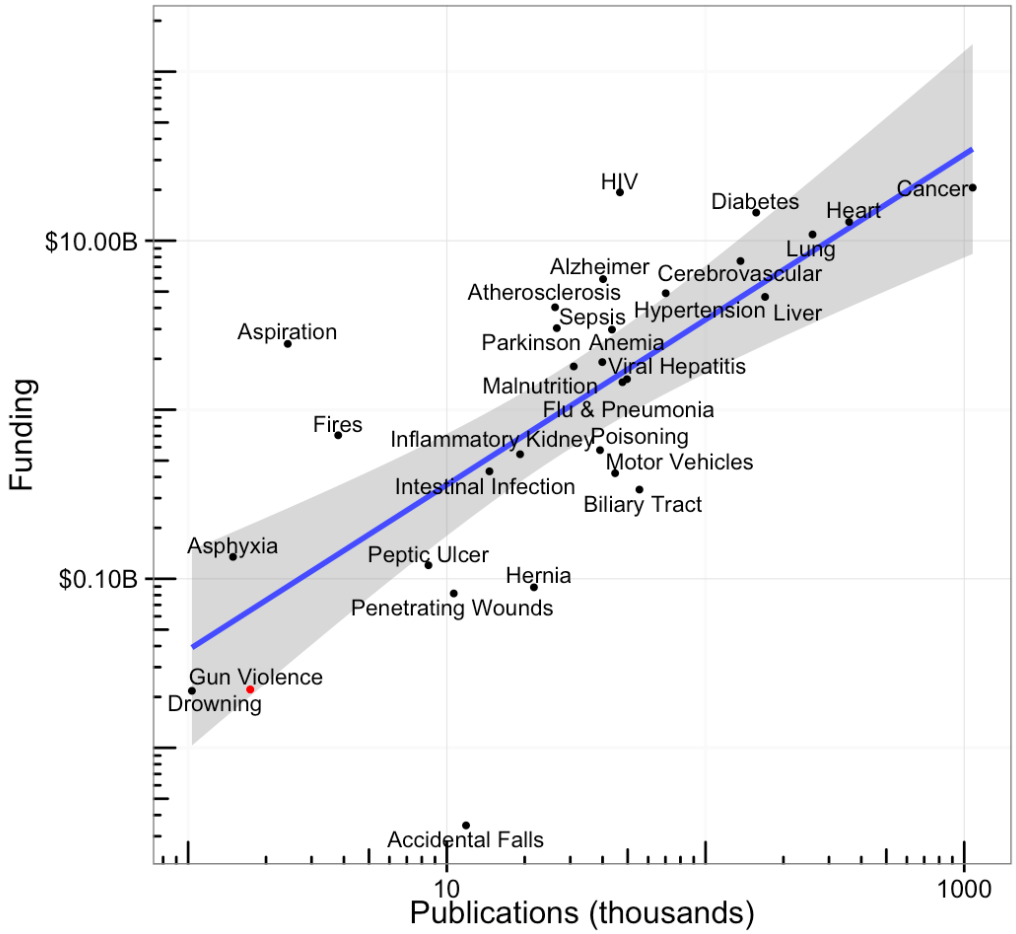

Figure 2: Funding versus publication volume for leading causes of death.
Funding versus publication volume is plotted for 30 leading causes of death. Plotting is on a log-log scale with line of best fit included.

```
# Regress publications on funding
lm.fit <- lm(log(Total.Funding)~ log(Publications), data = data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = log(Total.Funding) ~ log(Publications), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8066 -0.7926 -0.0231  0.8348  3.3037
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.6999     1.8305   5.845 2.78e-06 ***
## log(Publications)  0.9772     0.1761   5.550 6.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.549 on 28 degrees of freedom
## Multiple R-squared:  0.5239, Adjusted R-squared:  0.5069
## F-statistic: 30.81 on 1 and 28 DF,  p-value: 6.179e-06
```

```
paste("Pearson's r: ", cor(log(data$Publications), log(data$Total.Funding)))
```

```
## [1] "Pearson's r:  0.723778611115899"
```

Analysis: Calculating residuals

To determine how research funding and publication volume correlated with mortality, two linear regression analyses were performed using mortality 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 is an outlier in terms of research funding or publication volume.

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

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

```
##
## Call:
## lm(formula = log(Publications) ~ log(Crude.Rate), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1082 -0.4593  0.1643  0.9583  1.8159
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9.0652     0.3865  23.453  < 2e-16 ***
## log(Crude.Rate)   0.6420     0.1604   4.003 0.000417 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.326 on 28 degrees of freedom
## Multiple R-squared:  0.3639, Adjusted R-squared:  0.3412
## F-statistic: 16.02 on 1 and 28 DF,  p-value: 0.0004172
```

```
# 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
## -5.8611 -0.9973  0.6127  1.1078  3.4776
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19.3797     0.5656  34.265  < 2e-16 ***
## log(Crude.Rate)   0.7227     0.2347   3.079  0.00461 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.94 on 28 degrees of freedom
## Multiple R-squared:  0.253,  Adjusted R-squared:  0.2263
## F-statistic: 9.481 on 1 and 28 DF,  p-value: 0.004614
```

Table 1: Publication Residuals.

Cause of Death	Publications	Predicted Publications	Residual
Heart	358,915	260,875	0.27
Cancer	1,078,144	248,160	1.25
Lung	259,196	99,170	0.75
Cerebrovascular	136,502	98,070	0.26
Alzheimer	40,179	70,085	-0.43
Diabetes	156,921	66,114	0.66
Flu & Pneumonia	47,732	55,458	-0.11
Inflammatory Kidney	19,196	49,650	-0.73
Poisoning	39,101	46,252	-0.13
Motor Vehicles	44,710	43,390	0.02
Sepsis	43,477	41,792	0.03
Gun Violence	1,738	38,897	-2.63
Liver	169,832	38,657	1.14
Hypertension	70,178	34,609	0.54
Accidental Falls	11,864	34,043	-0.8
Parkinson	26,605	30,388	-0.1
Aspiration	2,426	26,350	-1.92
Asphyxia	1,491	24,833	-2.32

HIV	46,672	18,030	0.73
Intestinal Infection	14,608	16,751	-0.1
Atherosclerosis	26,198	16,247	0.36
Viral Hepatitis	49,703	15,132	0.92
Anemia	39,895	11,554	0.96
Drowning	1,034	10,586	-1.9
Fires	3,801	9,087	-0.68
Biliary Tract	55,518	9,033	1.46
Malnutrition	30,941	8,759	0.99
Peptic Ulcer	8,488	8,759	-0.02
Penetrating Wounds	10,637	7,967	0.22
Hernia	21,715	6,163	1.01

Table 2: Funding Residuals.

Cause of Death	Funding	Predicted Funding	Residual
Heart	$ 12,910,927,202	$ 12,075,841,121	0.04
Cancer	$ 20,596,612,634	$ 11,415,379,454	0.33
Lung	$ 10,881,388,337	$ 4,065,188,916	0.52
Cerebrovascular	$ 7,577,203,625	$ 4,014,434,824	0.34
Alzheimer	$ 5,925,875,891	$ 2,750,314,629	0.4
Diabetes	$ 14,659,678,132	$ 2,575,510,477	0.92
Flu & Pneumonia	$ 1,455,413,419	$ 2,113,200,268	-0.19
Inflammatory Kidney	$ 545,228,484	$ 1,865,780,117	-0.64
Poisoning	$ 576,633,974	$ 1,722,680,056	-0.57
Motor Vehicles	$ 421,039,553	$ 1,603,146,022	-0.7
Sepsis	$ 2,978,734,825	$ 1,536,837,955	0.34
Gun Violence	$ 22,131,926	$ 1,417,564,256	-2.36
Liver	$ 4,651,595,537	$ 1,407,700,121	0.62
Hypertension	$ 4,889,944,561	$ 1,242,904,513	0.71
Accidental Falls	$ 3,474,852	$ 1,220,029,999	-3.71
Parkinson	$ 3,038,436,363	$ 1,073,615,675	0.54
Aspiration	$ 2,452,057,137	$ 914,413,397	0.51

Asphyxia	$ 134,714,254	$ 855,395,597	-0.97
HIV	$ 19,318,328,159	$ 596,560,678	1.92
Intestinal Infection	$ 432,035,547	$ 549,140,907	-0.12
Atherosclerosis	$ 4,037,397,953	$ 530,593,605	1.07
Viral Hepatitis	$ 1,516,673,869	$ 489,767,131	0.59
Anemia	$ 1,913,274,324	$ 361,488,866	0.88
Drowning	$ 21,728,913	$ 327,585,866	-1.48
Fires	$ 706,367,767	$ 275,848,561	0.5
Biliary Tract	$ 337,021,767	$ 274,000,251	0.11
Malnutrition	$ 1,802,887,247	$ 264,685,579	1.03
Peptic Ulcer	$ 120,214,685	$ 264,685,579	-0.42
Penetrating Wounds	$ 81,875,047	$ 237,898,350	-0.57
Hernia	$ 89,024,769	$ 178,199,091	-0.38

```
# plot funding residuals x publication residuals
residual_plot <- ggplot(data, aes(x = Publications.Residuals, y = Funding.Residuals)) + ge
om_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 Viol
ence')+1) + scale_color_brewer(type = 'qual', palette = 'Set1') + geom_text_repel(aes(labe
l = Abbreviation), size = 3, segment.size = 0, box.padding = unit(0.1, "lines")) + theme_b
w(base_size = 10) + labs(y = "Funding (Studentized Residuals)") + labs(x = "Publications
 (Studentized Residuals)") + theme(aspect.ratio=1)

residual_plot
```

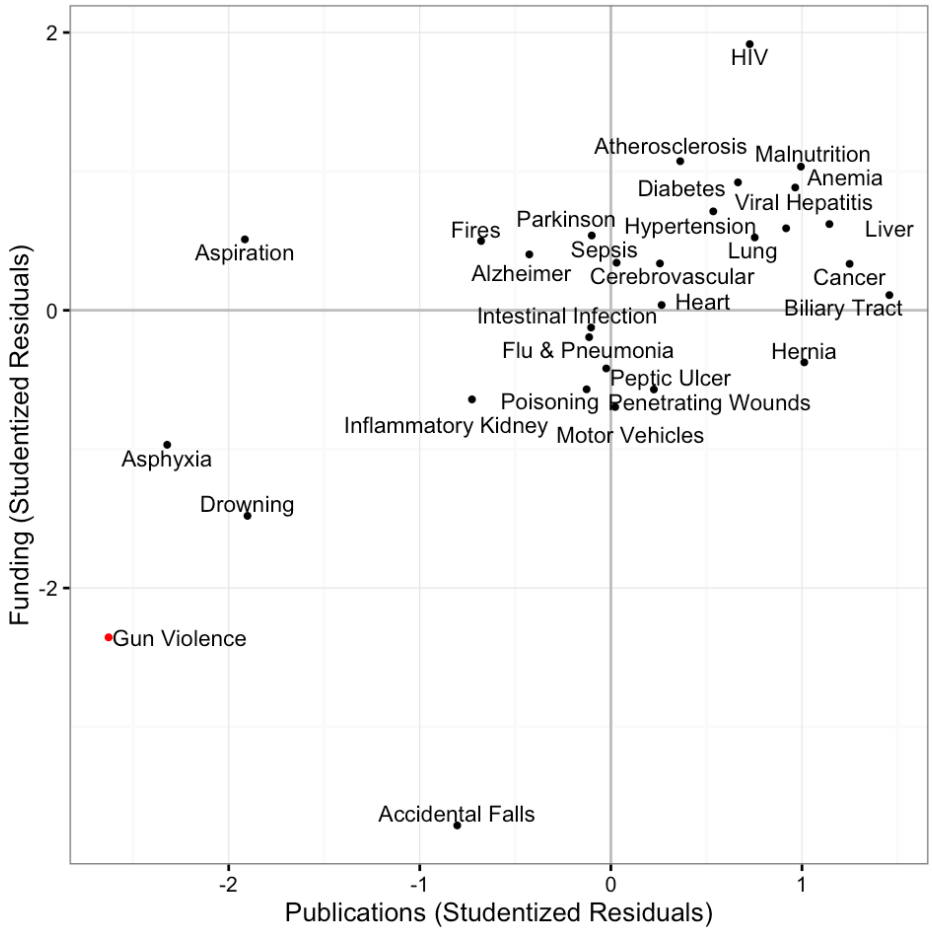

Figure 3: Residual predicted versus observed funding and publication volumes for leading causes of death. Mortality rate was used to predict funding and research volume for 30 leading causes of death. Studentized residuals (residual divided by estimated standard error) were calculated to give a standardized estimate of predicted versus observed funding and publication volume. Publication residuals and funding residuals were plotted along the x- and y- axes, respectively.