Heading 1

Every now and again we wonder down a rabbit hole and end up asking ourselves, "how did I get here?", that is the exact scenario I found myself which led to this project.

I somehow got into a conversation with a coworker about beekeeping and in a matter of minutes we were on the Illinois Department of Agriculture Apiary Annual report page. Here

Here, you will find an Annual Apiary Report which is published in a PDF format. As you look through the report, you will see many different tables representing the number of registered beekeepers, number of colonies, and number of apiaries in Illinois. A table representation of the data does not lend itself to being able to quickly identify changes over time.

As an aspiring Data Scientist, I saw a new challenge, could I extract the data in the report, in a programmatic way?

My goal of this project was to extract the data from the PDF, manipulate the data, engineer new features, create visualizations, and extract new insights.

First, could we extract the data, long story short the answer is yes, yes we can. There happens to be a nifty Python library called Tabula. With Tabula you can identify the boundaries of the table you are attempting to extract within the PDF, and it will do its best at providing you with an accurate representation of that table, though not perfect, as you can see below.

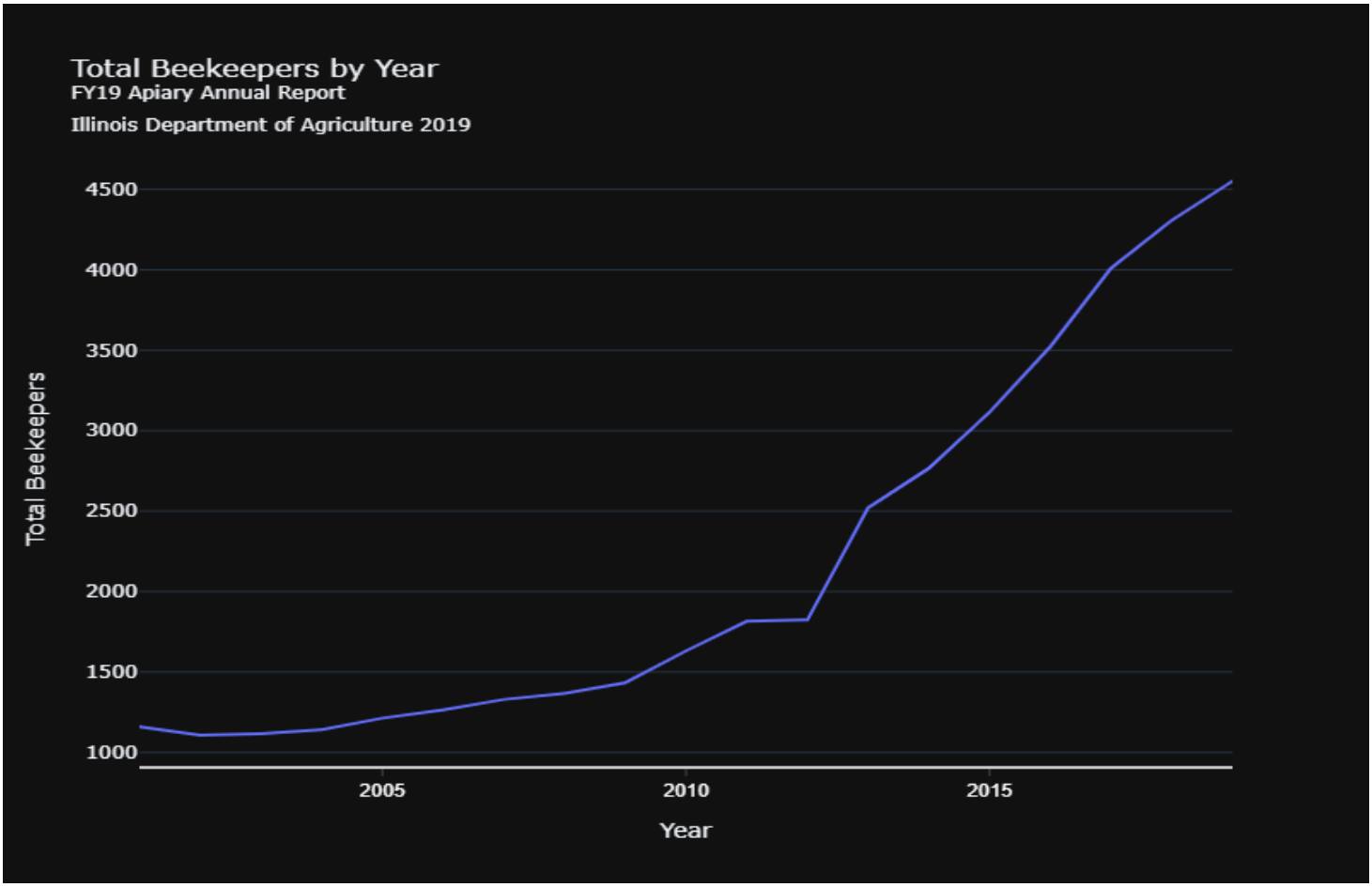
	0	1	2	3	4	5
0	NaN	Number of	Number	Number	NaN	NaN
1	NaN	Registered	of	of	Colonies/	Colonies/
2	Year	Beekeepers	Apiaries	Colonies	Beekeeper	Apiary
3	2019	4,551	6,202	32,268	7.1	5.2
4	2018	4,308	6,000	30,017	7.0	5.0

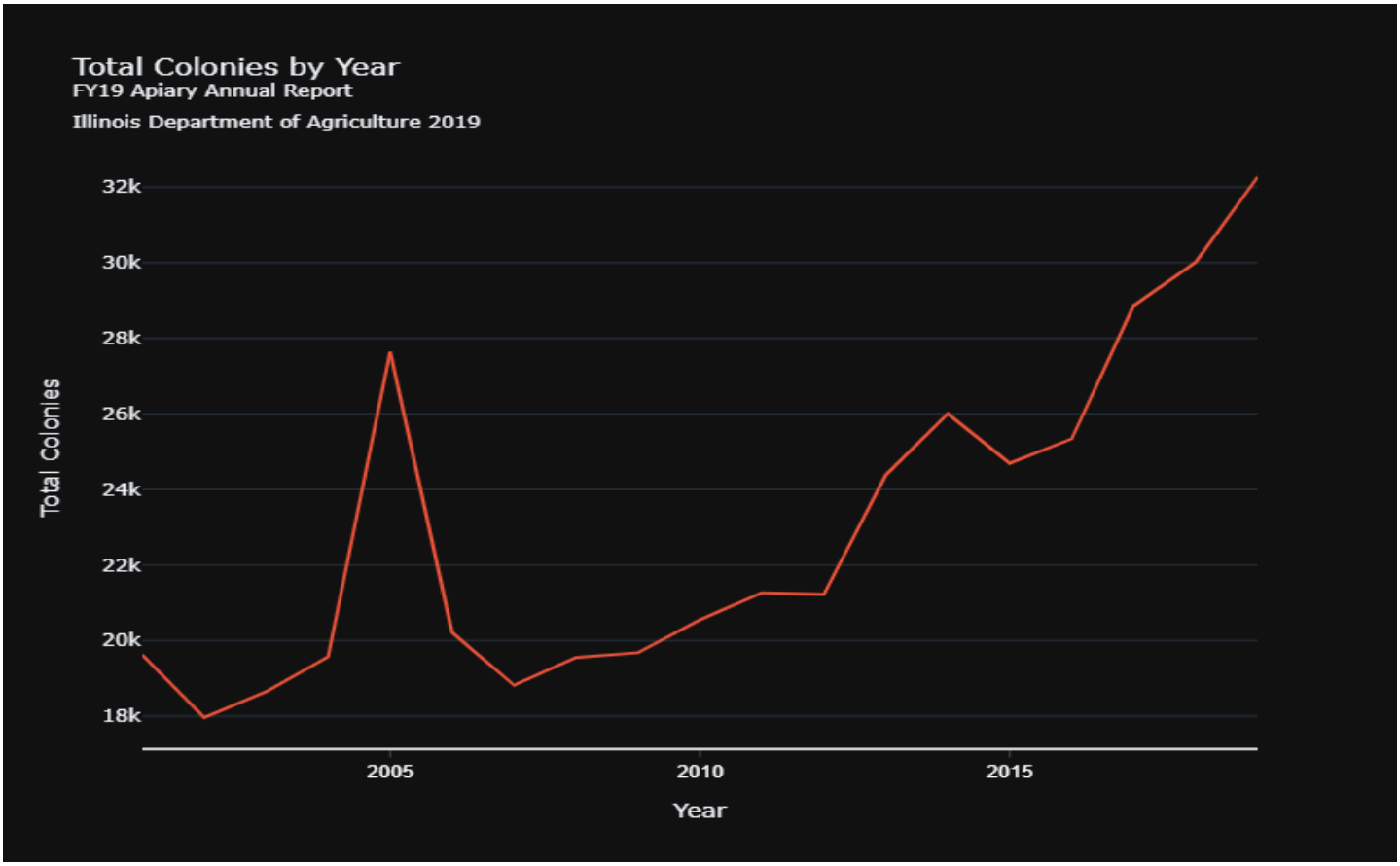
The main problem I encountered was that Tabula had trouble extracting the table headers. To overcome this, I deleted all of the rows that had the subsets of the complete column headers and replaced them with new headers that would be easier to use programmatically.

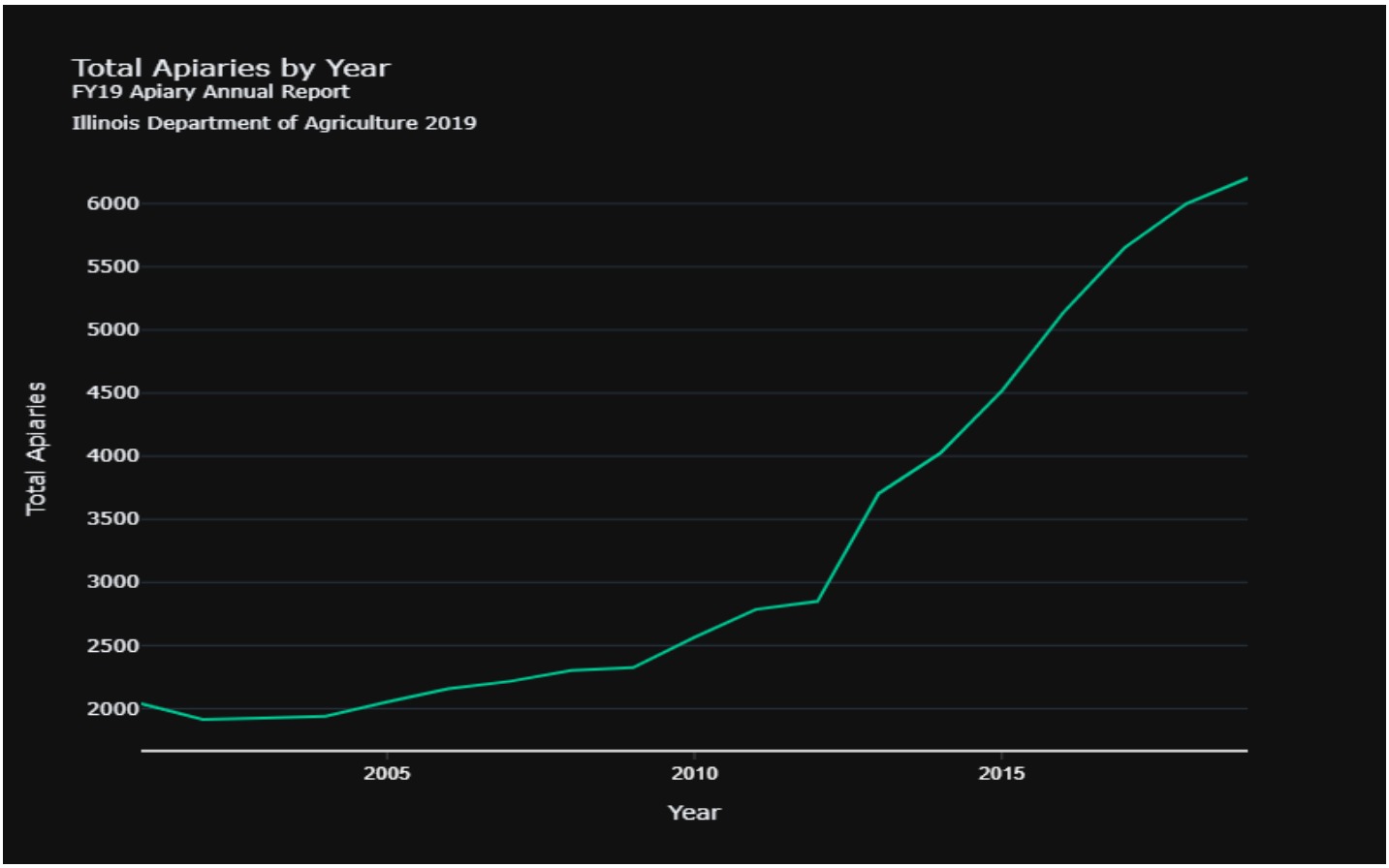
After this was accomplished, I flipped the data frame so years were represented from the minimum value (2001) to the maximum value (2019). I did this in order to utilize Python's Panda libraries built in percent change function as I wanted to be able to visualize the year over year change. The resulting data frame can be seen below.

	year	num_keepers	num_apiaries	num_colonies	keepers_pct_chg	colonies_pct_chg	apiaries_pct_chg
0	2001	1160	2038	19627	NaN	NaN	NaN
1	2002	1107	1914	17963	-0.045690	-0.084781	-0.060844
2	2003	1117	1926	18649	0.009033	0.038190	0.006270
3	2004	1141	1940	19572	0.021486	0.049493	0.007269
4	2005	1213	2054	27646	0.063103	0.412528	0.058763

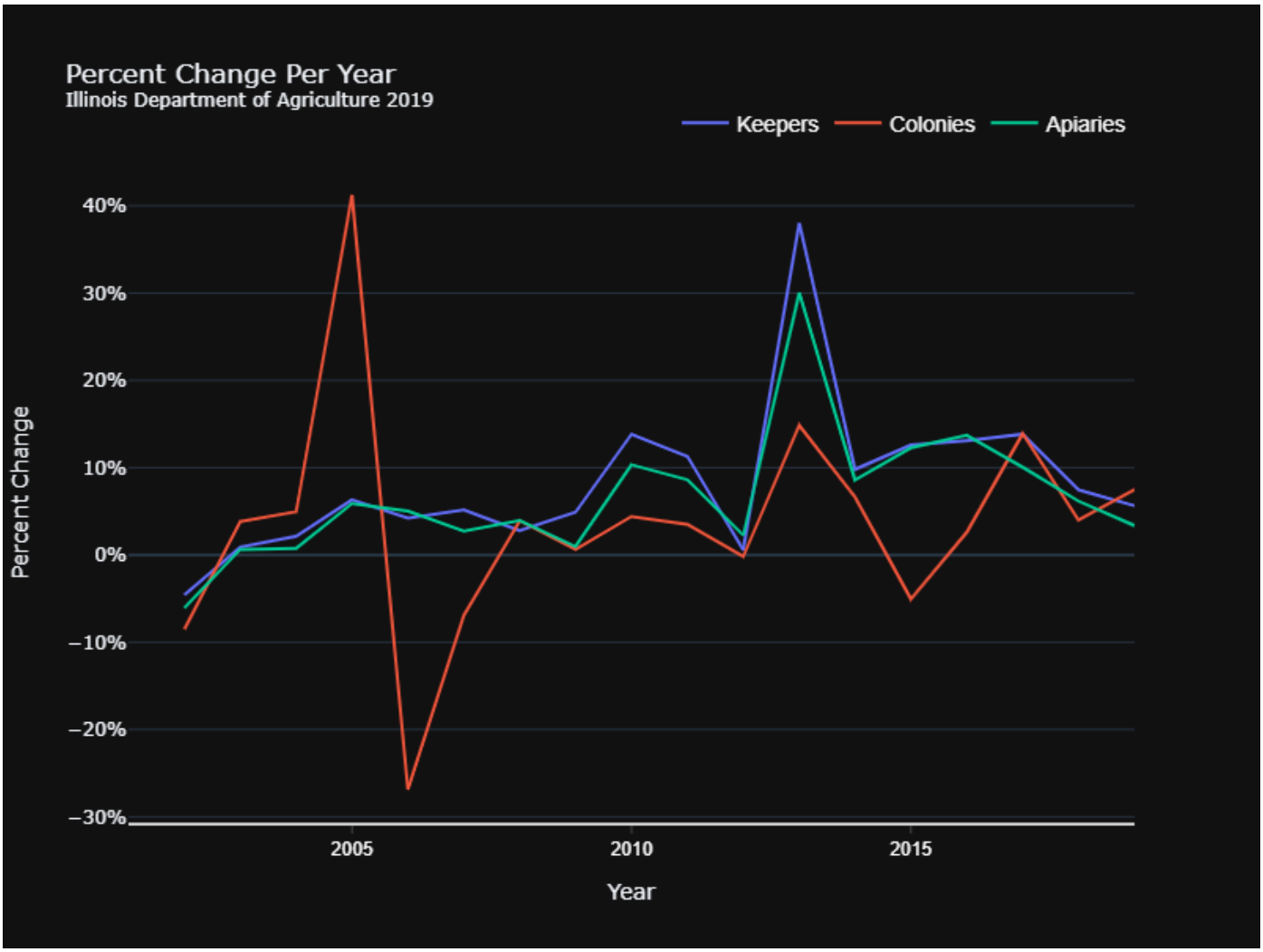
Now we have ourselves a nice and tidy data set that we can use to start visualizing the data in a way that will lend itself to insights that are difficult to see when the data is in a table format. The first visualizations I produced were the change in the number of keepers, colonies, and apiaries from 2001 to 2019. For the most part all of these variables follow a similar slope except for our total colonies. There is a sharp rise from 2004 to 2005 and a sharp decrease from 2005 to 2006.




Total Colonies by Year
FY19 Apiary Annual Report
Illinois Department of Agriculture 2019
32k
30k
28k
26k
24k
22k
20k
18k
Total Colonies
2005
2010
2015
Year


Total Apiaries by Year
FY19 Apiary Annual Report
Illinois Department of Agriculture 2019
6000
5500
5000
4500
4000
3500
3000
2500
2000
Total Apiaries
2005
2010
2015
Year

I then plotted the percent change year over year for these three variables on the same plot. There is a very clear question that comes from this. Why was there a 27% decrease in the number of colonies from 2005 to 2006? In that same period keepers and apiaries increased by about 4%.



The last thing to do was to export our data frame to excel. This is a format that more people are familiar with, lends itself to further manipulation, and can be imported into Python for further use.

And with all that, that is how a rabbit hole that led to looking at annual apiary reports, turned into a learning adventure and a new project. We were able to successfully extract a table from a PDF, manipulate the date, engineer new features, and create visualizations that led to new insights

Sources:

Annual Apiary reports. Illinois.gov. (n.d.). Retrieved October 17, 2021, from https://www2.illinois.gov/sites/agr/Insects/Bees/Pages/Annual-Apiary-Reports.aspx.