Can a State Tax Incentive Increase Solar Panel Adoption Rates in the State of Colorado?

James Nelson

MSDS692 Data Science Practicum I

Professor Gannous, Aiman

Regis University

Photo by Matthew Smith on Unsplash

Introduction:

The original question was could the CO2 emissions produced by Colorado power plants be alleviated if enough rooftop solar panels were installed. The obvious answer to this question is yes, but then I redirected the question and asked how many carbon metric tons could be lifted? However, that question there is just a quantifiable measure after some calculations and so I needed to ask a more in depth and puzzling question. After some brainstorming the I decided I needed to know if it is possible to predict rooftop solar panel installation rates with stat tax incentives?

Contents

1. Understanding the issue

1.1 The Problem

The state of Colorado doesn't currently have any state tax incentives when it comes to purchasing and installing solar panels. The current incentives are federal solar tax credit and solar renewable energy credits. The federal tax credit equates to taking off 26% from the total upfront cost which can be a lot depending on the system, but it doesn't cover enough to really spark an interest in others when calculating the system year's payback. In my sample of random household addresses and their assumed solar data estimates based on a default $80 electric bill, the average total years until system payback is 15. So, waiting 15 years to finally start seeing positive return for an investment that is supposed to save money doesn't really provide the willpower to make that purchase. However, I want to present a solution.

1.2 The Solution

What if the government of Colorado were to introduce state solar tax credits or incentives of their own? I personally believe that if this were the case, Colorado would see an increase in adoption rates for installing rooftop solar systems. For one, installing solar systems is better for the environment when talking about alleviating CO2 emissions from power plants and it is sustainable.

1.3The Plan

The plan for this project is to gather data on total energy consumption, production, price expenditure, CO2 emissions, randomized house address sample, and solar data for those houses to engineer the features necessary to create this hypothetical solution. The state tax incentive will be $2,000 addition to the subtracting from the initial upfront cost of this random house address solar data sample. The calculated average for system years payback for this sample size is 13 years.

2. Mining the Data


"A coal mine" Image by hangela from Pixabay

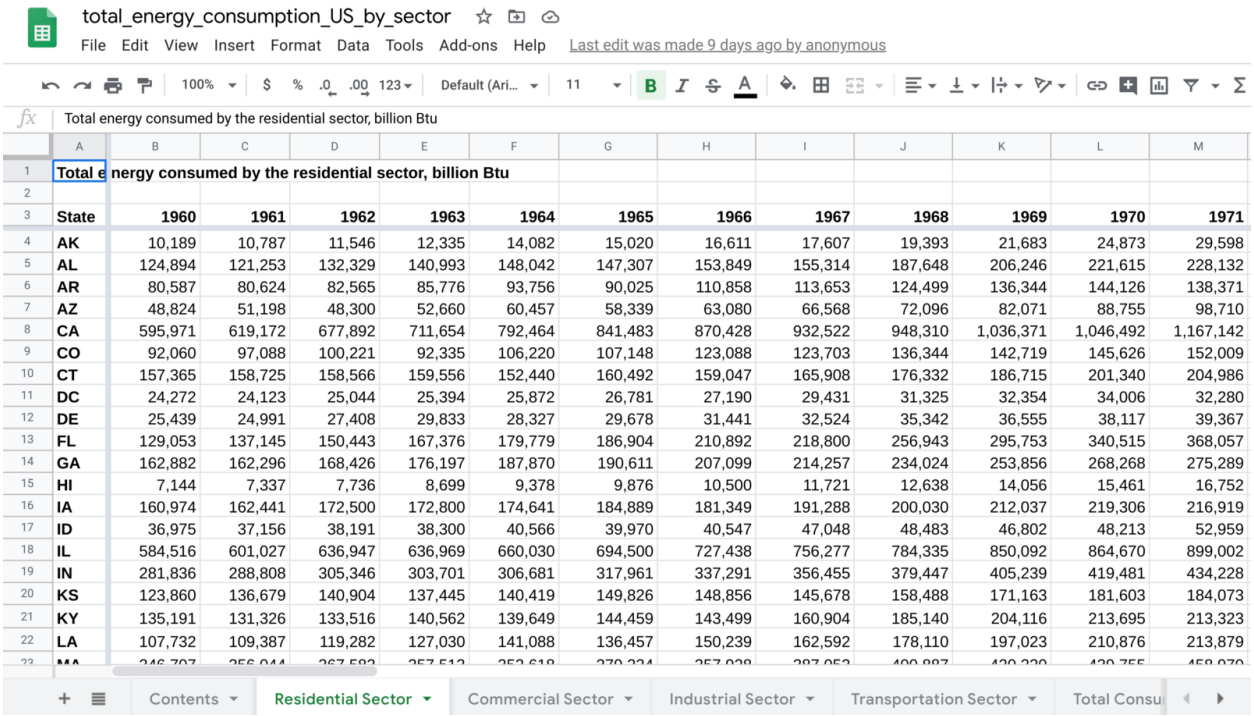
One of the most important parts of the process begins with mining the data. It is important to keep in mind the data that is being collected has consistency; while there will always be outliers, the data needs to be able to group together nicely for a clean integration, so a sufficient model can actually forecast the answer to this great question.

2.1 U.S. Energy Information Administration

Colorado

*Released: June 28, 2019 | Next release: June 26, 2020 Comprehensive state-level estimates of energy production…*www.eia.gov

The main source where I scraped a lot of data was from The U.S. Energy Information Administration's page. The formation of the EIA was a response to the energy crisis that the U.S. was facing in the 1970s. The purpose of the EIA was and still is to collect, analyze, and disseminate independent and impartial energy data in order to give way for justified policymaking, high-performance markets, public transparency with the energy, and the impact it has with the economy and the environment ("About EIA — U.S. Energy Information Administration (EIA) — U.S. Energy Information Administration (EIA)," 2016).



total_energy_consumption_US_by_sector

File Edit View Insert Format Data Tools Add-ons Help Last edit was made 9 days ago by anonymous

Total energy consumed by the residential sector, billion Btu

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Total energy consumed by the residential sector, billion Btu												
2													
3	State	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971
4	AK	10,189	10,787	11,546	12,335	14,082	15,020	16,611	17,607	19,393	21,683	24,873	29,598
5	AL	124,894	121,253	132,329	140,993	148,042	147,307	153,849	155,314	187,648	206,246	221,615	228,132
6	AR	80,587	80,624	82,565	85,776	93,756	90,025	110,858	113,653	124,499	136,344	144,126	138,371
7	AZ	48,824	51,198	48,300	52,660	60,457	58,339	63,080	66,568	72,096	82,071	88,755	98,710
8	CA	595,971	619,172	677,892	711,654	792,464	841,483	870,428	932,522	948,310	1,036,371	1,046,492	1,167,142
9	CO	92,060	97,088	100,221	92,335	106,220	107,148	123,088	123,703	136,344	142,719	145,626	152,009
10	CT	157,365	158,725	158,566	159,556	152,440	160,492	159,047	165,908	176,332	186,715	201,340	204,986
11	DC	24,272	24,123	25,044	25,394	25,872	26,781	27,190	29,431	31,325	32,354	34,006	32,280
12	DE	25,439	24,991	27,408	29,833	28,327	29,678	31,441	32,524	35,342	36,555	38,117	39,367
13	FL	129,053	137,145	150,443	167,376	179,779	186,904	210,892	218,800	256,943	295,753	340,515	368,057
14	GA	162,882	162,296	168,426	176,197	187,870	190,611	207,099	214,257	234,024	253,856	268,268	275,289
15	HI	7,144	7,337	7,736	8,699	9,378	9,876	10,500	11,721	12,638	14,056	15,461	16,752
16	IA	160,974	162,441	172,500	172,800	174,641	184,889	181,349	191,288	200,030	212,037	219,306	216,919
17	ID	36,975	37,156	38,191	38,300	40,566	39,970	40,547	47,048	48,483	46,802	48,213	52,959
18	IL	584,516	601,027	636,947	636,969	660,030	694,500	727,438	756,277	784,335	850,092	864,670	899,002
19	IN	281,836	288,808	305,346	303,701	306,681	317,961	337,291	356,455	379,447	405,239	419,481	434,228
20	KS	123,860	136,679	140,904	137,445	140,419	149,826	148,856	145,678	158,488	171,163	181,603	184,073
21	KY	135,191	131,326	133,516	140,562	139,649	144,459	143,499	160,904	185,140	204,116	213,695	213,323
22	LA	107,732	109,387	119,282	127,030	141,088	136,457	150,239	162,592	178,110	197,023	210,876	213,879

Contents | Residential Sector | Commercial Sector | Industrial Sector | Transportation Sector | Total Consu

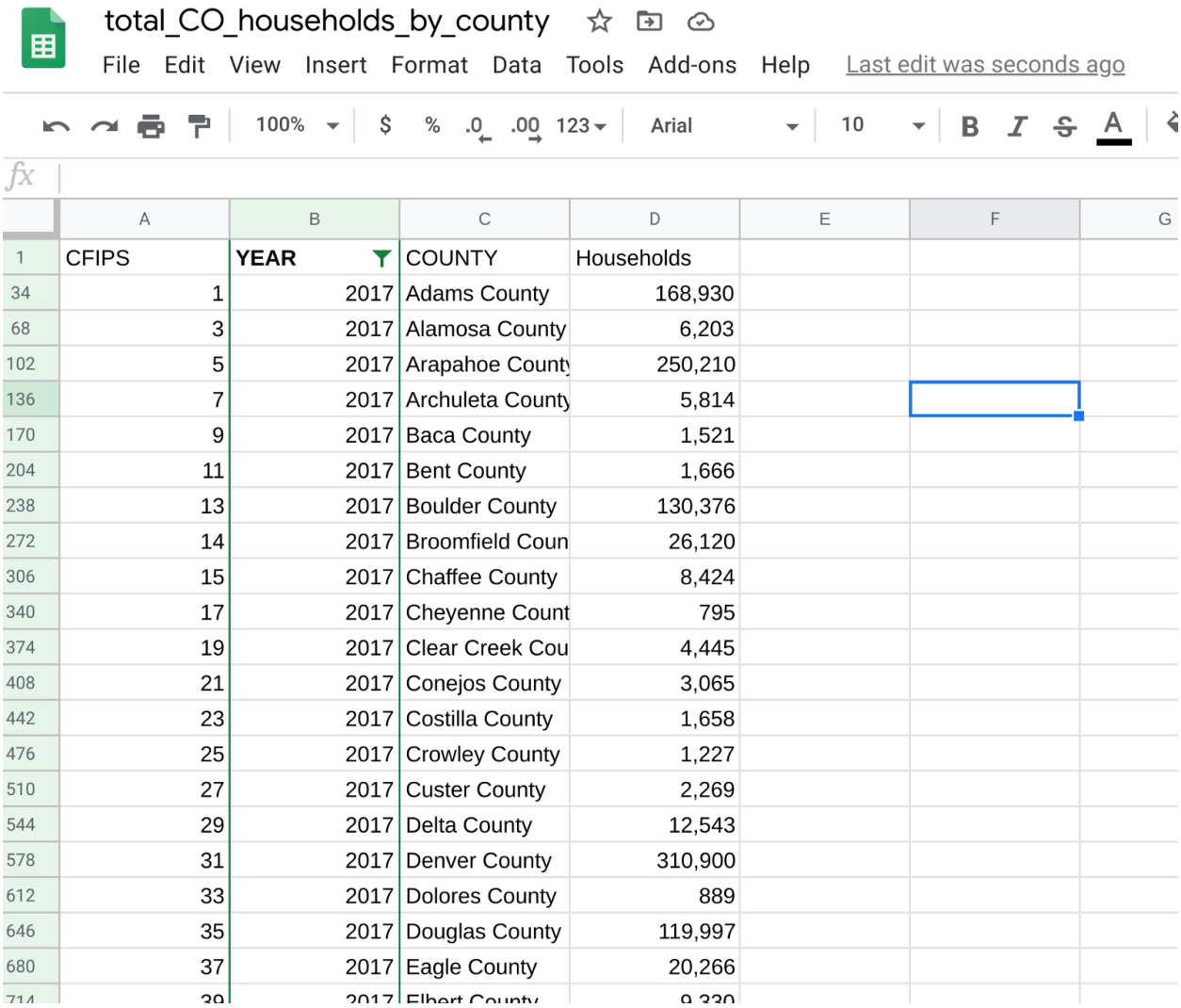
"A screenshot of the total energy consumption of the residential sector in the US"

The format of the data is CSV. The page where I got the data from was the Colorado SEDS (State Energy Data System). From this page, I can not only collect data from the state of Colorado, but also from the entire nation as well. All the data I mined from Colorado and the U.S. (between 1960–2017) was total energy consumption, total energy production, total net generation for solar energy, total net generation for all fuels, total energy expenditure, total renewable energy consumption, and Colorado CO2 emissions. The purpose of gathering this data is to have historical info on how the distinct types of energy fuel sources were produced and consumed to investigate the correlation between CO2 level emissions when more renewable energy sources were introduced. It will be extremely beneficial with the pricing data as well to showcase how feasible it would be for the state of Colorado to install solar panels incrementally over the years.

2.2 The Demography of Household Data in Colorado Counties

County Data Lookup

*Search for information on total population, births, deaths, natural increase, net migration, building permits, group…*demography.dola.colorado.gov



total_CO_households_by_county

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

100% $ % .0 .00 123 Arial 10

	A	B	C	D	E	F	G
1	CFIPS	YEAR	COUNTY	Households			
34	1	2017	Adams County	168,930			
68	3	2017	Alamosa County	6,203			
102	5	2017	Arapahoe County	250,210			
136	7	2017	Archuleta County	5,814			
170	9	2017	Baca County	1,521			
204	11	2017	Bent County	1,666			
238	13	2017	Boulder County	130,376			
272	14	2017	Broomfield Coun	26,120			
306	15	2017	Chaffee County	8,424			
340	17	2017	Cheyenne Count	795			
374	19	2017	Clear Creek Cou	4,445			
408	21	2017	Conejos County	3,065			
442	23	2017	Costilla County	1,658			
476	25	2017	Crowley County	1,227			
510	27	2017	Custer County	2,269			
544	29	2017	Delta County	12,543			
578	31	2017	Denver County	310,900			
612	33	2017	Dolores County	889			
646	35	2017	Douglas County	119,997			
680	37	2017	Eagle County	20,266			
714	39	2017	Elbert County	9,330			

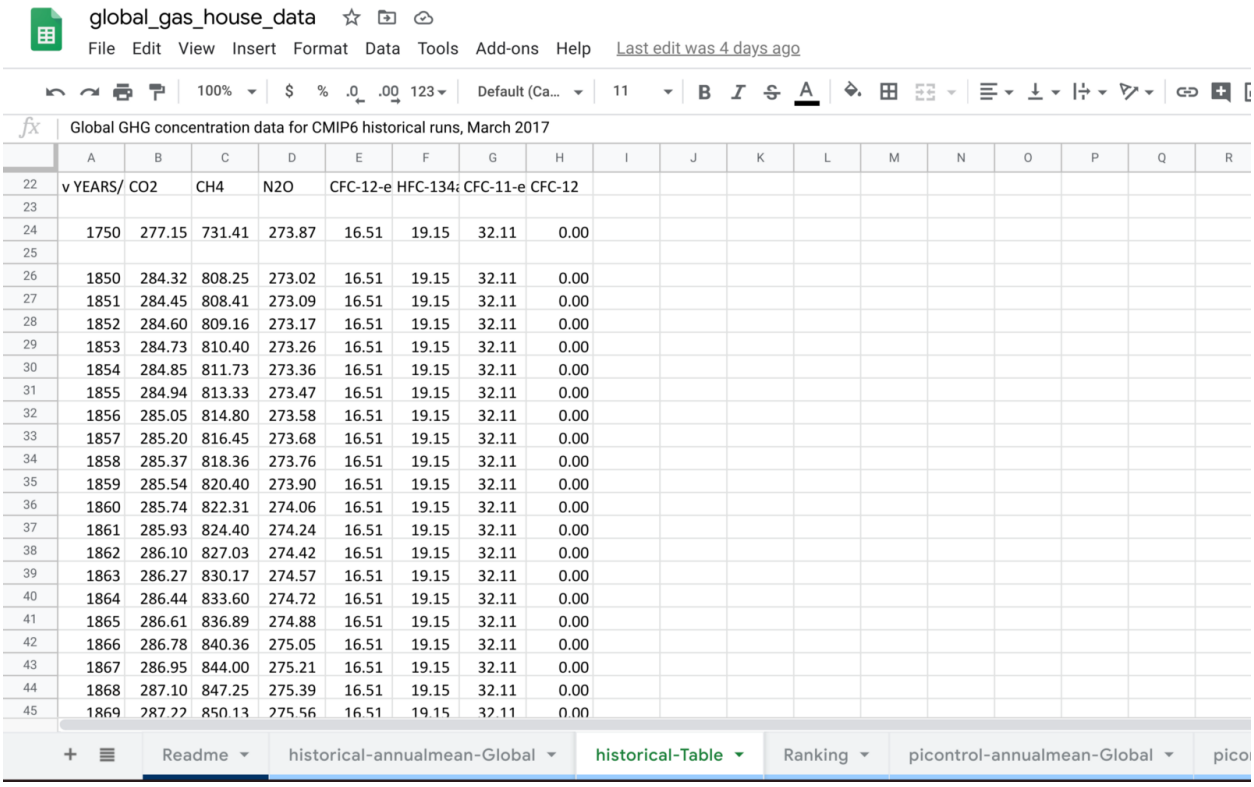
“A screenshot of the total households in the state of Colorado in 2017”

The next source of data where I collected from was the demographic information on households in Colorado counties on Colorado.gov. “The State Demography Office is committed to increasing data accessibility for the public” (*COLORADO STATE DEMOGRAPHY OFFICE*, 2020). The format of this data is in CSV, and only data I collected from this source was the number of households per county back in 2017. I will use this to create an assumption for the relevant location of the houses in those counties to estimate how much solar energy they potentially could produce.

2.3 Greenhouse Gas CMIP6 Data

CMIP6 Data

*Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., Fraser, P. J., Montzka, S…*climatecollege.unimelb.edu.au



global_gas_house_data

File Edit View Insert Format Data Tools Add-ons Help Last edit was 4 days ago

Global GHG concentration data for CMIP6 historical runs, March 2017

	A	B	C	D	E	F	G	H
22	v YEARS/	CO2	CH4	N2O	CFC-12-e	HFC-134a	CFC-11-e	CFC-12
23								
24	1750	277.15	731.41	273.87	16.51	19.15	32.11	0.00
25								
26	1850	284.32	808.25	273.02	16.51	19.15	32.11	0.00
27	1851	284.45	808.41	273.09	16.51	19.15	32.11	0.00
28	1852	284.60	809.16	273.17	16.51	19.15	32.11	0.00
29	1853	284.73	810.40	273.26	16.51	19.15	32.11	0.00
30	1854	284.85	811.73	273.36	16.51	19.15	32.11	0.00
31	1855	284.94	813.33	273.47	16.51	19.15	32.11	0.00
32	1856	285.05	814.80	273.58	16.51	19.15	32.11	0.00
33	1857	285.20	816.45	273.68	16.51	19.15	32.11	0.00
34	1858	285.37	818.36	273.76	16.51	19.15	32.11	0.00
35	1859	285.54	820.40	273.90	16.51	19.15	32.11	0.00
36	1860	285.74	822.31	274.06	16.51	19.15	32.11	0.00
37	1861	285.93	824.40	274.24	16.51	19.15	32.11	0.00
38	1862	286.10	827.03	274.42	16.51	19.15	32.11	0.00
39	1863	286.27	830.17	274.57	16.51	19.15	32.11	0.00
40	1864	286.44	833.60	274.72	16.51	19.15	32.11	0.00
41	1865	286.61	836.89	274.88	16.51	19.15	32.11	0.00
42	1866	286.78	840.36	275.05	16.51	19.15	32.11	0.00
43	1867	286.95	844.00	275.21	16.51	19.15	32.11	0.00
44	1868	287.10	847.25	275.39	16.51	19.15	32.11	0.00
45	1869	287.22	850.13	275.56	16.51	19.15	32.11	0.00

Readme | historical-annualmean-Global | historical-Table | Ranking | picontrol-annualmean-Global

"A screenshot from the CMIP6 data historical-Table sheet on greenhouse gases"

I gathered data on historical figures for the diverse types of greenhouse gases in our atmosphere from the University of Melbourne Climate College website. The University of Melbourne Climate College has been around since 1853 and making contributions toward multiple research communities ("The University of Melbourne, Australia — Australia's best university and one of the world's finest," 2019). The format of this dataset is in CSV and is a combined effort from numerous contributing researchers from various parts of the globe to provide the most up to date and accurate information on current greenhouse gases. Hopefully, the greenhouse gas' data and the data on CO2 emissions from various sources across the U.S. will integrate smoothly.

2.4 Zip code, Random Address, and Sunlight for Address Data

Listing of all Zip Codes in the state of Colorado

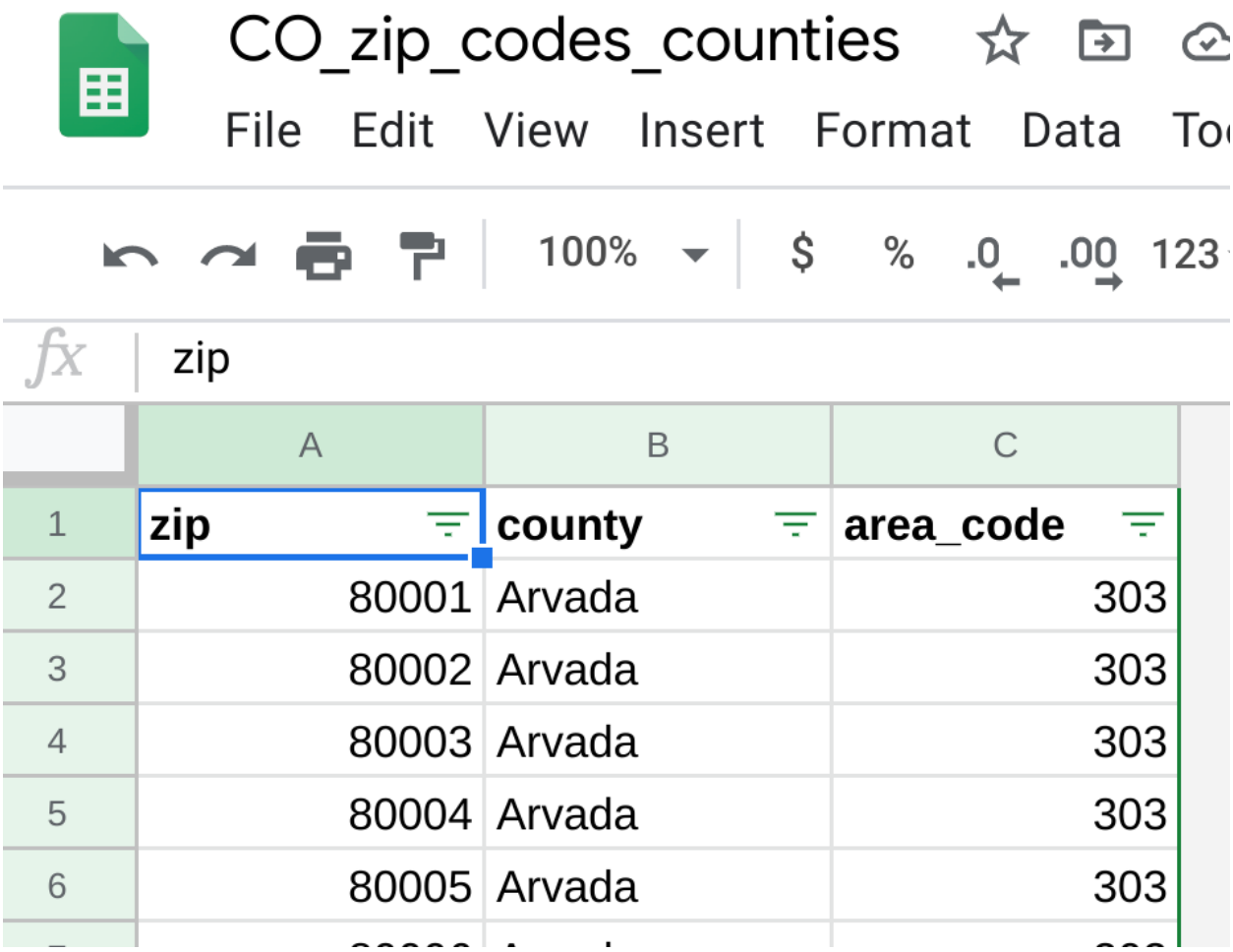
*List of all Zip Codes for the state of Colorado, CO. Includes all counties and cities in Colorado.*www.zip-codes.com

Real Estate, Homes for Sale, MLS Listings, Agents | Redfin

*Search all real estate listings. Tour homes and make offers with the help of local Redfin real estate agents.*www.redfin.com

Project Sunroof
*Enter a state, county, city, or zip code to see a solar estimate for the area, based on the amount of usable sunlight…*www.google.com

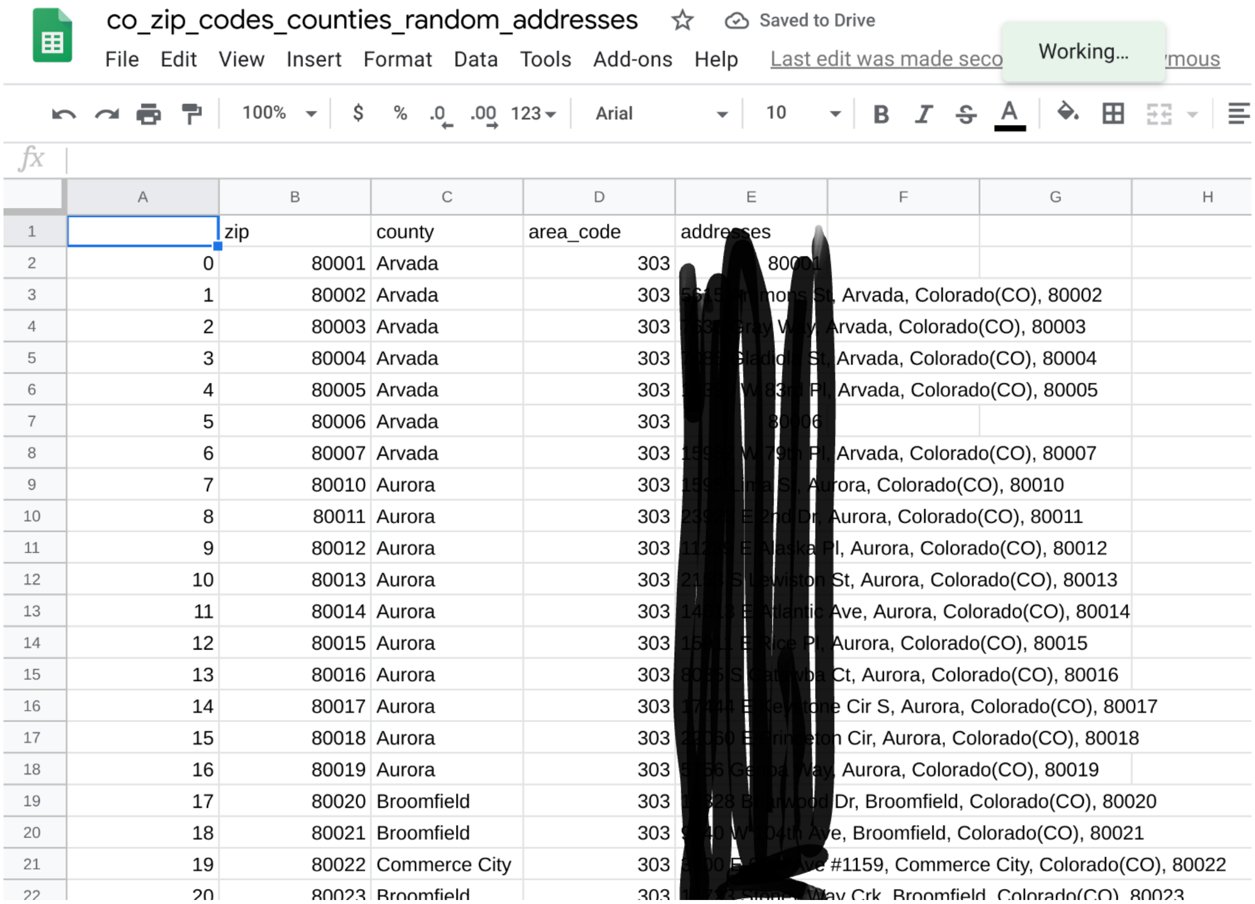
The next set of data will be more of a combination of three websites Project Sunroof by Google, a random address generator by redfin.com, and a listing of all zip codes in Colorado at mongabay.com. This will be a pipelined pre-integrated dataset right off the bat. Project sunroof provides a calculator based on the address provided to give an estimate on hours of useable sunlight per year, square footage available on the roof, the estimated savings after 20 years, estimated necessary kW based off of the electric bill (I will use the default on the page as an assumption), and the big one estimation for an environmental impact on in carbon metric tons. It should also be mentioned that some areas are not covered on Project Sunroof so it is possible not every zip code will provide data for solar energy.



	A	B	C
1	zip	county	area_code
2	80001	Arvada	303
3	80002	Arvada	303
4	80003	Arvada	303
5	80004	Arvada	303
6	80005	Arvada	303

How I got the zip code data was copying and pasting every Colorado county zip code from Mongabay on a CSV sheet. To compile a list of random addresses, I used the list of zip codes in each county to generate the random addresses on the redfin website with the help of the program module Selenium Webdriver. Selenium is helpful because it creates a web driver that automates browsing based on

programmed scripts. I was able to do this inside of Google Collaboratory, but the video above shows what is happening without hiding the driver's activity.

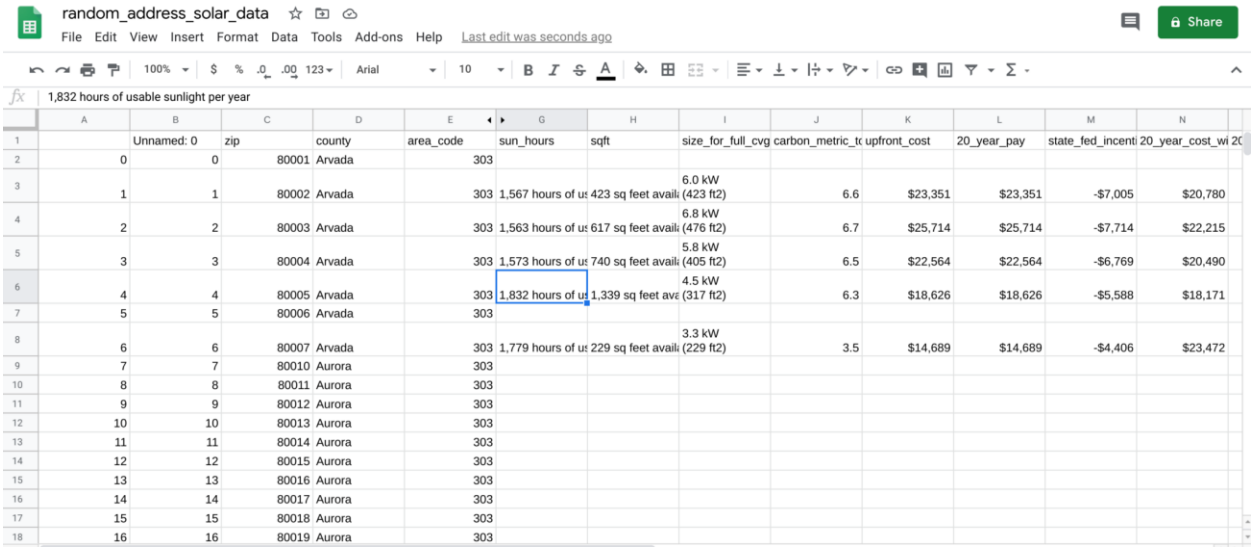


	zip	county	area_code	addresses
0	80001	Arvada	303	[redacted]
1	80002	Arvada	303	[redacted], Arvada, Colorado(CO), 80002
2	80003	Arvada	303	[redacted], Colorado(CO), 80003
3	80004	Arvada	303	[redacted] Arvada, Colorado(CO), 80004
4	80005	Arvada	303	[redacted] Arvada, Colorado(CO), 80005
5	80006	Arvada	303	[redacted]
6	80007	Arvada	303	[redacted] Arvada, Colorado(CO), 80007
7	80010	Aurora	303	[redacted], Colorado(CO), 80010
8	80011	Aurora	303	[redacted] Aurora, Colorado(CO), 80011
9	80012	Aurora	303	[redacted] Pl, Aurora, Colorado(CO), 80012
10	80013	Aurora	303	[redacted] St, Aurora, Colorado(CO), 80013
11	80014	Aurora	303	[redacted] Ave, Aurora, Colorado(CO), 80014
12	80015	Aurora	303	[redacted] Aurora, Colorado(CO), 80015
13	80016	Aurora	303	[redacted] Ct, Aurora, Colorado(CO), 80016
14	80017	Aurora	303	[redacted] Cir S, Aurora, Colorado(CO), 80017
15	80018	Aurora	303	[redacted] Cir, Aurora, Colorado(CO), 80018
16	80019	Aurora	303	[redacted] Aurora, Colorado(CO), 80019
17	80020	Broomfield	303	[redacted] Dr, Broomfield, Colorado(CO), 80020
18	80021	Broomfield	303	[redacted], Broomfield, Colorado(CO), 80021
19	80022	Commerce City	303	[redacted] #1159, Commerce City, Colorado(CO), 80022
20	80023	Broomfield	303	[redacted] Way Crk, Broomfield, Colorado(CO), 80023

"A screenshot of randomly generated addresses from bestrandoms"

I have scribbled out the addresses at the top for the sake of maintaining privacy. Some of the data is missing complete addresses because the zip code the bot suggested didn't trigger any addresses to be generated by the website.

After gathering (scraping) the addresses information, I then used Selenium to ascertain the relative hours of annual sunlight.



random_address_solar_data

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

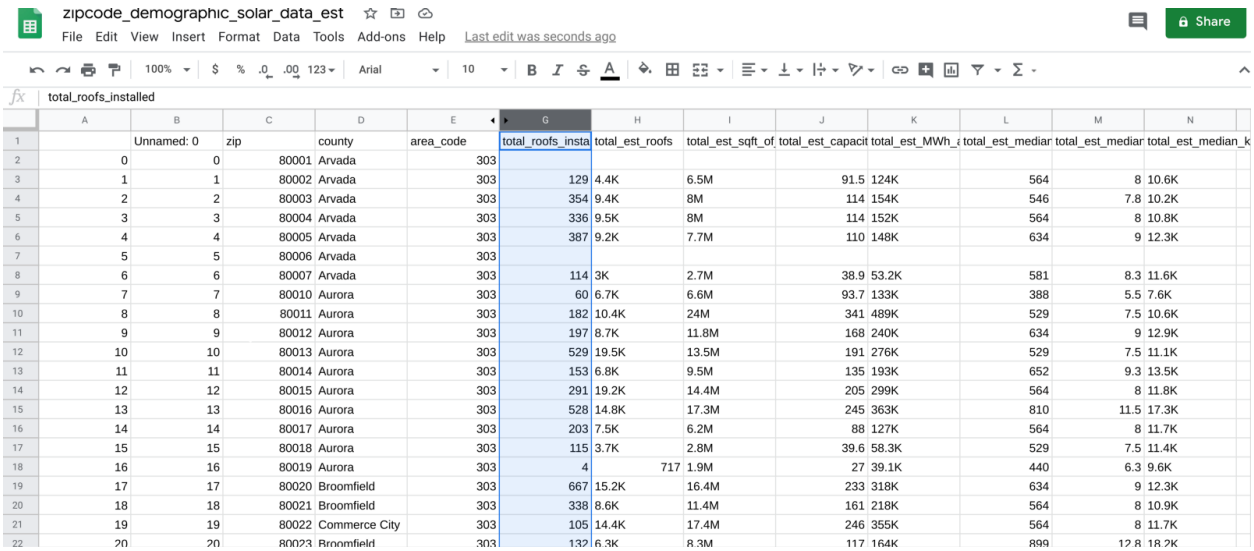
Share

1,832 hours of usable sunlight per year

	Unnamed: 0	zip	county	area_code	sun_hours	sqft	size_for_full_cvg	carbon_metric_t	upfront_cost	20_year_pay	state_fed_incenti	20_year_cost_wi
0	0	80001	Arvada	303								
1	1	80002	Arvada	303	1,567 hours of us	423 sq feet avail	6.0 kW (423 ft2)	6.6	$23,351	$23,351	-$7,005	$20,780
2	2	80003	Arvada	303	1,563 hours of us	617 sq feet avail	6.8 kW (476 ft2)	6.7	$25,714	$25,714	-$7,714	$22,215
3	3	80004	Arvada	303	1,573 hours of us	740 sq feet avail	5.8 kW (405 ft2)	6.5	$22,564	$22,564	-$6,769	$20,490
4	4	80005	Arvada	303	1,832 hours of us	1,339 sq feet ava	4.5 kW (317 ft2)	6.3	$18,626	$18,626	-$5,588	$18,171
5	5	80006	Arvada	303								
6	6	80007	Arvada	303	1,779 hours of us	229 sq feet avail	3.3 kW (229 ft2)	3.5	$14,689	$14,689	-$4,406	$23,472
7	7	80010	Aurora	303								
8	8	80011	Aurora	303								
9	9	80012	Aurora	303								
10	10	80013	Aurora	303								
11	11	80014	Aurora	303								
12	12	80015	Aurora	303								
13	13	80016	Aurora	303								
14	14	80017	Aurora	303								
15	15	80018	Aurora	303								
16	16	80019	Aurora	303								

"A screenshot of the solar energy estimation from Project Sunroof"

I have hidden the column of addresses to maintain the privacy of those owners, but as you can see there are some rows without data because those random addresses didn't yield real properties. Below is a video of the process without hiding the bot's activities. Of course, the zip code pulled back an address from Florida, but that will be taken care of when cleaning the data.



zipcode_demographic_solar_data_est

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

Share

total_roofs_installed

	Unnamed: 0	zip	county	area_code	total_roofs_insta	total_est_roofs	total_est_sqft_of	total_est_capacit	total_est_MWh_a	total_est_median	total_est_median	total_est_median_k
0	0	80001	Arvada	303								
1	1	80002	Arvada	303	129	4.4K	6.5M	91.5	124K	564	8	10.6K
2	2	80003	Arvada	303	354	9.4K	8M	114	154K	546	7.8	10.2K
3	3	80004	Arvada	303	336	9.5K	8M	114	152K	564	8	10.8K
4	4	80005	Arvada	303	387	9.2K	7.7M	110	148K	634	9	12.3K
5	5	80006	Arvada	303								
6	6	80007	Arvada	303	114	3K	2.7M	38.9	53.2K	581	8.3	11.6K
7	7	80010	Aurora	303	60	6.7K	6.6M	93.7	133K	388	5.5	7.6K
8	8	80011	Aurora	303	182	10.4K	24M	341	489K	529	7.5	10.6K
9	9	80012	Aurora	303	197	8.7K	11.8M	168	240K	634	9	12.9K
10	10	80013	Aurora	303	529	19.5K	13.5M	191	276K	529	7.5	11.1K
11	11	80014	Aurora	303	153	6.8K	9.5M	135	193K	652	9.3	13.5K
12	12	80015	Aurora	303	291	19.2K	14.4M	205	299K	564	8	11.8K
13	13	80016	Aurora	303	528	14.8K	17.3M	245	363K	810	11.5	17.3K
14	14	80017	Aurora	303	203	7.5K	6.2M	88	127K	564	8	11.7K
15	15	80018	Aurora	303	115	3.7K	2.8M	39.6	58.3K	529	7.5	11.4K
16	16	80019	Aurora	303	4	717	1.9M	27	39.1K	440	6.3	9.6K
17	17	80020	Broomfield	303	667	15.2K	16.4M	233	318K	634	9	12.3K
18	18	80021	Broomfield	303	338	8.6K	11.4M	161	218K	564	8	10.9K
19	19	80022	Commerce City	303	105	14.4K	17.4M	246	355K	564	8	11.7K
20	20	80023	Broomfield	303	132	6.3K	8.3M	117	164K	899	12.8	18.2K

"Screenshot of demographic data on solar energy for zip codes"

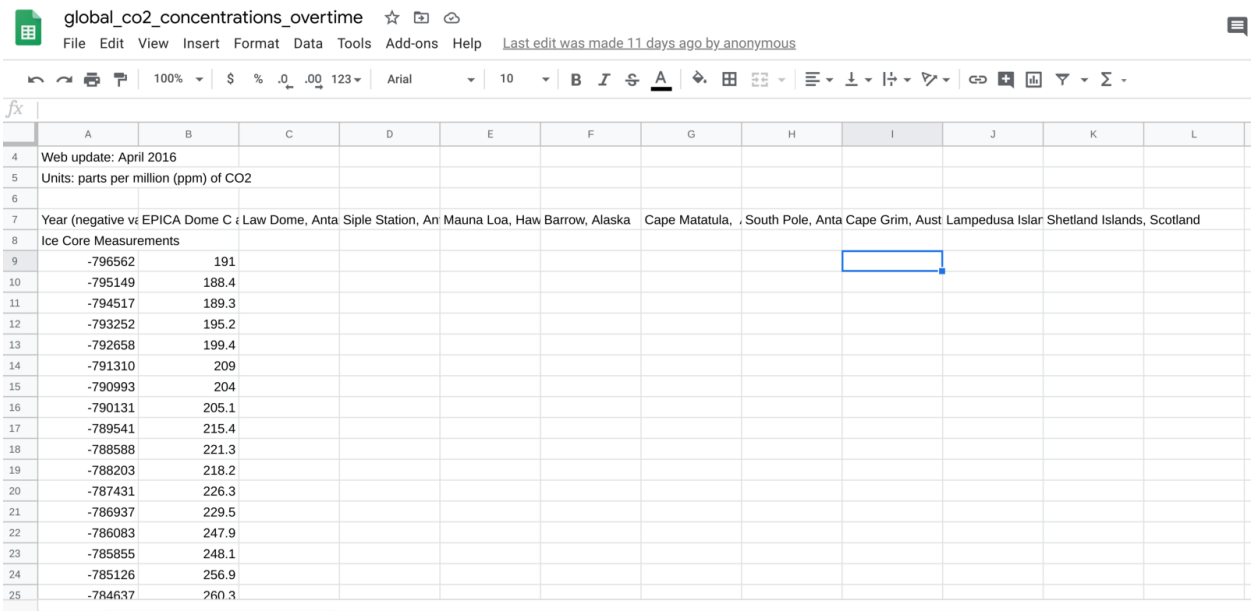
However, this data will be essential for estimating and forecasting. I did one more pull of data from Project Sunroof upon discovering that there was another page that provided the demographic data of each county with the total number of roofs that have solar panels, the estimated number of roofs in that county without solar panels, the estimated sqft roof available, the estimated total capacity of MW DC, the estimated electricity generate in MWh AC per year, and the median estimate per house with the previously listed metrics but estimated with kW DC for capacity and kWH AC for electricity generated per year.

2.5 Environmental Protection Agency on 800,000 Years on Ice Core Measurements

Climate Change Indicators: Atmospheric Concentrations of Greenhouse Gases | US EPA

*Global atmospheric concentrations of carbon dioxide, methane, nitrous oxide, and certain manufactured greenhouse gases…*www.epa.gov

The last piece of data I got was from the Environmental Protection Agency climate change indicators page with concentrations on greenhouse gases.



	A	B	C	D	E	F	G	H	I	J	K	L
4	Web update: April 2016											
5	Units: parts per million (ppm) of CO2											
6												
7	Year (negative va	EPICA Dome C a	Law Dome, Anta	Siple Station, An	Mauna Loa, Haw	Barrow, Alaska	Cape Matatula,	South Pole, Anta	Cape Grim, Aust	Lampedusa Islar	Shetland Islands,	Scotland
8	Ice Core Measurements											
9	-796562	191										
10	-795149	188.4										
11	-794517	189.3										
12	-793252	195.2										
13	-792658	199.4										
14	-791310	209										
15	-790993	204										
16	-790131	205.1										
17	-789541	215.4										
18	-788588	221.3										
19	-788203	218.2										
20	-787431	226.3										
21	-786937	229.5										
22	-786083	247.9										
23	-785855	248.1										
24	-785126	256.9										
25	-784637	260.3										

“A screenshot of the EPA ice core measurement data”

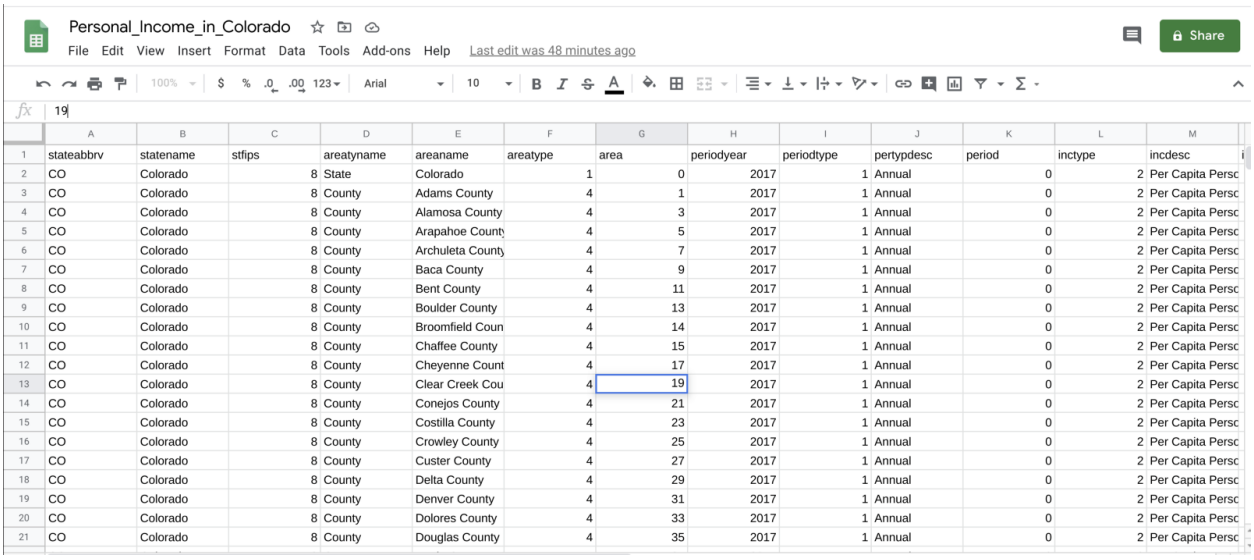
The data I collected has ice core measurements for the last 800,000 years from these locations EPICA Dome C and Vostok Station, AntarcticaLaw Dome, Antarctica (75-year smoothed)Siple Station, Antarctica Mauna Loa, Hawaii Barrow, Alaska Cape Matatula, American Samoa South Pole, Antarctica Cape Grim, Australia Lamped USA Island, Italy, Shetland Islands, Scotland (US EPA, OAR, 2016). I hope to use this data for my time series forecast of climate change after the implementation of the incremental solution.

2.6 Per Capita Personal Income for Each County in Colorado Data

Colorado Information Marketplace | Colorado Information Marketplace | data.colorado.gov

*Home Data Catalog Help Video Tutorials Feedback Status Blog*data.colorado.gov

The next data set I mined was the annual per capita personal income of Colorado residence by county.



Personal_Income_in_Colorado

File Edit View Insert Format Data Tools Add-ons Help Last edit was 48 minutes ago

Share

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	stateabbrv	statename	stfips	areatyname	areaname	areatype	area	periodyear	periodtype	pertypdesc	period	inctype	incdesc
2	CO	Colorado	8	State	Colorado	1	0	2017	1	Annual	0	2	Per Capita Persc
3	CO	Colorado	8	County	Adams County	4	1	2017	1	Annual	0	2	Per Capita Persc
4	CO	Colorado	8	County	Alamosa County	4	3	2017	1	Annual	0	2	Per Capita Persc
5	CO	Colorado	8	County	Arapahoe County	4	5	2017	1	Annual	0	2	Per Capita Persc
6	CO	Colorado	8	County	Archuleta County	4	7	2017	1	Annual	0	2	Per Capita Persc
7	CO	Colorado	8	County	Baca County	4	9	2017	1	Annual	0	2	Per Capita Persc
8	CO	Colorado	8	County	Bent County	4	11	2017	1	Annual	0	2	Per Capita Persc
9	CO	Colorado	8	County	Boulder County	4	13	2017	1	Annual	0	2	Per Capita Persc
10	CO	Colorado	8	County	Broomfield Coun	4	14	2017	1	Annual	0	2	Per Capita Persc
11	CO	Colorado	8	County	Chaffee County	4	15	2017	1	Annual	0	2	Per Capita Persc
12	CO	Colorado	8	County	Cheyenne Count	4	17	2017	1	Annual	0	2	Per Capita Persc
13	CO	Colorado	8	County	Clear Creek Cou	4	19	2017	1	Annual	0	2	Per Capita Persc
14	CO	Colorado	8	County	Conejos County	4	21	2017	1	Annual	0	2	Per Capita Persc
15	CO	Colorado	8	County	Costilla County	4	23	2017	1	Annual	0	2	Per Capita Persc
16	CO	Colorado	8	County	Crowley County	4	25	2017	1	Annual	0	2	Per Capita Persc
17	CO	Colorado	8	County	Custer County	4	27	2017	1	Annual	0	2	Per Capita Persc
18	CO	Colorado	8	County	Delta County	4	29	2017	1	Annual	0	2	Per Capita Persc
19	CO	Colorado	8	County	Denver County	4	31	2017	1	Annual	0	2	Per Capita Persc
20	CO	Colorado	8	County	Dolores County	4	33	2017	1	Annual	0	2	Per Capita Persc
21	CO	Colorado	8	County	Douglas County	4	35	2017	1	Annual	0	2	Per Capita Persc

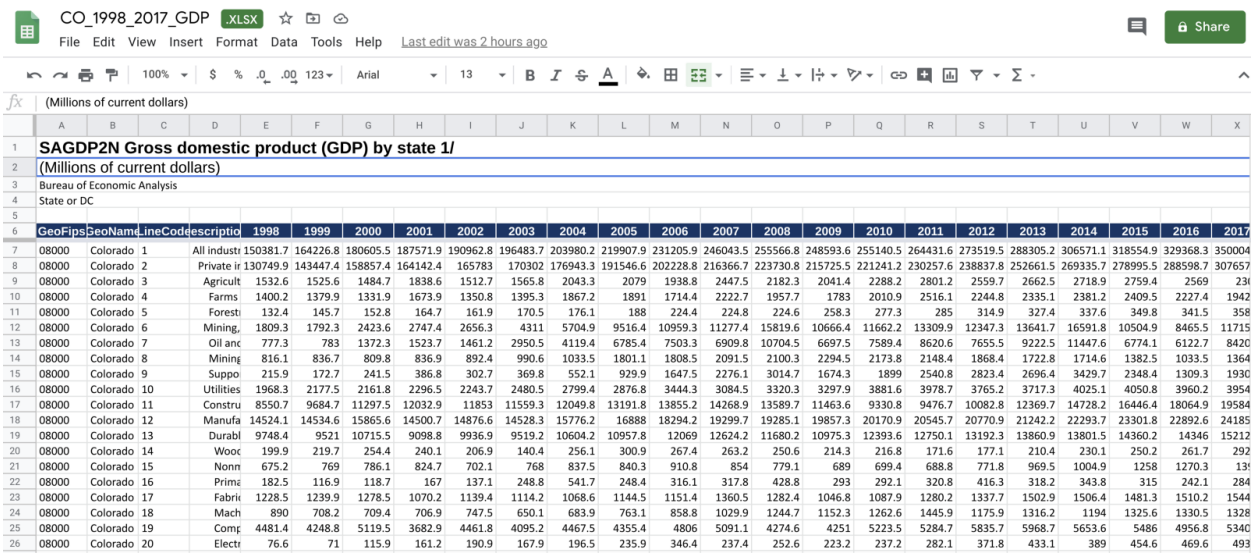
"A screenshot of the data for per capita personal income for Colorado by county"

I got this data from a data table off of Colorado Information Marketplace. Colorado Information Marketplace is a site full of different data sources such as COVID-19, business, education, health, public safety, recreation, water, agriculture, etc. The dataset I collected was filtered to personal per capita income by county in Colorado and set to the year 2017.

2.7 Colorado Annual GDP in Current Dollars Data

Bureau of Economic Analysis

*The U.S. monthly international trade deficit increased in April 2020 according to the U.S. Bureau of Economic Analysis…*www.bea.gov



CO_1998_2017_GDP .XLSX

File Edit View Insert Format Data Tools Help Last edit was 2 hours ago

Share

(Millions of current dollars)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	SAGDP2N Gross domestic product (GDP) by state 1/																							
2	(Millions of current dollars)																							
3	Bureau of Economic Analysis																							
4	State or DC																							
5																								
6	GeoFips	GeoNam	LineCode	Descriptio	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
7	08000	Colorado	1	All industr	150381.7	164226.8	180605.5	187571.9	190962.8	196483.7	203980.2	219907.9	231205.9	246043.5	255566.8	248593.6	255140.5	264431.6	273519.5	288305.2	306571.1	318554.9	329368.3	350004
8	08000	Colorado	2	Private ir	130749.9	143447.4	158857.4	164142.4	165783	170302	176943.3	191546.6	202228.8	216366.7	223730.8	215725.5	221241.2	230257.6	238837.8	252661.5	269335.7	278995.5	288598.7	307657
9	08000	Colorado	3	Agricult	1532.6	1525.6	1484.7	1838.6	1512.7	1565.8	2043.3	2079	1938.8	2447.5	2182.3	2041.4	2288.2	2801.2	2559.7	2662.5	2718.9	2759.4	2569	23(
10	08000	Colorado	4	Farms	1400.2	1379.9	1331.9	1673.9	1350.8	1395.3	1867.2	1891	1714.4	2222.7	1957.7	1783	2010.9	2516.1	2244.8	2335.1	2381.2	2409.5	2227.4	1942
11	08000	Colorado	5	Forest	132.4	145.7	152.8	164.7	161.9	170.5	176.1	188	224.4	224.8	224.6	258.3	277.3	285	314.9	327.4	337.6	349.8	341.5	358
12	08000	Colorado	6	Mining,	1809.3	1792.3	2423.6	2747.4	2656.3	4311	5704.9	9516.4	10959.3	11277.4	15819.6	10666.4	11662.2	13309.9	12347.3	13641.7	16591.8	10504.9	8465.5	11715
13	08000	Colorado	7	Oil and	777.3	783	1372.3	1523.7	1461.2	2950.5	4119.4	6785.4	7503.3	6909.8	10704.5	6697.5	7589.4	8620.6	7655.5	9222.5	11447.6	6774.1	6122.7	8420
14	08000	Colorado	8	Mining	816.1	836.7	809.8	836.9	892.4	990.6	1033.5	1801.1	1808.5	2091.5	2100.3	2294.5	2173.8	2148.4	1868.4	1722.8	1714.6	1382.5	1033.5	1364
15	08000	Colorado	9	Suppo	215.9	172.7	241.5	386.8	302.7	369.8	552.1	929.9	1647.5	2276.1	3014.7	1674.3	1899	2540.8	2823.4	2696.4	3429.7	2348.4	1309.3	1930
16	08000	Colorado	10	Utilities	1968.3	2177.5	2161.8	2296.5	2243.7	2480.5	2799.4	2876.8	3444.3	3084.5	3320.3	3297.9	3881.6	3978.7	3765.2	3717.3	4025.1	4050.8	3960.2	3954
17	08000	Colorado	11	Constru	8550.7	9684.7	11297.5	12032.9	11853	11559.3	12049.8	13191.8	13855.2	14268.9	13589.7	11463.6	9330.8	9476.7	10082.8	12369.7	14728.2	16446.4	18064.9	19584
18	08000	Colorado	12	Manufa	14524.1	14534.6	15865.6	14500.7	14876.6	14528.3	15776.2	16888	18294.2	19299.7	19285.1	19857.3	20170.9	20545.7	20770.9	21242.2	22293.7	23301.8	22892.6	24185
19	08000	Colorado	13	Durabl	9748.4	9521	10715.5	9098.8	9936.9	9519.2	10604.2	10957.8	12069	12624.2	11680.2	10975.3	12393.6	12750.1	13192.3	13860.9	13801.5	14360.2	14346	15212
20	08000	Colorado	14	Wooc	199.9	219.7	254.4	240.1	206.9	140.4	256.1	300.9	267.4	263.2	250.6	214.3	216.8	171.6	177.1	210.4	230.1	250.2	261.7	292
21	08000	Colorado	15	Nonm	675.2	769	786.1	824.7	702.1	768	837.5	840.3	910.8	854	779.1	689	699.4	688.8	771.8	969.5	1004.9	1258	1270.3	13(
22	08000	Colorado	16	Prima	182.5	116.9	118.7	167	137.1	248.8	541.7	248.4	316.1	317.8	428.8	293	292.1	320.8	416.3	318.2	343.8	315	242.1	284
23	08000	Colorado	17	Fabri	1228.5	1239.9	1278.5	1070.2	1139.4	1114.2	1068.6	1144.5	1151.4	1360.5	1282.4	1046.8	1087.9	1280.2	1337.7	1502.9	1506.4	1481.3	1510.2	1544
24	08000	Colorado	18	Mach	890	708.2	709.4	706.9	747.5	650.1	683.9	763.1	858.8	1029.9	1244.7	1152.3	1262.6	1445.9	1175.9	1316.2	1194	1325.6	1330.5	1328
25	08000	Colorado	19	Comp	4481.4	4248.8	5119.5	3682.9	4461.8	4095.2	4467.5	4355.4	4806	5091.1	4274.6	4251	5223.5	5284.7	5835.7	5968.7	5653.6	5486	4956.8	5340
26	08000	Colorado	20	Electr	76.6	71	115.9	161.2	190.9	167.9	196.5	235.9	346.4	237.4	252.6	223.2	237.2	282.1	371.8	433.1	389	454.6	469.6	493

"A screenshot of the Colorado GDP dataset from the BEA"

The seventh dataset I mined was from the Bureau of Economic Analysis website gathering data for GDP data for the state of Colorado from 1980 to 2017. The BEA provided accurate and objective information on the nation's economy for public viewing ("U.S. Bureau of Economic Analysis (BEA)," 2020).

2.8 Annual kWh per Random Address Data

PVWatts

*Estimates the energy production and cost of energy of grid-connected photovoltaic (PV) energy systems throughout the…*pvwatts.nrel.gov

The last dataset I gathered was the annual estimated kWh of the random addresses I scraped from Redfin.com. I got this data from https://pvwatts.nrel.gov/. The website is a solar energy calculator provided by the National Renewable Energy Laboratory that takes an address and the inputs of the desired solar panel size, tilt, azimuth, module type, array type, and system loss. I left the tilt, azimuth, module type, array type, and system loss default when using Selenium to input the randomly generated addresses. Then I changed the size of the kW panel to the respective recommended size that Project Sunroof from Google recommended for each address for total coverage for the default electricity bill of $80 a month. This will give me an estimate on how much kWh can be generated from the selected panel size.

2.9 Average Household Savings Based on Income Data

Average U.S. Savings Account Balance 2019: A Demographic Breakdown

*American households with savings accounts have a median balance of $7,000 and an average balance of $30,600, according…*www.valuepenguin.com

This was a quick dataset I mined based on the fact that I came up with a quick idea on how to compute numbers within the sample size of scraped random address solar data with the main integrated dataset that will be used for machine learning and prediction. The data is already defined and cleaned up by the author Chris Moon and the sources of the data come from uscensus.gov, fdic.gov, federalreserve.gov, and fred.org.

Household Income	2016 Average Savings	2013 Average Savings	3-Year Change
0 - 25,000	6,021	5,079	19%
25,000 - 44,999	11,719	9,565	23%
45,000 - 69,999	13,179	8,932	48%
70,000 - 114,999	15,333	17,305	-11%
115,000 - 159,999	37,645	20,925	80%
160,000+	117,771	82,917	42%

"A screenshot of the Average Savings Based on Household Income dataset"

The data is on the household incomes defined with their respective ranges and the average figures for savings in 2013, 2016, and the calculated 3–year percentage change.

2.10 Annual Installed Capacity MW Dc Data

Power and Renewables: Transforming the power markets with renewable energy technologies

*With an unparalleled level of depth, our integrated power, solar, wind, storage and grid edge market intelligence…*www.woodmac.com

I received this dataset from a contact I was connected with through one of my professors at Regis University, Professor Ksenia Polson. The contact (who wishes to remain anonymous) receives data from the company Wood Mackenzie Power & Renewables for their organization and was kind enough to give me a sample of the annual installed capacity of MW Dc for solar installations in Colorado. The data is collected from the years 2010 to 2019 with the following estimated years afterward from 2020 to 2025.

2.11 NREL's Assumed Levelized Cost of Energy Data

National Renewable Energy Laboratory (NREL) Home Page
*The only federal laboratory dedicated to research, development, commercialization, and deployment of renewable energy…*www.nrel.gov

The last dataset I scraped was mentioned to me by my point of contact from the annual installed capacity MW Dc dataset. The data comes from National Renewable Energy Laboratory and provides data on the assumed Levelized cost of energy per MW.

3. Data Cleansing

"A Lego man cleaning up" Image by Michael Schwarzenberger from Pixabay

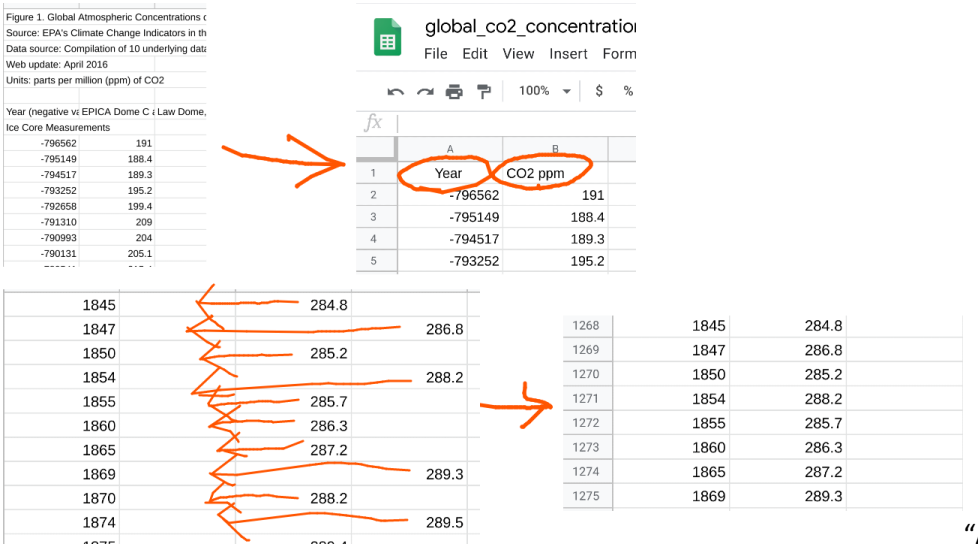
Part of the Data Science life cycle is data hygiene. Cleaning the data comes with a lot of different methods from removing outliers, to scaling everything to the same measurement, removing duplicates, removing irrelevant data, etc. This is a necessary process before modeling, analyzing, and visualizing the data for precise results.

3.1 Global CO2 Concentrations

The first dataset that I will be cleansing is the Global CO2 concentrations over the last 800,000 years. The data collection on ice core measurements is for the most part pretty clean but when getting closer to the year mark 1010 the first column of data "EPICA Dome C and Vostok Station, Antarctica" cuts off where the ice core measurements pick back up in the "Law Dome, Antarctica (75-year smoothed)",

"Siple Station, Antarctica", and "Mauna Loa, Hawaii" columns to the right of it. Plus, the ice core measurements cease in the year 1955 and now are direct measurements of CO2 levels.

Now I don't necessarily want to get rid of all the data because I still want to have something to reference back to just in case a question arises. So, I will just make a separate sheet for the new data.

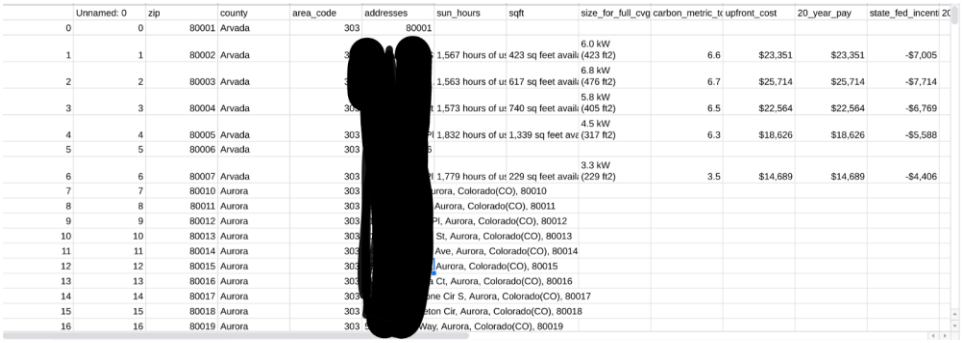


"A diagram of how I cleaned the data for Global CO2 concentrations"

After creating a separate sheet, I took all of the data on ice core measurements from the "EPICA Dome C and Vostok Station, Antarctica" column, the "Year (negative values = BC)" column, and appended them side by side on the new sheet. I then took the rest of the values from the other columns and I merged them together to have a relevant dataset for measurements.

3.2 Solar Energy Data (Random Addresses)

I am not cleaning data in any particular order at this point but by only opening the datasets I have and noticing issues that need to be cleaned and so the next set that needs cleaning would be the random addresses with solar data.



"Screenshot of the solar energy data for random addresses"

Right away I notice that there are some null values for some of the addresses and that is the first thing that needs to be cleaned for this dataset. What I did to clean up this data was I used Python. After reading in the data and precoding a script that would remove all the empty values the next irrelevant

data would be the zip codes that have values. The representation in this dataset is actual home addresses and just zip codes is incomplete data. So in order to remove those I coded in a python script that would change the former dtype(int64) “addresses” column to dtype(str), and then had the program remove values that had a shorter than the length of five characters (the length of a zip code).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	nnamed: 0	zip	county	area_code	addresses	sun_hours	sqft	e_for_full_	on_metric	pfront_cos	0_year_pa	fed_incer	r_cost_wit	cost_witho	0_year_save
2	1	80002	Arvada	303	Amn	1,567 hou	423 sq fee	6.0 kW (423 ft2)	6.6	23351	23351	-7005	20780	23250	2470
3	2	80003	Arvada	303	Gray	1,563 hou	617 sq fee	6.8 kW (476 ft2)	6.7	25714	25714	-7714	22215	23250	1035
4	3	80004	Arvada	303	Glac	1,573 hou	740 sq fee	5.8 kW (405 ft2)	6.5	22564	22564	-6769	20490	23250	2760
5	4	80005	Arvada	303	W	1,832 hou	1,339 sq f	4.5 kW (317 ft2)	6.3	18626	18626	-5588	18171	23250	5079
6	6	80007	Arvada	303	W	1,779 hou	229 sq fee	3.3 kW (229 ft2)	3.5	14689	14689	-4406	23472	23250	-222
7	50	80111	Centennia	303	Mi	1,795 hou	1,938 sq f	5.5 kW (388 ft2)	7.5	21776	21776	-7323	18471	23707	5236
8	51	80112	Centennia	303	Kr	1,796 hou	1,022 sq f	6.3 kW (440 ft2)	7.8	24139	24139	-8058	19474	23707	4233
9	52	80113	Cherry Hil	303	W	1,867 hou	440 sq fee	6.3 kW (440 ft2)	6.9	24139	24139	-7970	21616	23707	2091
10	56	80120	Littleton	303	W Li	1,737 hou	916 sq fee	5.0 kW (352 ft2)	6.2	20201	20201	-6060	17659	23432	5773
11	59	80123	Bow Mar	303	S Be	1,821 hou	1,374 sq f	4.3 kW (300 ft2)	5.9	17839	17839	-5351	16839	23432	6593
12	63	80127	Denver	303	S All	1,707 hou	3,436 sq f	4.5 kW (317 ft2)	5.9	18626	18626	-5588	17413	23432	6019
13								5.3 kW							

“A cleaned version of random addresses and their solar energy estimated data”

I also got rid of all the unnecessary strings with the integers in columns such as “sun_hours,” “sqft,” “size_for_full_cvg_kW,” etc.

3.3 County Demographics Solar Energy Data

Following the random address solar energy dataset, I figured it would be easy to clean the demographic solar energy data for all County in the state of Colorado. What I did was normalize the letter format for representing thousands and millions with floats and converted it with python to an integer of the actual value. All of the counties that don’t have data I have placed 0s in and plan on removing this from the dataset before training, but I am keeping them there for reference.

county_demographic_solar_data_est

File Edit View Insert Format Data Tools Add-ons Help Last edit was yesterday at 1:26 PM

Share

100% $ % .0 .00 123 Arial 10

fx 0

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		CFIPS	YEAR	COUNTY	Households	total_roofs_insta	total_est_roofs	total_est_sqft_of	total_est_capacit	total_est_MWh_	total_est_mediar	total_est_mediar	total_est_mediar	to
2	0	1	2017	Adams County	168930	2200	117000	144000000	2000	2900000	5640	80	11500	
3	1	3	2017	Alamosa County	6203	0	0	0	0	0	0	0	0	
4	2	5	2017	Arapahoe County	250210	3400	146000	168000000	2400	3400000	5990	85	12400	
5	3	7	2017	Archuleta County	5814	0	0	0	0	0	0	0	0	
6	4	9	2017	Baca County	1521	0	0	0	0	0	0	0	0	
7	5	11	2017	Bent County	1666	0	0	0	0	0	0	0	0	
8	6	13	2017	Boulder County	130376	3800	44600	52900000	750	996000	5290	75	10200	
9	7	14	2017	Broomfield Coun	26120	685	18200	23200000	330	452000	7220	103	14200	
10	8	15	2017	Chaffee County	8424	0	0	0	0	0	0	0	0	
11	9	17	2017	Cheyenne Count	795	0	0	0	0	0	0	0	0	
12	10	19	2017	Clear Creek Cou	4445	0	0	0	0	0	0	0	0	
13	11	21	2017	Conejos County	3065	0	0	0	0	0	0	0	0	
14	12	23	2017	Costilla County	1658	0	0	0	0	0	0	0	0	
15	13	25	2017	Crowley County	1227	0	0	0	0	0	0	0	0	
16	14	27	2017	Custer County	2269	0	0	0	0	0	0	0	0	
17	15	29	2017	Delta County	12543	0	0	0	0	0	0	0	0	
18	16	31	2017	Denver County	310900	3600	129000	171000000	2400	3400000	4400	63	9000	
19	17	33	2017	Dolores County	889	0	0	0	0	0	0	0	0	
20	18	35	2017	Douglas County	119997	84	22800	35800000	508	636000	7930	113	14100	
21	19	37	2017	Eagle County	20266	0	0	0	0	0	0	0	0	
22	20	39	2017	Elbert County	9330	0	0	0	0	0	0	0	0	

“A screenshot of the cleaned data for the demographics on solar energy for Colorado counties”

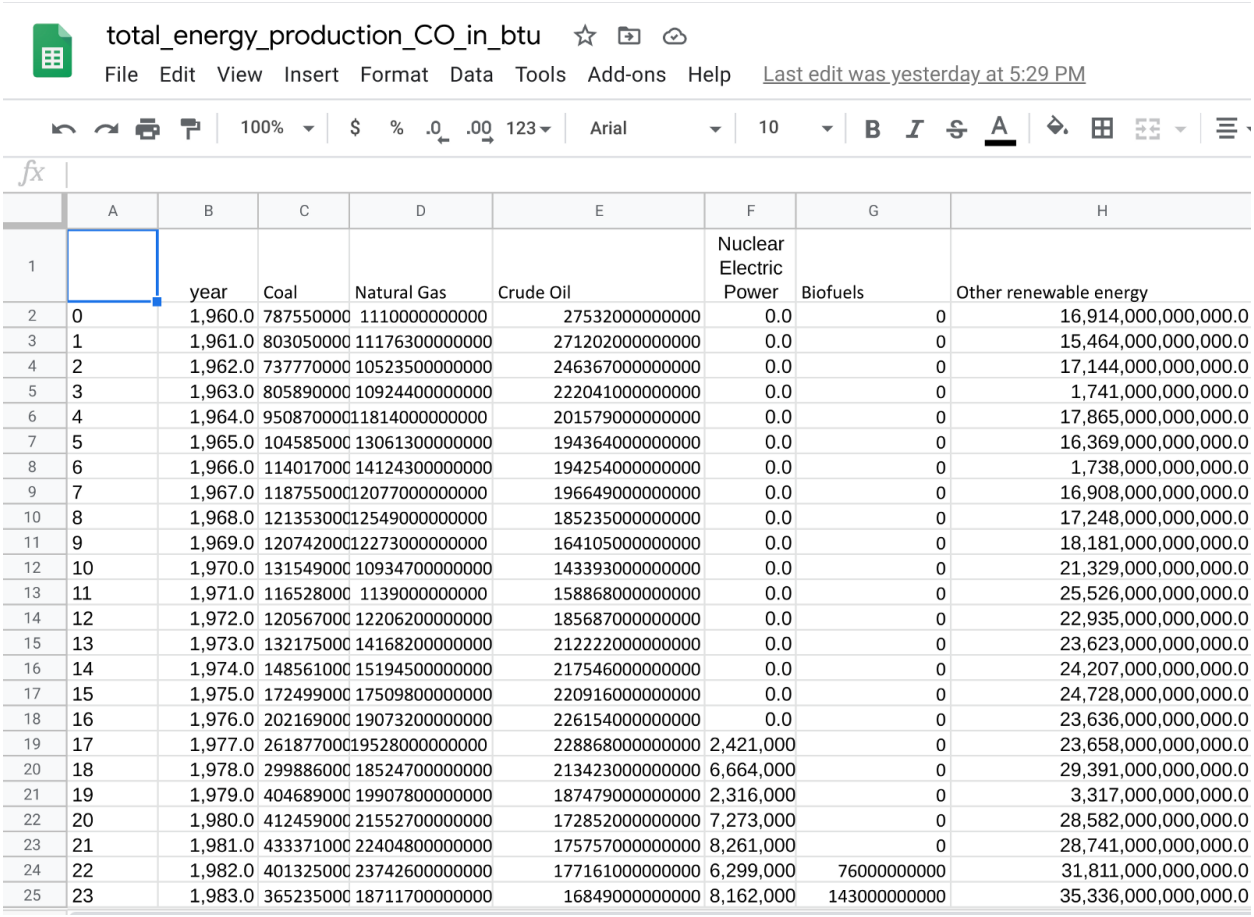
3.4 Colorado CO2 Emissions

This data has CO2 emissions in million metric tons from the year 1980 to 2017 and of multiple sectors. So, the only thing I really needed to do was remove all the other sectors besides residential and consolidate the data on another sheet.

3.5 Total Renewable Energy Consumption for the US by Sector Data (1960–2017)

This dataset didn't need much cleansing since it was already pretty clean when I first mined it. I won't know its usefulness until I start exploring it, but the data may no longer necessarily be relevant to the question I am trying to answer.

3.6 Total Energy Production for Colorado and the US by Sector in BTU

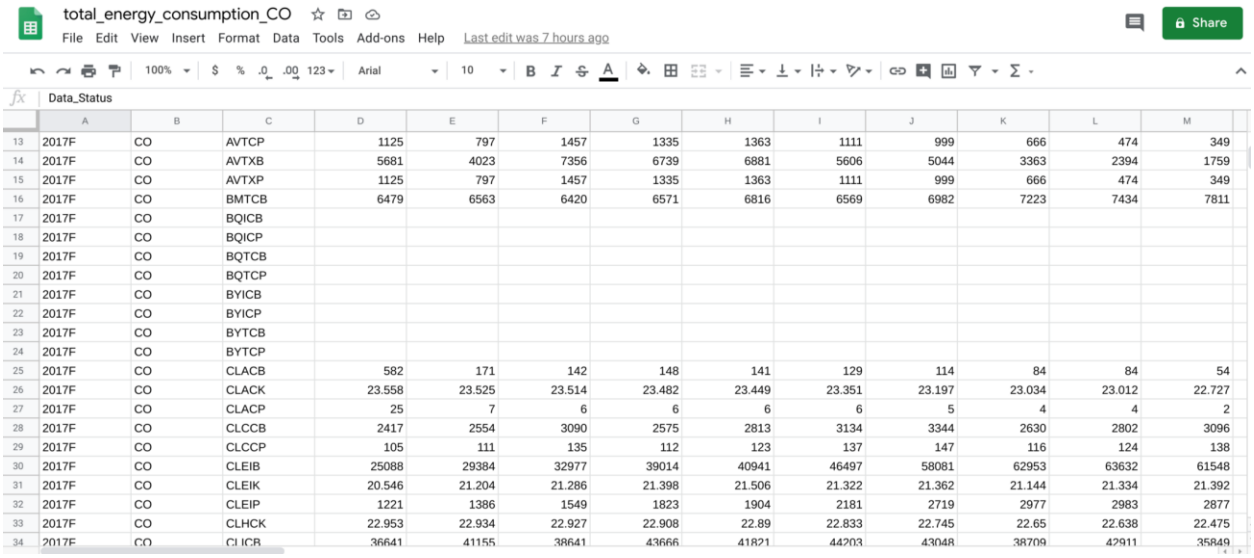


	A	B	C	D	E	F	G	H
1		year	Coal	Natural Gas	Crude Oil	Nuclear Electric Power	Biofuels	Other renewable energy
2	0	1,960.0	787550000	1110000000000	27532000000000	0.0	0	16,914,000,000,000.0
3	1	1,961.0	803050000	11176300000000	271202000000000	0.0	0	15,464,000,000,000.0
4	2	1,962.0	737770000	10523500000000	246367000000000	0.0	0	17,144,000,000,000.0
5	3	1,963.0	805890000	10924400000000	222041000000000	0.0	0	1,741,000,000,000.0
6	4	1,964.0	950870000	11814000000000	201579000000000	0.0	0	17,865,000,000,000.0
7	5	1,965.0	104585000	13061300000000	194364000000000	0.0	0	16,369,000,000,000.0
8	6	1,966.0	114017000	14124300000000	194254000000000	0.0	0	1,738,000,000,000.0
9	7	1,967.0	118755000	12077000000000	196649000000000	0.0	0	16,908,000,000,000.0
10	8	1,968.0	121353000	12549000000000	185235000000000	0.0	0	17,248,000,000,000.0
11	9	1,969.0	120742000	12273000000000	164105000000000	0.0	0	18,181,000,000,000.0
12	10	1,970.0	131549000	10934700000000	143393000000000	0.0	0	21,329,000,000,000.0
13	11	1,971.0	116528000	1139000000000	158868000000000	0.0	0	25,526,000,000,000.0
14	12	1,972.0	120567000	12206200000000	185687000000000	0.0	0	22,935,000,000,000.0
15	13	1,973.0	132175000	14168200000000	212222000000000	0.0	0	23,623,000,000,000.0
16	14	1,974.0	148561000	15194500000000	217546000000000	0.0	0	24,207,000,000,000.0
17	15	1,975.0	172499000	17509800000000	220916000000000	0.0	0	24,728,000,000,000.0
18	16	1,976.0	202169000	19073200000000	226154000000000	0.0	0	23,636,000,000,000.0
19	17	1,977.0	261877000	19528000000000	228868000000000	2,421,000	0	23,658,000,000,000.0
20	18	1,978.0	299886000	18524700000000	213423000000000	6,664,000	0	29,391,000,000,000.0
21	19	1,979.0	404689000	19907800000000	187479000000000	2,316,000	0	3,317,000,000,000.0
22	20	1,980.0	412459000	21552700000000	172852000000000	7,273,000	0	28,582,000,000,000.0
23	21	1,981.0	433371000	22404800000000	175757000000000	8,261,000	0	28,741,000,000,000.0
24	22	1,982.0	401325000	23742600000000	177161000000000	6,299,000	76000000000	31,811,000,000,000.0
25	23	1,983.0	365235000	18711700000000	16849000000000	8,162,000	143000000000	35,336,000,000,000.0

"A screenshot of the cleaned total energy production for Colorado by sector in BTU"

The only thing that was wrong with both of these datasets was the condense number format of having smaller integers representing BTU in the trillions. So, with python, I converted all the floats to integers of the value.

3.7 Total Energy Consumption for Colorado and the US by Sector

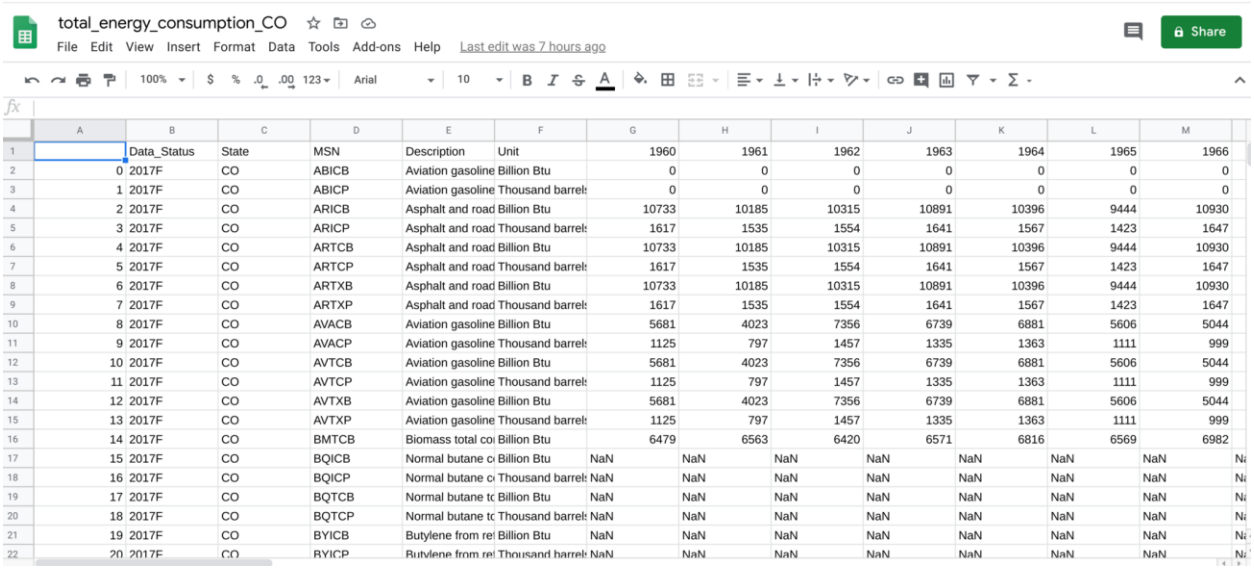


	A	B	C	D	E	F	G	H	I	J	K	L	M
13	2017F	CO	AVTCP	1125	797	1457	1335	1363	1111	999	666	474	349
14	2017F	CO	AVTXB	5681	4023	7356	6739	6881	5606	5044	3363	2394	1759
15	2017F	CO	AVTXP	1125	797	1457	1335	1363	1111	999	666	474	349
16	2017F	CO	BMTCB	6479	6563	6420	6571	6816	6569	6982	7223	7434	7811
17	2017F	CO	BQICB										
18	2017F	CO	BQICP										
19	2017F	CO	BQTCB										
20	2017F	CO	BQTCP										
21	2017F	CO	BYICB										
22	2017F	CO	BYICP										
23	2017F	CO	BYTCB										
24	2017F	CO	BYTCP										
25	2017F	CO	CLACB	582	171	142	148	141	129	114	84	84	54
26	2017F	CO	CLACK	23.558	23.525	23.514	23.482	23.449	23.351	23.197	23.034	23.012	22.727
27	2017F	CO	CLACP	25	7	6	6	6	6	5	4	4	2
28	2017F	CO	CLCCB	2417	2554	3090	2575	2813	3134	3344	2630	2802	3096
29	2017F	CO	CLCCP	105	111	135	112	123	137	147	116	124	138
30	2017F	CO	CLEIB	25088	29384	32977	39014	40941	46497	58081	62953	63632	61548
31	2017F	CO	CLEIK	20.546	21.204	21.286	21.398	21.506	21.322	21.362	21.144	21.334	21.392
32	2017F	CO	CLEIP	1221	1386	1549	1823	1904	2181	2719	2977	2983	2877
33	2017F	CO	CLHCK	22.953	22.934	22.927	22.908	22.89	22.833	22.745	22.65	22.638	22.475
34	2017F	CO	CLICB	36641	41155	38641	43666	41821	44203	43048	38709	42911	35849

"A screenshot of the total energy consumption for Colorado by sector"

This data required a little bit of innovation to clean. Since there are a bunch of random codes for different sectors on how energy was consumed, I had to create a dictionary out of the consumption code values from what the EIA provided. The consumption codes provided the description along with the units of measure as well. I created two columns inside of Google sheets and then saved to an excel format. After that, I read both datasets for Colorado and the US along with the consumption codes data into data frames within Google Collaboratory. Then I created a dictionary out of the consumption code data with a loop.

This gave two clean columns next to the consumption codes for both US and Colorado datasets that way it will make exploration easier when having the descriptions next to the codes. I also filled in all the empty cells with NaN values until I figure out what to do with the data.

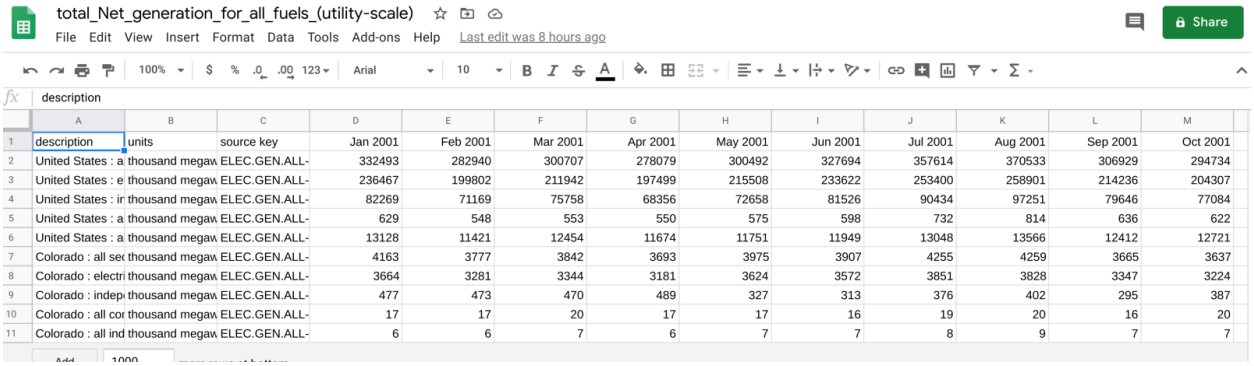


	A	Data_Status	State	MSN	Description	Unit	1960	1961	1962	1963	1964	1965	1966
2	0	2017F	CO	ABICB	Aviation gasoline	Billion Btu	0	0	0	0	0	0	0
3	1	2017F	CO	ABICP	Aviation gasoline	Thousand barrels	0	0	0	0	0	0	0
4	2	2017F	CO	ARICB	Asphalt and road	Billion Btu	10733	10185	10315	10891	10396	9444	10930
5	3	2017F	CO	ARICP	Asphalt and road	Thousand barrels	1617	1535	1554	1641	1567	1423	1647
6	4	2017F	CO	ARTCB	Asphalt and road	Billion Btu	10733	10185	10315	10891	10396	9444	10930
7	5	2017F	CO	ARTCP	Asphalt and road	Thousand barrels	1617	1535	1554	1641	1567	1423	1647
8	6	2017F	CO	ARTXB	Asphalt and road	Billion Btu	10733	10185	10315	10891	10396	9444	10930
9	7	2017F	CO	ARTXP	Asphalt and road	Thousand barrels	1617	1535	1554	1641	1567	1423	1647
10	8	2017F	CO	AVACB	Aviation gasoline	Billion Btu	5681	4023	7356	6739	6881	5606	5044
11	9	2017F	CO	AVACP	Aviation gasoline	Thousand barrels	1125	797	1457	1335	1363	1111	999
12	10	2017F	CO	AVTCB	Aviation gasoline	Billion Btu	5681	4023	7356	6739	6881	5606	5044
13	11	2017F	CO	AVTCP	Aviation gasoline	Thousand barrels	1125	797	1457	1335	1363	1111	999
14	12	2017F	CO	AVTXB	Aviation gasoline	Billion Btu	5681	4023	7356	6739	6881	5606	5044
15	13	2017F	CO	AVTXP	Aviation gasoline	Thousand barrels	1125	797	1457	1335	1363	1111	999
16	14	2017F	CO	BMTCB	Biomass total co	Billion Btu	6479	6563	6420	6571	6816	6569	6982
17	15	2017F	CO	BQICB	Normal butane c	Billion Btu	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18	16	2017F	CO	BQICP	Normal butane c	Thousand barrels	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19	17	2017F	CO	BQTCB	Normal butane tc	Billion Btu	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20	18	2017F	CO	BQTCP	Normal butane tc	Thousand barrels	NaN	NaN	NaN	NaN	NaN	NaN	NaN
21	19	2017F	CO	BYICB	Butylene from re	Billion Btu	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22	20	2017F	CO	BYICP	Butylene from re	Thousand barrels	NaN	NaN	NaN	NaN	NaN	NaN	NaN

"A screenshot of the cleaned total energy consumption data for Colorado by sector"

After I was able to designate a clean sheet of data for the Total Energy Consumption for Colorado, I transposed the data on an entirely new sheet and got rid of the columns that weren't related to the residential sector.

3.8 Total Energy Net Generation for all Fuels



total_Net_generation_for_all_fuels_(utility-scale)

File Edit View Insert Format Data Tools Add-ons Help Last edit was 8 hours ago

Share

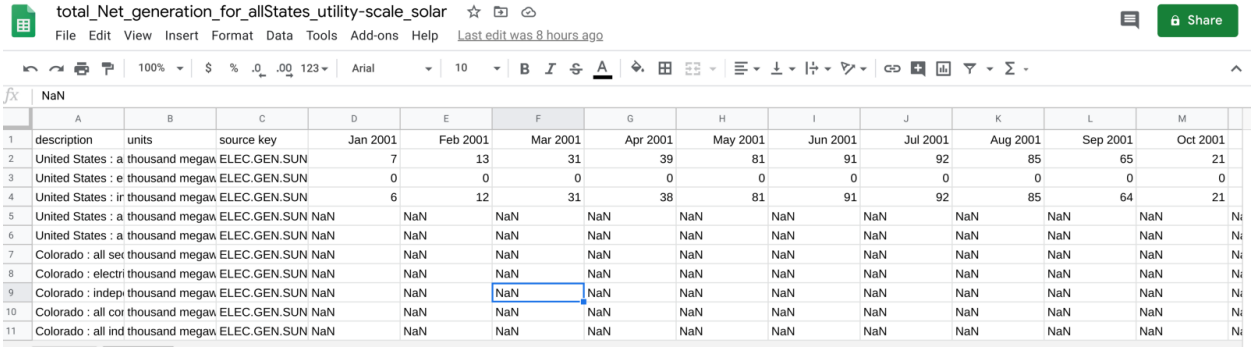
description

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	description	units	source key	Jan 2001	Feb 2001	Mar 2001	Apr 2001	May 2001	Jun 2001	Jul 2001	Aug 2001	Sep 2001	Oct 2001
2	United States : a	thousand megaw	ELEC.GEN.ALL-	332493	282940	300707	278079	300492	327694	357614	370533	306929	294734
3	United States : e	thousand megaw	ELEC.GEN.ALL-	236467	199802	211942	197499	215508	233622	253400	258901	214236	204307
4	United States : ir	thousand megaw	ELEC.GEN.ALL-	82269	71169	75758	68356	72658	81526	90434	97251	79646	77084
5	United States : a	thousand megaw	ELEC.GEN.ALL-	629	548	553	550	575	598	732	814	636	622
6	United States : a	thousand megaw	ELEC.GEN.ALL-	13128	11421	12454	11674	11751	11949	13048	13566	12412	12721
7	Colorado : all se	thousand megaw	ELEC.GEN.ALL-	4163	3777	3842	3693	3975	3907	4255	4259	3665	3637
8	Colorado : electri	thousand megaw	ELEC.GEN.ALL-	3664	3281	3344	3181	3624	3572	3851	3828	3347	3224
9	Colorado : indep	thousand megaw	ELEC.GEN.ALL-	477	473	470	489	327	313	376	402	295	387
10	Colorado : all cor	thousand megaw	ELEC.GEN.ALL-	17	17	20	17	17	16	19	20	16	20
11	Colorado : all ind	thousand megaw	ELEC.GEN.ALL-	6	6	7	6	7	7	8	9	7	7

"A screenshot of the clean data for total energy net generation for all fuels"

This dataset was mostly clean when I mined it from the EIA website. All I had to do was create a separate sheet without all of the other 49 states in the US and store the data for the US and Colorado on one sheet.

3.9 Total Energy Net Generation for Utility Solar Scales Data



total_Net_generation_for_allStates_utility-scale_solar

File Edit View Insert Format Data Tools Add-ons Help Last edit was 8 hours ago

Share

NaN

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	description	units	source key	Jan 2001	Feb 2001	Mar 2001	Apr 2001	May 2001	Jun 2001	Jul 2001	Aug 2001	Sep 2001	Oct 2001
2	United States : a	thousand megaw	ELEC.GEN.SUN	7	13	31	39	81	91	92	85	65	21
3	United States : e	thousand megaw	ELEC.GEN.SUN	0	0	0	0	0	0	0	0	0	0
4	United States : ir	thousand megaw	ELEC.GEN.SUN	6	12	31	38	81	91	92	85	64	21
5	United States : a	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	United States : a	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Colorado : all se	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	Colorado : electri	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	Colorado : indep	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	Colorado : all cor	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	Colorado : all ind	thousand megaw	ELEC.GEN.SUN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

"A screenshot of the clean data for total energy net generation for utility solar scales"

This data has the same format as the previous dataset so cleaning it involved the same process only this time there were some missing values. I replaced the missing values with NaN and I figure I will know what to do with these once I explore them. However, it looks pretty obvious with most of the datasets revolving around 2017 and I will not be using the NaN values since the years in which these NaNs fall under aren't 2017.

3.10 Total Households in Colorado Counties Data

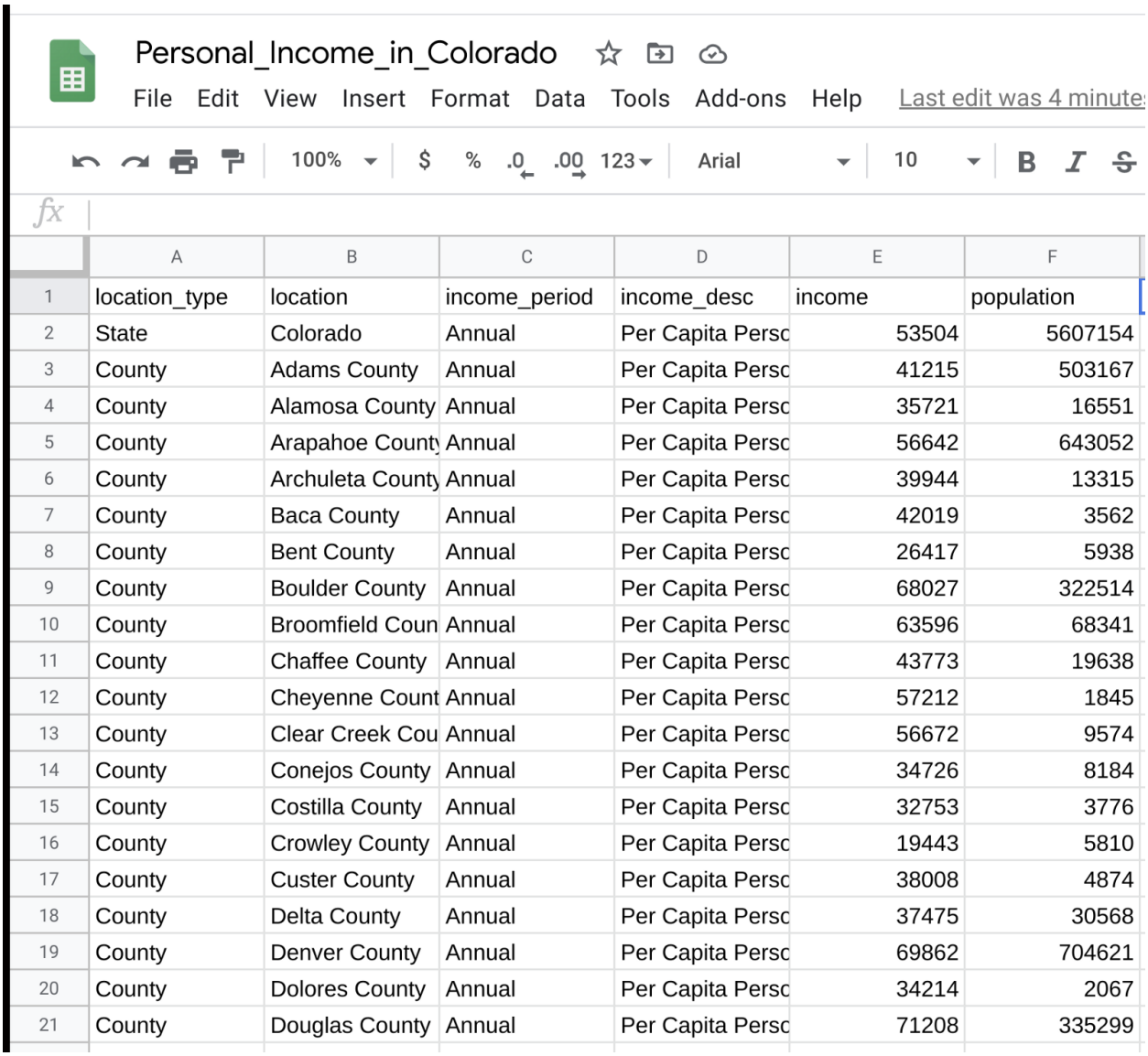
This data didn't really need any cleaning other than the fact that I needed to move the column containing the total households nine spaces over to the left so it will not create an issue when opening it in python.

3.11 Global Greenhouse Gases Data

The only cleaning that was necessary for this dataset was the removal of the titles and descriptions at the top of the CO2 sheet. Removing these in Google sheets is a lot easier for reading the data later with Python.

3.12 Per Capita Personal Income for Each County in Colorado Data

The last dataset I had to clean was the per capital of personal income for the state of Colorado by county data.



Personal_Income_in_Colorado

File Edit View Insert Format Data Tools Add-ons Help Last edit was 4 minute

	A	B	C	D	E	F
1	location_type	location	income_period	income_desc	income	population
2	State	Colorado	Annual	Per Capita Persc	53504	5607154
3	County	Adams County	Annual	Per Capita Persc	41215	503167
4	County	Alamosa County	Annual	Per Capita Persc	35721	16551
5	County	Arapahoe County	Annual	Per Capita Persc	56642	643052
6	County	Archuleta County	Annual	Per Capita Persc	39944	13315
7	County	Baca County	Annual	Per Capita Persc	42019	3562
8	County	Bent County	Annual	Per Capita Persc	26417	5938
9	County	Boulder County	Annual	Per Capita Persc	68027	322514
10	County	Broomfield Coun	Annual	Per Capita Persc	63596	68341
11	County	Chaffee County	Annual	Per Capita Persc	43773	19638
12	County	Cheyenne Count	Annual	Per Capita Persc	57212	1845
13	County	Clear Creek Cou	Annual	Per Capita Persc	56672	9574
14	County	Conejos County	Annual	Per Capita Persc	34726	8184
15	County	Costilla County	Annual	Per Capita Persc	32753	3776
16	County	Crowley County	Annual	Per Capita Persc	19443	5810
17	County	Custer County	Annual	Per Capita Persc	38008	4874
18	County	Delta County	Annual	Per Capita Persc	37475	30568
19	County	Denver County	Annual	Per Capita Persc	69862	704621
20	County	Dolores County	Annual	Per Capita Persc	34214	2067
21	County	Douglas County	Annual	Per Capita Persc	71208	335299

“A screenshot of the cleaned per capita personal income Colorado dataset”

This dataset was already clean for the most part, and all I had to do was create a separate sheet labeled “cleaned” made up of all the columns I really wanted to focus on which were the location type, the county, the period for the income, the description of the income, the income, and the population. So, it

was easier to understand relabeled the column names to “location_type,” “location,” “income_period,” “income_desc,” “income,” and “population.”

3.13 Colorado Annual GDP in Current Dollars Data

The dataset on Colorado annual GDP in current dollars was mostly already cleaned. The only thing I really needed to do with the data was to integrate the two datasets on different time periods 1980–1997 and 1998–2017 together and remove all the clutter since all I really cared about was the industry total GDP for all the years.

3.14 Data That Didn’t Need to be Cleaned

The Annual kWh per Random Address Data didn’t necessarily need to be cleaned since it was a clean scrape and appended right onto the random solar address data. The NREL data for LCOEs was a clean copy and paste scrape and so the data from there was appended onto the main dataset. The annual installed MW Dc capacity for Colorado was also already clean when delivered to me. The average household savings based on income was also a clean scrape and didn’t require any data hygiene.

4. Exploring the Data and Feature Selection

Image by Colin Behrens from Pixabay

Exploring the data is crucial as it allows me the opportunity to discover certain values and aspects of the data that gives. Even though I initially have looked at the data before during mining and cleaning, it is more of a first-hand investigation of the value that the data may hold.

4.1 Global CO2 Concentrations

The first dataset I explored I thought would be the easiest of all the sets since there are only two columns, with only 1200+ rows. I loaded the data with Pandas and used df.head() to get the first 5 rows of data.

	Year	CO2 ppm
0	-796562	191.0
1	-795149	188.4
2	-794517	189.3
3	-793252	195.2
4	-792658	199.4


“A screenshot of the Global CO2 Concentrations dataset head”

The two columns of data are “Year” and “CO2 ppm (parts per million)” Next I used df.shape to give me a data shape of (1359, 2) Meaning there are 1359 recorded Ice core measurements. The types of data for both columns after using df.info() are int64 for Year and float64 for CO2 ppm with matching nonnull counts of 1359.

	Year	CO2 ppm
count	1359.000000	1359.000000
mean	-313069.128035	243.926571
std	281570.433674	38.915326
min	-796562.000000	171.600000
25%	-571299.500000	214.700000
50%	-277370.000000	239.500000
75%	-5416.500000	273.700000
max	2015.000000	400.830000

“An annotated screenshot of a quick summary of statistics for the Global CO2 Concentrations dataset”

Using df.describe() I have annotated out the summary of statistics for the “Year” column since the integers are not values of any sort but only years. The mean of the CO2 ppm is 243.92 which would mean that for the duration of 800,000 years the average of Earth’s CO2 levels has remained pretty normal. The Standard deviation is 38.91 ppm, the min 800,000 years ago was 171.6 ppm, 25th percentile was 214.7 ppm, 50th percentile was 239.5 ppm, 75th percentile was 273.7 ppm, and the max is 400.83 ppm.

```
283.2    8
276.4    8
239.1    7
208.1    6
283.1    6
         ..
262.0    1
273.8    1
215.2    1
221.0    1
214.6    1
Name: CO2 ppm, Length: 799, dtype: int64
```

"A screenshot of the value counts for the Global CO2 Concentrations dataset"

Since the data is already a time-series it would be pointless to try and see if there was anything categorical about the data. So I ignored using df.['CO2 ppm'].unique() and used to df['CO2 ppm'].value_counts() to discover that there are multiple values within the data leaving the total length of unique values at 799.

The next step after exploring this dataset would be to explore the other datasets that might be worth integrating with this dataset in order to show a correlation to the given rise of CO2 levels.

4.2 Colorado CO2 Emissions

The next dataset I started to explore was the Colorado CO2 emissions data. Calling df.head() revealed that the dataset has five columns "Years," "Coals," "Petroleum Products," "Natural Gas," and "Total"

	Years	Coal	Petroleum Products	Natural Gas	Total
0	1980	0.043	0.442	4.734	5.219
1	1981	0.035	0.541	3.961	4.537
2	1982	0.044	0.635	4.498	5.177
3	1983	0.026	0.813	4.494	5.333
4	1984	0.049	0.385	4.969	5.403

“A Screenshot of the head of the data for the Colorado CO2 Emissions dataset”

The year 2017 can’t be seen but is of immense importance since most of the data I have for houses and energy revolve around that year. The shape of the data is (39,5), so an exceedingly small dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38 entries, 0 to 37
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Years               38 non-null     int64
 1   Coal                38 non-null     float64
 2   Petroleum Products  38 non-null     float64
 3   Natural Gas         38 non-null     float64
 4   Total               38 non-null     float64
dtypes: float64(4), int64(1)
memory usage: 1.6 KB
```

“A screenshot of the info on the data for the Colorado CO2 Emissions dataset”

Every column within the data besides the first is a float64 Dtype since the data is in millions of metric tons. I will have to figure a way to convert this data when integrating it with the global co2 concentrations and solar housing data.

	Years	Coal	Petroleum Products	Natural Gas	Total
count	38.000000	38.000000	38.000000	38.000000	38.000000
mean	1998.500000	0.022947	0.581921	5.923842	6.528816
std	11.113055	0.022820	0.202009	1.006623	1.127644
min	1980.000000	0.000000	0.059000	3.961000	4.537000
25%	1989.250000	0.000750	0.430750	4.963750	5.381250
50%	1998.500000	0.019500	0.642500	6.068500	6.478000
75%	2007.750000	0.041750	0.741750	6.825000	7.584000
max	2017.000000	0.077000	0.926000	7.402000	8.224000

"A screenshot of the basic statistics for the Colorado CO2 Emissions dataset"

I don't think that df.describe() will show anything useful since the main focus of mine is the year 2017, but I do believe there holds some value in this. The mean of million metric tons for all the years is 6.52, for standard deviation is 1.12, for min is 4.53, for 25th percentile is 5.38, for 50th percentile is 6.47, for 75th percentile is 7.58, and for the max is 8.22.

Since these values are all floats I think that there won't be any unique values since all of the floats end in the millionths place, but I believe I can integrate this data with the global CO2 concentrations data and the solar housing data/solar data by counties.

Total Energy Consumption for Colorado

This data I can tell already will be pretty beneficial when it comes to integration for all of the datasets.

	Years	ESRCB	ESRCP	GERCB	HLRCB	HLRCP	KSRCB	KSRCP	LORCB	NGRCB	NGRCP	PARCB	PARCP	PQRCB	PQRCP	SFRCB	SOR7P	SORCB	TERCB	TERPB	TNRCB	WDRCB
0	1960	6061	1776	0	8034	2092	282	50	14988	54125	52295	9176	2289	NaN	NaN	NaN	NaN	0	92060	52.0	77072	4231
1	1961	6529	1913	0	7969	2075	305	54	15866	57691	55740	9166	2282	NaN	NaN	NaN	NaN	0	97088	105.7	81222	4162
2	1962	7007	2054	0	7986	2079	224	40	16832	59175	57174	8908	2238	NaN	NaN	NaN	NaN	0	100221	105.8	83389	3853
3	1963	7601	2228	0	7943	2068	426	75	18170	50330	55430	8999	2251	NaN	NaN	NaN	NaN	0	92335	95.7	74166	3681
4	1964	8300	2433	0	10130	2637	673	119	19723	59225	65952	11200	2824	NaN	NaN	NaN	NaN	0	106220	107.9	86496	3887

"A screenshot of the head of the data for the Total Energy Consumption for Colorado dataset"

A df.head() shows that the dataset itself contains lots of values by year with different codes (that are referenced to on another sheet) to represent how different energies were consumed within the residential sector. Using df.shape() shows that the dataset is (58,22) which is small but still holds valuable data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58 entries, 0 to 57
Data columns (total 22 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Years   58 non-null     int64  
 1   ESRCB   58 non-null     int64  
 2   ESRCP   58 non-null     int64  
 3   GERCB   58 non-null     int64  
 4   HLRCB   58 non-null     int64  
 5   HLRCP   58 non-null     int64  
 6   KSRCB   58 non-null     int64  
 7   KSRCP   58 non-null     int64  
 8   LORCB   58 non-null     int64  
 9   NGRCB   58 non-null     int64  
 10  NGRCP   58 non-null     int64  
 11  PARCB   58 non-null     int64  
 12  PARCP   58 non-null     int64  
 13  PQRCB   8 non-null      float64
 14  PQRCP   8 non-null      float64
 15  SFRCB   38 non-null     float64
 16  SOR7P   29 non-null     float64
 17  SORCB   58 non-null     int64  
 18  TERCB   58 non-null     int64  
 19  TERPB   58 non-null     float64
 20  TNRCB   58 non-null     int64  
 21  WDRCB   58 non-null     int64  
dtypes: float64(5), int64(17)
memory usage: 10.1 KB
```

“A screenshot of

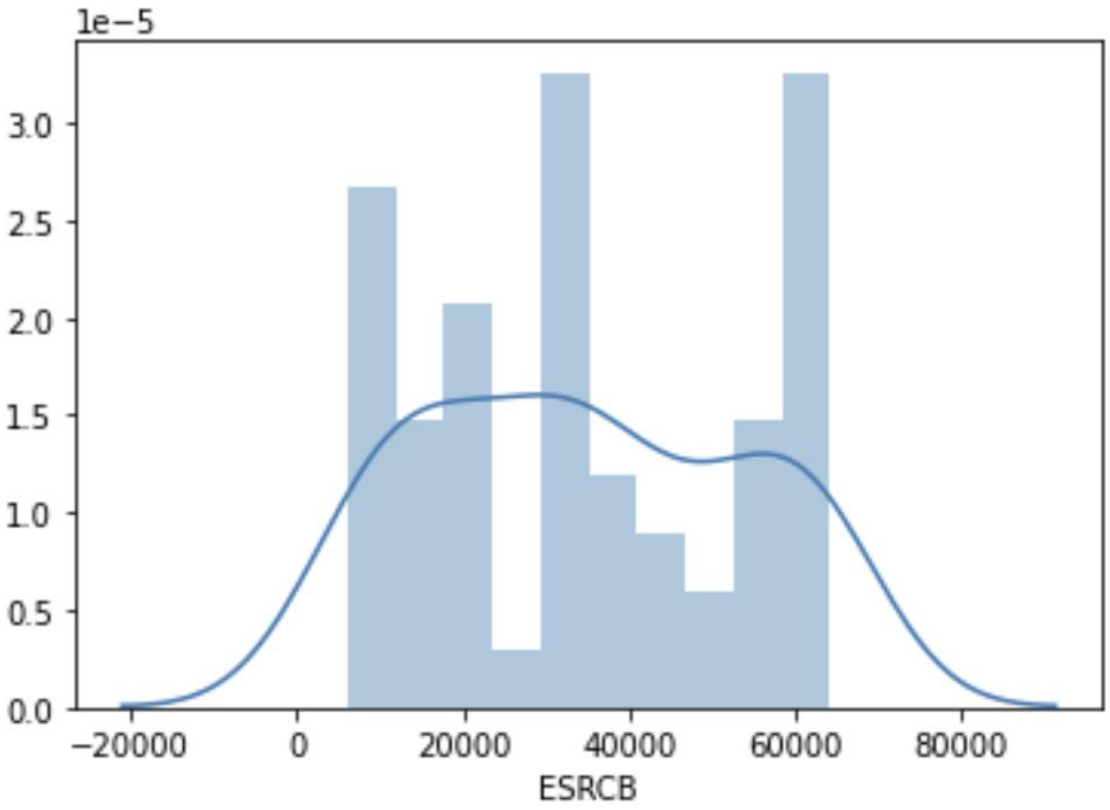
the data info for the Total Energy Consumption of Colorado dataset”

The columns revealed that none of them have non-null values but as far as I saw when looking at the head of the data there were quite a few NaNs. Most of the columns are of Dtype int64 format, but some are float64 as well.

	Years	ESRCB	ESRCP	GERCB	HLRCB	HLRCP	KSRCB	KSRCP	LORCB	NGRCB	NGRCP	PARCB
count	[redacted]	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000	58.000000
mean	[redacted]	34521.137931	10117.620690	76.706897	9483.310345	2468.965517	316.896552	55.862069	77949.327586	99311.793103	99653.206897	10342.051724
std	[redacted]	19180.470954	5621.470107	96.170537	3173.984618	826.275961	400.269720	70.533916	40558.785138	24506.639652	22239.298187	3391.171316
min	[redacted]	6061.000000	1776.000000	0.000000	656.000000	171.000000	3.000000	1.000000	14988.000000	50330.000000	52295.000000	901.000000
25%	[redacted]	17875.500000	5239.250000	0.000000	7386.000000	1922.750000	102.750000	18.250000	42510.750000	83170.750000	84732.000000	7620.500000
50%	[redacted]	32663.000000	9573.000000	33.000000	10271.000000	2674.000000	202.000000	35.500000	75957.500000	95047.500000	98102.500000	10847.500000
75%	[redacted]	52904.500000	15505.250000	121.750000	11959.750000	3113.250000	324.500000	57.000000	118930.250000	122494.500000	119098.000000	12808.750000
max	[redacted]	64261.000000	18834.000000	271.000000	14552.000000	3789.000000	1679.000000	296.000000	136659.000000	139516.000000	134936.000000	15999.000000

“A screenshot of basic statistics for the Total Energy Consumption for Colorado dataset”

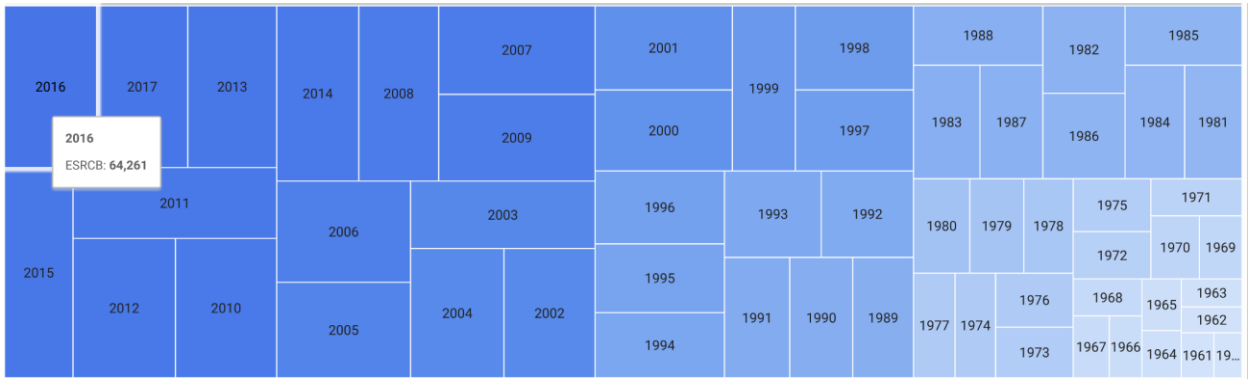
A quick screenshot of df.describe() shows the basic statistical values for each of the areas of consumption through the years. For example, ESRCB which is “Electricity consumed by (i.e., sold to) the residential sector” has a mean of 34521.14 in which the unit metric for this is a billion BTU.



“A screenshot of a distribution plot on ESRCB for the Total Energy of Consumption for Colorado dataset”

Using sns.distplot(df[‘ESCRB’]) A distribution plot here shows that there are more values of energy consumed in billion BTU from 5 billion to 35 billion and then the distribution picks back up from 50

billion to 60 billion. The distribution line shows this also where the first hump is the 5 billion to 35 billion and the second smaller hump is 50 billion to 60 billion. The small dip where 20 billion to 30 billion is a big dip between the distribution of values.



"A screenshot of a treemap on the distribution of values over the years for the Total Energy Consumption for Colorado dataset"

Above is a great visual on where the values begin to increase over the years. The value increase is shown through the box size increase.

4.3 Solar Energy Data (Random Addresses)

The Solar Energy Data of random addresses should be easy to explore as well since it is fairly small and when I did most of the cleaning, I got a feel for what it holds. df.head() shows a number of columns with integer values.

	addresses	county	sun_hours	sqft	size_for_full_cvg_kW	carbon_metric_ton	upfront_cost	20_year_pay	state_fed_incentives	20_year_cost_with_solar	20_y
0	13715 W 51st Ave, Arvada, CO 80002	Jefferson	1715	1216	6.3	7.7	22979	22979	-6784	19535	
1	5295 Gladiola St, Arvada, CO 80002	Jefferson	1496	634	7.0	6.6	25229	25229	-6559	23117	
2	5722 Xenon Way, Arvada, CO 80002	Jefferson	1824	2995	4.8	6.6	18479	18479	-4804	17953	
3	5955 Carr St, Arvada, CO 80004	Jefferson	1652	1039	5.5	6.4	20729	20729	-5389	20171	
4	7439 W 74th Ave, Arvada, CO 80003	Jefferson	1781	1163	4.8	6.4	18479	18479	-4804	18642	

"A screenshot of the data head for the Solar Energy Data of Random Addresses for Colorado dataset"

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 499 entries, 0 to 498
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   addresses                   499 non-null    object
 1   county                      499 non-null    object
 2   sun_hours                   499 non-null    int64
 3   sqft                        499 non-null    int64
 4   size_for_full_cvg_kW        499 non-null    float64
 5   carbon_metric_ton           499 non-null    float64
 6   upfront_cost                499 non-null    int64
 7   20_year_pay                 499 non-null    int64
 8   state_fed_incentives        499 non-null    int64
 9   20_year_cost_with_solar     499 non-null    int64
 10  20_year_cost_without_solar  499 non-null    int64
 11  20_year_save                499 non-null    int64
dtypes: float64(2), int64(8), object(2)
memory usage: 46.9+ KB
```

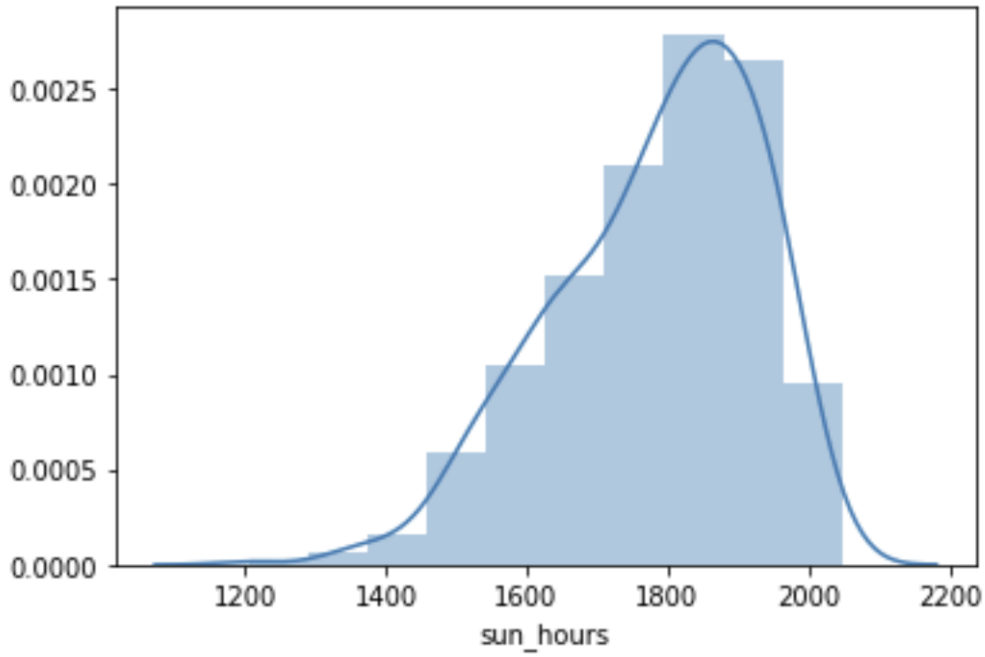
“A screenshot of the info for the Solar Energy Data of Random Addresses for Colorado dataset”

The dataset has a shape of (499, 12) and df.info () shows that majority of the columns are Dtype int64 (eight to be exact), with the exception of two being float64 and two being an object. Each column has 499 non-null values for each column which is good considering the amount of NaN values I had to clean up before.

	sun_hours	sqft	size_for_full_cvg	carbon_metric_ton	upfront_cost	20_year_pay	state_fed_incentives	20_year_cost_with_solar
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	1783.572000	1234.912000	5.311600	6.728800	20079.650000	20079.650000	-5641.284000	19344.806000
std	146.410248	721.294653	0.944335	0.813273	2844.228384	2844.228384	991.384651	1873.698465
min	1207.000000	211.000000	3.000000	3.400000	13229.000000	13229.000000	-8932.000000	16008.000000
25%	1688.000000	788.500000	4.500000	6.300000	17729.000000	17729.000000	-6376.250000	17928.000000
50%	1808.000000	1092.000000	5.300000	6.600000	19979.000000	19979.000000	-5401.000000	18965.500000
75%	1897.000000	1484.500000	5.800000	7.600000	21479.000000	21479.000000	-4804.000000	20468.750000
max	2047.000000	5145.000000	9.000000	7.800000	31229.000000	31229.000000	-3634.000000	29767.000000

“A screenshot of basic statistics for the Solar Energy Data of Random Addresses for Colorado dataset”

The basic statistics of the dataset shows the mean, standard deviation, min, 25th percentile, 50th percentile, 75th percentile, and the max of all columns.



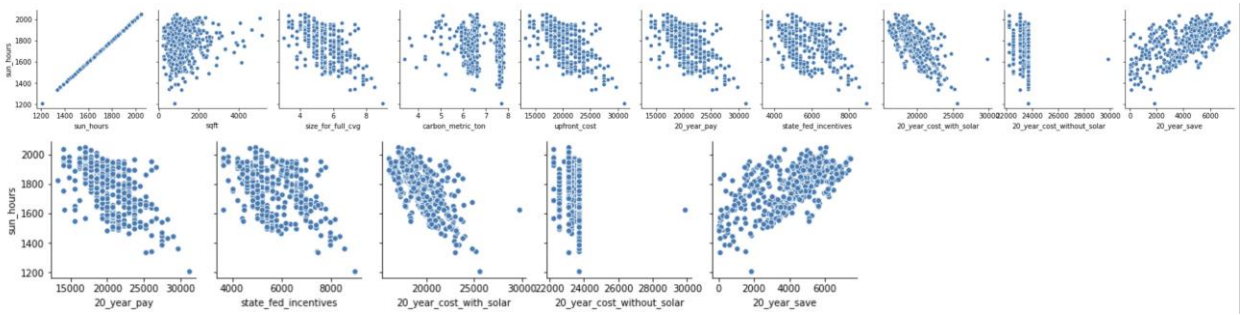
“A screenshot of a distribution plot for the Solar Energy Data of Random Addresses for Colorado dataset”

A distribution plot shows that there is a higher frequency for values between 1800 and 1950 in sun hours.

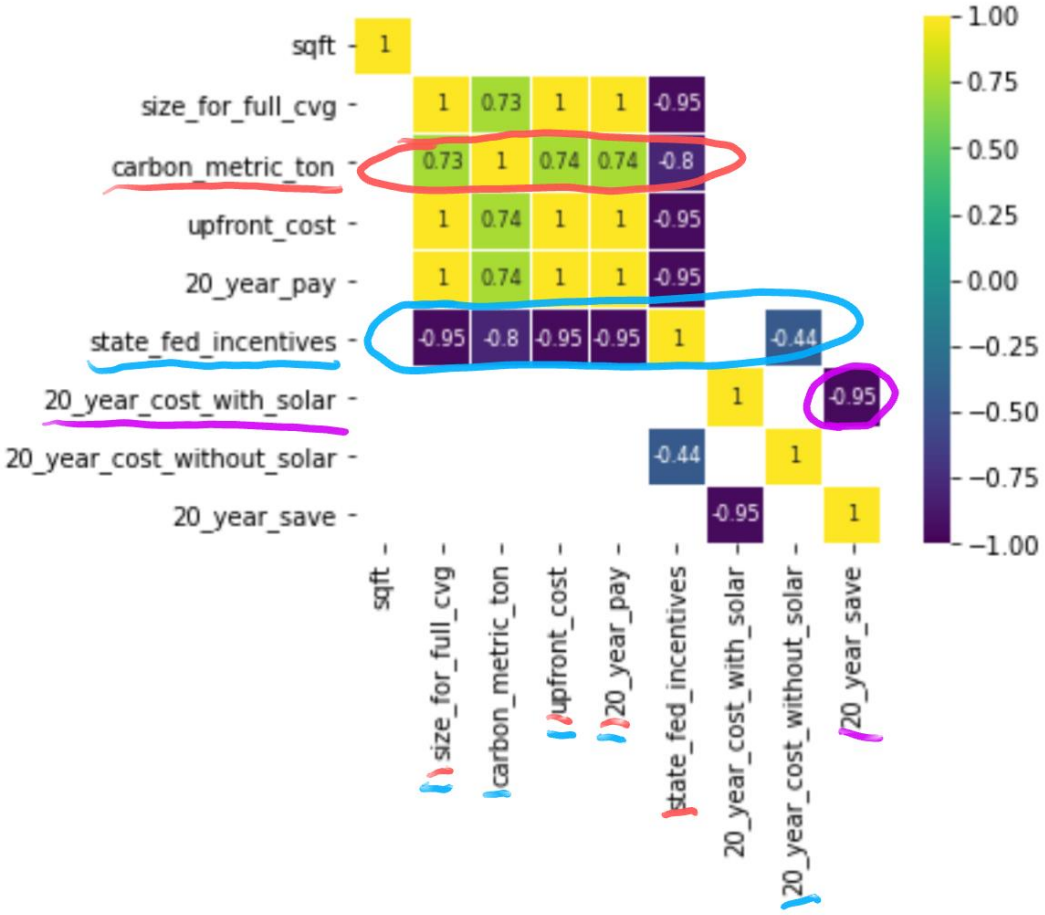
```
sun_hours                1.000000
sqft                     0.096564
size_for_full_cvg       -0.604378
carbon_metric_ton       -0.090017
upfront_cost            -0.600153
20_year_pay             -0.600153
state_fed_incentives     0.431451
20_year_cost_with_solar -0.672827
20_year_cost_without_solar -0.067334
20_year_save             0.626154
Name: sun_hours, dtype: float64
```

“A screenshot of the correlation strength with sun_hours for the Solar Energy Data of Random Addresses for Colorado dataset”

The strength of correlation with the variable/column “sun_hours” and as far as I can see that there are only two variables with positive correlation to sun_hours, but one is sort of strong (20_year_save) and the other is very weak (sqft). The other variables have an inverse correlation to sun_hours with both “size_for_full_cvg” and “20_year_cost_with_solar” being the strongest. Below is a pair plot of that correlation visual.

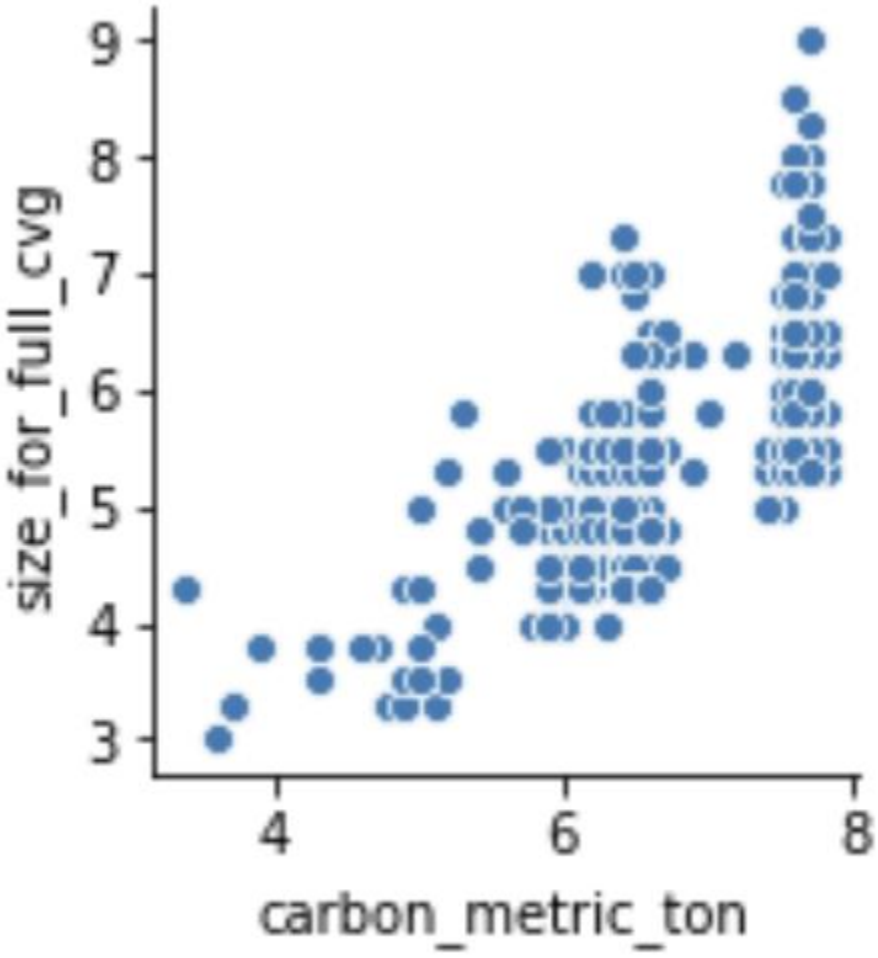


“A screenshot of the correlation strength with sun_hours for the Solar Energy Data of Random Addresses for Colorado dataset”



“A screenshot of the correlation strength for the Solar Energy Data of Random Addresses for Colorado dataset”

Dropping the variable/column “sun_hours” I am able to see the correlation strength between the other variables on a heatmap. The majority of these have an extraordinarily strong correlation because they are all mathematical calculations for the financing of solar panels. In the red, I have highlighted the correlation between the removal of CO2 emissions in carbon metric tons in relation to most of the financial calculations and the size of the solar panel. In blue, I highlighted the state and federal incentives having the strongest correlation between all the financial calculations and the removal of CO2 emissions in carbon metric tons which all have an inverse correlation. The final correlation I highlighted was in the purple in relation to the twenty-year savings calculation and the total cost after the solar panel has been installed for twenty years and this is an inverse correlation making it the other strongest inverse correlation.



“A screenshot of the correlation strength between size for full coverage panels and CO2 emission removal in carbon metric ton for the Solar Energy Data of Random Addresses for Colorado dataset”

This I think will be greatly beneficial when trying to come up with a test hypothesis even though these two have a very obvious correlation. None the less it will be particularly good to define one size panel for every house when modeling.

4.4 Solar Energy Data (County Demographics)

The next dataset I explored was the demographics of solar energy for Colorado by county.

	CFIPS	YEAR	COUNTY	Households	total_roofs_installed	total_est_roofs	total_est_sqft_of_overall_roofs	total_est_capacity_of_mW_dc	total_est_MWh_ac_p
0	1	2017	Adams County	168930	2200	117000	144000000	2000	
1	5	2017	Arapahoe County	250210	3400	146000	168000000	2400	
2	13	2017	Boulder County	130376	3800	44600	52900000	750	
3	14	2017	Broomfield County	26120	685	18200	23200000	330	
4	31	2017	Denver County	310900	3600	129000	171000000	2400	

“A screenshot of the data head for the Solar Energy Data of Random Addresses for Colorado dataset

There isn’t anything much revealing when using df.head() other than the fact the high numbers I have generated for the total Est of roofs and Est sqft of overall roofs after cleaning the data. The shape of the dataset is (17,15) so another smaller dataset for, but this could be due to the counties I removed that had null values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17 entries, 0 to 16
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   CFIPS                                 17 non-null     int64
 1   YEAR                                  17 non-null     int64
 2   COUNTY                                17 non-null     object
 3   Households                            17 non-null     int64
 4   total_roofs_installed                 17 non-null     int64
 5   total_est_roofs                       17 non-null     int64
 6   total_est_sqft_of_overall_roofs       17 non-null     int64
 7   total_est_capacity_of_mW_dc           17 non-null     int64
 8   total_est_MWh_ac_per_year             17 non-null     int64
 9   total_est_median_roof_space           17 non-null     int64
 10  total_est_median_kW_dc_capacity       17 non-null     int64
 11  total_est_median_kWH_ac_per_year      17 non-null     int64
 12  total_est_carbon_metric_tons          17 non-null     int64
 13  total_est_cars_off_road__per_yr       17 non-null     int64
 14  total_est_number_tree_seedlings_grown_10_yrs  17 non-null  int64
dtypes: int64(14), object(1)
memory usage: 2.1+ KB
```

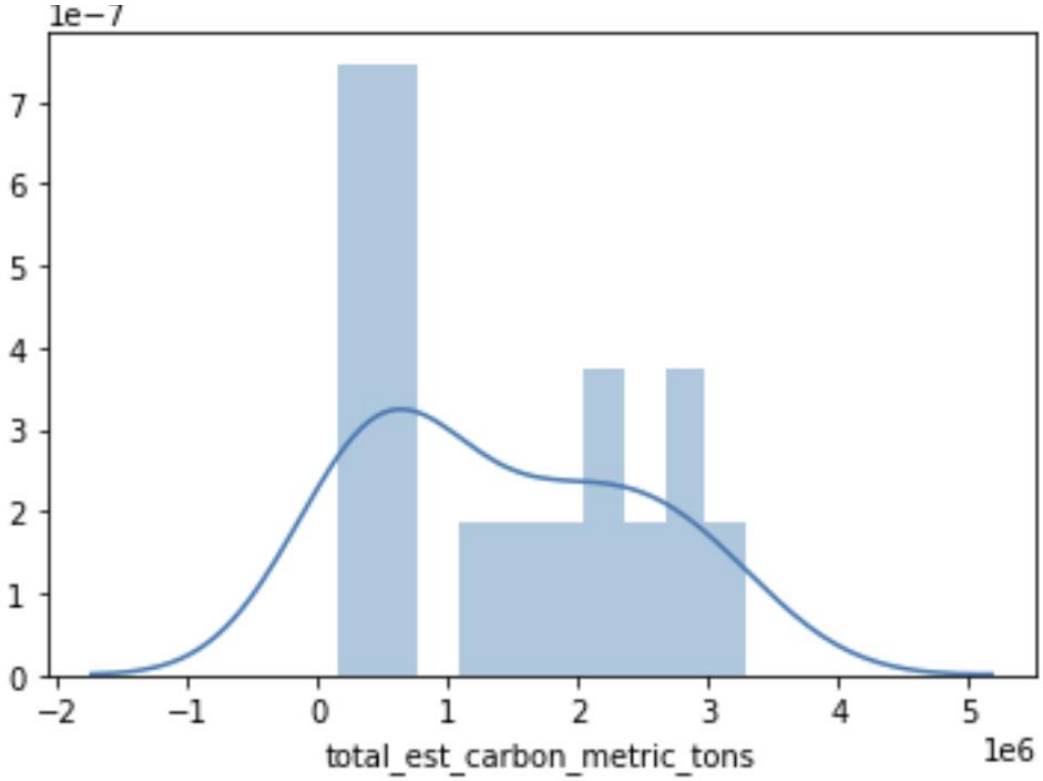
“A screenshot of the info for the Solar Energy Data of Random Addresses for Colorado dataset”

The information on the data shows that every single column is Dtype int64 (14 total) except county which is since the values of that column are all strings.

an_kW_dc_capacity	total_est_median_kWH_ac_per_year	total_est_carbon_metric_tons	total_est_cars_off_road__per_yr	total_est_number_tree_seedlings_grown_10_
17.000000	17.000000	1.700000e+01	17.000000	1.700000e
96.352941	13494.117647	1.420059e+06	298935.294118	3.626471e
17.779698	3168.491476	1.042335e+06	218889.880822	2.655384e
63.000000	9000.000000	1.490000e+05	31500.000000	3.800000e
85.000000	11500.000000	6.060000e+05	128000.000000	1.550000e
98.000000	13300.000000	1.100000e+06	235000.000000	2.850000e
110.000000	15400.000000	2.100000e+06	443000.000000	5.370000e
133.000000	21900.000000	3.300000e+06	694000.000000	8.420000e

"A screenshot of the basic statistics for the Solar Energy Data of Random Addresses for Colorado dataset"

The interesting find from using df.describe() shows on average each county with solar panels on viable roofs would remove 1,420,059 metric tons of CO2 emissions from the electricity sector annually.



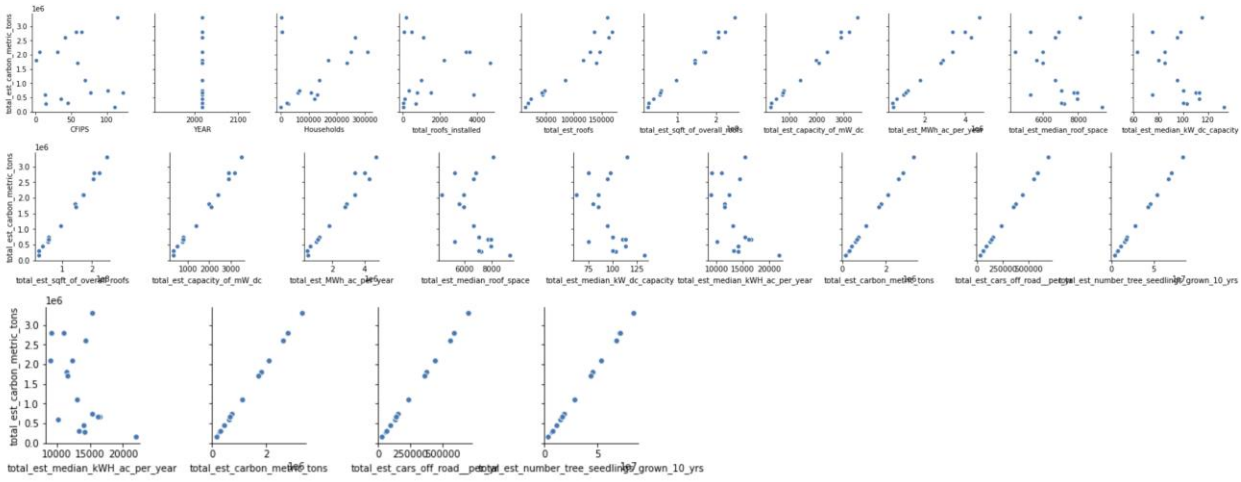
"A screenshot of the distribution of total_est_carbon_metric_tons values for the Solar Energy Data of Random Addresses for Colorado dataset"

I chose the variable of total estimated carbon metric ton values to run with after seeing on average how much CO2 emissions could be removed annually by each county. The distribution, however, shows an extremely high frequency of values in the hundred thousand. This is possibly due to the number of viable roofs in that area that can support solar panels.

```
CFIPS                                -0.031328
YEAR                                       NaN
Households                            0.196321
total_roofs_installed                 0.115842
total_est_roofs                       0.961266
total_est_sqft_of_overall_roofs       0.997313
total_est_capacity_of_mW_dc           0.996228
total_est_MWh_ac_per_year             0.984821
total_est_median_roof_space          -0.388978
total_est_median_kW_dc_capacity      -0.389856
total_est_median_kWH_ac_per_year     -0.463316
total_est_carbon_metric_tons          1.000000
total_est_cars_off_road__per_yr       0.999919
Name: total_est_carbon_metric_tons, dtype: float64
```

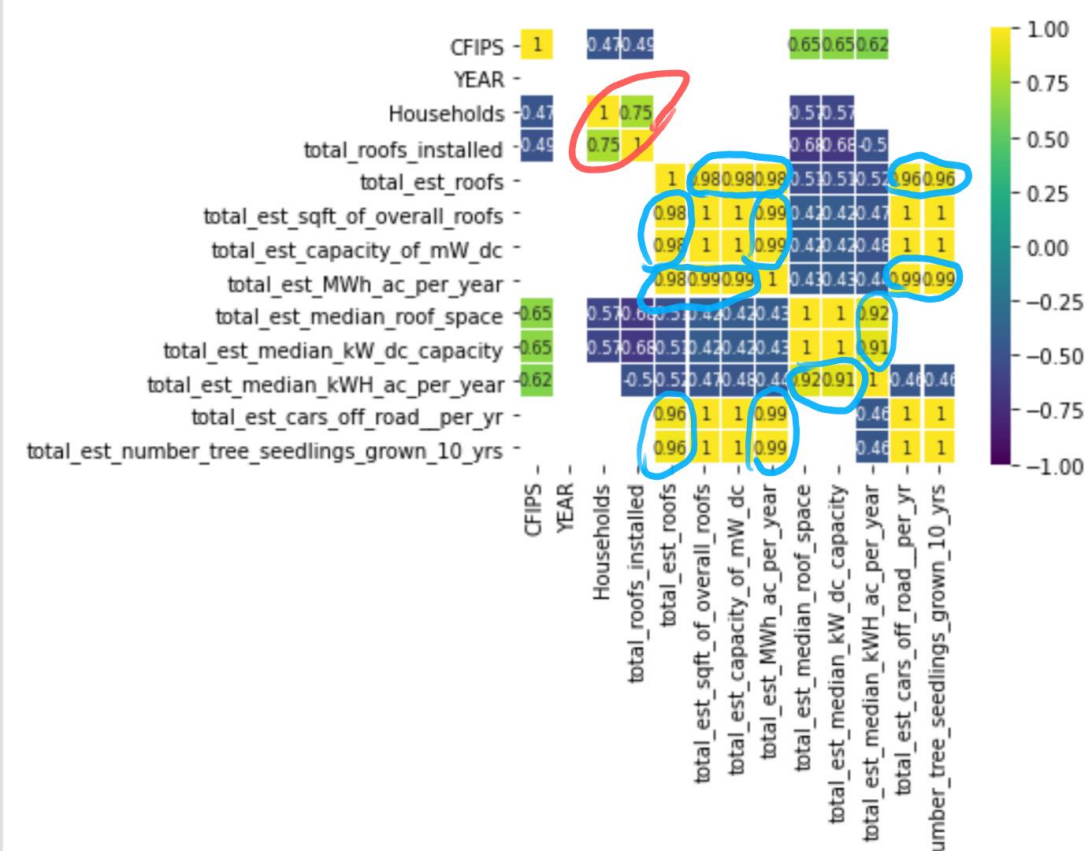
“A screenshot of correlation strength of total_est_carbon_metric_tons values for the Solar Energy Data of Random Addresses for Colorado dataset”

My suspicions on the estimation of total viable roofs is correct and there is a strong correlation there when it comes to the total number of CO2 emissions in metric tons being removed annually from the energy sector. There is also strong correlation with the square footage of all the roofs, the capacity of MW dc, and the estimated electricity generation of MWh ac per year.



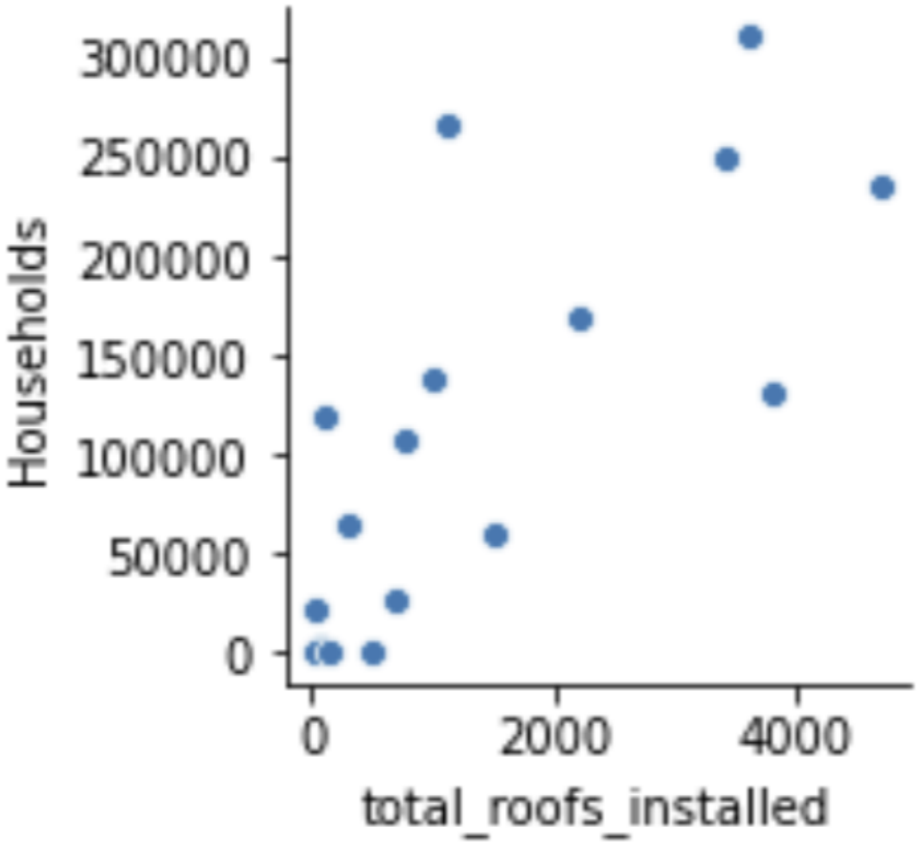
“A screenshot of correlation strength of total_est_carbon_metric_tons values for the Solar Energy Data of Random Addresses for Colorado dataset”

Next, I used a heatmap to show any other correlations I may be overlooking.



"A screenshot of correlation strength of total_est_carbon_metric_tons values for the Solar Energy Data of Random Addresses for Colorado dataset"

I highlighted in blue what I already saw in the visualizations and statistics in the other two before, but in red is what stood out the most. Since the households per county, data is an integrated set of data and wasn't part of the data being scraped for the county demographic data for solar energy I found it remarkably interesting how strong of positive .75 correlation it was.



“A screenshot of correlation strength between total_roofs_installed and Households values for the Solar Energy Data of Random Addresses for Colorado dataset”

This is what the .75 positive correlation looks like on a pair plot. I am really curious about what this possibly means, and maybe it is an anomaly since the number of households per county and the total of number of roofs with installed solar panels or county don’t really have anything in common other than it being a fraction. What really should have been a strong correlation was total roofs with installed solar panels and an estimated number of viable roofs for installations since they both are from the same dataset. However, that part of the heatmap was blank.

4.5 Per Capita Personal Income for Each County in Colorado Data

Exploring this dataset should be fairly easy since the data isn’t that large.

	location_type	location	income_period	income_desc	income	population
0	County	Adams County	Annual	Per Capita Personal Income - Bureau of Economi...	41215	503167
1	County	Alamosa County	Annual	Per Capita Personal Income - Bureau of Economi...	35721	16551
2	County	Arapahoe County	Annual	Per Capita Personal Income - Bureau of Economi...	56642	643052
3	County	Archuleta County	Annual	Per Capita Personal Income - Bureau of Economi...	39944	13315
4	County	Baca County	Annual	Per Capita Personal Income - Bureau of Economi...	42019	3562

“A screenshot of what the head of the data for Per Capita Personal Income for Each County in Colorado dataset”

Only six columns of data with a shape of (64, 6) for the dataset means this will be a fairly simple dataset to explore. A quick glance at the info of the data with df.info() shows that most all of the columns are of Dtype objects (strings from what I saw) and the last two which I feel will be important later on are int64 objects.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64 entries, 0 to 63
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   location_type   64 non-null     object
 1   location        64 non-null     object
 2   income_period   64 non-null     object
 3   income_desc     64 non-null     object
 4   income          64 non-null     int64 
 5   population      64 non-null     int64 
dtypes: int64(2), object(4)
memory usage: 3.1+ KB
```

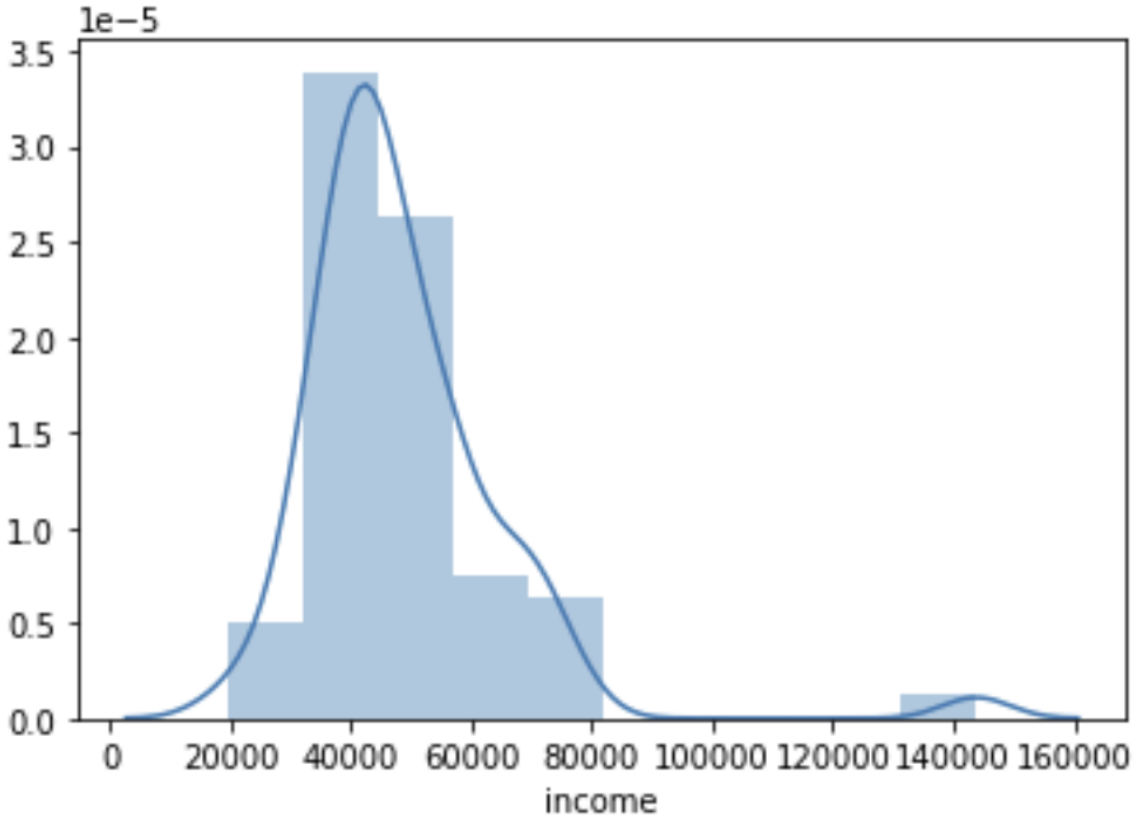
“A screenshot of the info for the Per Capita Personal Income for Each County in Colorado dataset”

The file size for the dataset is fairly small too with it only being 3.1+ KB.

	income	population
count	64.00000	64.000000
mean	48509.81250	87611.781250
std	17164.51282	178045.401336
min	19443.00000	715.000000
25%	38785.00000	5906.000000
50%	44948.00000	14779.500000
75%	55224.00000	43227.750000
max	143812.00000	704621.000000

“A screenshot of the basic statistics for the Per Capita Personal Income for Each County in Colorado dataset”

The basic statistics of the data indicate that the mean for the per capita personal income for all counties is 48,509.81 which is a little off in comparison to the per capita for personal income for the state of Colorado at 53,504. The standard deviation is 17,164.51, the minimum is 19,443, the 25th percentile is 38,785, the 50th percentile is 44,948, the 75th percentile is 55,224, and the maximum is 143,812.



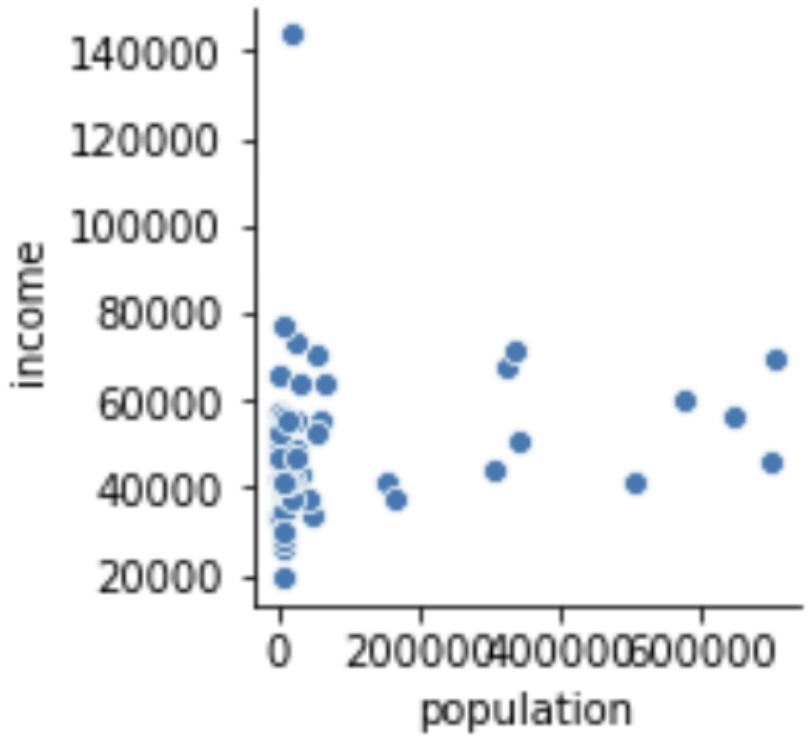
“A screenshot of the distribution for the Per Capita Personal Income for Each County in Colorado dataset”

The distribution plot for the data shows that the frequency of values is significantly high within the 40000 to 60000 range and that one of the counties is a possible outlier to this data. However, the outlier isn’t a problem since houses in Pitkin county can install solar panels too.

```
income        1.00000
population    0.17872
Name: income, dtype: float64
```

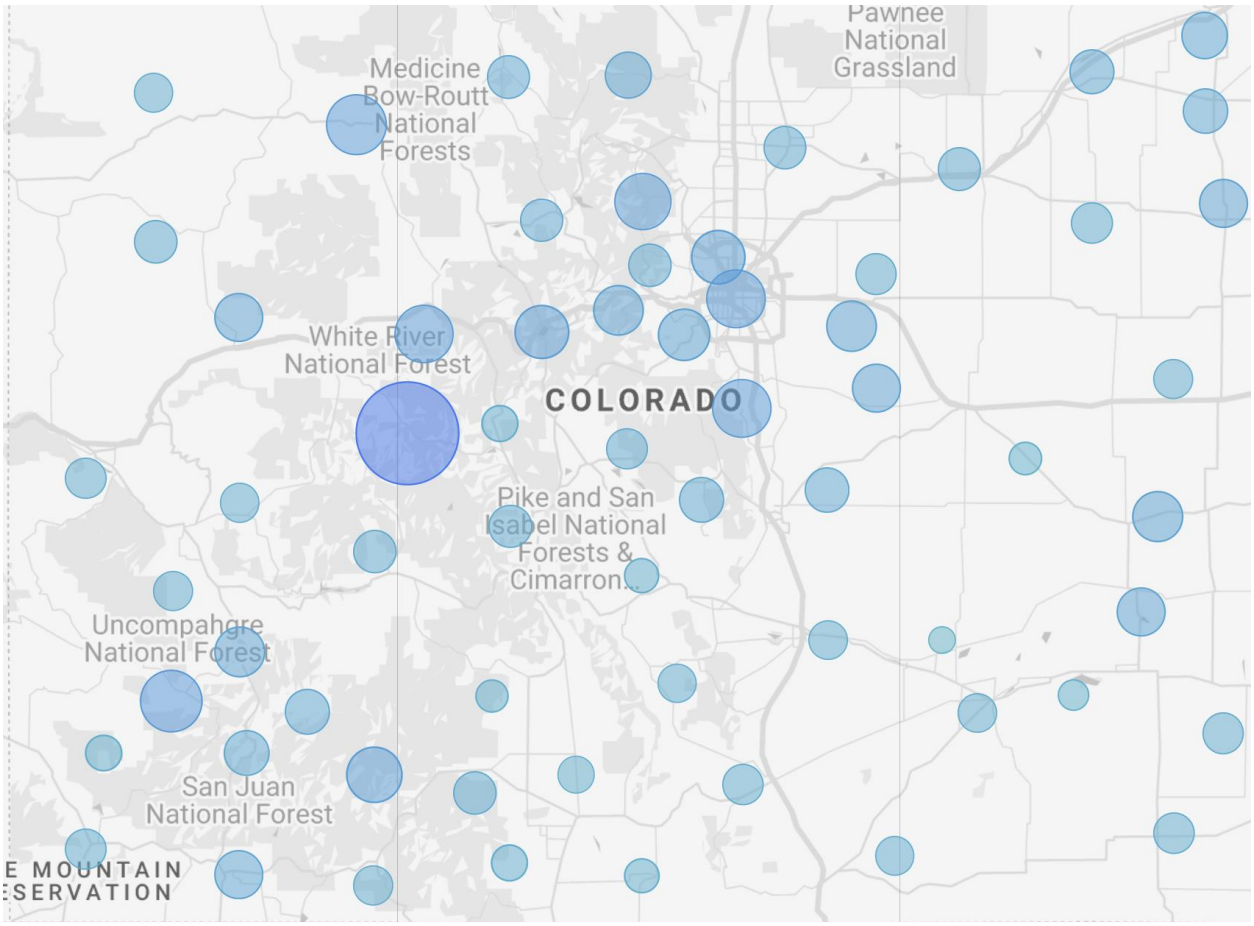
“A screenshot for the correlation between income and population for the Per Capita Personal Income for Each County in Colorado dataset”

I didn’t expect to see much of a correlation here since there really is no determination for the income of a county based on the population count.



“A screenshot of a pair plot showing the correlation between income and population per county for the Per Capita Personal Income for Each County in Colorado dataset”

Even the pair plot confirms that there is a lack of correlation between the two variables, but this isn’t a terrible thing.



"A screenshot of a map showing the income distribution through bubbles for the Per Capita Personal Income for Each County in Colorado dataset"

4.6 Data Integration and Feature Engineering

This next part is pretty crucial in setting up before the machine learning can commence. Data integration, feature engineering, and feature selection are all important and necessary steps to complete so that the data can be properly fitted and scaled for machine learning. The first dataset that I integrated and engineered was the solar energy data for random addresses.

*4.7 Feature Engineering and Integrating Data for the Solar Energy Data (Random **Addresses)***

This dataset was the most crucial and important set to my project for the main dataset would rely on the integration of engineered features and calculations from this data to prepare the final dataset for integration and feature engineering.

M	N	O	P	Q
upfront_cost_after_incentives	cost_w	ost_wit	year_s	assumed_income
16195	19535	23706	4171	60398
18670	23117	23250	133	60398
13675	17953	23250	5297	60398
15340	20171	23250	3079	60398
13675	18642	23250	4608	60398
13675	18325	23250	4925	60398
14785	19634	23250	3616	60398
16450	20746	23250	2504	60398
15895	20466	23250	2784	60398

“A screenshot of the feature engineered upfront_cost_after_incentives and the integrated feature for assumed_income”

The first thing I did was create an assumed feature of the average household electricity bill for $80 a month totaling to $960 a year. Then I calculated a new feature for the upfront cost after the federal incentives were applied and the third one was the integrated feature engineered for assumed income per household based on the household’s county, they reside in. The next feature I integrated into the dataset was from the average household savings based on income dataset and depending on the conditional ranges of their income was what dictated their assumed savings. After that engineered a feature with the calculated number of years for system payback before the incentive which is calculated by taking the initial upfront cost minus the federal tax incentive and then dividing it by the annual cost of electricity.

R	S	T	U	V	W	X	Y
avg_savings_by_income	years_payback	Increase_state_incentives	est_new_upfront_cost	kWh_yr_est	new_ROI_(system payback)	2019_affordable(YoN)	2020_incentive_affordable(Yo
19504.92	16.86979167	-2000	14195	9,504	14.78645833	Y	Y
19504.92	19.44791667	-2000	16670	10,502	17.36458333	Y	Y
19504.92	14.24479167	-2000	11675	7,247	12.16145833	Y	Y
19504.92	15.97916667	-2000	13340	6058	13.89583333	Y	Y
19504.92	14.24479167	-2000	11675	7,287	12.16145833	Y	Y
19504.92	14.24479167	-2000	11675	7,282	12.16145833	Y	Y
19504.92	15.40104167	-2000	12785	6,068	13.31770833	Y	Y
19504.92	17.13541667	-2000	14450	6,072	15.05208333	Y	Y
19504.92	16.55729167	-2000	13895	6,068	14.47395833	Y	Y

“A screenshot of what the remaining seven features engineered for the integrated random address solar dataset”

The most important but one of the easiest features I created was the Colorado state tax incentive of $2000 which would be calculated into the newly calculated/engineered feature of the new upfront cost. Then after those two features are made, I constructed through calculation the return of investment (system payback). The remaining two columns I engineered with the assumption that this sample of random addresses wanted to install solar systems.

year	yes	no	solar_installed	total
2019	406	93	406	499
2020	474	25	68	499

“A screenshot of the calculation of which households could assumedly purchase a solar system before and after state tax incentives”

So with the use of python, I set up a function that would check to see if before the state tax incentive households could afford to buy the estimated solar system based off of project sunroof’s estimation of what the house could install to almost cover their assumed default electricity bill of $80 a month. If the household could afford it, then the household would receive a Y for yes, and if they couldn’t an N for no. The next year is when the incentives would be applied and then the function would go back and be applied to the remaining homes that could then purchase a home with the assumed savings they had. The first year the 406 out of 499 houses could purchase a solar system leaving 93 remaining households assumedly still want to. The seconds year after the hypothetical roll out of incentives 68 households could no purchase solar systems which would give be calculated as a **73% increase** which would be the assumed percentage of increase to be applied to the main dataset for integration for the increase in buying solar systems.

4.8 Feature Engineering, Integration and Exploration of Data for the Main Dataset to Predict Adoption Rates

This dataset was mostly all feature engineering and integration from other collected data. Some of the features had data dating back to 1960, but to keep everything fairly consistent to where there weren’t too many imputations for the other datasets with missing data I decided to decrease the range of years from 1960 to 2025 to 2010 to 2025.

Years	Total_emissions_	Total_emissions_	Solar_kWh_gene	Solar_kWh_gene	Total energy con	electric_cost	est_installed_cos	state_level_gdp	est_annual_insta	est_annual_insta	est_cumul_instal	est_cumul_instal	cumul_est_	cumul_est_	est_pct_ch	est_pct_ch	adoption_rates	adoption_rates_incentives_
2010	7900000	7900000	49000000	49000000	35231500000000	32.35	7.34	255140500000	18.65	18.65	18.65	18.65	3391	3391	0.000	0.000	0.00000000	0.00000000
2011	7900000	7900000	65000000	65000000	35426200000000	33.02	6.44	264431600000	13.8	13.8	32.45	32.45	5900	5900	0.740	0.740	0.00250909	0.00250909
2012	7100000	7100000	88000000	88000000	33494900000000	33.58	4.55	273519500000	17.7	17.7	50.15	50.15	9118	9118	0.545	0.545	0.00321818	0.00321818
2013	8200000	8200000	124000000	124000000	36336800000000	34.96	3.97	288305200000	28.2	28.2	78.35	78.35	14245	14245	0.562	0.562	0.00512727	0.00512727
2014	8100000	8100000	177000000	177000000	35333400000000	35.7	3.49	306571100000	41.8	41.8	120.15	120.15	21845	21845	0.534	0.534	0.00760000	0.00760000
2015	7500000	7500000	210000000	210000000	34737600000000	35.52	3.23	318554900000	41.3108046	41.3108046	161.4608046	161.4608046	29357	29357	0.344	0.344	0.00751106	0.00751106
2016	7500000	7500000	265000000	265000000	34830800000000	35.39	3.02	329368300000	35.31514368	35.31514368	196.7759483	196.7759483	35777	35777	0.219	0.219	0.00642094	0.00642094
2017	7300000	7300000	317000000	317000000	34332600000000	35.65	2.84	350004400000	51.90093276	51.90093276	248.676881	248.676881	45214	45214	0.264	0.264	0.00943653	0.00943653
2018	7414285.714	7085892.151	338392857.1	464441686.7	34614871428571	34.23203414	2.7	359421021429	58.30897126	58.30897126	306.9858523	306.9858523	55816	55816	0.234	0.234	0.01060163	0.01060163
2019	7353571.429	6799432.508	377619047.6	577291153.1	34536595523809	34.72459459	1.731268385	373017484524	74.59104253	74.59104253	381.5768948	381.5768948	69378	69378	0.243	0.243	0.01356201	0.01356201
2020	7292857.143	6479192.155	416845238.1	682455416.9	34459047619047	35.21715505	1.201089714	386613947619	69.51129085	120.3240445	451.0881857	501.9009393	82016	91255	0.182	0.315	0.01263842	0.02187710
2021	7232142.857	6114080.503	456071428.6	802342677.7	34381135714285	35.89563067	0.5296973966	400210410714	79.24287157	137.1694107	530.3310573	639.07035	96424	116195	0.176	0.273	0.01440779	0.02493989
2022	7171428.571	5723463.252	495297619	907999440.1	34303223809523	36.11545118	0.004156688847	413806873810	69.83682165	120.8875383	600.1678789	759.9578883	109121	138174	0.132	0.189	0.01269760	0.02197955
2023	7110714.286	5297356.523	534523809.5	1021052176	34225311904761	36.33527169	0.01	427403336905	74.72539916	129.3496659	674.8932781	889.3075542	122708	161692	0.125	0.170	0.01358644	0.02351812
2024	7050000	4829365.196	573750000	1151062822	34147400000000	36.5550922	0.01	440999800000	85.93420904	148.7521158	760.8274871	1038.05967	138332	188738	0.127	0.167	0.01562440	0.02704584
2025	6989285.714	4308901.104	612976190.5	1310891035	34069488095238	36.77491271	0.01	454596263095	105.6429725	182.8679854	866.4704596	1220.927655	157540	221987	0.139	0.176	0.01920781	0.03324872

“A screenshot of the main dataset used for predicting adoption rates”

With a shape of (16,19) the dataset contains seventeen Dtype float64 columns and two int64 columns with no non-null values. This dataset will probably cause underfitting when applying to the model’s, but in order to limit the number of forecasted/back casted imputed values I would have to do with a dataset ranging back to 1960 or 1980 I had to go with this.

1	Years	Total_emissions_	Total_emissions_
2	2010	7900000	7900000
3	2011	7900000	7900000
4	2012	7100000	7100000
5	2013	8200000	8200000
6	2014	8100000	8100000
7	2015	7500000	7500000
8	2016	7500000	7500000
9	2017	7300000	7300000
10	2018	7414285.714	7085892.151
11	2019	7353571.429	6799432.508
12	2020	7292857.143	6479192.155
13	2021	7232142.857	6114080.503
14	2022	7171428.571	5723463.252
15	2023	7110714.286	5297356.523
16	2024	7050000	4829365.196
17	2025	6989285.714	4308901.104

"A screenshot of the first, second, and third column of the dataset for predicting adoption rates"

The first column was data is the index of years from 2010 to 2025. The second column is pulled from the Colorado CO2 emissions data in carbon metric tons and was scaled from the year 1980 - 2017 to 2010 to 2025. The years from 2018 to 2025 were imputed values using the "forecast.linear" function in sheets. The sister of this column (third column) is an engineered feature which is the total emissions with forecasts and alleviated emissions that have the same format of imputations. However, the calculations for the assumed alleviated carbon metric tons were performed based on the assumption that the average solar system is 5.5 kW which generates about 8,321 kWh on average.

Solar_kWh_gene	Solar_kWh_gene	Total energy cons
49000000	49000000	3523150000000(
65000000	65000000	3542620000000(
88000000	88000000	3349490000000(
124000000	124000000	3633680000000(
177000000	177000000	3533340000000(
210000000	210000000	3473760000000(
265000000	265000000	3483080000000(
317000000	317000000	3433260000000(
338392857.1	464441686.7	34614871428571
377619047.6	577291153.1	3453695952380
416845238.1	682455416.9	3445904761904
456071428.6	802342677.7	343811357142
495297619	907999440.1	3430322380952
534523809.5	1021052176	3422531190476
573750000	1151062822	3414740000000(
612976190.5	1310891035	340694880952

“A screenshot of the next three columns in the dataset for predicting adoption rates”

The fourth column solar kWh Generation by residential had data from the total consumption of energy in Colorado dataset. Ranging between 1960 and 2017 but was dropped down ranges between 2010 and 2017. The imputations here were also done with the “forecast.linear “function on sheets, but for the

sister (fifth) column, the 2020 data until 2025 data is calculated after the implementation of the state tax incentives.

The sixth column which was the total energy consumption by the residential sector from the same dataset as the fourth column. Brought from ranges 1960 to 2017 down to 2010 to 2017 with forecasts from 2018 to 2025. This column is assumed to stay consistent regardless of incentives since it is data for consumption, and not generation or expenditure.

electric_cost	est_installed_cos	state_level_gdp
32.35	7.34	255140500000
33.02	6.44	264431600000
33.58	4.55	273519500000
34.96	3.97	288305200000
35.7	3.49	306571100000
35.52	3.23	318554900000
35.39	3.02	329368300000
35.65	2.84	350004400000
34.23203414	2.7	359421021429
34.72459459	1.731268385	373017484524
35.21715505	1.201089714	386613947619
35.89563067	0.5296973966	400210410714
36.11545118	0.004156688475	413806873810
36.33527169	0.001	427403336905
36.5550922	0.001	440999800000

“A screenshot of the seventh and eighth columns for the dataset predicting the adoption rates”

The seventh column is the cost of electricity over the years coming from the total price expenditures of energy for the Colorado dataset. Also dating back to 1960 to 2017 this data was scaled down for 2010 to 2017 and forecasted from 2018 to 2025 also.

The eighth column is the annual estimated installation cost of a dollar per watt from the National Renewable Energy Laboratory LCOE (Levelized cost of energy) data. This data was estimated from 2010 to 2018 where I forecasted the years from 2019 to 2022 for imputations, and then marked the remaining values to .001. These imputations may be the most eccentric since I don't think dollar per watt will ever cost a penny, but the forecast was starting to go into the negatives which who knows maybe the government will pay for entire solar installations in the future.

state_level_gdp	est_annual_insta	est_annual_insta
255140500000	18.65	18.65
264431600000	13.8	13.8
273519500000	17.7	17.7
288305200000	28.2	28.2
306571100000	41.8	41.8
318554900000	41.3108046	41.3108046
329368300000	35.31514368	35.31514368
350004400000	51.90093276	51.90093276
359421021429	58.30897126	58.30897126
373017484524	74.59104253	74.59104253
386613947619	69.51129085	120.3240445
400210410714	79.24287157	137.1694107
413806873810	69.83682165	120.8875383
427403336905	74.72539916	129.3496659
440999800000	85.93420904	148.7521158
454596263095	105.6429725	182.8679854

“A screenshot of the ninth, tenth, and eleventh columns for the dataset predicting the adoption rates”

The ninth column was the state-level GDP data from the Bureau of Economic Analysis dataset. This data was combined together from 1980 to 2017 but was scaled to 2010 to 2017 for consistency. The years 2018 to 2025 were forecasted through sheets as well.

The tenth column “Annual Installed Capacity of MW dc” didn’t involve any forecasts or calculations since it was already all done by Wood Mackenzie Power & Renewables. The data ranges from 2010 to 2025, but with the years from 2020 to 2025 being forecasted estimates by Wood Mackenzie. This column is what I used to engineer a lot of the next features like its sister column (eleventh). The sister column contains all the same values until 2020 where the state incentives would hypothetically take place and I calculated with the remainder of those estimated total installed MW dc capacity values with a 73% increase all the way until 2025.

est_cumul_instal	est_cumul_instal	cumul_est_	cumul_est
18.65	18.65	3391	3391
32.45	32.45	5900	5900
50.15	50.15	9118	9118
78.35	78.35	14245	14245
120.15	120.15	21845	21845
161.4608046	161.4608046	29357	29357
196.7759483	196.7759483	35777	35777
248.676881	248.676881	45214	45214
306.9858523	306.9858523	55816	55816
381.5768948	381.5768948	69378	69378
451.0881857	501.9009393	82016	91255
530.3310573	639.07035	96424	116195
600.1678789	759.9578883	109121	138174
674.8932781	889.3075542	122708	161692
760.8274871	1038.05967	138332	188738
866.4704596	1220.927655	157540	221987

“A screenshot of the twelfth, thirteenth, fourteenth, and fifteenth columns for the dataset predicting the adoption rates”

The twelfth column for estimated cumulative installed MW dc is a set of engineered values based on the annual installed capacity of MW dc values and the same is done for its sister column (thirteenth) based off the tenth column’s sister column calculated with the incentive percentage increase.

The fourteenth column is the estimated cumulative number of roofs with installed solar panels. This column is calculated based on the estimated cumulative installed MW dc with the rough average of all

households having 5.5 kW solar systems. To get the cumulative household values by taking the cumulative value of the installed MW dc and dividing it by .0055 which is equivalent to 5.5 kW in MW. The same is done for column fourteen's sister column (fifteen) based on the implementation for column thirteen with the assumed percentage increase from the state tax incentives.

The sixteenth column is the estimated percentage of change on a year to year basis from how many homes installed solar systems. This feature is engineered with the calculation of the present year's value minus the previous years and then divided by the previous year's value. This column has a sister (seventeen) which has the same calculations in relation to column fifteen.

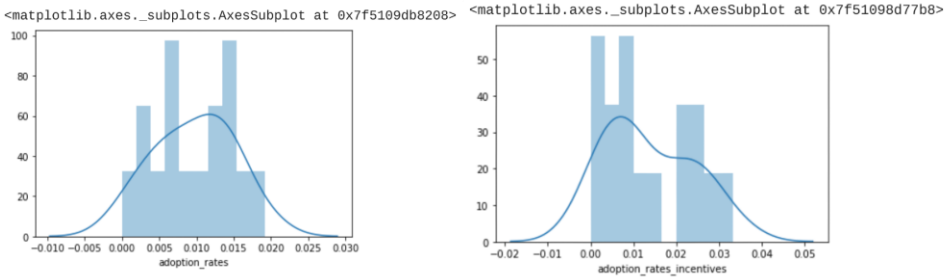
The final two columns (eighteen and nineteen) are the engineered features of adoption rates. The eighteenth column being the adoption rates without the incentive implementation, and the nineteenth column being the one with the incentive implementation. The calculation for these adoption rates is done by taking the value of presently installed solar systems minus the previous year's installed solar systems to give the number of new systems installed divided by the total viable number of roofs that can install solar systems which according to Google's [Project Sunroof] is 1 million.

	Years	Total_emissions_wforecast	Total_emissions_wforecast_aleviated_emissions	Solar_kWh_generation_residential	Solar_kWh_generation_resident
count	16.000000	1.600000e+01	1.600000e+01	1.600000e+01	
mean	2017.500000	7.444643e+06	6.758605e+06	3.187798e+08	
std	4.760952	3.818590e+05	1.197875e+06	1.870458e+08	
min	2010.000000	6.989286e+06	4.308901e+06	4.900000e+07	
25%	2013.750000	7.156250e+06	6.016426e+06	1.637500e+08	
50%	2017.500000	7.326786e+06	7.092946e+06	3.276964e+08	
75%	2021.250000	7.600000e+06	7.600000e+06	4.658780e+08	
max	2025.000000	8.200000e+06	8.200000e+06	6.129762e+08	

"A screenshot of what the basic statistics for the dataset for predicting adoption rates"

Here is a sample of what the basic statistics for the data set looks like for the first four columns. It is clear that the numbers for the total emissions with forecast for alleviated emissions are lower than the original total emissions with forecasts. If you would like to see the full df.head() here is a link to it.

(insert link)



"A screenshot of a distribution plot comparison between the two target variables for the main dataset"

The distribution of the values between the two indicates that the original adoption rate calculations are pretty distributed with the one hump, but the adoption rate with the incentives shows a little bit of uneven distribution. This is obviously because of the incentive calculation.

Years	0.969220
Solar_kWh_generation_residential	0.964168
Solar_kWh_generation_residential_incentives	0.929946
electric_cost	0.820672
state_level_gdp	0.969144
est_annual_installed_capacity	0.988772
est_annual_install_cap_after_incentives	0.931272
est_cumul_installed_capacity	0.938054
est_cumul_installed_capacity_after_incentives	0.901856
cumul_est_roofs	0.938054
cumul_est_roofs_after_incentives	0.901856
adoption_rates	1.000000
adoption_rates_incentives	0.946671

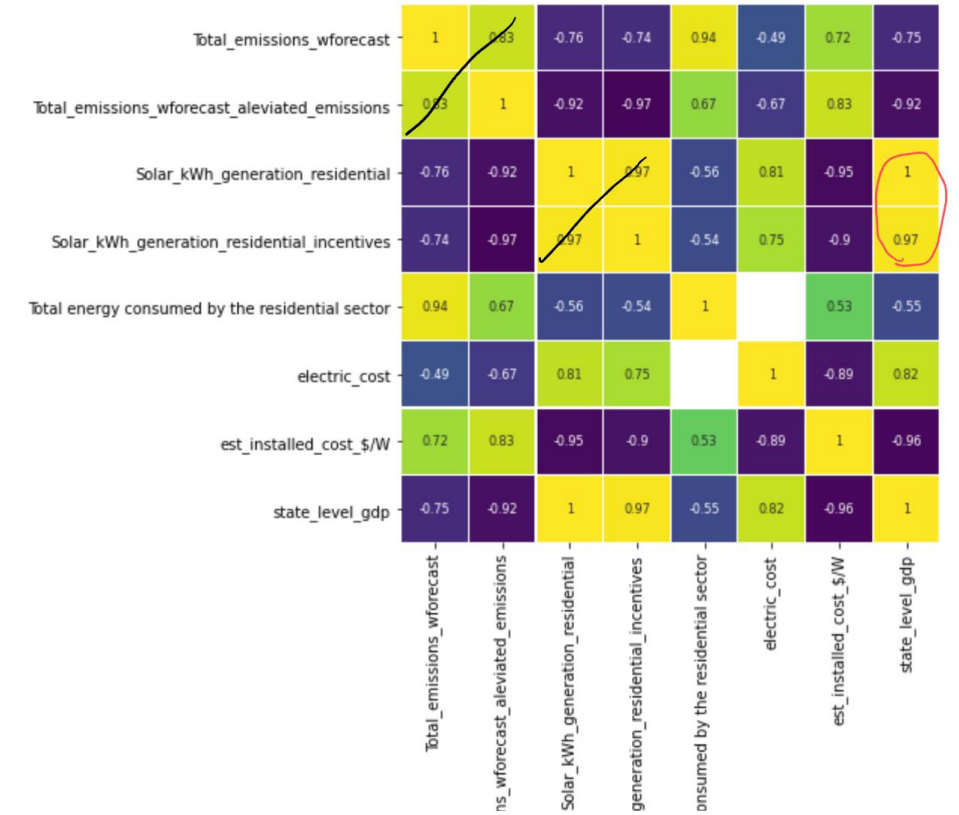
Name: adoption_rates, dtype: float64

Years	0.963599
Solar_kWh_generation_residential	0.960256
Solar_kWh_generation_residential_incentives	0.978318
electric_cost	0.784066
state_level_gdp	0.963280
est_annual_installed_capacity	0.953281
est_annual_install_cap_after_incentives	0.996834
est_cumul_installed_capacity	0.978577
est_cumul_installed_capacity_after_incentives	0.967353
cumul_est_roofs	0.978577
cumul_est_roofs_after_incentives	0.967353
adoption_rates	0.946671
adoption_rates_incentives	1.000000

Name: adoption_rates_incentives, dtype: float64

"A screenshot of the calculated correlation between features both target variables on the main dataset"

Both target variables have high correlations for features that weren't engineered to calculated to give the adoption rates except the Solar kWh generation with incentives may be pushing it since those remaining values were estimated based on assumed kW panels used for household solar systems on average. I probably will leave out the solar kWh generation calculated after the incentives just to avoid researcher bias.



"A screenshot of a heatmap for correlations between all variables that aren't directly related to the target variables"

I plotted a heatmap to see the correlations between all of the variables that weren't directly related to the target variable in regards of calculations. What struck me as odd were the correlations between solar kWh generation and state-level GDP. Mostly because these two came from completely different

datasets, and different agencies who supplied them. I would assume that somewhere in the total calculated GDP there is calculated values that have to do with solar energy but looking at the dataset for the state-level GDP I couldn't find anything on energy.

After exploring this integrated dataset, it is clear to me that there is quite a bit of features with multi collinearity going on. However, I wasn't going to use some of those features anyways for the machine learning, but rather had them there to give me more insight into what features would be better correlated with the engineered adoption rates.

5. Predictive Analysis Using Machine Learning and Visualization


Image by mohamed Hassan from Pixabay

Machine learning is probably the smallest part of the data science lifecycle. Applying the correct algorithm for fitting the data is crucial in this step-in order to gain the most insight from the data as possible.

5.1 Choosing Machine Learning Algorithms

Since the data is a time series, I have decided to use Scikit-Learn's Linear Regression, an RNN LSTM (Recurrent Neural Network Long-Short Term Memory), and ARIMA (Autoregressive Integrated Moving Average). I chose Scikit-Learn's Linear Regression model because I am familiar with Sklearn libraries in Python, and the time series data is very linear regardless. I chose the LSTM RNN because I am very fascinated about how neural networks operate and have only used a densely (fully) connected layer, and convolutional layer neural networks. ARIMA is the newest addition for me in my collection of machine learning algorithms.

5.2 Scikit-Learn's Linear Regression

Linear Regression is a well-known and simple modeling algorithm used to fit find and fit linear datasets. It is a supervised machine learning algorithm as it needs for the dataset to be split in two, one for training and the other for testing. What linear regression does is it takes the values of the training data

to train the machine with the algorithm to predict output points and test against them with the testing data. There are two ways to apply linear regression.

One application is simple regression meaning one target feature (dependent variable) and one predictor feature (independent variable). The algorithm uses basic slope-intercept form y=mx+b, where m is the weight (coefficients) and b is the bias (intercept). Then the algorithm uses the training data to learn the best value for the weight and the bias in order to predict the target.

Another application of linear regression is multivariate regression and is the method I will be using.

$$f(x, y, z) = w_1 x + w_2 y + w_3 z$$

"Multivariate Regression algorithm from ("Linear Regression — ML Glossary documentation," 2014)"

Multivariate regression is the use of multiple features (x,y,z) in order to predict the target feature. For both of these applications of linear regression, the weights will use a cost function in order to optimize weights and this is done with MSE (mean squared error). The objective is to minimize the MSE so the accuracy of the model can be improved, and this done through gradient descent.

$$MSE = \frac{1}{2N}\sum_{i=1}^{n}(y_i - (W_1x_1 + W_2x_2 + W_3x_3))^2$$

Cost Function

Chain Rule

$$f'(W_1) = -x_1(y - (W_1x_1 + W_2x_2 + W_3x_3))$$
$$f'(W_2) = -x_2(y - (W_1x_1 + W_2x_2 + W_3x_3))$$
$$f'(W_3) = -x_3(y - (W_1x_1 + W_2x_2 + W_3x_3))$$

"The cost function being calculated with gradient descent, ("Linear Regression — ML Glossary documentation, annotated by James Nelson" 2014)"

Gradient descent uses the chain rule in order to iterate through and minimize the MSE. Gradient descent can be improved though by vectorizing it and turning it into a matrix for these calculations.

Then a bias term just like simple regression is added to this matrix of values that are calculated from gradient descent to be prepared for iterating through the training data with a learning rate to get the MSE as minimal as possible ("Linear Regression — ML Glossary documentation," 2014).

5.3 ARIMA (Auto Regressive Integrates Moving Average)

The second machine learning model I fitted was ARIMA. I know about simple moving averages and exponential moving averages, but never heard of ARIMA, so I was interested when I read it could predict time series data with machine learning. ARIMA is actually a linear regression model made up of a few different models which take past values to predict future values. ARIMA can be characterized by three terms p, d, and q (Prabhakaran, 2019). The P stands for AR order, the Q is for the MA order, and the D is for the differencing of time series to make it stationary.

In order to train the data on ARIMA the orders of p, q, and d need to be defined when fitting the model. P is needed for the AR model which is when the model will only be fitted based on its own lags which is Yt (Prabhakaran, 2019).

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

"The equation for lags of Yt for the AR model, (Prabhakaran, 2019)"

The q for the order of the MA model is when the Yt is using lags based on the lagged forecast of errors.

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

"The equation for Yt dependence on lags forecasted errors for the MA model, (Prabhakaran, 2019)"

The two models combined together make up the ARIMA algorithm so that the time series data can be differenced at least once in order to become stationary so that Yt can now be predicted based off of the past lags from the AR equation and the lagged forecasted errors from MA .

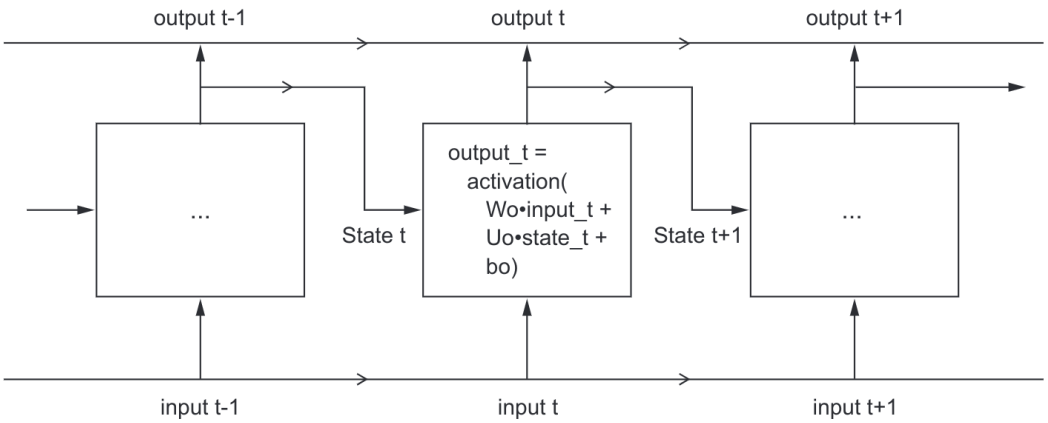
$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

"The algorithm for ARIMA to make a time series stationary in order to, (Prabhakaran, 2019)"

5.4 Long-Short Term Memory Recurrent Neural Network

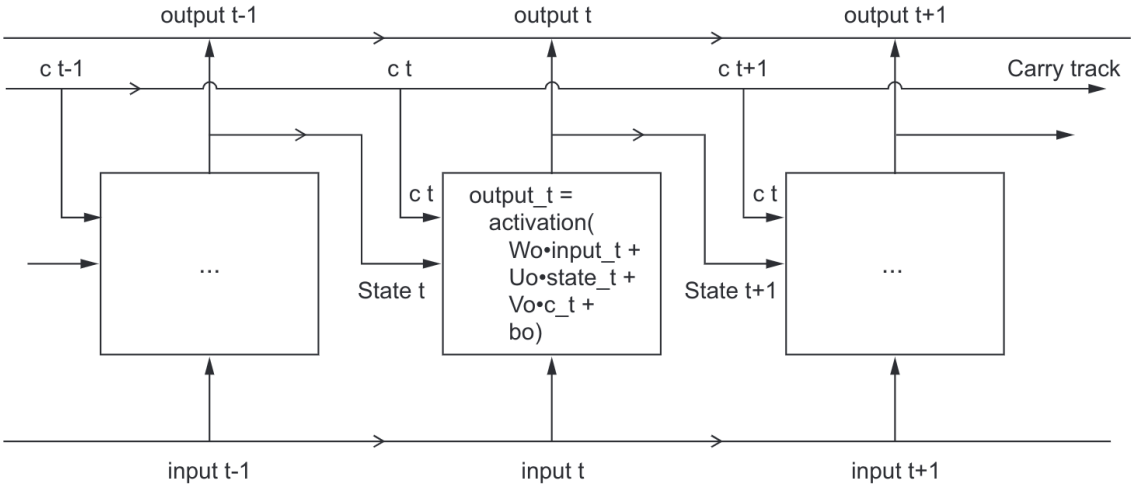
An RNN is a representation of how biological intelligence processes data by sequentially observing events and then remembering these events with internal memory (Chollet, 2018). So, an RNN will pass through the data multiple times and rather like other neural networks that are viewing data at each iteration but not storing the information internally, an RNN will store this data internally. However, the

problem with an RNN when the layers pile on and as iteration goes forward it begins to lose its trainability and therefore useless. The solution is an LSTM RNN where information is stored separately from where the training takes place and is then being carried forward as the model continues to iterate forward. Below is a visualization of how a regular RNN works.



"A visualization of How a simple RNN works, (Chollet, 2018)"

The information processing flow above shows what happens with a normal RNN and how data is stored for in its recurrent memory, but with the introduction of adding another flow to the process that allows the data to be carried forward is what makes an LSTM. The carried data is then reintroduced back into the iterative flow.



"A visualization of how an LSTM RNN works, (Chollet, 2018)"

The top recurrent flow is the representation of a long-term memory, while the carry flow is the representation of the short-term memory. Both are used to then compute the next output after taking in the input data for that next state. The simple RNN code is

y = activation(dot(state_t, U) + dot(input_t, W) + b)

but the LSTM is a series of calculations mixed into the code above. The pseudocode is

output_t = activation(dot(state_t, Uo) + dot(input_t, Wo) + dot(C_t, Vo) + bo)

how C_t is obtained is by calculating three different weight matrices i, f, and k. And how those are obtained is the bottom calculation below.

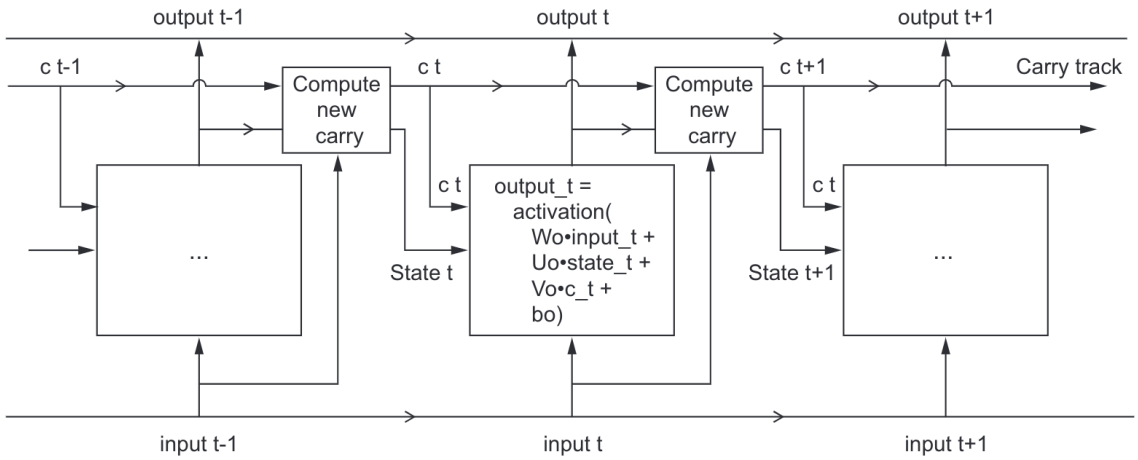
i_t = activation(dot(state_t, Ui) + dot(input_t, Wi) + bi)

f_t = activation(dot(state_t, Uf) + dot(input_t, Wf) + bf)

k_t = activation(dot(state_t, Uk) + dot(input_t, Wk) + bk)

Once the weighted matrices of i, f, and k are computed they are then combined and to give the next carry state c_t +1 like so,

c_t+1 = i_t * k_t + c_t * f_t. Multiplying c_t and f_t is represented as forgetting irrelevant information and multiplying i_t and k_t are providing present information so the c_t can be updated with new information (Chollet, 2018).



"A screenshot of the entire process of LSTM, (Chollet, 2018)"

5.5 The Plan

Since I know that the dataset is small, I already know that these models are going to bet very understatedly, but I still believe some sort of insight can be gained from this. Even if there is not credible accuracy given from training the small datasets, I want to show that with the proper amount of data, it will and should be possible to predict and forecast adoption rates.

What I want to do is compare the prediction results of the adoption rates versus the adoption rates with incentives. The data may not accurately display correct predictions, but it is my hope that this will give the audience an idea of the possibilities that could come from an appropriate dataset.

With linear regression, I am going to use multivariate regression for both predictions with the predictor features of choice Solar kWh generation of each year, and the electricity cost of each year for the first prediction; then the sister columns of those two with the incentives for the other prediction. The target

feature will obviously be the adoption rates for the first one and the adoption rated with incentives for the second. I will visualize the predictions for both.

With ARIMA it will be a univariate prediction and the target feature for the first prediction will be adoption rates and the second prediction's target feature will be adoption rates with the incentive. The orders, p will be defined as 0 for the late order, d will be defined as 1 for the order for the degree of difference, and q will be defined as 1 for the order of the moving average. I will visualize the predictions for both.

For LSTM I will do two univariate predictions one for the adoption rates feature, and another for the adoption rates with the incentives feature. I will visualize the predictions for both.

The next thing I will do is compare the results of all the predictions with their metrics included to give an interpreted understanding of which how well the models performed. I'll bear in mind that these models will be under fitted but the importance here is to show the audience the potential and give a reason for the effort to collect the necessary data for modeling in the future.

5.6 Applying Linear Regression Machine Learning

What I did to apply linear regression was load the scikit-learn library linear_model. With the linear model, I will be able to easily train the data with ease rather than coding the linear regression from scratch. After I imported the necessary libraries, I read in the data frame with python library module read_excel from the Pandas library. Once in a data frame, I defined the predictor features and the target features for both predictions. After that, I split the data up into training and testing datasets before defining the model. Then after defining the model I fitted both training datasets for both predictions and trained the model so I could run the predictions. After the predictions, I print the metrics of the model accuracy and then I plot the results on a graph so the results can be visually seen easier.

5.7 Applying ARIMA Machine Learning

For applying the ARIMA machine learning model to the dataset, I load the necessary library module ARIMA from the statsmodel library and make sure that the required modules are also installed as well. The next thing I did was reload the data into a different data frame just like I did the first time with linear regression, but this time only chose two features for both predictions and that was the years and adoption rates (adoption rates after incentives for the second prediction). I also transform the years Dtype from int64 to datetime. Then I split the data frame into training and testing before I apply the model ARIMA to the entire data frame with the order of pdq as 0,1,1. After ARIMA is applied to the entire data frame I then fit it the data and print a summary of the fitted data before running arima predict to get the results. Once the predictions are acquired, like what I did for linear regression I print the model metrics for its credibility and then I plot the results on a line graph to visually show the predictions.

5.8 Applying LSTM RNN Machine Learning

The LSTM training requires a little bit more preprocessing and data normalization than the other two. First, I have to make sure that I load the LSTM layers module from the TensorFlow Keras library, and all the other required modules in order to run the LSTM neural network model. I can still use the same dataset I used for ARIMA since I am only doing a univariate prediction of the adoption rates and the adoption rates with incentives. Next, I use the module MinMaxScaler from sci-kit learn's library to fit the

data and normalize/transform the training/testing data. The last step of preprocessing I do is I use the TimeSeriesGenerator module from the TensorFlow Keras library to transform the data into a two-dimensional tensor to pass through the neural network. Then I begin to set up the neural network with defining the environment with the Sequential module from the TensorFlow Keras library and add the LSTM layer to it with a relu activation and input data dimensions. I add a Dense layer to the end of the neural network model for the output. Then I add the ending piece to the neural network, which is the compiler with Adam optimization algorithm, mean square error as a loss function with the metrics for validation being mean absolute error. After that i will print a summary of the model before I run the model with 25 epochs (iterations of the neural network updating the weights for training) on the training data against the validation data. When the neural network is done training and validating, I then will have to plot the metrics of the loss function against the validation loss function to verify if the model is underfitting/overfitting and if it is reliable at all. I then will take the predictions and inverse transform the scaled predictions back to normal values so that I can gather the metrics of the model accuracy for the predictions and then plot them for visual understanding.

6. Machine Learning Results

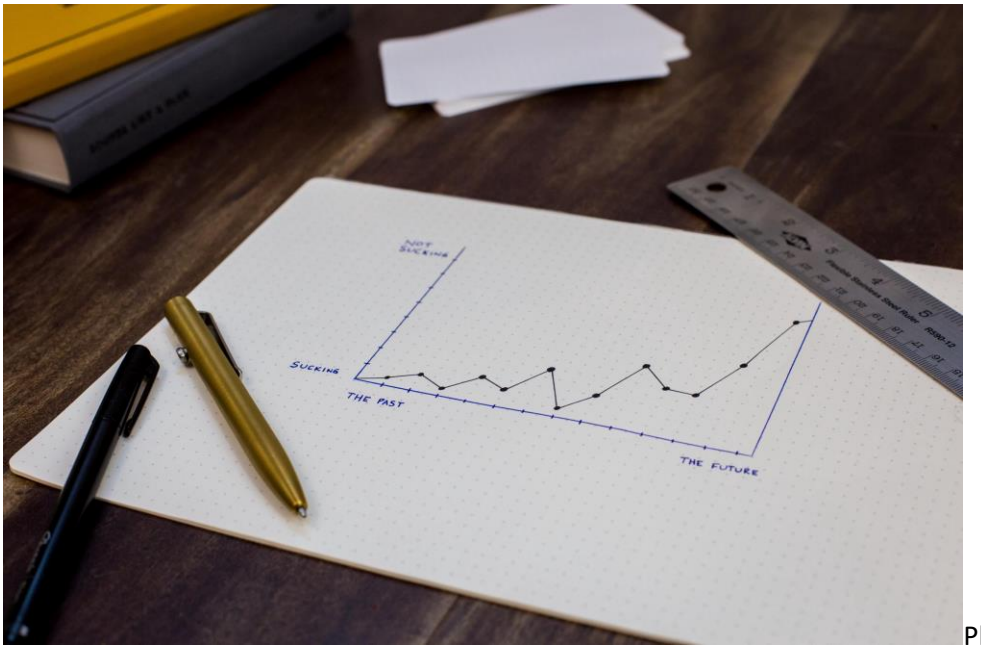


Photo by Isaac Smith on Unsplash

Going into viewing the metrics and results of the machine learning modeling is important because from here I can gain two steps. Viewing the metrics is useful to evaluate model accuracy. The other valuable step I gain from this is the storytelling and visualization of predictive accuracy. However, my only caution is that viewing these results should be taken with a grain of salt because the models may be severely under fitted.

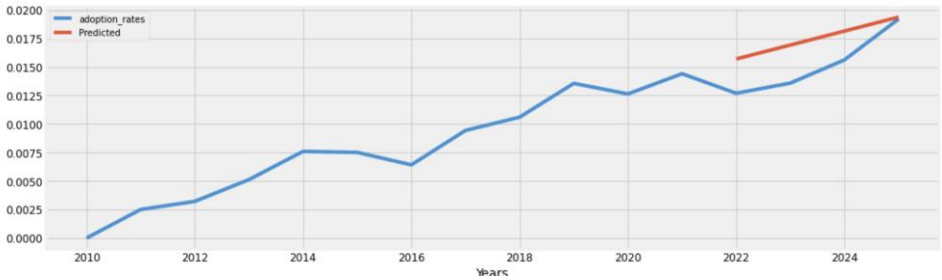
6.1 Linear Regression Machine Learning Results

The first model predicting the adoption rates without incentives came out pretty close.

Years	Actual	Predicted
2022	0.012698	0.015702
2023	0.013586	0.016925
2024	0.015624	0.018148
2025	0.019208	0.019371

“A screenshot of the actual vs predicted for the machine learning of adoption rates with linear regression”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 6.642303952692676e-06, the mean absolute error is 0.00226, the root mean squared error is 0.0026, the mean value is 0.0096, and the r squared value is -0.059. Going off of the r squared I would say right away that this model doesn’t perform accurately at all, but the mean absolute error indicates a little differently.



“A visualization of the predicted results for adoption rates with linear regression”

Years	Solar_kWh_gene	est_annual_insta	adoption_rates
2021	456071428.6	79.24287157	0.01440779
2022	495297619	69.83682165	0.01269760
2023	534523809.5	74.72539916	0.01358644
2024	573750000	85.93420904	0.01562440
2025	612976190.5	105.6429725	0.01920781

“A screenshot of the probable explanation for the linear predictions”

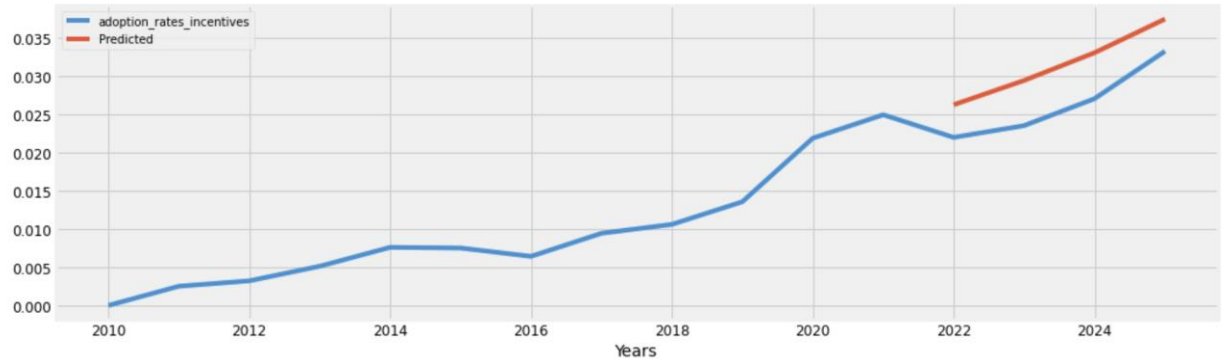
The training of the data was split right at where the year 2022 is and so that huge gap makes sense when looking at the graph and I would say that these predictions are actually fairly linear. The dip right there at 2022 is because the projections from Wood Mackenzie said that year would probably have a drop in installed solar panels. Plus, the training features both keep increasing which as well is a probable indicator of why the machine probably predicted the continual increase.

The second model with running machine learning with linear regression on the adoption rates with incentives is where things may be interesting.

Years	Actual	Predicted
2022	0.021980	0.026249
2023	0.023518	0.029428
2024	0.027046	0.033053
2025	0.033249	0.037461

“A screenshot of the actual vs the predicted for the machine learning on adoption rates with incentives with linear regression”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 2.6749494491856575e-05, the mean absolute error is 0.0051, the root mean squared error is 0.0052, the mean value is 0.012, and the r squared value is -0.424. Once again, I feel like this model isn’t a good fit because of the r squared value indicating that the model is doing worse than the mean value, but then again, the mean absolute error indicates that it isn’t bad either.



"A visualization of the predicted results for adoption rates with incentives with linear regression"

Years	Solar_kWh_gene	electric_cost	est_annual_insta	adoption_rates_incentives
2020	682455416.9	35.21715505	120.3240445	0.02187710
2021	802342677.7	35.89563067	137.1694107	0.02493989
2022	907999440.1	36.11545118	120.8875383	0.02197955
2023	1021052176	36.33527169	129.3496659	0.02351812
2024	1151062822	36.5550922	148.7521158	0.02704584
2025	1310891035	36.77491271	182.8679854	0.03324872

A screenshot of the probable explanation for the linear predictions"

I would give the same reasoning I used to explain the ending predictions of the model for adoption rates without incentives. The line is still pretty linear and even if it is off because of that dip in 2022.

Even though both predictions with linear regression machine learning are not predicting that dip, they still are predicting values in an upward trend. The 2025 predictions for the adoption rates without the incentives are pretty spot on if that dip didn't occur. Also bear in mind the predictor features were shown to have a lot of correlation when is exploring the data but aren't features that lead to the adoption rate calculations. So, the predictor features are training the machine on this consistent upward trend and 2022 is a pretty big outlier for the testing data that the machine never sees. However, I don't think I would put all my hopes in linear regression for these predictions

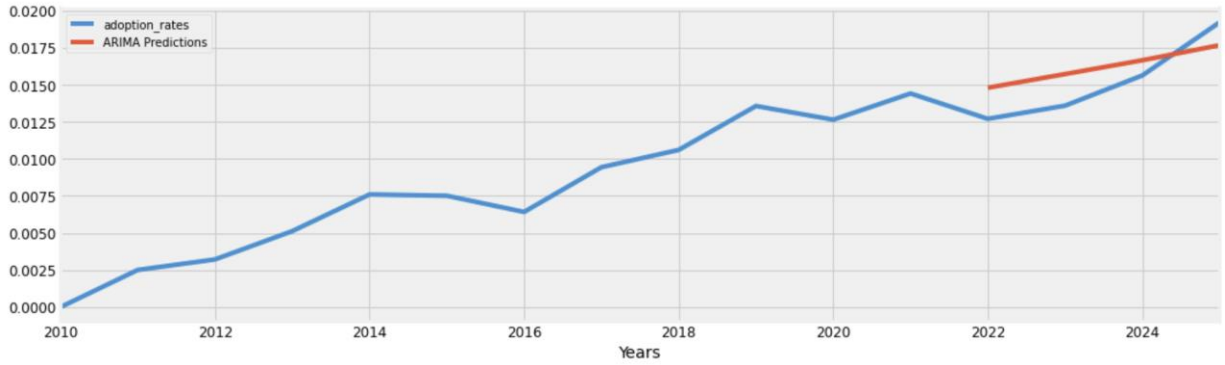
6.2 ARIMA Machine Learning Results

The results for the first set of predictions were also close.

Years	adoption_rates	ARIMA_Predictions
2022-01-01	0.012698	0.014795
2023-01-01	0.013586	0.015717
2024-01-01	0.015624	0.016648
2025-01-01	0.019208	0.017663

“A screenshot of the actual vs predicted for the machine learning of adoption rates with ARIMA”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 3.092111739895624e-06, the mean absolute error is 0.0017, the root mean squared error is 0.0018, the mean value is 0.0096, and the r squared value is 0.5069. All the metrics for evaluation of the model would indicate that it is a fairly good fit.



“A visualization of the predicted results for adoption rates with ARIMA”

As was the same issue for linear regression it probably will be the same problem for all the other models as well. The year 2022 is throwing off the predictions with that small dip because the machine that is learning the data doesn’t receive that extra time step of data.

The second predictions that ARIMA machine learning made were rather good as well.

Years	adoption_rates_incentives	ARIMA_Predictions
2022-01-01	0.021980	0.025755
2023-01-01	0.023518	0.021795
2024-01-01	0.027046	0.027163
2025-01-01	0.033249	0.029409

“A screenshot of the actual vs predicted for the machine learning of adoption rates with incentives using ARIMA”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 7.995021109432833e-06, the mean absolute error is 0.0024, the root mean squared error is 0.0028, the mean value is 0.0137, and the r squared value is 0.5745. The evaluation metrics would definitely indicate that the model is a good fit.

“A visualization of the predicted results for adoption rates with incentives with ARIMA”

The funny thing with this model prediction though is that it looks as if the model is predicting the dip in rates in 2023 rather than 2022. Even though the predictions fall under the actual for 2025 I would still say both ARIMA models are still doing a surprisingly respectable job at predicting the data.

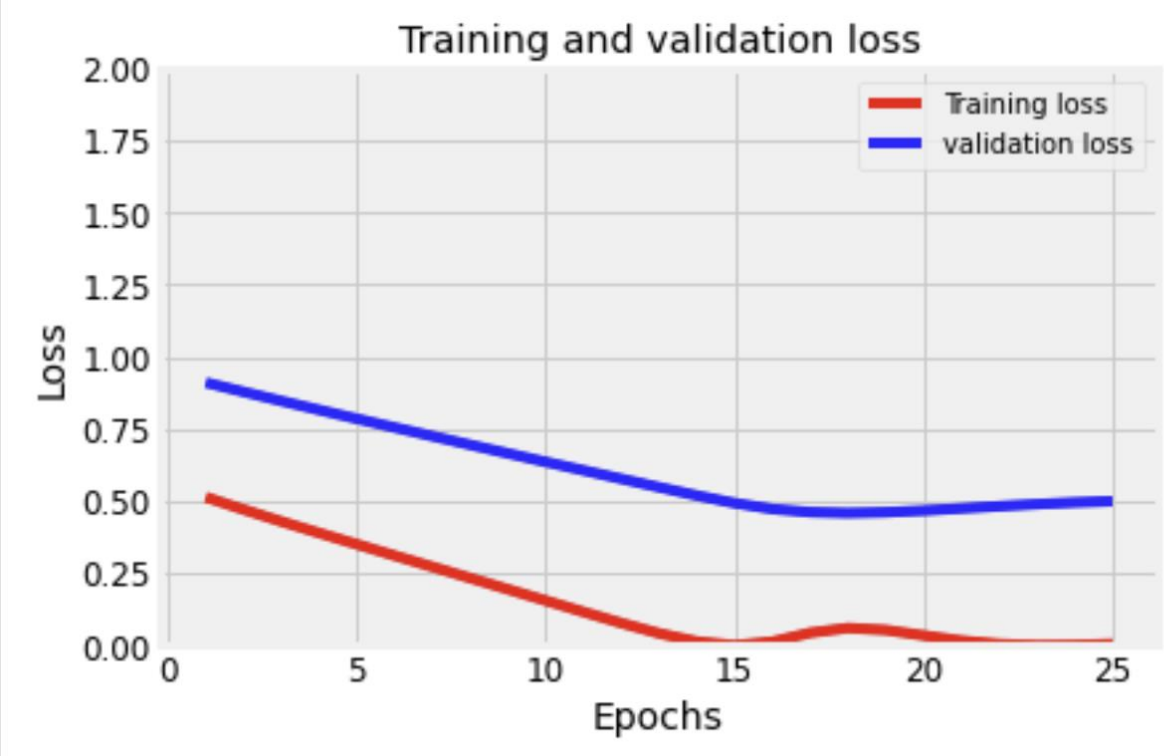
6.3 LSTM Machine Learning Results

Although I am pretty bias about neural networks, I won’t make any arguments for the results of the LSTM if they are bad. The first model predictions are fairly good.

Years	adoption_rates	LSTM_Predictions
2022-01-01	0.012698	0.014091
2023-01-01	0.013586	0.015363
2024-01-01	0.015624	0.016763
2025-01-01	0.019208	0.018158

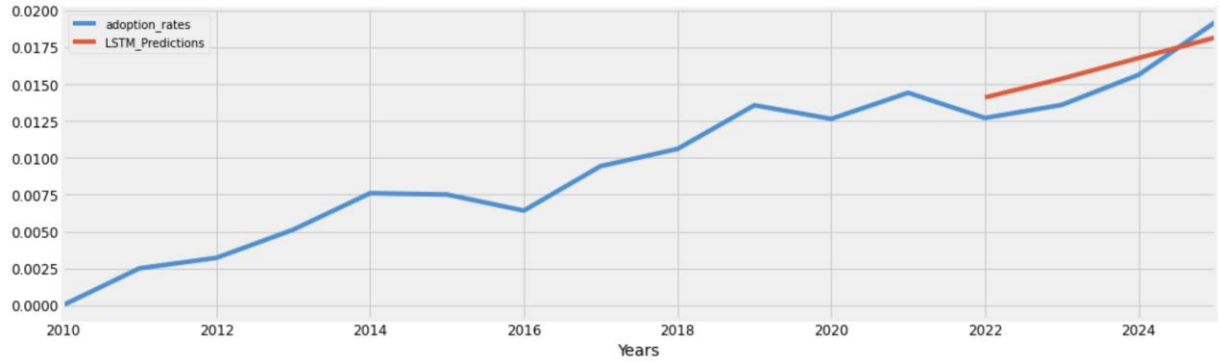
“A screenshot of the actual vs predicted for the machine learning of adoption rates using LSTM”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 1.8736872707858614e-06, the mean absolute error is 0.0013, the root mean squared error is 0.0014 the mean value is 0.0096 and the r squared value is 0.7012. All these values would indicate that the model evaluation is rather good, however, the model is pretty under fitted, just like most of them are and what I speculated from the very beginning.



"A visualization of the values of the loss function for the training data versus the validation data"

If the model were a good fit the two lines would eventually meet up and follow each other all the way until the 25th epoch, but they do not. However, this is what the predictions look like.



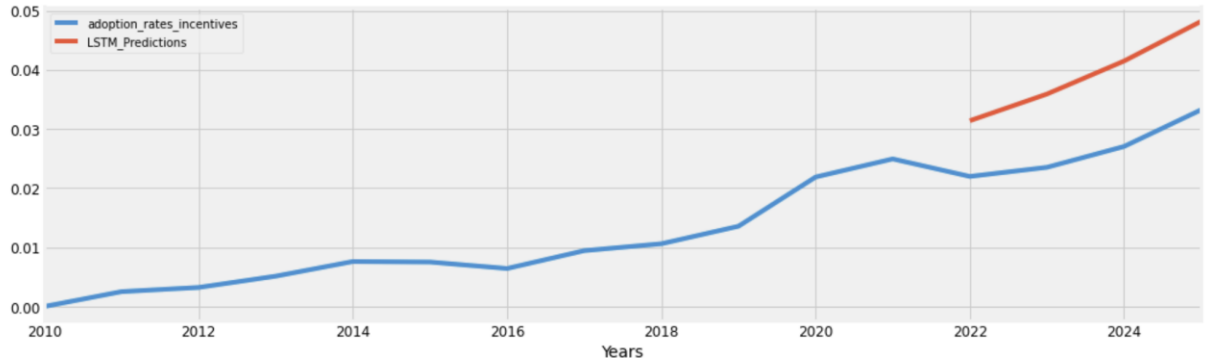
"A visualization of the predicted results for adoption rates with LSTM"

All the models may be underfitted, but they are doing a surprisingly decent job with predicting given the fact that the dataset is small. The second model prediction of LSTM on the adoption rate incentives, however, is not that good.

Years	adoption_rates_incentives	LSTM_Predictions
2022-01-01	0.021980	0.031398
2023-01-01	0.023518	0.035907
2024-01-01	0.027046	0.041485
2025-01-01	0.033249	0.048200

“A screenshot of the actual vs predicted for the machine learning of adoption rates with incentives using LSTM”

The metrics for evaluating the model are the mean squared error, the mean absolute error, the root mean squared error, the mean value, and the r squared. The mean square error is 0.00016855723752781803, the mean absolute error is 0.0128, the root mean squared error is 0.013 the mean value is 0.0137 and the r squared value is -7.9708. The evaluation of this model does not look so good in regard to the r squared value, but this could be because of the dip in 2022.

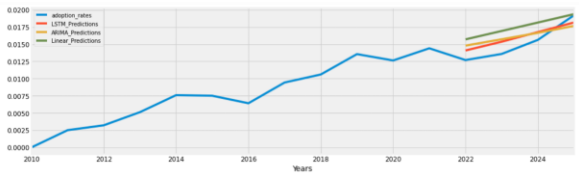


“A screenshot of the actual vs predicted for the machine learning of adoption rates with incentives using LSTM”

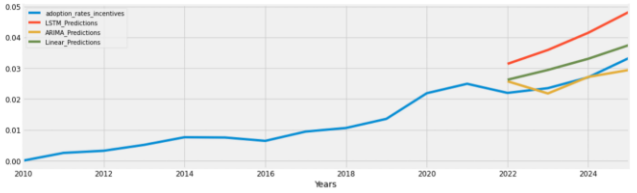
The way the predictions are trending upwards would be reason to believe that the neural network was predicting for rates that probably would have been the trend. The fact that there is not enough data for any of these models to really train for credible accuracy at all makes it a wonder how some of them did pretty well.

6.4 Comparing All Models

Years	adoption_rates	LSTM_Predictions	Linear_Predictions	ARIMA_Predictions
2022-01-01	0.012698	0.014091	0.015702	0.014795
2023-01-01	0.013586	0.015363	0.016925	0.015717
2024-01-01	0.015624	0.016763	0.018148	0.016648
2025-01-01	0.019208	0.018158	0.019371	0.017663



Years	adoption_rates_incentives	LSTM_Predictions	Linear_Predictions	ARIMA_Predictions
2022-01-01	0.021980	0.031398	0.026249	0.025755
2023-01-01	0.023518	0.035907	0.029428	0.021795
2024-01-01	0.027046	0.041485	0.033053	0.027163
2025-01-01	0.033249	0.048200	0.037461	0.029409



"Adoption rates vs all model predictions"

The one thing to keep in mind is all these models are extremely underfitted. However, for the most part, all the models predicting the values for adoption rates before the incentives did fairly well and I would say that the best performing one given the evaluation metrics would be the LSTM model. But the model that kept consistent with predictions over the years was the ARIMA modeling. I would say that the dip for the adoption rate incentives from the ARIMA model is probably predicted because of the dip that occurred back from 2014 to 2016, but it is still something to be confused about. Overall, the use of any of these algorithms with more data are an excellent choice in my opinion. I would lean more toward LSTM and linear regression, but ARIMA may possibly be the one to turn towards when there is not enough data.

If I was to rerun these model again though with a 90/10 split though values like r squared wouldn't be in the negative and probably point to .7 but the reason for using splits that don't give a smaller sample size for testing is because the objective is to train the machine to predict values over a greater length of time and not a short amount of time.

7. Conclusion and Proposal

7.1 Disposition

After reviewing the evaluation of the models and seeing their predictions my conclusion is that there needs to be more data. The data that is needed cannot underfit or overfit the models in order to get a pretty credible prediction. The data explored for solar photovoltaic generation on a small-scale in the residential sector suggests that there were houses out there that were generating solar energy all the back to 1989.

7.2 Solution

Someone might say "Well that would still be too small of a dataset, 1989 until now would only be 31 samples of installed capacity of MWdc data". My solution is simple, data does not have to be annual, it can be monthly. 31 Samples turned into 372 samples of monthly reports of how many household installations are enough data to train a model to predict adoption rates accurately to where there is not over or underfitting.

7.3 Proposal

How to acquire this data may be the only tough part. Wood Mackenzie has data on annual installation capacity in which they may have more data on it monthly too. If Wood Mackenzie did have monthly

data for installed capacity and data on the number of households as well, then those two features are the only data that are not currently public that would be required to piece with together a new dataset based on monthly values. Of course, the other data in the dataset from solar generation to electric cost, and the dollar per watt will have to be converted to monthly as well if we want to go for a multivariate prediction.

Also, there will need to be a larger sample size and probably an improved quality of collection on the number of households selected at random for solar energy data. For the random household data, I would recommend an anonymous survey Asking questions like “Are you interested in installing solar panels on top of your house if it meant you could save money on your electric bill?.” Once we get the yes and nos from that then the survey would ask “Can you afford to install one right now?.” After that question, the survey will ask the final question “If there were $2000 state tax incentives in Colorado to install solar panels on your house, would you consider doing so?”

The survey would be all voluntary and those who participate will have their information kept private. The incentive to also get more people to participate maybe actually providing them with information from project sunroof by Google on what the estimated figures for the number of years on system payback would look like for them, and then with the new state tax incentives providing more financial information on what the new estimated number of years on system payback would look like. With the results from the survey, this could give a more accurate idea of what a percentage increase may look like and can recalculate that new value back into the dataset.

Most importantly the data that is absolutely necessary though is the total number of households with solar panels already installed and the installation capacity MWdc by month. With that data and new sample size from survey results, I think that there would be enough data to train the models, make accurate predictions on the testing data, and then forecast what the adoption rates will look like in the future.

References

About EIA - U.S. Energy Information Administration (EIA) - U.S. Energy Information Administration (EIA). (2016). Retrieved May 19, 2020, from Eia.gov website: https://www.eia.gov/about/

About NREL. (2019). Retrieved June 9, 2020, from Nrel.gov website: https://www.nrel.gov/about/

blickpixel. (2014, December 15). Lego Legomaennchen Males - Free photo on Pixabay. Retrieved June 23, 2020, from Pixabay.com website: https://pixabay.com/photos/lego-legomaennchen-males-workers-568039/

Chollet, F. (2018). *Deep Learning with Python*. Shelter Island (New York, Estados Unidos): Manning, Cop.

CMIP6 Data | Climate and Energy College. (2017a). Retrieved May 19, 2020, from Unimelb.edu.au website: http://climatecollege.unimelb.edu.au/cmip6

ColiN00B. (2018, January 8). Graph Chart Growth - Free image on Pixabay. Retrieved June 2, 2020, from Pixabay.com website: https://pixabay.com/illustrations/graph-chart-growth-report-analyst-3068300/

Colorado - SEDS - U.S. Energy Information Administration (EIA). (2016). Retrieved May 19, 2020, from Eia.gov website: https://www.eia.gov/state/seds/seds-data-complete.php?sid=CO

Colorado State Demography Office. (2019a). County Data Lookup. Retrieved May 19, 2020, from Colorado Demography website: https://demography.dola.colorado.gov/population/data/county-data-lookup/

COLORADO STATE DEMOGRAPHY OFFICE. (2020).

FDIC: Weekly National Rates and Rate Caps - Weekly Update. (2020). Retrieved June 17, 2020, from Fdic.gov website: https://www.fdic.gov/regulations/resources/rates/

Federal Reserve Board - Survey of Consumer Finances (SCF). (2016). Retrieved June 17, 2020, from Board of Governors of the Federal Reserve System website: https://www.federalreserve.gov/econres/scfindex.htm

Fu, R., Feldman, D., & Margolis, R. (2018). *U.S. Solar Photovoltaic System Cost Benchmark: Q1 2018*. Retrieved from https://www.nrel.gov/docs/fy19osti/72399.pdf

hangela. (2016, August 30). Coal Black Mineral - Free photo on Pixabay. Retrieved May 19, 2020, from Pixabay.com website: https://pixabay.com/photos/coal-black-mineral-underground-1626368/

Homepage - U.S. Energy Information Administration (EIA). (2016). Retrieved May 19, 2020, from Eia.gov website: https://www.eia.gov/

Linear Regression — ML Glossary documentation. (2014). Retrieved June 19, 2020, from Readthedocs.io website: https://ml-cheatsheet.readthedocs.io/en/latest/linear_regression.html

Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., Fraser, P. J., Montzka, S. A., Rayner, P. J., Trudinger, C. M., Krummel, P. B., Beyerle, U., Canadell, J. G., Daniel, J. S., Enting, I. G., Law, R. M., Lunder, C. R., O'Doherty, S., Prinn, R. G., Reimann, S., Rubino, M., Velders, G. J. M., Vollmer, M. K., Wang, R. H. J., and Weiss, R.: Historical greenhouse gas concentrations for climate modelling (CMIP6), Geosci. Model Dev., 10, 2057-2116, 2017.

Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., … Weiss, R. (2017a). Historical greenhouse gas concentrations for climate modelling (CMIP6). *Geoscientific Model Development*, *10*(5), 2057–2116. https://doi.org/10.5194/gmd-10-2057-2017

mohamed_hassan. (2019, March 26). Artificial Intelligence Machine - Free image on Pixabay. Retrieved June 16, 2020, from Pixabay.com website: https://pixabay.com/illustrations/artificial-intelligence-machine-4082314/

Moon, C. (2016, November 3). Average U.S. Savings Account Balance 2019: A Demographic Breakdown. Retrieved June 12, 2020, from ValuePenguin website: https://www.valuepenguin.com/banking/average-savings-account-balance

Personal Income in Colorado | Colorado Information Marketplace | data.colorado.gov. (2018a). Retrieved June 7, 2020, from Colorado Information Marketplace website: https://data.colorado.gov/Labor-and-Employment/Personal-Income-in-Colorado/2cpa-vbur/data

Prabhakaran, S. (2019, February 18). ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+. Retrieved June 20, 2020, from Machine Learning Plus website: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

Project Sunroof. (2020a). Retrieved May 21, 2020, from Google.com website: https://www.google.com/get/sunroof

PVWatts Calculator. (2020). Retrieved June 9, 2020, from Nrel.gov website: https://pvwatts.nrel.gov/

Real Estate, Homes for Sale, MLS Listings, Agents | Redfin. (2020a). Retrieved June 5, 2020, from Redfin.com website: https://www.redfin.com/

Schwarzenberger, M. (2014, December 15). Lego Legomaennchen Males - Free photo on Pixabay. Retrieved May 26, 2020, from Pixabay.com website: https://pixabay.com/photos/lego-legomaennchen-males-workers-568039/

Smith, I. (2018, November 26). pen om paper. Retrieved June 21, 2020, from Unsplash.com website: https://unsplash.com/photos/AT77Q0Njnt0

Smith, M. (2015, January 20). selective photography of green leaf plant. Retrieved June 23, 2020, from Unsplash.com website: https://unsplash.com/photos/Rfflri94rs8

State Carbon Dioxide Emissions Data - U.S. Energy Information Administration (EIA). (2016a). Retrieved June 2, 2020, from Eia.gov website: https://www.eia.gov/environment/emissions/state/

State Carbon Dioxide Emissions Data - U.S. Energy Information Administration (EIA). (2016b). Retrieved June 2, 2020, from Eia.gov website: https://www.eia.gov/environment/emissions/state/

Sunrun. (2019, June 27). Colorado Solar Incentives. Retrieved June 23, 2020, from Sunrun website: https://www.sunrun.com/solar-by-state/co/colorado-solar-tax-incentives

The University of Melbourne, Australia - Australia's best university and one of the world's finest. (2019, January 7). Retrieved May 19, 2020, from The University of Melbourne website: https://www.unimelb.edu.au/

Total Savings Deposits at all Depository Institutions. (2020). Retrieved June 17, 2020, from Stlouisfed.org website: https://fred.stlouisfed.org/series/SAVINGS

U.S. Bureau of Economic Analysis (BEA). (2020). Retrieved June 9, 2020, from Bea.gov website: https://www.bea.gov/

U.S. Census Buerau. (2020). Retrieved May 20, 2020, from Census.gov website: https://data.census.gov/cedsci/table?q=Housing%20Units&hidePreview=false&t=Housing%20Units&tid=ACSDP1Y2017.DP04&g=0400000US01,02,04,05,06,08,09,10,12,13,15,17,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,44,45,46,47,48,49,50,51,53,54,55,56&tp=true

US Census Bureau. (2017, September). Income and Poverty in the United States: 2016. Retrieved June 17, 2020, from The United States Census Bureau website: https://www.census.gov/data/tables/2017/demo/income-poverty/p60-259.html

US EPA, OAR. (2016, June 27). Climate Change Indicators: Atmospheric Concentrations of Greenhouse Gases | US EPA. Retrieved May 23, 2020, from US EPA website: https://www.epa.gov/climate-indicators/climate-change-indicators-atmospheric-concentrations-greenhouse-gases

Wood Mackenzie. (2017, June 23). Retrieved June 17, 2020, from Woodmac.com website: https://www.woodmac.com/research/products/power-and-renewables/

Zip-Codes.com. (2010a). Retrieved May 27, 2020, from Zip-codes.com website: https://www.zip-codes.com/