A Study of Factors Affecting Heart Disease Mortality Rate in the United States

Ashraful Islam, Nicholas Luczak, Saswata Paul, and Yiyun Su
Rensselaer Polytechnic Institute, Troy, New York, 12180
{islama6, luczan, pauls4, suy4}@rpi.edu

Abstract

Coronary heart disease has become one of the major complications that ail the American population. So, in this paper, we investigate the factors that may be responsible for coronary heart disease in the United States. We perform two types of analysis - an analysis of coronary heart disease and median household income for New York State and an analysis of coronary heart disease and social determinants for the entire United States. We obtain public domain data from www.cdc.gov and www.data.gov to perform our analysis. Our preliminary analysis shows interesting patterns between coronary heart disease mortality and social factors. We then train a machine learning model to see if it is possible to correctly predict coronary heart disease from the various factors.

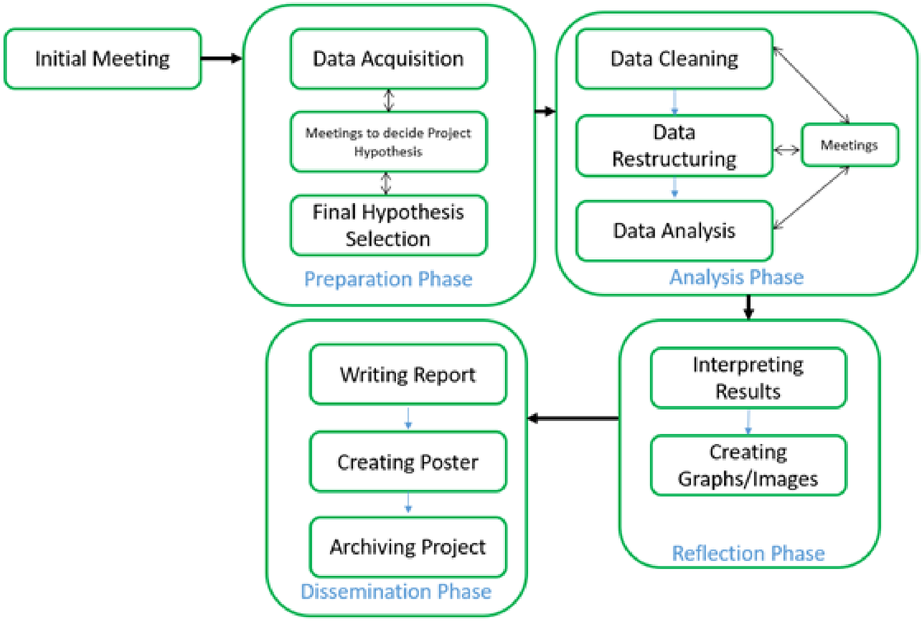
I. Choosing the Investigation and Identifying a Pre-Existing Source of Data

A. *The Goal and Reasons behind Choice of Datasets, How They Were Found and Managed*

We wanted to find datasets that were collected between the same time period, so that the analysis would be accurate. We searched for suitable datasets in data.gov and cdc.gov and came across the heart disease mortality dataset. We realized that it was a rich dataset since it contained detailed information about heart disease mortality rate by county for the entire United States. Then we decided that we wanted to do an analysis of how heart disease mortality is related to different social determinants. We found two more rich datasets, the first of which contained information about median income for the state of New York and the other contained information about social determinants for the entire United States.

We stored and managed the datasets in a Github repository that we had created for the project. All other materials related to the project were also stored there.

Fig. 1: Project workflow



Given below is a description of the various stages of our project work:

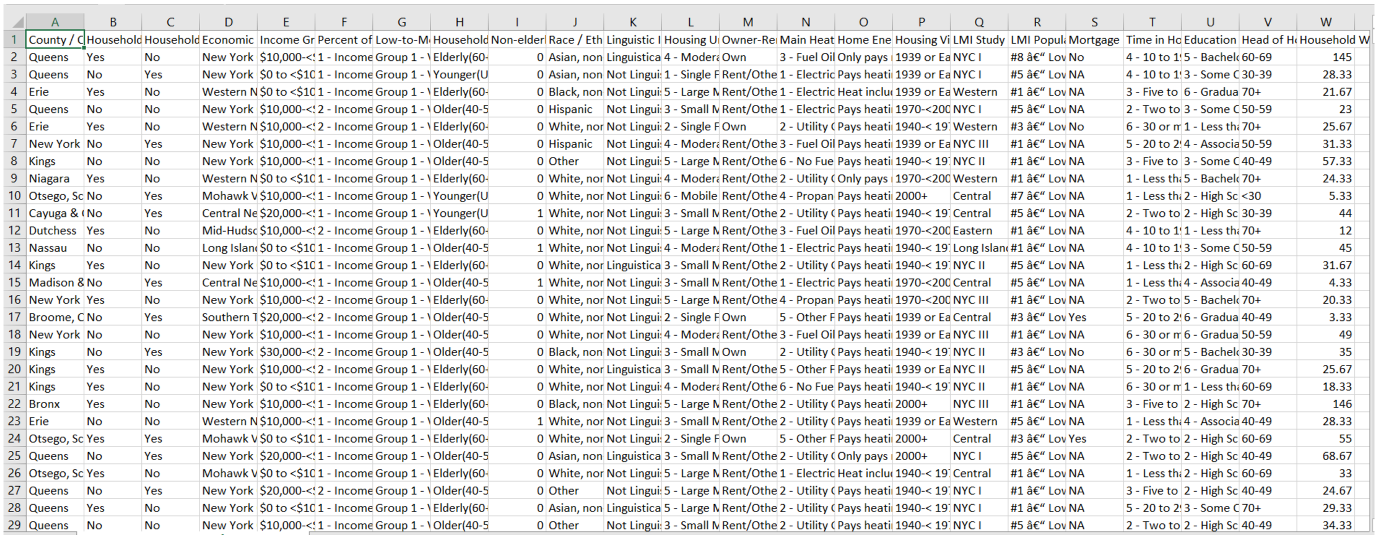
- **Preparation Phase:** In this phase, we downloaded several datasets from data.gov and cdc.gov. We looked at different types of datasets before deciding what the hypothesis of our project would be. We finalized a hypothesis after rejecting several options.
- **Analysis Phase:** In this phase, we cleaned the data and preprocessed it so that we could analyze it. For the median income dataset, there were multiple values for each county in the dataset, so we took an average of all the values for each county. We also had to make sure that the county names in the median income dataset were written exactly the same as in the heart disease dataset so that we could easily compare them using Excel. Moreover, there were erroneous and missing values that needed to be filtered.
- **Reflection Phase:** In this phase, we generated visual representations to support our hypothesis. We generated graphs, heatmaps and geo-distribution maps that corroborate our findings from the data. This helped us articulate and express our results in a way that is easily understandable by others.
- **Dissemination Phase:** In this phase, we aim to create the final report and the poster. We will also properly archive our project in the Github repository of the course. Data preservation will be handled by data.gov as the data was collected directly from that website. We will be providing copies of the datasets along with metadata and provenance information, but the most well documented source will still be the data.gov websites from where the data was originally collected.

B. *Data Formats and Metadata Standards*

A detailed description about the datasets is presented below.

1) **NYSERDA Low- to Moderate-Income New York State Census Population Analysis Dataset: Average for 2013-2015**
   - Collected from – catalog.data.gov
   - Type of file – comma separated values
   - Publisher - data.ny.gov
   - Maintainer – NY Open data
   - Maintainer email - openny@nyserda.ny.gov
   - Unique Identifier - https://data.ny.gov/api/views/bui8-bb6g
   - Metadata updated date – Nov 21, 2019
   - Metadata created date – March 28, 2018

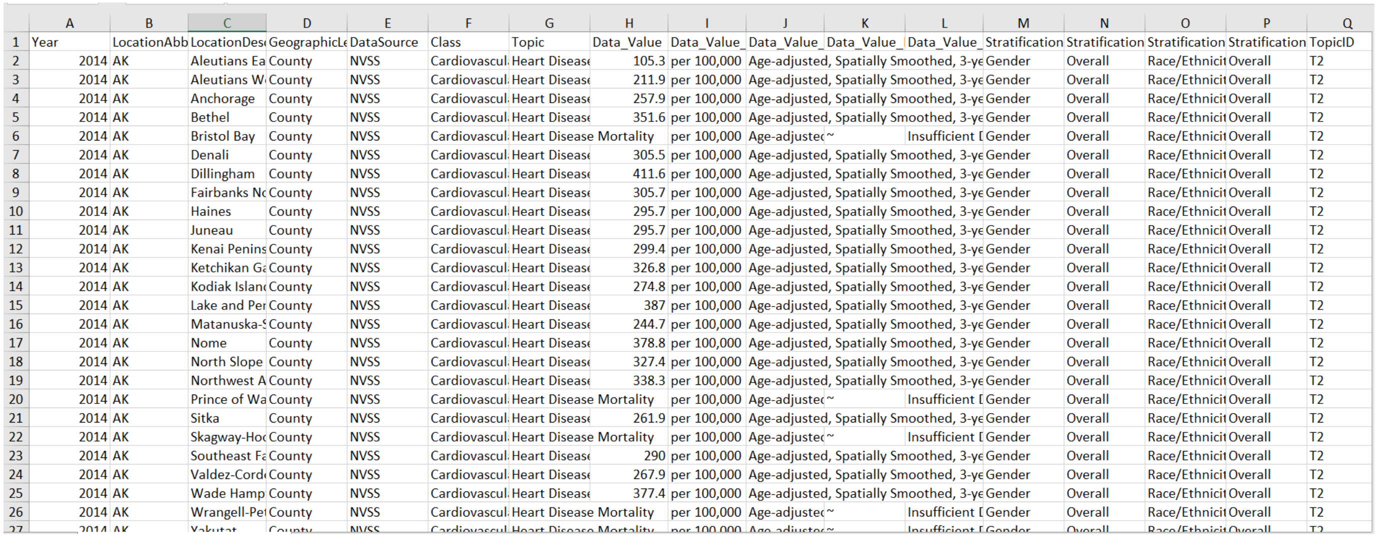
Fig. 2: A snippet of the NYSERDA Low- to Moderate-Income New York State Census Population Analysis Dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	County / C	Household	Household	Economic	Income Gr	Percent of	Low-to-M	Household	Non-elderl	Race / Eth	Linguistic I	Housing U	Owner-Re	Main Heat	Home Ene	Housing Vi	LMI Study	LMI Popula	Mortgage	Time in Ho	Education	Head of H	Household W
2	Queens	Yes	No	New York	$10,000-<$	1 - Income	Group 1 - V	Elderly(60-	0	Asian, non	Linguistica	4 - Modera	Own	3 - Fuel Oil	Only pays	1939 or Ea	NYC I	#8 â€“ Low	No	4 - 10 to 1	5 - Bachelo	60-69	145
3	Queens	No	Yes	New York	$0 to <$10	1 - Income	Group 1 - V	Younger(U	0	Asian, non	Not Lingui	1 - Single F	Rent/Othe	1 - Electric	Pays heati	1939 or Ea	NYC I	#5 â€“ Low	NA	4 - 10 to 1	3 - Some C	30-39	28.33
4	Erie	Yes	No	Western N	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	Black, non	Not Lingui	5 - Large M	Rent/Othe	1 - Electric	Heat inclu	1939 or Ea	Western	#1 â€“ Low	NA	3 - Five to	6 - Gradua	70+	21.67
5	Queens	No	No	New York	$10,000-<$	2 - Income	Group 1 - V	Older(40-5	0	Hispanic	Not Lingui	3 - Small M	Rent/Othe	1 - Electric	Pays heati	1970-<200	NYC I	#5 â€“ Low	NA	2 - Two to	3 - Some C	50-59	23
6	Erie	Yes	No	Western N	$10,000-<$	2 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	2 - Single F	Own	2 - Utility (	Pays heati	1940-< 19	Western	#3 â€“ Low	No	6 - 30 or m	1 - Less tha	70+	25.67
7	New York	No	Yes	New York	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	0	Hispanic	Not Lingui	4 - Modera	Rent/Othe	3 - Fuel Oil	Pays heati	1939 or Ea	NYC III	#1 â€“ Low	NA	5 - 20 to 2	4 - Associa	50-59	31.33
8	Kings	No	No	New York	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	0	Other	Not Lingui	5 - Large M	Rent/Othe	6 - No Fue	Pays heati	1940-< 19	NYC II	#1 â€“ Low	NA	3 - Five to	3 - Some C	40-49	57.33
9	Niagara	Yes	No	Western N	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	4 - Modera	Rent/Othe	2 - Utility (	Only pays	1970-<200	Western	#1 â€“ Low	NA	1 - Less tha	5 - Bachelo	70+	24.33
10	Otsego, Sc	No	Yes	Mohawk V	$10,000-<$	1 - Income	Group 1 - V	Younger(U	0	White, non	Not Lingui	6 - Mobile	Rent/Othe	4 - Propan	Pays heati	2000+	Central	#7 â€“ Low	NA	1 - Less tha	2 - High Sc	<30	5.33
11	Cayuga & (	No	Yes	Central Ne	$20,000-<$	1 - Income	Group 1 - V	Younger(U	1	White, non	Not Lingui	3 - Small M	Rent/Othe	2 - Utility (	Pays heati	1940-< 19	Central	#5 â€“ Low	NA	2 - Two to	2 - High Sc	30-39	44
12	Dutchess	Yes	No	Mid-Hudso	$10,000-<$	2 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	5 - Large M	Rent/Othe	3 - Fuel Oil	Pays heati	1970-<200	Eastern	#1 â€“ Low	NA	4 - 10 to 1	1 - Less tha	70+	12
13	Nassau	No	No	Long Islan	$0 to <$10	1 - Income	Group 1 - V	Older(40-5	1	White, non	Not Lingui	4 - Modera	Rent/Othe	1 - Electric	Pays heati	1940-< 19	Long Islan	#1 â€“ Low	NA	4 - 10 to 1	3 - Some C	50-59	45
14	Kings	Yes	No	New York	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	White, non	Linguistica	3 - Small M	Rent/Othe	2 - Utility (	Pays heati	1940-< 19	NYC II	#5 â€“ Low	NA	1 - Less tha	2 - High Sc	60-69	31.67
15	Madison &	No	Yes	Central Ne	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	1	White, non	Not Lingui	3 - Small M	Rent/Othe	1 - Electric	Pays heati	1970-<200	Central	#5 â€“ Low	NA	1 - Less tha	4 - Associa	40-49	4.33
16	New York	Yes	No	New York	$10,000-<$	2 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	5 - Large M	Rent/Othe	4 - Propan	Pays heati	1970-<200	NYC III	#1 â€“ Low	NA	2 - Two to	5 - Bachelo	70+	20.33
17	Broome, C	No	Yes	Southern T	$20,000-<$	2 - Income	Group 1 - V	Older(40-5	0	White, non	Not Lingui	2 - Single F	Own	5 - Other F	Pays heati	1939 or Ea	Central	#3 â€“ Low	Yes	5 - 20 to 2	6 - Gradua	40-49	3.33
18	New York	No	No	New York	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	0	White, non	Not Lingui	4 - Modera	Rent/Othe	3 - Fuel Oil	Pays heati	1939 or Ea	NYC III	#1 â€“ Low	NA	6 - 30 or m	6 - Gradua	50-59	49
19	Kings	No	Yes	New York	$30,000-<$	2 - Income	Group 1 - V	Older(40-5	0	Black, non	Not Lingui	3 - Small M	Own	2 - Utility (	Pays heati	1940-< 19	NYC II	#3 â€“ Low	No	6 - 30 or m	5 - Bachelo	30-39	35
20	Kings	Yes	No	New York	$10,000-<$	2 - Income	Group 1 - V	Elderly(60-	0	White, non	Linguistica	3 - Small M	Rent/Othe	5 - Other F	Pays heati	1939 or Ea	NYC II	#5 â€“ Low	NA	5 - 20 to 2	6 - Gradua	70+	25.67
21	Kings	Yes	No	New York	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	4 - Modera	Rent/Othe	6 - No Fue	Pays heati	1940-< 19	NYC II	#1 â€“ Low	NA	6 - 30 or m	1 - Less tha	60-69	18.33
22	Bronx	Yes	No	New York	$10,000-<$	1 - Income	Group 1 - V	Elderly(60-	0	Black, non	Not Lingui	5 - Large M	Rent/Othe	2 - Utility (	Pays heati	2000+	NYC III	#1 â€“ Low	NA	3 - Five to	2 - High Sc	70+	146
23	Erie	No	No	Western N	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	1	White, non	Not Lingui	3 - Small M	Rent/Othe	2 - Utility (	Pays heati	1939 or Ea	Western	#5 â€“ Low	NA	1 - Less tha	4 - Associa	40-49	28.33
24	Otsego, Sc	Yes	Yes	Mohawk V	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	2 - Single F	Own	5 - Other F	Pays heati	2000+	Central	#3 â€“ Low	Yes	2 - Two to	2 - High Sc	60-69	55
25	Queens	No	Yes	New York	$20,000-<$	1 - Income	Group 1 - V	Older(40-5	0	Asian, non	Linguistica	3 - Small M	Rent/Othe	2 - Utility (	Only pays	2000+	NYC I	#5 â€“ Low	NA	2 - Two to	2 - High Sc	40-49	68.67
26	Otsego, Sc	Yes	No	Mohawk V	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	White, non	Not Lingui	5 - Large M	Rent/Othe	1 - Electric	Heat inclu	1940-< 19	Central	#1 â€“ Low	NA	1 - Less tha	2 - High Sc	60-69	33
27	Queens	No	Yes	New York	$20,000-<$	2 - Income	Group 1 - V	Older(40-5	0	Other	Not Lingui	5 - Large M	Rent/Othe	2 - Utility (	Pays heati	1940-< 19	NYC I	#1 â€“ Low	NA	3 - Five to	2 - High Sc	40-49	24.67
28	Queens	Yes	No	New York	$0 to <$10	1 - Income	Group 1 - V	Elderly(60-	0	Asian, non	Linguistica	5 - Large M	Rent/Othe	2 - Utility (	Pays heati	1940-< 19	NYC I	#1 â€“ Low	NA	5 - 20 to 2	3 - Some C	70+	29.33
29	Queens	No	No	New York	$10,000-<$	1 - Income	Group 1 - V	Older(40-5	0	Other	Not Lingui	3 - Small M	Rent/Othe	2 - Utility (	Pays heati	1940-< 19	NYC I	#5 â€“ Low	NA	2 - Two to	2 - High Sc	40-49	34.33

2) **Heart Disease Mortality Data Among US Adults (35+) by State/Territory and County**
   - Collected from – catalog.data.gov
   - Type of file – comma separated values
   - Publisher - Centers for Disease Control and Prevention
   - Maintainer – DHDSP Requests
   - Maintainer email - dhdsprequests@cdc.gov
   - Unique Identifier - https://data.cdc.gov/api/views/i2vk-mgdh
   - Metadata updated date – June 11, 2019
   - Metadata created date – September 2, 2019

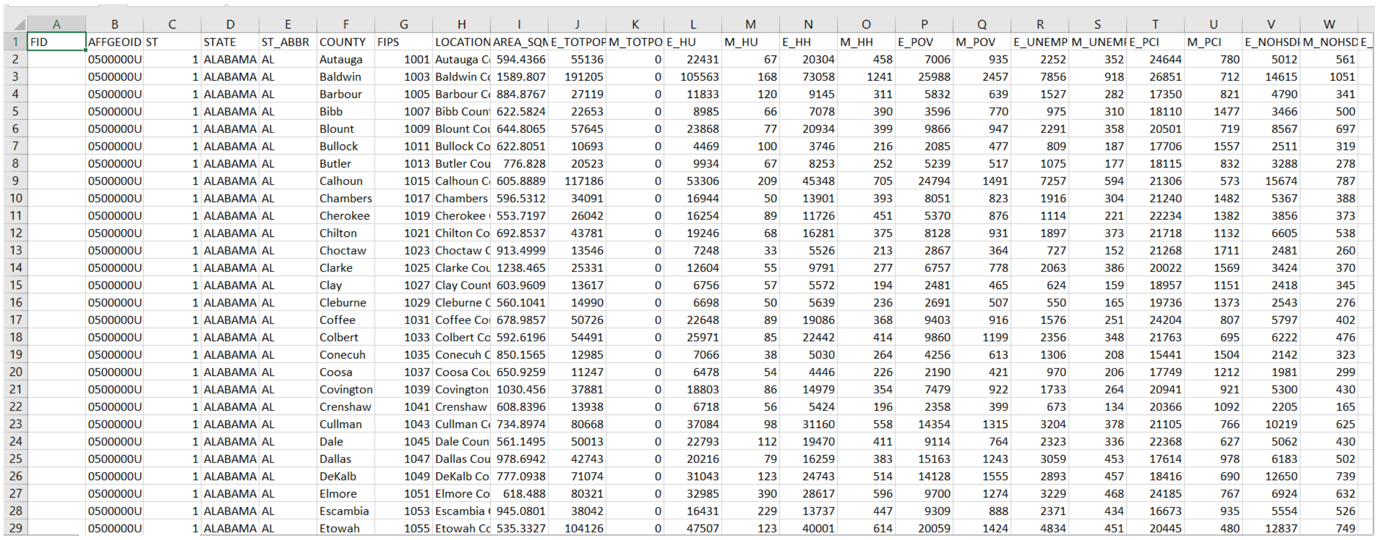
Fig. 3: A snippet of the Heart Disease Mortality Data Among US Adults Dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Year	LocationAbb	LocationDes	GeographicLe	DataSource	Class	Topic	Data_Value	Data_Value_	Data_Value_	Data_Value_	Data_Value_	Stratification	Stratification	Stratification	Stratification	TopicID
2	2014	AK	Aleutians Ea	County	NVSS	Cardiovascul	Heart Disease	105.3	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
3	2014	AK	Aleutians W	County	NVSS	Cardiovascul	Heart Disease	211.9	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
4	2014	AK	Anchorage	County	NVSS	Cardiovascul	Heart Disease	257.9	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
5	2014	AK	Bethel	County	NVSS	Cardiovascul	Heart Disease	351.6	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
6	2014	AK	Bristol Bay	County	NVSS	Cardiovascul	Heart Disease Mortality		per 100,000	Age-adjustec	~	Insufficient I	Gender	Overall	Race/Ethnicit	Overall	T2
7	2014	AK	Denali	County	NVSS	Cardiovascul	Heart Disease	305.5	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
8	2014	AK	Dillingham	County	NVSS	Cardiovascul	Heart Disease	411.6	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
9	2014	AK	Fairbanks Nc	County	NVSS	Cardiovascul	Heart Disease	305.7	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
10	2014	AK	Haines	County	NVSS	Cardiovascul	Heart Disease	295.7	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
11	2014	AK	Juneau	County	NVSS	Cardiovascul	Heart Disease	295.7	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
12	2014	AK	Kenai Penins	County	NVSS	Cardiovascul	Heart Disease	299.4	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
13	2014	AK	Ketchikan Ga	County	NVSS	Cardiovascul	Heart Disease	326.8	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
14	2014	AK	Kodiak Island	County	NVSS	Cardiovascul	Heart Disease	274.8	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
15	2014	AK	Lake and Per	County	NVSS	Cardiovascul	Heart Disease	387	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
16	2014	AK	Matanuska-S	County	NVSS	Cardiovascul	Heart Disease	244.7	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
17	2014	AK	Nome	County	NVSS	Cardiovascul	Heart Disease	378.8	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
18	2014	AK	North Slope	County	NVSS	Cardiovascul	Heart Disease	327.4	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
19	2014	AK	Northwest A	County	NVSS	Cardiovascul	Heart Disease	338.3	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
20	2014	AK	Prince of Wa	County	NVSS	Cardiovascul	Heart Disease Mortality		per 100,000	Age-adjustec	~	Insufficient I	Gender	Overall	Race/Ethnicit	Overall	T2
21	2014	AK	Sitka	County	NVSS	Cardiovascul	Heart Disease	261.9	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
22	2014	AK	Skagway-Hoc	County	NVSS	Cardiovascul	Heart Disease Mortality		per 100,000	Age-adjustec	~	Insufficient I	Gender	Overall	Race/Ethnicit	Overall	T2
23	2014	AK	Southeast Fa	County	NVSS	Cardiovascul	Heart Disease	290	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
24	2014	AK	Valdez-Cord	County	NVSS	Cardiovascul	Heart Disease	267.9	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
25	2014	AK	Wade Hamp	County	NVSS	Cardiovascul	Heart Disease	377.4	per 100,000	Age-adjusted, Spatially Smoothed, 3-ye			Gender	Overall	Race/Ethnicit	Overall	T2
26	2014	AK	Wrangell-Pet	County	NVSS	Cardiovascul	Heart Disease Mortality		per 100,000	Age-adjustec	~	Insufficient I	Gender	Overall	Race/Ethnicit	Overall	T2
27	2014	AK	Yakutat	County	NVSS	Cardiovascul	Heart Disease Mortality		per 100,000	Age-adjustec	~	Insufficient I	Gender	Overall	Race/Ethnicit	Overall	T2

3) **Social Vulnerability Index 2012 - 2014**
   - Collected from – svi.cdc.gov
   - Type of file – comma separated values
   - Publisher - cdc.gov
   - Maintainer – Centers for Disease Control and Prevention
   - Maintainer email - dhdsprequests@cdc.gov

Fig. 4: A snippet of the Social Vulnerability Index Dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	FID	AFFGEOID	ST	STATE	ST_ABBR	COUNTY	FIPS	LOCATION	AREA_SQM	E_TOTPOP	M_TOTPO	E_HU	M_HU	E_HH	M_HH	E_POV	M_POV	E_UNEMP	M_UNEMP	E_PCI	M_PCI	E_NOHSDP	M_NOHSD	E_
2		0500000U	1	ALABAMA	AL	Autauga	1001	Autauga C	594.4366	55136	0	22431	67	20304	458	7006	935	2252	352	24644	780	5012	561	
3		0500000U	1	ALABAMA	AL	Baldwin	1003	Baldwin C	1589.807	191205	0	105563	168	73058	1241	25988	2457	7856	918	26851	712	14615	1051	
4		0500000U	1	ALABAMA	AL	Barbour	1005	Barbour C	884.8767	27119	0	11833	120	9145	311	5832	639	1527	282	17350	821	4790	341	
5		0500000U	1	ALABAMA	AL	Bibb	1007	Bibb Count	622.5824	22653	0	8985	66	7078	390	3596	770	975	310	18110	1477	3466	500	
6		0500000U	1	ALABAMA	AL	Blount	1009	Blount Cou	644.8065	57645	0	23868	77	20934	399	9866	947	2291	358	20501	719	8567	697	
7		0500000U	1	ALABAMA	AL	Bullock	1011	Bullock Co	622.8051	10693	0	4469	100	3746	216	2085	477	809	187	17706	1557	2511	319	
8		0500000U	1	ALABAMA	AL	Butler	1013	Butler Cou	776.828	20523	0	9934	67	8253	252	5239	517	1075	177	18115	832	3288	278	
9		0500000U	1	ALABAMA	AL	Calhoun	1015	Calhoun C	605.8889	117186	0	53306	209	45348	705	24794	1491	7257	594	21306	573	15674	787	
10		0500000U	1	ALABAMA	AL	Chambers	1017	Chambers	596.5312	34091	0	16944	50	13901	393	8051	823	1916	304	21240	1482	5367	388	
11		0500000U	1	ALABAMA	AL	Cherokee	1019	Cherokee (	553.7197	26042	0	16254	89	11726	451	5370	876	1114	221	22234	1382	3856	373	
12		0500000U	1	ALABAMA	AL	Chilton	1021	Chilton Co	692.8537	43781	0	19246	68	16281	375	8128	931	1897	373	21718	1132	6605	538	
13		0500000U	1	ALABAMA	AL	Choctaw	1023	Choctaw C	913.4999	13546	0	7248	33	5526	213	2867	364	727	152	21268	1711	2481	260	
14		0500000U	1	ALABAMA	AL	Clarke	1025	Clarke Cou	1238.465	25331	0	12604	55	9791	277	6757	778	2063	386	20022	1569	3424	370	
15		0500000U	1	ALABAMA	AL	Clay	1027	Clay Count	603.9609	13617	0	6756	57	5572	194	2481	465	624	159	18957	1151	2418	345	
16		0500000U	1	ALABAMA	AL	Cleburne	1029	Cleburne C	560.1041	14990	0	6698	50	5639	236	2691	507	550	165	19736	1373	2543	276	
17		0500000U	1	ALABAMA	AL	Coffee	1031	Coffee Co	678.9857	50726	0	22648	89	19086	368	9403	916	1576	251	24204	807	5797	402	
18		0500000U	1	ALABAMA	AL	Colbert	1033	Colbert Co	592.6196	54491	0	25971	85	22442	414	9860	1199	2356	348	21763	695	6222	476	
19		0500000U	1	ALABAMA	AL	Conecuh	1035	Conecuh C	850.1565	12985	0	7066	38	5030	264	4256	613	1306	208	15441	1504	2142	323	
20		0500000U	1	ALABAMA	AL	Coosa	1037	Coosa Cou	650.9259	11247	0	6478	54	4446	226	2190	421	970	206	17749	1212	1981	299	
21		0500000U	1	ALABAMA	AL	Covington	1039	Covington	1030.456	37881	0	18803	86	14979	354	7479	922	1733	264	20941	921	5300	430	
22		0500000U	1	ALABAMA	AL	Crenshaw	1041	Crenshaw	608.8396	13938	0	6718	56	5424	196	2358	399	673	134	20366	1092	2205	165	
23		0500000U	1	ALABAMA	AL	Cullman	1043	Cullman C	734.8974	80668	0	37084	98	31160	558	14354	1315	3204	378	21105	766	10219	625	
24		0500000U	1	ALABAMA	AL	Dale	1045	Dale Coun	561.1495	50013	0	22793	112	19470	411	9114	764	2323	336	22368	627	5062	430	
25		0500000U	1	ALABAMA	AL	Dallas	1047	Dallas Cou	978.6942	42743	0	20216	79	16259	383	15163	1243	3059	453	17614	978	6183	502	
26		0500000U	1	ALABAMA	AL	DeKalb	1049	DeKalb Co	777.0938	71074	0	31043	123	24743	514	14128	1555	2893	457	18416	690	12650	739	
27		0500000U	1	ALABAMA	AL	Elmore	1051	Elmore Co	618.488	80321	0	32985	390	28617	596	9700	1274	3229	468	24185	767	6924	632	
28		0500000U	1	ALABAMA	AL	Escambia	1053	Escambia (	945.0801	38042	0	16431	229	13737	447	9309	888	2371	434	16673	935	5554	526	
29		0500000U	1	ALABAMA	AL	Etowah	1055	Etowah Cc	535.3327	104126	0	47507	123	40001	614	20059	1424	4834	451	20445	480	12837	749	

The datasets that we used for this project were collected from data.gov and cdc.gov. They were stored in an organized manner in csv formats, which made them easy to use without much pre-processing. They were well documented, which made it easy to work with them without much confusion.

II. Data Analysis

A. *The Questions/Hypotheses we Sought to Answer from the Data*

Since the datasets we chose had rich information about heart disease mortality rate, median income for New York State, and social determinants, we decided to answer the two following questions:

1) *How is heart disease mortality rate connected to the median income in New York State?*
2) *How is heart disease mortality rate connected to social vulnerability in the United States?*
3) *How is heart disease mortality rate connected to ethnicity in New York?*

We first decided to check using basic exploratory data analysis if there was actually any relationship between heart disease mortality rate and the other factors as explained above. For that we planed to use MS Excel. After a pattern was found, we decided to clean the data and do some pre-processing so that the relationships could be studied in detail. After we were certain that there were some clear relationships between the factors, we decided to train a machine learning model to try and see if we could predict heart disease mortality from social determinants.

B. *Description of Tools and Methods used for the Analysis*

- Python was used for cleaning and pre-processing the data for analysis of the first hypothesis. We first cleaned the data by filtering all missing and erroneous fields, then we pre-processed it by taking only the counties that were common in both datasets. We also needed to manually split some county names which had been grouped together median income dataset during data cleaning. Two different pieces of code were written - *data_clean.py* for cleaning the data and *data_fix.py* for pre-processing the data The codes are given in Appendix I.
- For creating a machine learning model, we used the scikit-learn tool and the Pandas tool to read the data and create data frames in Python. The entire code for the machine learning part is given in the Appendix II.
- For the visualizations of the analysis of the relationship between heart disease mortality rate and median income, Microsoft Excel was used.
- To draw the choloropeth maps for the social vulbnerability index distribution, we used a software called arcGis.

C. Steps Taken to Perform the Analysis

1) For the Analysis of Relationship Between Heart Disease Mortality Rate and Median Income for New York State:

1) First, we had to clean the data and remove rows with missing fields.
2) In the NYSERDA dataset, the data was presented in a way in which multiple counties were bundled together. So, we had to write a Python script to manually separate the counties.
3) Once the NYSERDA dataset was cleaned, we had to save it as a new file.
4) Then we had to extract data for NY state counties which were common in both the NYSERDA and Heart Disease Mortality datasets. This was done with the help of a Python script.
5) For the analysis, we put the Heart Disease Mortality rate and Median income for each county as columns into a csv file. Then we used MS Excel to analyze the data.

2) For the Analysis of Relationship Between Heart Disease Mortality Rate and Social Determinants for the entire United States:

1) First we downloaded the shape file of the data from the data download page of CDC.
2) Next, we used arcGis, a proprietary mappoing application to create the county map for the entire United States.
3) Then we used the different scores in the social vulnerability index to divide them into quartiles.
4) The on the map, we plotted the value for each county, with the four scores represented by 4 varying degrees of colors.

3) For the Creation of a Machine Learning Model and Testing it on the Data:

1) First, we had to clean the data and remove rows with missing fields.
2) From the Social Vulnerability Index data, the following fields were selected for analysis:
    - EP_POV (person below poverty estimate),
    - EP_UNEMP (civilian unemployed),
    - EP_PCI (per capita income),
    - EP_NOHSDP (person with no high school diploma),
    - EP_AGE65 (person age 65+),
    - EP_AGE17 (17+),
    - EP_SNGPNT (single parent household with one children),
    - EP_MINRTY (minority estimate except white, non-Hispanic),
    - EP_LIMENG (person who speaks English less than well),
    - EP_NOVEH (household with no vehicle),
    - EP_GROUPQ (person in group quarters),
    - E_TOTPOP (total population).
3) Python Scikit Learn was used to see the corelation between the different metrics.
4) The data was pre-processed to convert heart rate disease values to categorical values.
5) Python Scikit Learn was used to train SVM and KNN and analyze their accuracy (Details given in Appendix II).

4) For the analysis of heart disease mortality rate and ethnicity in New York: For this analysis, we simply cleaned the data by removing rows with missing values and used Microsoft Excel to create a chart of how heart disease rate varies with ethnicity.

III. Presentation/Visualization of the Results

A. Results of the Analysis

1) Relationship Between Heart Disease Rate and Median Income for New York State: We used MS Excel to perform this analysis. First we created a chart that compared the heart disease mortality rate to the median income for each state. Since the heart disease mortality rate was too small compared to median income for the them to be clearly accommodated in a single column chart, we convert it from per 100000 to per 10000000. Then we plotted the average line for both median income and heart disease rate. We also created two choropleth maps of heart disease mortality rate and median income using MS Excel to see how they were related.

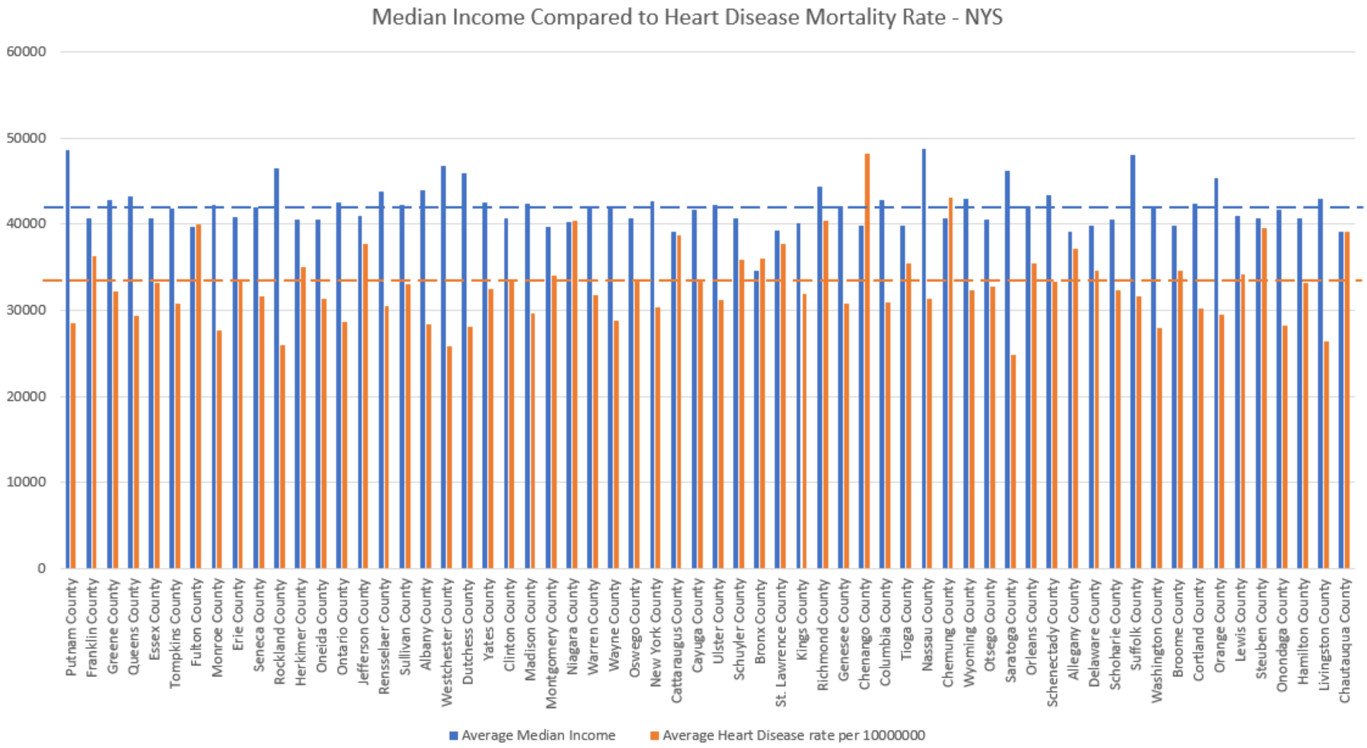
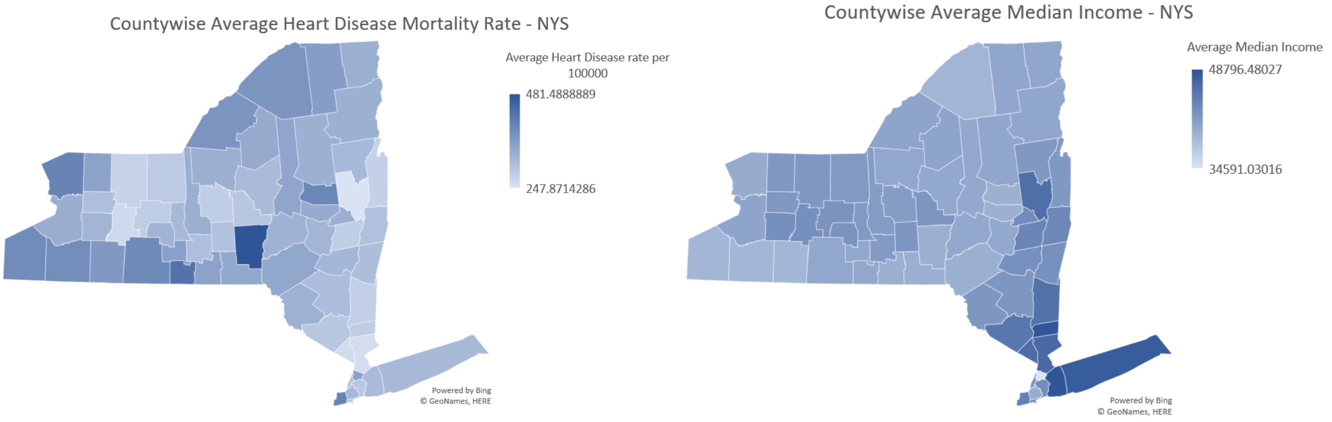


Fig. 5: Heart disease mortality rate vs median income (NYS).


Countywise Average Heart Disease Mortality Rate - NYS
Average Heart Disease rate per 100000
481.4888889
247.8714286
Powered by Bing
© GeoNames, HERE
Countywise Average Median Income - NYS
Average Median Income
48796.48027
34591.03016
Powered by Bing
© GeoNames, HERE

(a) Heart disease mortality rate

(b) Median income

Fig. 6: Median income vs heart disease mortality rate for New York

Results: From Fig. 5, it is clear that in in most counties where median income is below the average, heart disease rate is above average and vice versa. This is corroborated by Fig. 6a and Fig. **??** which show that in counties where median income is low, heart disease rate is high and vice versa. Therefore we can conclude that heart disease mortality rate is usually inversely related to median income.

2) Relationship Between Heart Disease Rate and Social Vulnerability:

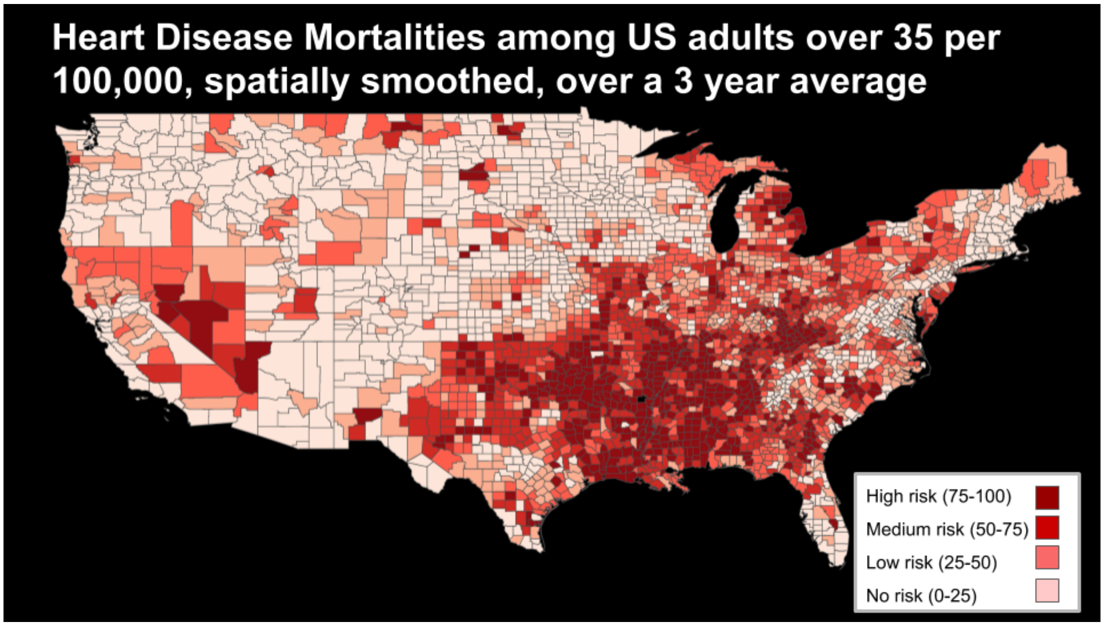

Heart Disease Mortalities among US adults over 35 per 100,000, spatially smoothed, over a 3 year average
High risk (75-100)
Medium risk (50-75)
Low risk (25-50)
No risk (0-25)

Fig. 7: Chloropeth map of heart disease mortality rate

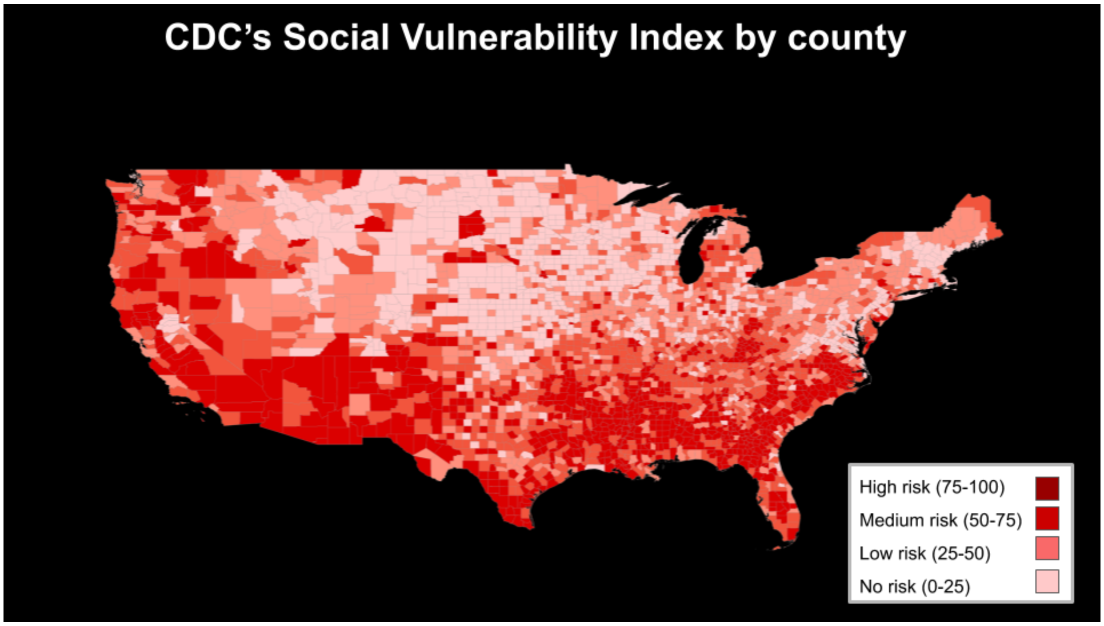


Fig. 8: Chloropeth map of overall social vulnerability index (2012-2014)

It uses these 15 US census variables to determine the Social vulnerability of each county. We believe that these indicators of social vulnerability also serve as strong indicators of social determinants. Our hypothesis is that the higher the overall vulnerability of a community the less likely they are to live a healthy lifestyle; therefore, their likelihood of coronary-related mortalities sees an increase as a result.

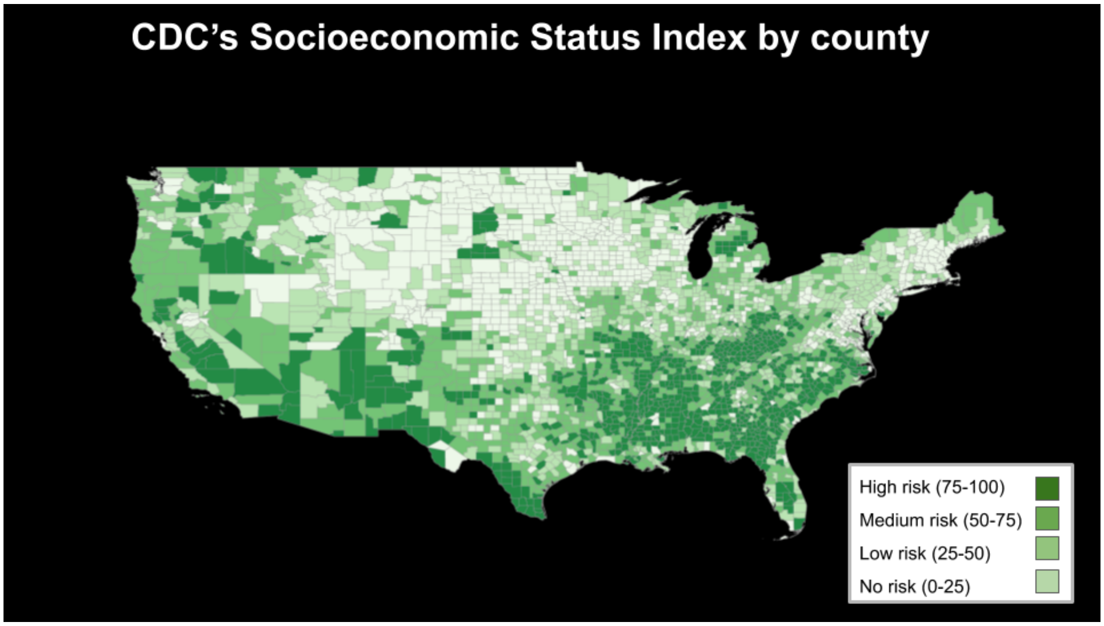


Fig. 9: Chloropeth map of Socioeconomic Status (2012-2014)

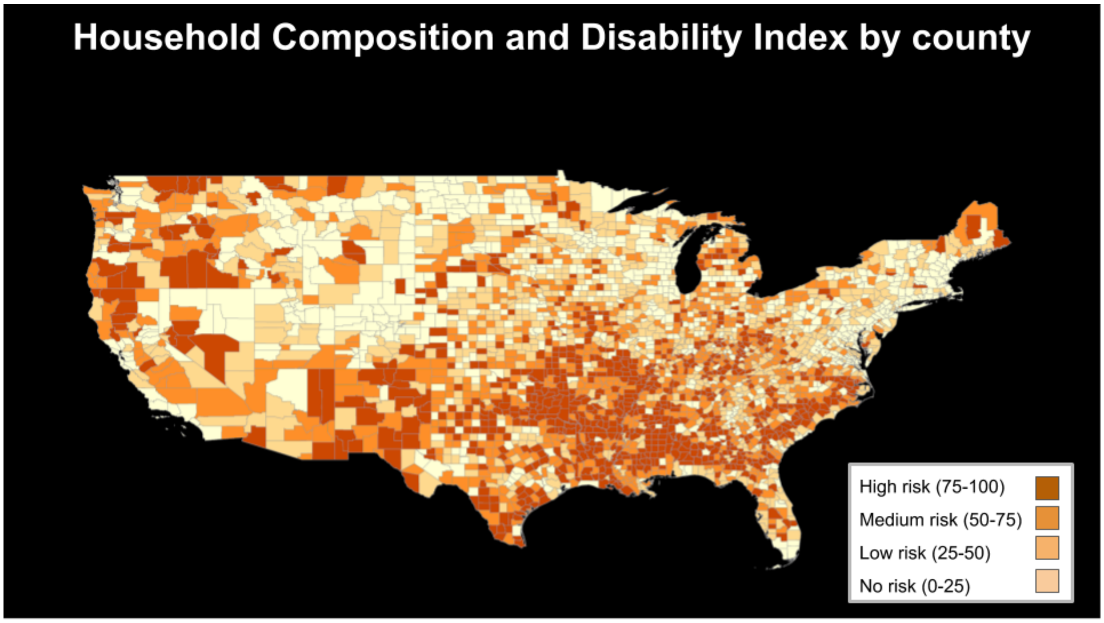


Fig. 10: Chloropeth map of Household Composition and Disability (2012-2014)

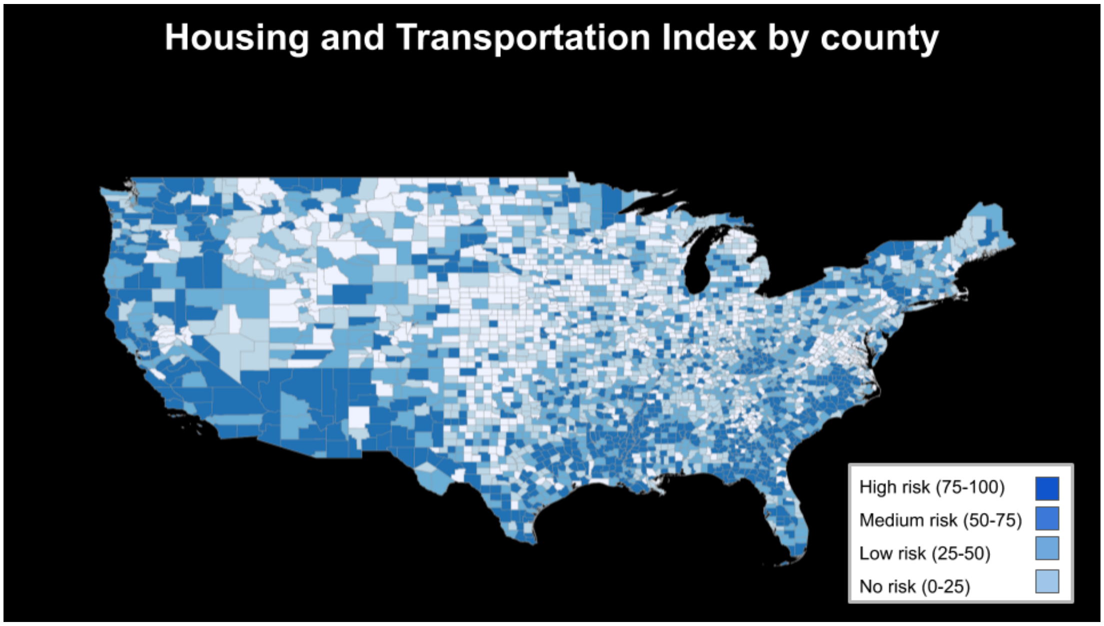


Fig. 11: Chloropeth map of Housing and Transportation (2012-2014)

From the Chloropeth maps we can conclude the following:

- Social vulnerability is a good indicator of Heart disease mortalities.
- Both these datasets are indicative of genuine problems as they are specially smoother and not affected by population size.
- There are some regions where there are few mortalities, but high SVI, Socioeconomic vulnerability, poor housing and transportation, and a high population which is disabled or unemployed. Certain counties in New Mexico, Arizona, Nevada, California, Oregon, and Washington score high on the SVI and all of it's subcategories but have relatively low rates of heart disease related mortalities.

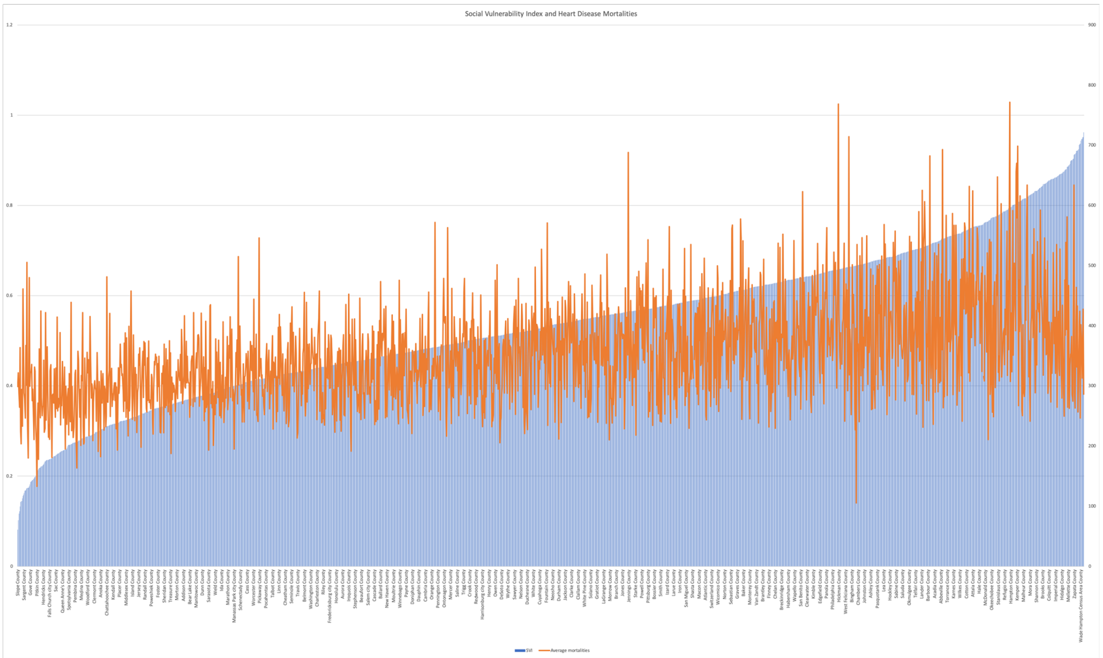


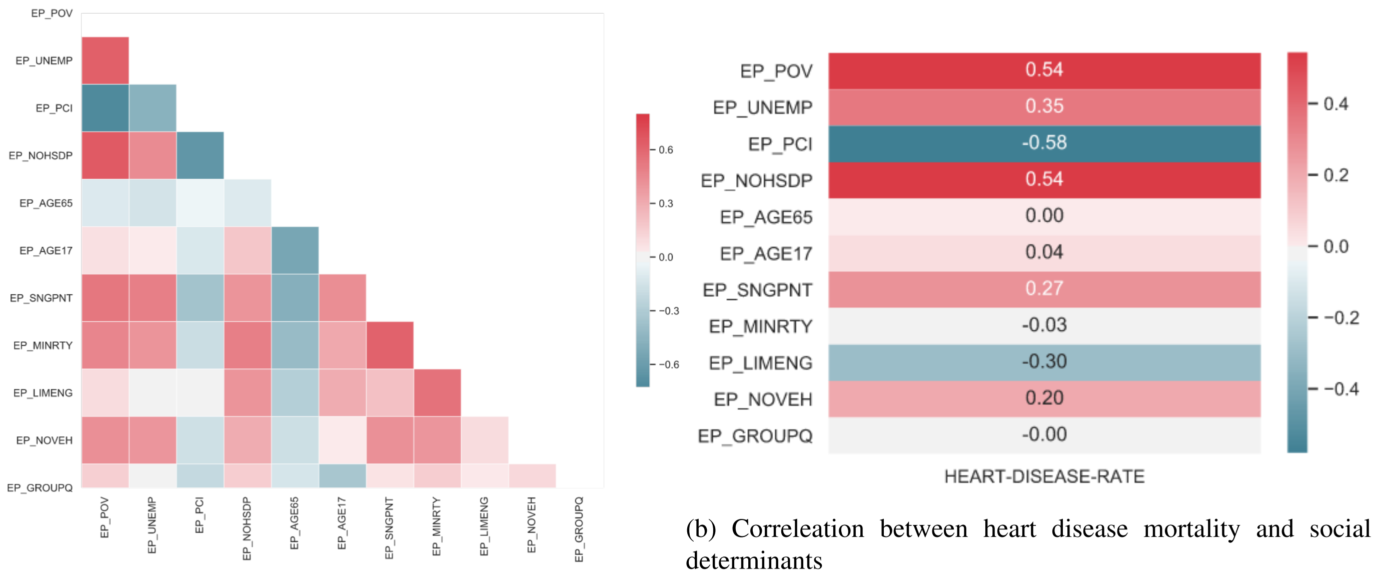
Fig. 12: Social Vulnerability and Heart Disease Mortality

Some conclusions we can draw from Fig. 12:

- It seems that regardless of SVI score heart disease stays pretty regular throughout the counties.

- The only trend is that counties with higher SVI values tend to have higher highs and slightly above average heart related mortalities.
- Surprisingly, it seems that the top 1% of counties with the highest SVI score seem to have relatively lower heart disease related deaths.

3) Results from Training a Machine Learning Model and Testing it on the Data: In order to determine which model to use, we first analyzed the correlation between various metrics in the social determinants data generating the heatmap given in Fig 13a.



(a) Correlation between various social determinants

(b) Correleation between heart disease mortality and social determinants

Fig. 13: Correlation heatmaps

From Fig. 13a, we concluded the following:

- There is high correlation between EP_UNEMP and EP_POV, i.e., poverty and unemployment rate highly correlates. Same goes for poverty vs no-high-school-diploma.
- High correlation between minority population (EP_MINRTY) and limited English (EP_LIMENG), minority population and no vehicle in household(EP_NOVEH), minority population and single parent household (EP_SNGPNT)
- High correlation between single parent household and poverty, which is interesting, as this is generally reversed in developing nations.

From Fig. 13b, we concluded the following:

- Heart-disease-death-rate highly correlates with poverty and no-high-school-diploma-household
- Heart-disease-death-rate inversely correlates with per-capita-income and limited-English-speaking-ability

We then train two kernels (an SVM kernel and a KNN kernel) on a portion of the data (test data). From the SVM classifier, we can get around 67% classification accuracy, while from the KNN classifier we get around 67% of accuracy as well.

4) Relationship Between Ethnicity and Heart Disease Mortality Rate in New York:

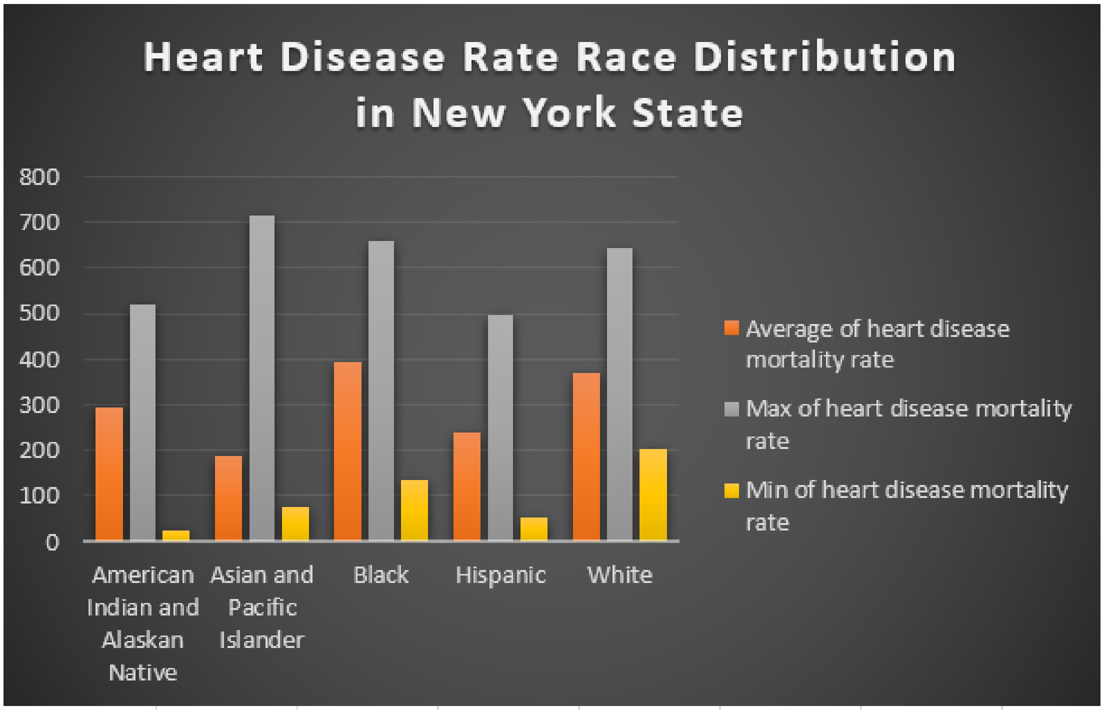


Fig. 14: Ethnicity and Heart Disease Mortality Rate

From the Fig.14, we can see that Asian and Pacific Islanders have remarkably high chances of getting heart disease compared to other races based on New York state data.

B. Management of the Presentation/Visualization

We had created a Github repository for storing all our project related work at the beginning of the project. All of us used this repository to store the codes and generated images/visualizations for the analysis. This way, everyone had access to all the materials anytime they needed. All the datasets were also stored in this repository for easy access. Since Github is a cloud platform, it ensured that in the event all our local machines crashed for some reason, all the project materials would still persist on the cloud and be easily accessible.

C. How the Visualization/Presentation Supports the Goal of the Data Science Project

In this project, we wanted to show the relationships between heart disease mortality rate and different social determinants like median income and social vulnerability index. The column chart with the average lines shows clearly that there is a pattern in how heart disease mortality rate varies with median income for counties in New York State. The Choropleth maps that we created make it very easy to understand that there is a relationship between heart disease mortality rate and the different factors. Since column charts and Choropleth maps can be easily understood by people with no technical background, they make it easier to present the results of the analysis is a coherent manner to people from all backgrounds. On the other hand, the confusion matrix for the SVM classifier (Fig. 15), even though it would be an important source of information for an experienced data scientist, would make little sense to someone not trained in the various data analytic methods.

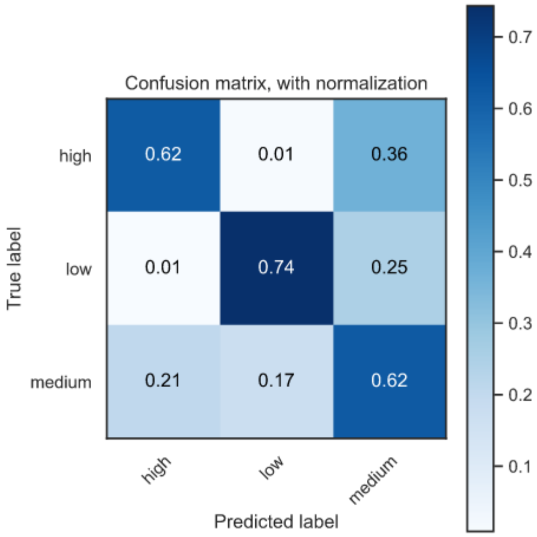


Fig. 15: Confusion matrix for the SVM classifier.

From this exercise, we realized that highly technical visualizations should be avoided if the goal is to make a study accessible to people from a wide range of backgrounds. It is better to use easily understandable statistical methods like columns charts and Choropleth maps to convey data in a meaningful manner to a wide range of people.

IV. Overall Data Management Plan

In this section, we describe our data management plan for the project. Tha plan includes the following:

1) **Interoperability support**

   We have put all the data used for the purpose of this project and analysis related codes and image sin a Github repository which can be accessed by the public. The datasets are all saved as csv files and, therefore, can be easily manipulated by data manipulation software like MS Excel.

2) **Security support**

   In the Github repository, have included a copy of this report. In Section I of this report, detailed information about the source of data, metadata conventions, and provenance information has been given in detail. Therefore, any potential user of the data can easily understand how they can validate the authenticity of the data by visiting the original source of the datasets. Since the Github repository will be open to the public, it can be accessed without restriction, but not modified without authorization.

3) **Data ownership**

   The data has been collected from public federal websites like data.gov and cdc.gov. Therefore, the actual owners of the data re the respective federal agents who had originally published them. However, the other project materials that will be put in the Github repository will be owned by Rensselaer Polytechnic Institute. Users will be able to access and use the resources, but will not be able to have ownership rights to the data.

4) **Creation of logical collections**

   The datasets are separated into logical collections. There is a separate csv file for each dataset. The other resources that we used for the analysis of the data have also been stored in logically separate directories, making them easier to find and access.

5) **Physical data handling**

   We have included back-up copies of the datasets in our public Github repository. The original datasets can be found on data.gov and cdc.gov. We also have a private repository that we had created for this project where we have back-ups of the datasets and all other materials associated with this project.

6) **Metadata collection, management and access**

   The metadata is stored in the csv files as the header of each column. Each column header has a name which makes it clear what the data of that column represents. Apart from that, the data.gov and cdc.gov websites have detailed descriptions about the data. We have included the unique identifiers of the datasets in our report, which will make it easy for people to access the original sources and read the detailed information there.

7) **Persistence**

   The datasets stored in the Github repository will still persist after the course is over. The Github repositories are permanent (until they are deleted by authorized personnel) and ca be accessed publicly. All the project materials will also be persistent as they will be similarly stored in the Github repository.

8) **Knowledge and information discovery**

   The datasets have useful metadata inside them in the form of column headers. This allows any user to understand the relationships between the different columns. Moreover, in the report, we have included interesting results that conclude that there is a relationship between the data in the different datasets. We have represented these relationships using easy to understand visualizations. This will give all users some preliminary knowledge about the datasets if they plan to use the datasets for some kind of analysis.

9) **Data distribution and publication**

   Since the Github repository will be public, users will be able to see when new changes have been made to the repository. Github has a very elaborate system in place that allows users to see detailed information about the changes that are made to a repository. Therefore, this system will make interested parties aware of the changes and additions to the project archive.

V. Conclusion

In this project, we have tried to study the factors which affect the heart disease mortality rate of different counties across the United States. Our results show some significant relationships between social determinant factors and heart disease mortality rates. Hopefully, this report will be able to coherently express these relationships to the concerned authorities so that they can use this a s a basis for further investigations in this field. Since heart disease is one of the primary killers of the US population, such investigations may help increase the average lifespan of people across the country.

Appendix I

- ***data_clean.py*** is given below:

```
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
import csv
mpl.rcParams['legend.fontsize'] = 10

fig = plt.figure()
axyz = fig.gca(projection='3d')
plt.axis('equal')


# Median Income dataset
county_income = [] #from csv

income_range = [] # from csv

income_median = [] #calculated by taking endpoint for ranges


with open("NYSERDA_Low-_to_Moderate-
   ↪ Income_New_York_State_Census_Population_Analysis_Dataset__Average_for_2013-2015.csv"
   ↪ ) as csvfile:
  readCSV = csv.reader(csvfile, delimiter=',')
  for row in readCSV:
    if (str(row[0]) == "Otsego, Schoharie, Oneida, & Herkimer"):
      county_income.append(("Otsego" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Schoharie" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Oneida" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Schoharie" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Herkimer" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Broome, Chenango, Delaware, & Tioga"):
      county_income.append(("Broome" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Chenango" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Delaware" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Tioga" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Clinton, Franklin, Essex & Hamilton"):
      county_income.append(("Clinton" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Franklin" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Essex" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Hamilton" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Steuben, Schuyler & Chemung"):
  county_income.append(("Steuben" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Schuyler" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Chemung" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Cattaraugus & Allegany"):
  county_income.append(("Cattaraugus" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Allegany" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Ontario & Yates"):
  county_income.append(("Ontario" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Yates" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Warren & Washington"):
  county_income.append(("Warren" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Washington" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Livingston & Wyoming"):
  county_income.append(("Livingston" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Wyoming" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Genesee & Orleans"):
  county_income.append(("Genesee" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Orleans" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Sullivan & Ulster"):
  county_income.append(("Sullivan" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Ulster" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Fulton & Montgomery"):
  county_income.append(("Fulton" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
  county_income.append(("Montgomery" + " County"))
  income_range.append((row[4]))
  income_median.append(0)
elif (str(row[0]) == "Wayne & Seneca"):
      county_income.append(("Wayne" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Seneca" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Jefferson & Lewis"):
      county_income.append(("Jefferson" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Lewis" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Columbia & Greene"):
      county_income.append(("Columbia" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Greene" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Madison & Cortland"):
      county_income.append(("Madison" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Cortland" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    elif (str(row[0]) == "Cayuga & Onondaga"):
      county_income.append(("Cayuga" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
      county_income.append(("Onondaga" + " County"))
      income_range.append((row[4]))
      income_median.append(0)
    else:
      county_income.append((str(row[0]) + " County"))
      income_range.append((row[4]))
      income_median.append(0)



for x in range(len(income_range)):
  if (income_range[x] == "$0 to <$10,000"):
    income_median[x] = 5000
  elif (income_range[x] == "$10,000-<$20,000"):
    income_median[x] = 15000
  elif (income_range[x] == "$20,000-<$30,000"):
    income_median[x] = 25000
  elif (income_range[x] == "$30,000-<$40,000"):
    income_median[x] = 35000
  elif (income_range[x] == "$40,000-<$50,000"):
    income_median[x] = 45000
  elif (income_range[x] == "$50,000+"):
    income_median[x] = 55000


with open("county_median_income3.csv","w+") as csv_file:
    for x in range(len(county_income)):
        row = str(county_income[x]) +","+ str(income_median[x])
        csv_file.write(row+'\n')



##########################################################################
# Heart Diesease dataset
```

```
county_heart = []
heart_disease = [] # per 100000
# only saving those counties which are common to both datasets
with open("Heart_Disease_Mortality_Data_Among_US_Adults__35___by_State_Territory_and_County
   ↪ .csv") as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
      if not row[7]:
        continue
      state_name = str(row[1])
      county_name = str(row[2])
      if (state_name == "NY"):
        county_heart.append(county_name)
        heart_disease.append(float(row[7]))


with open("county_heart_disease3.csv","w+") as csv_file:
    for x in range(len(county_heart)):
        row = str(county_heart[x]) +","+ str(heart_disease[x])
        csv_file.write(row+'\n')
```

- ***data_fix.py*** is given below:

```
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
import csv
mpl.rcParams['legend.fontsize'] = 10

fig = plt.figure()
axyz = fig.gca(projection='3d')
plt.axis('equal')


# Income dataset
county_income = [] #from csv

county_income_median = []

with open("county_median_income3.csv") as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
        county_income.append((str(row[0])))
        county_income_median.append([str(row[0]),float(row[1])])

county_avg_income_median = []

unique_counties_income = (list(set(county_income)))

for county in unique_counties_income:
  total = 0
  frequency = 0
  for row in county_income_median:
    if (row[0] == county):
      total = total + row[1]
      frequency = frequency + 1
  average = total / frequency
  county_avg_income_median.append([county, average])


for row in county_avg_income_median:
  print (row)

with open("county_avg_income_median3.csv","w+") as csv_file:
    for x in county_avg_income_median:
        row = str(x[0]) +","+ str(x[1])
        csv_file.write(row+'\n')

#Heart disease dataset
county_heart = [] #from csv

county_heart_disease = []

with open("county_heart_disease3.csv") as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
        county_heart.append((str(row[0])))
        county_heart_disease.append([str(row[0]),float(row[1])])

county_avg_heart_disease = []

unique_counties_heart = (list(set(county_heart)))

for county in unique_counties_heart:
  total = 0
  frequency = 0
  for row in county_heart_disease:
    if (row[0] == county):
      total = total + row[1]
      frequency = frequency + 1
  average = total / frequency
  county_avg_heart_disease.append([county, average])


for row in county_avg_heart_disease:
  print (row)

with open("county_avg_heart_disease3.csv","w+") as csv_file:
    for x in county_avg_heart_disease:
        row = str(x[0]) +","+ str(x[1])
        csv_file.write(row+'\n')

# saving the data for common counties

with open("common_county_data2.csv","w+") as csv_file:
  for county in unique_counties_heart:
    if (county in unique_counties_income):
      income = 0
      heart = 0
      for x in county_avg_income_median:
        if (x[0] == county):
          income = x[1]
      for x in county_avg_heart_disease:
        if(x[0] == county):
          heart = x[1]
      row = str(county)+ "," + str(income) +","+ str(heart)
      csv_file.write(row+'\n')
```

Appendix II

The code used for the machine learning part of the analysis is given below:

```
%pylab
%matplotlib inline
import pandas as pd
import seaborn as sns
sns.set(style="white")

# import library for ML
import sklearn
```

```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn import svm
from sklearn.neighbors import KNeighborsClassifier
from sklearn import tree

hd_filename = "./
    ↪ Heart_Disease_Mortality_Data_Among_US_Adults__35___by_State_Territory_and_County.xls"
df_xls = pd.read_excel(hd_filename)  # original dataset
# get data for 'overall' gender and 'overall' ethnicity
df_xls = df_xls[(df_xls['Stratification1']=='Overall') & (df_xls['Stratification2']=='Overall')
    ↪ ]
print("Column names in the original dataset")
print(df_xls.columns)
df = pd.DataFrame()
df['COUNTY'] = df_xls.LocationDesc.apply(lambda name: name.lower().replace("county", "").strip
    ↪ ())
df['STATE'] = df_xls.LocationAbbr
df['RATE'] = df_xls.Data_Value
df_hr = df.dropna()  # clean data
df_hr = df_hr.sort_values(by=['STATE', 'COUNTY'])
df.head()

columns_to_chose = ["ST_ABBR", "COUNTY", "E_TOTPOP", "EP_POV", "EP_UNEMP", "EP_PCI", "EP_NOHSDP
    ↪ ", "EP_AGE65", "EP_AGE17", "EP_SNGPNT", "EP_MINRTY", "EP_LIMENG", "EP_NOVEH", "EP_GROUPQ
    ↪ "]
df_svi = df_xls_svi.filter(columns_to_chose).dropna()
df_svi['COUNTY'] = df_xls_svi.COUNTY.apply(lambda name: name.lower().replace("county", "").
    ↪ strip())
df_svi = df_svi.rename(columns={"ST_ABBR": "STATE"})
df_svi = df_svi.sort_values(by=['STATE', 'COUNTY'])
df_svi.head()

df_svi_ep = df_svi.filter(regex='EP') / 100.  # normalize
corr = df_svi_ep.corr()
# plot correlation matrix
# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))
# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)
# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
# Draw the heatmap with the mask and correct aspect ratio
plt.figure(1)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.8, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})

# Merge two different datasets along 'STATE' and 'COUNTY'
df_merge = pd.merge(df_hr, df_svi, on=['COUNTY', 'STATE'])
df_merge = df_merge.dropna()

# Plot heatmap of heart-rate disease vs EP_*

df_merge_ep = df_merge.filter(regex='EP') / 100.  # normalize
corr = df_merge_ep.corrwith(df_merge.RATE)
f, ax = plt.subplots()
# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)
# Draw the heatmap with the mask and correct aspect ratio
corr = pd.DataFrame(corr, columns=["HEART-DISEASE-RATE"])
cmap = sns.diverging_palette(220, 10, as_cmap=True)
```

```
sns.heatmap(corr, annot=True, fmt=".2f", cmap=cmap, ax=ax)
plt.autoscale()

# Convert heart rate disease values to categorical values
df_merge['RATE_CAT'] = pd.cut(df_merge.RATE.values, bins=[0, 320, 420, 800],
            labels=["low", "medium", "high"])
df_merge['RATE_CAT'].value_counts(sort=False)

# get feature and target
feature_columns = ["E_TOTPOP", "EP_POV", "EP_UNEMP", "EP_PCI", "EP_NOHSDP", "EP_AGE65", "
    ↪ EP_AGE17", "EP_SNGPNT", "EP_MINRTY", "EP_LIMENG", "EP_NOVEH", "EP_GROUPQ"]
target_column = ["RATE_CAT"]
X = df_merge.loc[:, feature_columns]
X_scale = preprocessing.scale(X)
Y = df_merge.loc[:, target_column].values.ravel()
le = preprocessing.LabelEncoder()
Y = le.fit_transform(Y)
n_target = len(np.unique(Y))

# test train split
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, test_size=0.3, random_state=1)
print("train sample: ", X_train.shape[0])
print("test sample: ", X_test.shape[0])

## SVM model

clf = svm.SVC(gamma='scale', decision_function_shape='ovo')
clf.fit(X_train, Y_train)

y_pred = clf.predict(X_test)
_score = accuracy_score(Y_test, y_pred, normalize=True)
print("Accuracy :", _score)

# helper function

from sklearn.utils.multiclass import unique_labels
def plot_confusion_matrix(y_true, y_pred, classes,
                          normalize=False,
                          title=None,
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if not title:
        if normalize:
            title = 'Normalized confusion matrix'
        else:
            title = 'Confusion matrix, without normalization'

    # Compute confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    # Only use the labels that appear in the data
    classes = classes[unique_labels(y_true, y_pred)]
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    fig, ax = plt.subplots(figsize=(5,5))
    im = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    ax.figure.colorbar(im, ax=ax)
    # We want to show all ticks...
    ax.set(xticks=np.arange(cm.shape[1]),
           yticks=np.arange(cm.shape[0]),
           # ... and label them with the respective list entries
           xticklabels=classes, yticklabels=classes,
           title=title,
           ylabel='True label',
           xlabel='Predicted label')

    # Rotate the tick labels and set their alignment.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")

    # Loop over data dimensions and create text annotations.
    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")
    fig.tight_layout()
    plt.autoscale()
    return ax

plot_confusion_matrix(y_pred, Y_test, classes=le.classes_, normalize=True,
                      title='Confusion matrix with normalization')


clf = KNeighborsClassifier(n_neighbors=10)
clf.fit(X_train, Y_train)

y_pred = clf.predict(X_test)
_score = accuracy_score(Y_test, y_pred, normalize=True)
print("Accuracy : ", _score)

clf = tree.DecisionTreeClassifier()
clf.fit(X_train, Y_train)

y_pred = clf.predict(X_test)
_score = accuracy_score(Y_test, y_pred, normalize=True)
print("Accuracy : ", _score)
```