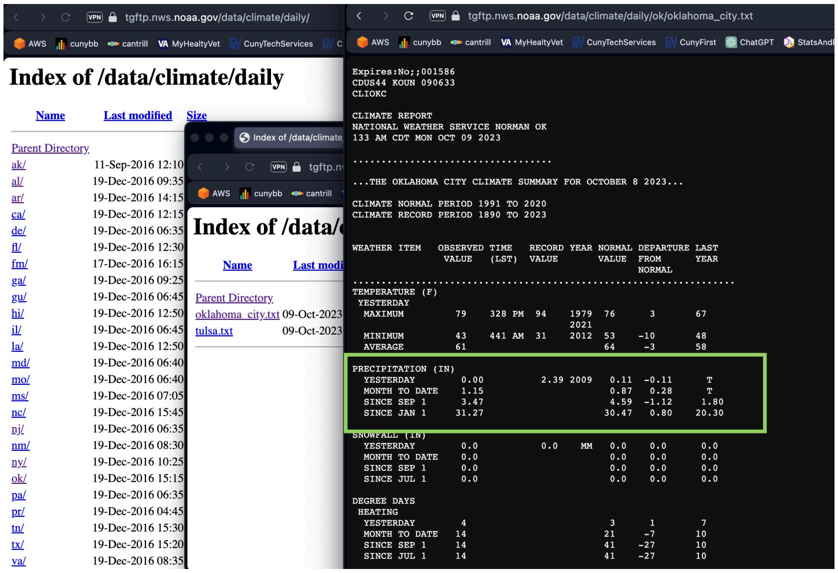
Tidy Datasets

Tony Fraser

Data Set 1: Rain Rain NOAAway

NOAA is one of the biggest publishers of global weather data. Most of the stuff the publish you have to FTP down, but I found this subset up on a web server. It's perfect for this project!



Index of /data/climate/daily

Name	Last modified	Size
Parent Directory		
ak/	11-Sep-2016 12:10	
al/	19-Dec-2016 09:35	
ar/	19-Dec-2016 14:15	
ca/	19-Dec-2016 12:15	
de/	19-Dec-2016 06:35	
fl/	19-Dec-2016 12:30	
fm/	17-Dec-2016 16:15	
ga/	19-Dec-2016 09:25	
gu/	19-Dec-2016 06:45	
hi/	19-Dec-2016 12:50	
il/	19-Dec-2016 06:45	
la/	19-Dec-2016 12:50	
md/	19-Dec-2016 06:40	
mo/	19-Dec-2016 06:40	
ms/	19-Dec-2016 07:05	
nc/	19-Dec-2016 15:45	
nj/	19-Dec-2016 06:35	
nm/	19-Dec-2016 08:30	
ny/	19-Dec-2016 10:25	
ok/	19-Dec-2016 15:15	
pa/	19-Dec-2016 06:35	
pr/	19-Dec-2016 04:45	
tn/	19-Dec-2016 15:30	
tx/	19-Dec-2016 15:20	
va/	19-Dec-2016 08:35	

Index of /data/

Name	Last modi
Parent Directory	
oklahoma_city.txt	09-Oct-2023
tulsa.txt	09-Oct-2023

```
Expires:No;;001586
CDUS44 KOUN 090633
CLIOKC

CLIMATE REPORT
NATIONAL WEATHER SERVICE NORMAN OK
133 AM CDT MON OCT 09 2023

...................................

...THE OKLAHOMA CITY CLIMATE SUMMARY FOR OCTOBER 8 2023...

CLIMATE NORMAL PERIOD 1991 TO 2020
CLIMATE RECORD PERIOD 1890 TO 2023


WEATHER ITEM   OBSERVED TIME   RECORD YEAR NORMAL DEPARTURE LAST
                VALUE   (LST)  VALUE       VALUE  FROM      YEAR
                                                  NORMAL
...................................................................
TEMPERATURE (F)
 YESTERDAY
  MAXIMUM         79    328 PM  94    1979  76      3       67
                                      2021
  MINIMUM         43    441 AM  31    2012  53    -10       48
  AVERAGE         61                        64     -3       58

PRECIPITATION (IN)
  YESTERDAY        0.00          2.39 2009   0.11  -0.11      T
  MONTH TO DATE    1.15                      0.87   0.28      T
  SINCE SEP 1      3.47                      4.59  -1.12     1.80
  SINCE JAN 1     31.27                     30.47   0.80    20.30

SNOWFALL (IN)
  YESTERDAY        0.0           0.0    MM   0.0    0.0      0.0
  MONTH TO DATE    0.0                       0.0    0.0      0.0
  SINCE SEP 1      0.0                       0.0    0.0      0.0
  SINCE JUL 1      0.0                       0.0    0.0      0.0

DEGREE DAYS
 HEATING
  YESTERDAY        4                         3      1        7
  MONTH TO DATE   14                        21     -7       10
  SINCE SEP 1     14                        41    -27       10
  SINCE JUL 1     14                        41    -27       10
```

NOAA Data Process Flow:

1. Go the main url, scrape all the US state links.
2. For each state, scrape the cities.
3. For each city, download the city file.
4. For each file, get just the precipitation data.
5. Finish by seeing who had the most rain yesterday and last month.

NOAA CODE

```r
packages <- c("rvest", "httr", "purrr", "tidyverse", "gt")
lapply(packages, library, character.only = TRUE)

## ###########################################
## All the functions for scraping and parsing
## ###########################################

get_content <- function(url, type = "html") {
  response <- GET(url)
  if (http_status(response)$category != "Success") {
    stop(paste("Failed to retrieve content from", url))
  }

  if (type == "html") {
    return(content(response, as = "text"))
  } else if (type == "text") {
    return(content(response, as = "text", encoding = "UTF-8"))
  } else {
    stop("Invalid type specified. Use 'html' or 'text'.")
  }
}

extract_links <- function(url, html_content, regex_filter = NULL) {
  html_parsed <- read_html(html_content)
  links <- html_parsed %>%
    html_nodes("td > a") %>%
    html_attr("href") %>%
    .[!. %in% c("/data/climate/")]  # Exclude the specific link pattern

  full_links <- paste0(url, links)
  if (!is.null(regex_filter)) {
    full_links <- full_links[!grepl(regex_filter, full_links)]
  }
  return(full_links)
}

extract_value <- function(pattern, lines) {
    #gets numbers out of the text file lines.
    line <- lines[grep(pattern, lines)]
    if (length(line) == 0) {
      return(NA)
    }
    # Extract the numeric value using regex
    value <- regmatches(line, regexpr("\\d+\\.?\\d*", line))
    if (length(value) == 0) {
      return(NA)
    } else {
      return(as.numeric(value[[1]]))
    }
}

extract_precipitation <- function(text) {
  # Finds the lines related to precipitation
  lines <- unlist(strsplit(text, "\n"))
  if (length(grep("PRECIPITATION", lines)) == 0) {
    return(data.frame(
      yesterday = NA,
      month_to_date = NA
    ))
  }

  precip_lines <- lines[grep("PRECIPITATION", lines):length(lines)]
  # Extract the values from the lines
  yesterday     <- extract_value("YESTERDAY", precip_lines)
  month_to_date <- extract_value("MONTH TO DATE", precip_lines)

  data <- data.frame(
    yesterday = yesterday,
    month_to_date = month_to_date
  )
  return(data)
}

## ###########################################
## NOAA Parsing Application
## ###########################################

url <- "https://tgftp.nws.noaa.gov/data/climate/daily/"
state_links <- extract_links(url, get_content(url))
state_stations <- state_links %>%
  map(get_content) %>%
  map2(state_links, ~extract_links(url = .y, .x, regex_filter="data/climate/daily/$")) %>%
  unlist()

all_data <- state_stations %>%
  map_df(function(link) {
    text <- get_content(link, type = "text")
    extract_precipitation(text) %>%
      mutate(station_file = link)
  }) %>%
  mutate(
    state_or_region = str_extract(station_file, "(?<=daily/)[^/]+"),
    station = str_extract(station_file, "(?<=/)[^/]+(?=.txt)")
  )%>%
  select(-station_file) %>%
  mutate(yesterday = ifelse(is.na(yesterday), 0, yesterday)) %>%
  mutate(month_to_date = ifelse(is.na(month_to_date), 0, month_to_date))

findings_month <- all_data %>%
  arrange(desc(month_to_date)) %>%
  slice_head(n = 10) %>%
  gt() %>%
  tab_header(
    title = "Highest Rainfall Last 30 Days"
  )

findings_yesterday <- all_data %>%
  arrange(desc(yesterday)) %>%
  slice_head(n = 10) %>%
  gt() %>%
  tab_header(
    title = "Highest Rainfall Yesterday"
  )
```

NOAA Findings: Station File Precipitation Results

Apparently Guam had the most rainfall over the month, and New Orleans had it for yesterday.!

```
findings_month
```

Highest Rainfall Last 30 Days

yesterday	month_to_date	state_or_region	station
0.25	9.40	gu	tiyan
0.34	3.95	fm	ck_t11
0.00	3.90	fl	miami
0.00	3.44	tx	houston
0.00	3.35	fl	west_palm_beach
0.00	3.24	tx	corpus_christi
0.00	2.99	fl	tallahassee
0.00	2.89	ak	valdez
0.00	2.87	ga	columbus
0.00	2.85	ga	macon

```
findings_yesterday
```

Highest Rainfall Yesterday

yesterday	month_to_date	state_or_region	station
2.53	2.46	la	new_orleans
2.05	2.17	pr	san_juan
0.36	2.73	ak	st_paul_island
0.34	3.95	fm	ck_t11
0.25	9.40	gu	tiyan
0.17	1.00	ak	king_salmon
0.02	0.02	ak	barrow
0.00	0.00	ak	bethel
0.00	0.00	ak	kodiak
0.00	0.00	ak	kotzebue

Dataset 2: NSF, Sex, Programs and PHDs

nsf.gov is a great resource for for learning about PHD level degrees. Since I am considering one, and since most of my data scientist coworkers are female, I though I'd take a look at PHDs by program and sex.

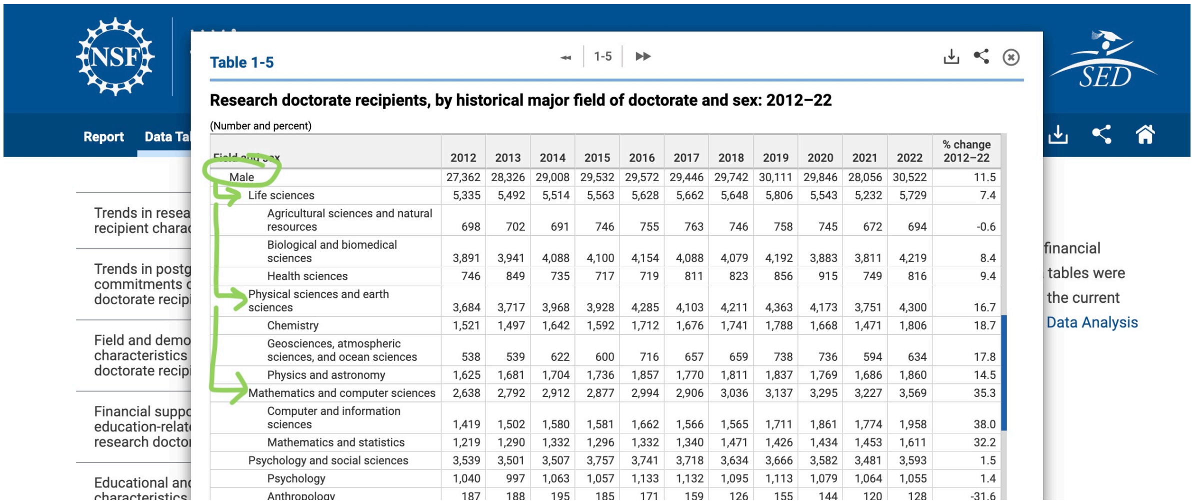


Table 1-5

Research doctorate recipients, by historical major field of doctorate and sex: 2012–22

(Number and percent)

Field and sex	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	% change 2012–22
Male	27,362	28,326	29,008	29,532	29,572	29,446	29,742	30,111	29,846	28,056	30,522	11.5
Life sciences	5,335	5,492	5,514	5,563	5,628	5,662	5,648	5,806	5,543	5,232	5,729	7.4
Agricultural sciences and natural resources	698	702	691	746	755	763	746	758	745	672	694	-0.6
Biological and biomedical sciences	3,891	3,941	4,088	4,100	4,154	4,088	4,079	4,192	3,883	3,811	4,219	8.4
Health sciences	746	849	735	717	719	811	823	856	915	749	816	9.4
Physical sciences and earth sciences	3,684	3,717	3,968	3,928	4,285	4,103	4,211	4,363	4,173	3,751	4,300	16.7
Chemistry	1,521	1,497	1,642	1,592	1,712	1,676	1,741	1,788	1,668	1,471	1,806	18.7
Geosciences, atmospheric sciences, and ocean sciences	538	539	622	600	716	657	659	738	736	594	634	17.8
Physics and astronomy	1,625	1,681	1,704	1,736	1,857	1,770	1,811	1,837	1,769	1,686	1,860	14.5
Mathematics and computer sciences	2,638	2,792	2,912	2,877	2,994	2,906	3,036	3,137	3,295	3,227	3,569	35.3
Computer and information sciences	1,419	1,502	1,580	1,581	1,662	1,566	1,565	1,711	1,861	1,774	1,958	38.0
Mathematics and statistics	1,219	1,290	1,332	1,296	1,332	1,340	1,471	1,426	1,434	1,453	1,611	32.2
Psychology and social sciences	3,539	3,501	3,507	3,757	3,741	3,718	3,634	3,666	3,582	3,481	3,593	1.5
Psychology	1,040	997	1,063	1,057	1,133	1,132	1,095	1,113	1,079	1,064	1,055	1.4
Anthropology	187	188	195	185	171	159	126	155	144	120	128	-31.6

NSF Data Process Flow:

1. Download the Excel
2. Flatten the data by extracting all hierarchy from within
3. Tidy by adding columns and pivoting wider with categories
4. Tidy by combining years into the same column
5. Visualize overall trends, and specific programs

NSF CODE

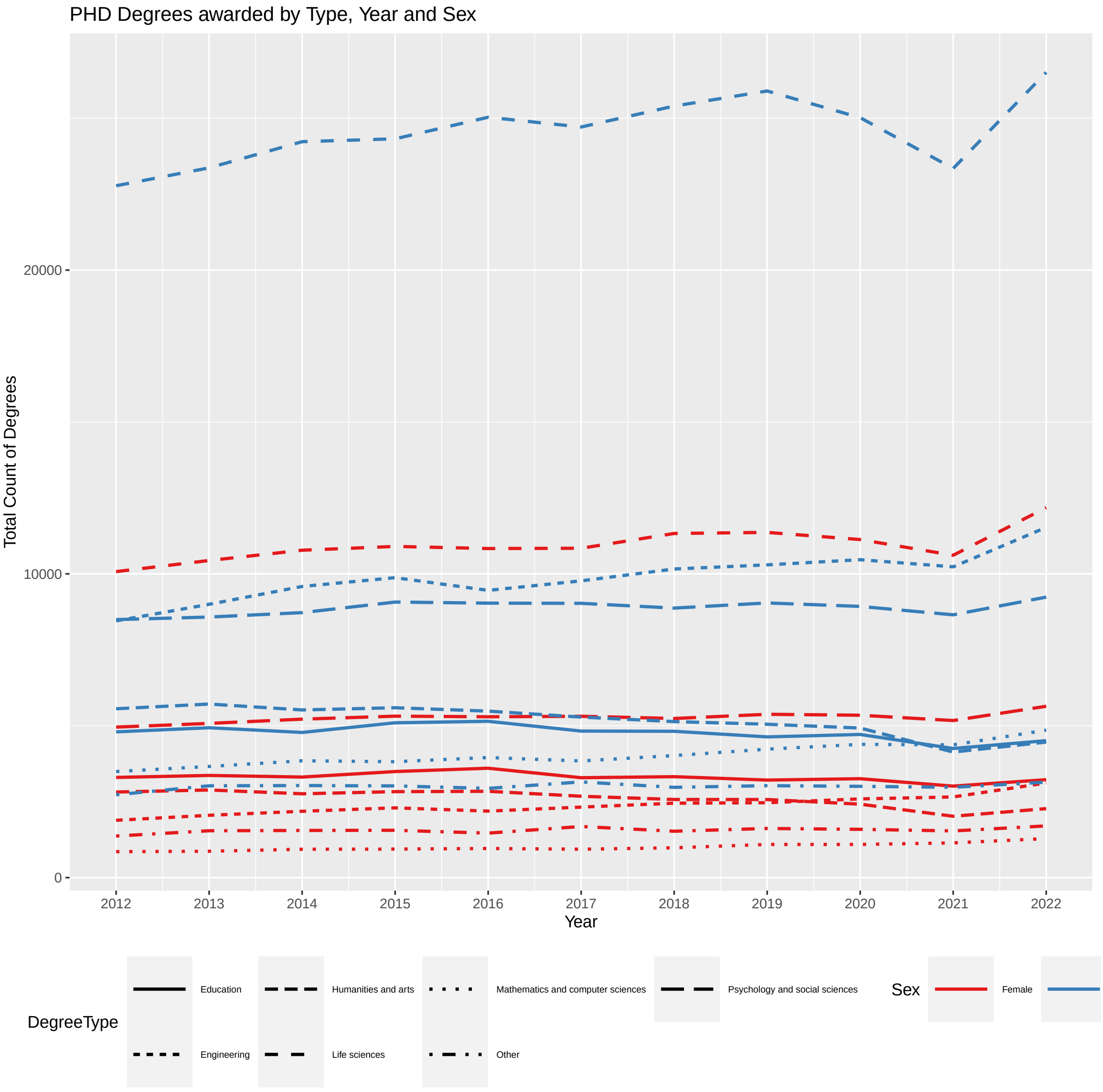
```
packages <- c("httr", "readxl", "dplyr", "tidyr", "gt", "ggplot2")
lapply(packages, library, character.only = TRUE)

r1 <- "https://ncses.nsf.gov/pubs/nsf24300/assets/"
r2 <- "data-tables/tables/nsf24300-tab001-005.xlsx"
url <- paste(r1, r2, sep="")
print(url)

#Load the file into a dataframe
temp_file <- tempfile(fileext = ".xlsx")
```

```r
download.file(url, temp_file, mode = "wb")
sex <- read_excel(temp_file, sheet = 1) %>%
  `colnames<-`(.[3, ]) %>%
  slice(-1:-3)

# Create the three dataframes
all_pos <- which(sex$`Field and sex` == "All doctorate recipientsa")
male_pos <- which(sex$`Field and sex` == "Male")
female_pos <- which(sex$`Field and sex` == "Female")
all_df <- sex[2:(male_pos-1), ]
male_df <- sex[male_pos+1:(female_pos-1), ]
female_df <- sex[(female_pos +1):nrow(sex), ]
all_df$Sex <- "Combined"
male_df$Sex <- "Male"
female_df$Sex <- "Female"

## Tidy, remove extra columns, fill, make long, etc.
cats <- c(
  "Life sciences",
  "Mathematics and computer sciences",
  "Psychology and social sciences",
  "Engineering",
  "Education",
  "Humanities and arts",
  "Other")

long_df <- bind_rows(all_df, male_df, female_df) %>%
  mutate(DegreeType = if_else(`Field and sex` %in% cats, `Field and sex`, NA_character_)) %>%
  fill(DegreeType) %>%
  filter(!(`Field and sex` %in% c(cats, "Female", "Male", "Combined"))  & !is.na(`Field and sex`)) %>%
  gather(key = "Year", value = "Count", `2012`:`2022`) %>%
  mutate(Year = as.numeric(Year)) %>%
  filter(Sex != "Combined")

##  Now prepare the two charts  #################
line_graph <- long_df %>%
  group_by(Year, DegreeType, Sex) %>%
  summarise(TotalCount = sum(Count, na.rm = TRUE)) %>%
  ungroup() %>%
  ggplot(aes(x = Year, y = TotalCount,
        color = Sex,
        linetype = DegreeType,
        group = interaction(DegreeType, Sex))) +
  geom_line(size = 1) +
  labs(title = "PHD Degrees awarded by Type, Year and Sex",
       x = "Year",
       y = "Total Count of Degrees",
       color = "Sex") +
  scale_x_continuous(breaks = 2012:2022) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "bottom",
        legend.key.size = unit(1.5, "cm"),
        legend.text = element_text(size = 6))

all_programs <- long_df %>%
  filter(Year == 2022) %>%
  group_by(`Field and sex`) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  ggplot(aes(x = reorder(`Field and sex`, Total), y = Count, fill = Sex)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(title = "Degree Counts by Field and Sex for 2022",
       x = "Degree Field",
       y = "Total Count of Degrees",
       fill = "Sex") +
  coord_flip() +
  theme_minimal()
```

PHD Findings 1

Mostly, I expected much closer to a 50/50 split between men and women. That is definitely not the case.

```
line_graph
```

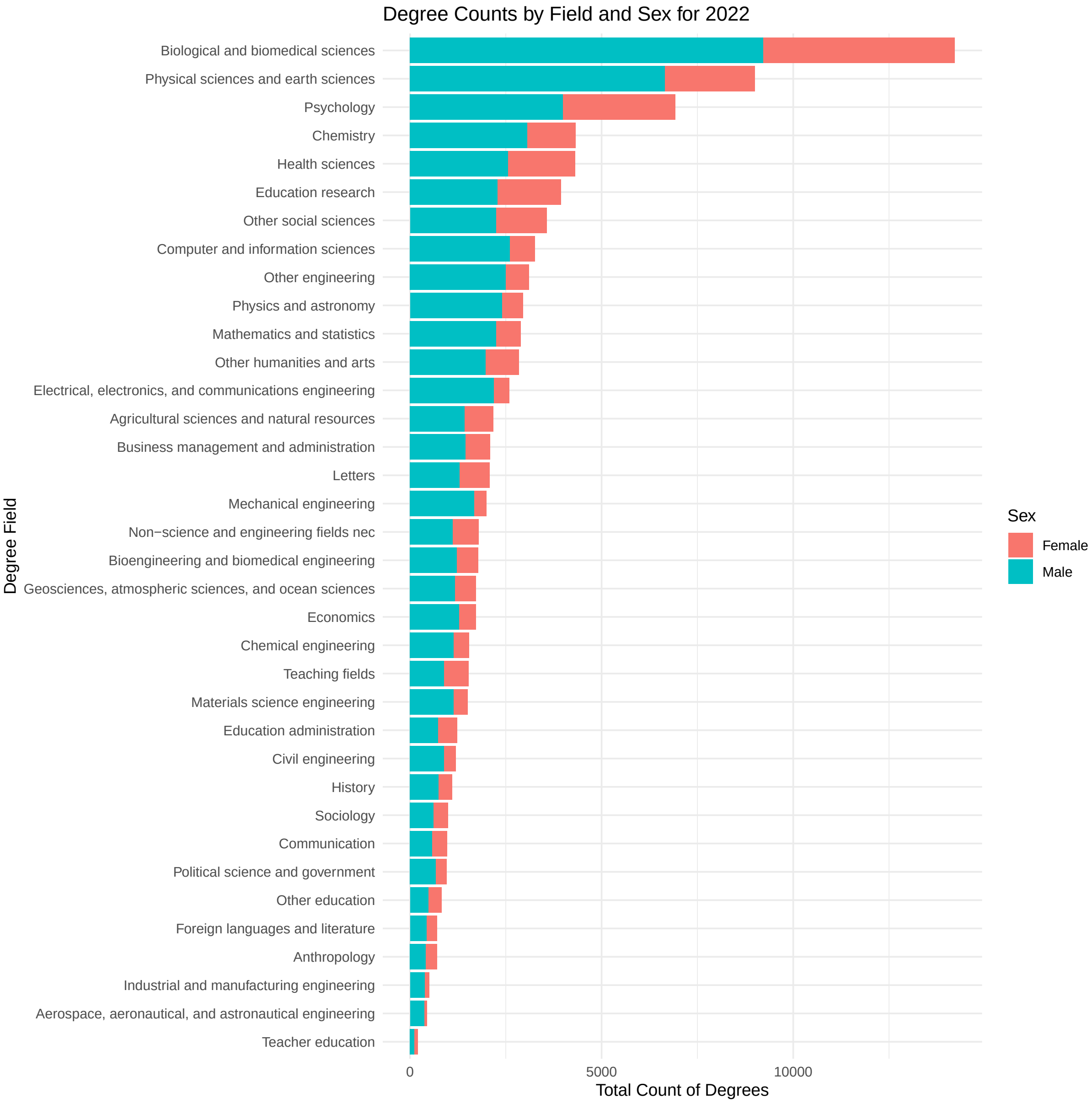


‘

PHD Findings 2

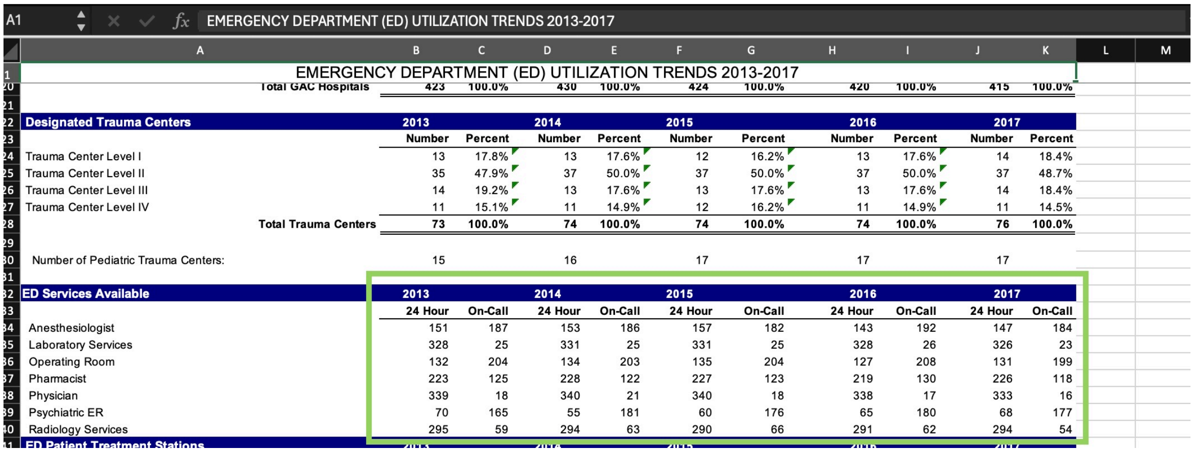
I took one year and drilled into which programs specifically offered PHD's. I figured business would be at the top of the list, but it is not.

```
all_programs
```



Data Set 3: Go to California yes, but not to the CA HHS

California HHS publishes a bunch of aggregated data about their hospitals. Of course they also publish raw data, but in the spirit of tidy data sets, let's work with this aggregated one. We'll do a plot to see if every office continues to grow year over year or not.



A1 EMERGENCY DEPARTMENT (ED) UTILIZATION TRENDS 2013-2017

EMERGENCY DEPARTMENT (ED) UTILIZATION TRENDS 2013-2017

	A	B	C	D	E	F	G	H	I	J	K
20	Total GAC Hospitals	423	100.0%	430	100.0%	424	100.0%	420	100.0%	415	100.0%
21											
22	Designated Trauma Centers	2013		2014		2015		2016		2017	
23		Number	Percent	Number	Percent	Number	Percent	Number	Percent	Number	Percent
24	Trauma Center Level I	13	17.8%	13	17.6%	12	16.2%	13	17.6%	14	18.4%
25	Trauma Center Level II	35	47.9%	37	50.0%	37	50.0%	37	50.0%	37	48.7%
26	Trauma Center Level III	14	19.2%	13	17.6%	13	17.6%	13	17.6%	14	18.4%
27	Trauma Center Level IV	11	15.1%	11	14.9%	12	16.2%	11	14.9%	11	14.5%
28	Total Trauma Centers	73	100.0%	74	100.0%	74	100.0%	74	100.0%	76	100.0%
29											
30	Number of Pediatric Trauma Centers:	15		16		17		17		17	
31											
32	ED Services Available	2013		2014		2015		2016		2017	
33		24 Hour	On-Call	24 Hour	On-Call	24 Hour	On-Call	24 Hour	On-Call	24 Hour	On-Call
34	Anesthesiologist	151	187	153	186	157	182	143	192	147	184
35	Laboratory Services	328	25	331	25	331	25	328	26	326	23
36	Operating Room	132	204	134	203	135	204	127	208	131	199
37	Pharmacist	223	125	228	122	227	123	219	130	226	118
38	Physician	339	18	340	21	340	18	338	17	333	16
39	Psychiatric ER	70	165	55	181	60	176	65	180	68	177
40	Radiology Services	295	59	294	63	290	66	291	62	294	54
41	ED Patient Treatment Stations										

HHS Data Process Flow

1. Download the excel file from the web.
2. Load the summary tab into a dataframe
3. Tidy
4. Show the adjusted table
5. Plot/review year over year growth

CA HHS Code

```r
packages <- c("httr", "readxl", "gt", "tidyverse", "dplyr")
lapply(packages, library, character.only = TRUE)

# URL of the Excel file
u1 <- "https://data.chhs.ca.gov/dataset/31ea2cfd-bc1c-4bde-9626-f41b97cc1b93/"
u2 <- "resource/a05be197-1cea-4685-9dfd-0afc7712f81f/download/ed_ut2013_2017_20181030.xlsx"
url <- paste(u1, u2, sep="")

#download and read in the excel file.
temp_file <- tempfile(fileext = ".xlsx")
download.file(url, temp_file, mode = "wb")
```

```r
df <- read_excel(temp_file, sheet = 1)
colnames(df) <- c("Service",
                  "2013-24Hour","2013-OnCall",
                  "2014-24Hour","2014-0nCall",
                  "2015-24Hour","2015-OnCall",
                  "2016-24Hour","2016-OnCall",
                  "2017-24Hour","2017-OnCall")
unlink(temp_file)

#get just the bit we are looking for.
start_row <- which(df$Service == "ED Services Available")
end_row <- which(df$Service == "ED Patient Treatment Stations")

reshaped_df <- df[start_row:(end_row-1),] %>%
  slice(-(1:2)) %>%
  # now we tidy
  pivot_longer(
    cols = -Service,
    names_to = "temp",
    values_to = "count"
  ) %>%
  separate(temp, into = c("Year", "ServiceType"), sep = "-") %>%
  select(Service, Year, ServiceType, count) %>%
  filter(!is.na(count)) %>%
  mutate(
    Year = as.numeric(Year),
    count = as.numeric(count)
  )

## build the outputs
myplot <- reshaped_df %>%
  ggplot(aes(x = Year, y = count, color = Service, linetype = ServiceType, group = interaction(Service, S
  geom_line(size = 1) +
  labs(
    title = "Service Counts over Years",
    x = "Year",
    y = "Count",
    color = "Service",
    linetype = "Service Type"
  ) +
  scale_linetype_manual(values = c("24Hour" = "solid", "OnCall" = "dashed")) +
  theme_minimal() +
  theme(legend.position = "bottom")

table <- reshaped_df %>%
  slice_head(n = 12) %>%
  gt() %>%
  tab_header(
    title = "After: California HHS Emergency Services Overview"
  )
```

CA HHS Findings

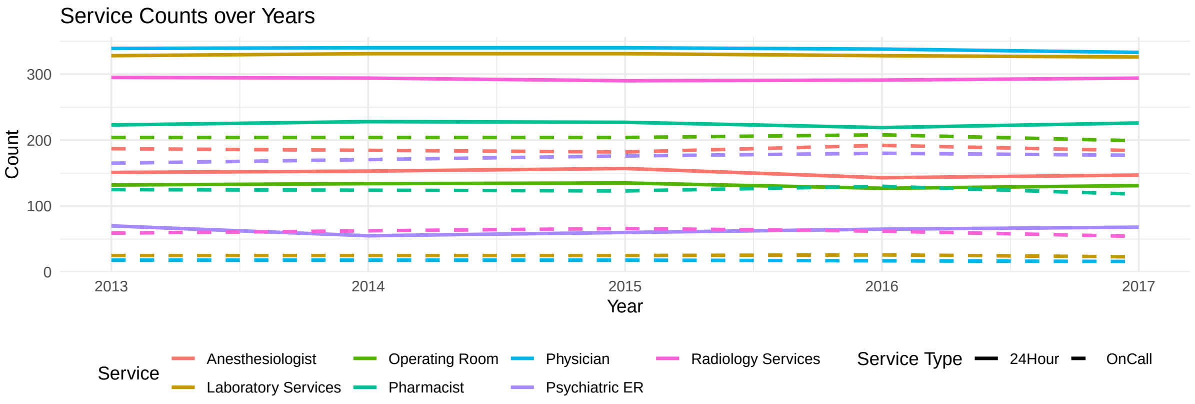
It looks flat. There are not a lot of ups and downs in terms of emergency service.

```
table
```

After: California HHS Emergency Services Overview

Service	Year	ServiceType	count
Anesthesiologist	2013	24Hour	151
Anesthesiologist	2013	OnCall	187
Anesthesiologist	2014	24Hour	153
Anesthesiologist	2014	0nCall	186
Anesthesiologist	2015	24Hour	157
Anesthesiologist	2015	OnCall	182
Anesthesiologist	2016	24Hour	143
Anesthesiologist	2016	OnCall	192
Anesthesiologist	2017	24Hour	147
Anesthesiologist	2017	OnCall	184
Laboratory Services	2013	24Hour	328
Laboratory Services	2013	OnCall	25

```
myplot
```


Service Counts over Years
Count
Year
Service
Anesthesiologist
Laboratory Services
Operating Room
Pharmacist
Physician
Psychiatric ER
Radiology Services
Service Type
24Hour
OnCall