Initial Set-Up

Activating Project Environment

```julia
using DrWatson
@quickactivate "CDC WONDER"
```

Loading Packages

```julia
using CSV
using DataFrames
using Latexify
using PrettyTables
using PyPlot
using Statistics
using TimeSeries
using Weave
```

Here is a little run down on what these packages are used for in our analysis:

- `CSV` - Julia's multi-threaded CSV reader
- `DataFrames` - Julia's dataframe handler for easily manipulating data
- `DrWatson` - Incredibly helpful tool for managing Julia-based scientific experiments and exploration
- `Latexify` - Enables the conversion of Julia objects to other formats such as LaTeX or Markdown
- `PrettyTables` - Printing pretty tables for nice references
- `PyPlot` - Julia interface to the `matplotlib` plotting library
- `Statistics` - Julia's standard library for useful statistical methods
- `TimeSeries` - Package for working with time series data and functions such as moving averages
- `Weave` - Converts Julia Markdown files to other forms of output

Familiarizing Ourselves with CDC Wonder

Introduction

We are using data from the CDC Wonder Natality reporting. This data is derived from birth certificates and we are using data from 2007 - 2019. Some additional detail on this data:

> Beginning in 2007, data are reported from the the 2003 U.S. standard Certificate of Live Birth. With the implementation of the 2003 U.S. standard Certificate of Live Birth by the states, some data items are not comparable with the previous 1989 revision, resulting in changes to the data items available here. Beginning with year 2007, data for five new birth anomalies are available, and data for

five maternal risk factors are no longer available. Beginning with year 2016, data for many additional items (mostly medical and health items) are available; "bridged-race" categories are not available in the 2016-2019 (expanded) database.

Building Our Dataset

In particular, we formed our dataset from a 2007 - 2019 data request to CDC Wonder where we selected the following measures:

- Birth Rate
- Average Birth Weight

and grouped these results by "State" and "County". Here, we only selected the state of New York for analysis as well as all the years available and left default ranges for everything else

Generating Summary Trends

Pre-Processing Our Data

When the data is given to us raw from CDC Wonder, it is not ready for reading into dataframes yet as it contains additional rows and columns that are not in CSV format. We apply the following transformations here:

```
df = DataFrame(CSV.File(
    datadir("exp_raw", "ny_2007_2019.txt");
    limit = 29,
    drop = ["Notes"],
))
CSV.write(datadir("exp_pro", "ny_2007_2019_processed.csv"), df)
```

Per `DrWatson`, we use the `datadir` function to save our processed data to the `exp_pro` directory!

Loading Our Cleaned Data

Now we are able to load in our cleaned data as a DataFrame as follows:

```
df =
    CSV.File(datadir("exp_pro", "ny_2007_2019_processed.csv")) |>
    DataFrame
```

And then we can preview our table using the great `Latexify` package:

```
latexify(first(df, 3); env = :mdtable, latex = false)
```

State	State Code	County	County Code	Births	Total Population	Birth Rate	Average Birth Weight
New York	36	Albany County, NY	36001	39757	3981132	9.99	3298.8
New York	36	Bronx County, NY	36005	280749	18368943	15.28	3197.47
New York	36	Broome County, NY	36007	26430	2563102	10.31	3306.45

Finding Most Populuous Counties

As a first dive into our analysis, lets get a sense for some summary values! First, lets find the most populuous counties in New York!

```
sort!(df, Symbol("Total Population"), rev = true)
names = ["County", "Total Population"]
tab = Any[
    df[1, :County] df[1, Symbol("Total Population")]
    df[2, :County] df[2, Symbol("Total Population")]
    df[3, :County] df[3, Symbol("Total Population")]
    df[4, :County] df[4, Symbol("Total Population")]
    df[5, :County] df[5, Symbol("Total Population")]
]
pretty_table(tab; header = names, backend = :html)
```

<!DOCTYPE html>

County

Total Population

Kings County, NY

33263747

Queens County, NY

29520786

Unidentified Counties, NY

25588938

New York County, NY

21031579

Suffolk County, NY

19381641

As you can see, we get a group of counties lumped together in the "County" column called "Unidentified Counties, NY". Let's remove that from our dataframe going forward:

```
delete!(df, 3)
```

And let's check the county table again:

```
tab = Any[
    df[1, :County] df[1, Symbol("Total Population")]
    df[2, :County] df[2, Symbol("Total Population")]
    df[3, :County] df[3, Symbol("Total Population")]
    df[4, :County] df[4, Symbol("Total Population")]
    df[5, :County] df[5, Symbol("Total Population")]
]
pretty_table(tab; header = names, backend = :html)
```

<!DOCTYPE html>

County

Total Population

Kings County, NY

33263747

Queens County, NY

29520786

New York County, NY

21031579

Suffolk County, NY

19381641

Bronx County, NY

18368943

Great! Let's move into a bit more of a deeper analysis.

Finding Counties with the Most Births

As we are working with natality data, I think it would be wise to see where births are occurring! Let's find the top five counties with most births in New York similar to how we found largest populations:

```
sort!(df, Symbol("Births"), rev = true)
names = ["County", "Births"]
tab = Any[
    df[1, :County] df[1, Symbol("Births")]
    df[2, :County] df[2, Symbol("Births")]
    df[3, :County] df[3, Symbol("Births")]
    df[4, :County] df[4, Symbol("Births")]
    df[5, :County] df[5, Symbol("Births")]
]
pretty_table(
    tab;
    title = "Counties with Most Births",
    title_alignment = :c,
    header = names,
    backend = :html,
)
```

<!DOCTYPE html>

Counties with Most Births

County

Births

Kings County, NY

532784

Queens County, NY

390625

Bronx County, NY

280749

New York County, NY

244766

Suffolk County, NY

210532

Very similar results to our previous table, but this time, Bronx County jumped up to the third position!

Final Steps

Export Julia Markdown Document as Jupyter Notebook

```julia
convert_doc(
    "exploration.jmd",
    projectdir("notebooks", "exploration.ipynb"),
)
```

Export Julia Markdown Document as an HTML Document

```julia
weave("exploration.jmd", "exploration.html")
```