Recent asset distribution trends in estate planning among Americans

An introduction to joining and cleaning SAS files in Python

Kristopher Nerl

Spring, 2022

TABLE OF CONTENTS

GOALS DEFINED

Research/business goal defined:
Working in a law firm practicing estate planning and administration in Pennsylvania means working with many clients of an older demographic as they plan for what happens after their death. Clients have a variety of wishes for their estate, but the vast majority of planning can be boiled down to who gets your assets after you pass. If you have children, who gets your car, house, or bank account? What happens to your residue (everything not explicitly listed in a will and/or trust)? Knowing how individuals of certain demographics plan for this may lead us to be better legal advisors.

Therefore, our proposed research question is: What asset distribution trends in estate planning are occurring among Americans recently?

Our quantitative data collection technique will need to study the estate planning behaviors of Americans. Our collection will need to take into account the demographics of the individuals, and whether they represent the typical American (and hopefully client). This may lead us to a re-evaluation of the firm's (organization's) processes. We will review surveys collected by the HRS (see below) for responses regarding estate planning and especially for charitable giving themes (such as amounts outlined in a will), sampling and controlling for outliers in our populations.

Queries to be addressed:
1. What are our demographics of our respondents?
2. How many respondents have a valid will and/or trust?
3. What is the breakdown of beneficiaries for respondents' estates?
4. What amount of respondents include some form of charitable giving?
5. What amount of respondents have some form of life insurance as part of their estate planning?

Source for collecting data:
A single source of data will be utilized to accomplish our goals, as the HRS (Health and Retirement Study) is conducted by the University of Michigan and released every two years. It surveys roughly 20,000 respondents across the United States. Included in this massive dataset is a respondent section on wills and trusts, and the breakdown of beneficiaries and financial designations. *See* James. This information is publicly available via a single SAS (.sas7bdat), along with a corresponding codebook for reference.

COLLECTING DATA

Joining Datasets

Data is collected in three separate sas files via browser download from a zipped "Core" file available here: https://hrsdata.isr.umich.edu/data-products/2020-hrs-core

We start by joining three separate sas files from the HRS survey: 'Demographics', 'Assets and Income', and 'Wills and Life Insurance'. All three are outer joinable on the variables HHID (Household Identification Number), RSUBHH and QSUBHH (2018 and 2020 Sub Household Identification Number). Python code is provided in the Appendix of this document for the join after using pandas to read each sas file and export the joins to csv. Our merged dataset now includes 931 unique variables and 24,190 unique records. 19995268 values are null and will need addressing at a later stage.

The format of the variables is coded by HRS, and we will not rename these columns for the sake of brevity. For example, RT005 is "WILL PROVIDE FOR ALL CHILDREN EQUALLY?" and its distribution is below as an example of the surveys numeric coding:

N	N%	Numeric Code	Attribute Values
6	0.04%	"-8"	Web non-response
2938	18.69%	"1"	YES
280	1.78%	"5"	NO
10	0.06%	"8"	DK (Don't Know); NA (Not Ascertained)
5	0.03%	"9"	RF (Refused)
12484	79.40%	null	Blank. INAP (Inapplicable); Partial Interview

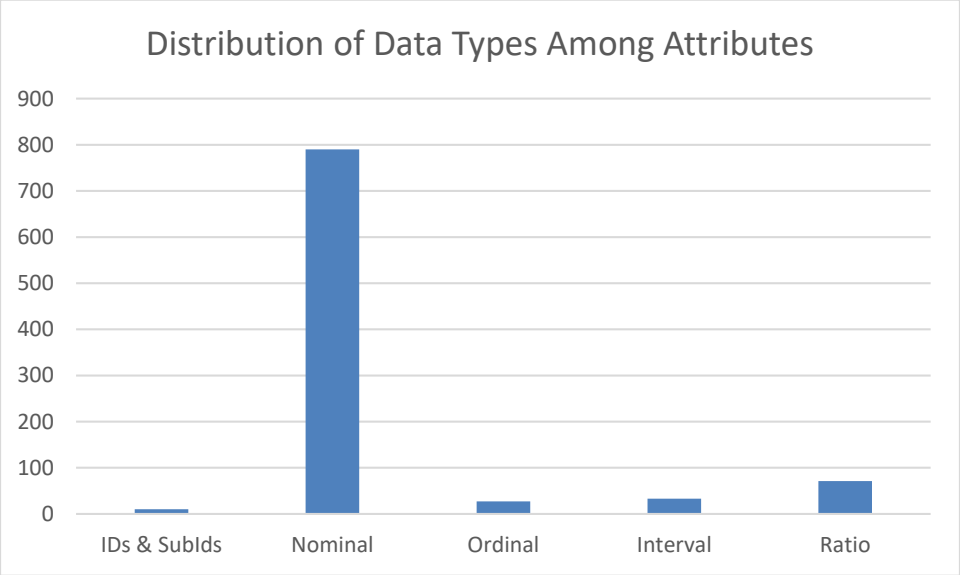
All attributes are numeric except for the three main identification attributes we joined on, along with seven additional sub-identification numbers used in the individual surveys as characters. Due to length, the relevant codebooks with number of instances and corresponding numeric values are included here for reference: https://drive.google.com/drive/folders/1WNTsB3EpRG3e85qX7CqwKeBt88t2D5N9?usp=sharing

Only 8 of the numeric attributes are not missing values at this time. In fact, 38 attributes contain no values at all and will be investigated and likely removed during cleaning. It can be expected that our merges created null values for attributes not shared by each sub-dataset. Also, many of our attributes are related to other attributes. If you do not have a valid Will, it is possible that the

survey did not code a response for "How many children are included in the Will. Further investigation will be needed to summarize these results. We have an average of 21,546 null values per numeric attribute. These will be dealt with during pre-processing of the data.

Distribution of Data Types Among Attributes

The vast majority (790) of numeric attributes are nominal in data type, while 27 are ordinal, 33 are interval scale and 71 of which are ratio scale. These ratio scale variables have a possibility for a zero value and their delta can be explored. An example attribute that conforms to this is "Dividends and Interest from Checking Accounts, Savings Accounts and, Money Market Funds", where the difference between respondents can be evaluated.



Data Reduction

Since we have so many attributes, we need to reduce our initial analysis to those responses that best help us answer our queries. We have identified key attributes that respond to our goals below (number of responses, followed by coded value):

1. What are our demographics of our respondents?
    a. RB017M (What is the highest degree you have earned?)
        i. 16 1. LESS THAN BACHELORS
        ii. 21 2. BACHELORS
        iii. 17 3. MASTERS/MBA
        iv. 13 7. OTHER (SPECIFY) (INCLUDES MD, LAW, AND PHD)
        v. 1 8. DK (Don't Know); NA (Not Ascertained)
        vi. 9. RF (Refused)
        vii. 15655 Blank. INAP (Inapplicable); Partial Interview
    b. RB020 (Would you say your family was pretty well off financially, about average, or poor?)
        i. 21 1. PRETTY WELL OFF FINANCIALLY
        ii. 116 3. ABOUT AVERAGE
        iii. 55 5. POOR
        iv. 2 8. DK (Don't Know); NA (Not Ascertained)
        v. 1 9. RF (Refused)
        vi. 15528 Blank. INAP (Inapplicable); Partial Interview
    c. RB022 (Before age 16, was there a time when you or your family received help from relatives because of financial difficulties?)
        i. 34 1. YES
        ii. 157 5. NO
        iii. 4 8. DK (Don't Know); NA (Not Ascertained)
        iv. 9. RF (Refused)
        v. 15528 Blank. INAP (Inapplicable); Partial Interview
    d. RB089M1M (What race do you consider yourself to be: White, Black or African American, American Indian, Alaska Native, Asian, Native Hawaiian, Pacific Islander, or something else?)
        i. 110 1. WHITE/CAUCASIAN
        ii. 50 2. BLACK/AFRICAN AMERICAN
        iii. 35 97. OTHER (SPECIFY) - Masked version includes American Indian, Alaskan Native, Asian, Native Hawaiian, and Pacific Islander
        iv. 0 98. DK (Don't Know); NA (Not Ascertained)
        v. 0 99. RF (Refused)
        vi. 15528 Blank. INAP (Inapplicable); Partial Interview

e. RQ020 (About how much wage and salary income did you receive in the Last Calendar Year, before taxes and other deductions?)

   i. N: 3118
   ii. Min: $0
   iii. Max: $2,500,000
   iv. Mean: $53,941.26
   v. SD: $69,723.87
   vi. Missing: 7815

Initial quantitative/qualitative analysis of responses for Query 1

Knowing the educational background of our clients could provide a key to serving them in their estate planning. It is interesting to note that the breakdown of degrees is relatively similar throughout, with advanced degrees being expectedly less of our population. However, with less than 0.5% of respondents answering this question, it might be hard to make inferences for what the data is telling us.

Asking respondents their opinion on their family's historical financial history might provide us with an interesting take on their present status. It is inherently biased by their opinion, rather than fact, but this is the nature of a voluntary survey and applies somewhat to all of the survey's questions. The vast majority of those answering the question think they were about average, with no financial help received when they were young. We will want to investigate later to see how these answers relate to later estate planning steps.

Our racial demographics are limited to essentially three categories (white, black, and other), which few chose to even answer. It appears this attribute may tell us little as to what kind of person will be a good client for our firm. On the other hand, our firm should caution itself on profiling clients based on race (making this question's impact potentially moot). More categories and actual responses here would certainly be beneficial.

For our income demographics, we see a decent amount of responses here in continuous, numeric dollar amounts. This gives us the ability to perform more robust statistical analysis, and see the mean appears to represent the average salary of an American. While, $2.5 million is a lot of money as a maximum response, we see this amount quite frequently. Rich clients want and need estate planning to protect their vast assets. Also, this is only for income in the last year and not lifetime or net worth. For retirees, this likely means a minimal amount.

In conclusion, we are at the mercy of the questions asked by the surveyors. I would have preferred to included questions about age, gender and location but these are not included in our datasets. All of the above questions are heavily skewed towards blank responses, which will inherently hurt our results. It's possible that respondents were not comfortable with these demographic questions and chose not to answer them. A quick review of these questions from the 2018 survey revealed similar levels of non-response. Hopefully, those responses we do have are able to correlate to each other or otherwise paint a picture post-cleaning. We might also think about reassigning coded values for a more ordinal or binary breakdown (ex. changing "5" to "2" for no financial help).

2. How many respondents have a valid will and/or trust?
    a. RT001 (Do you currently have a will that is written and witnessed?)
        i. 41 -8. Web non-response
        ii. 5987 1. [YES, WILL/YES, WILL ONLY]
        iii. 444 2. [YES, WILL AND TRUST/YES, BOTH WILL AND TRUST]
        iv. 144 3. [NO WILL, BUT HAVE TRUST/NO, TRUST ONLY]
        v. 8817 5. [NO WILL/NO, NEITHER WILL OR TRUST]
        vi. 56 8. DK (Don't Know); NA (Not Ascertained)
        vii. 65 9. RF (Refused)
        viii. 169 Blank. INAP (Inapplicable); Partial Interview

Initial quantitative/qualitative analysis of responses for Query 2

With this nominal response being our only relevant one to our query, it is refreshing to see the vast majority of responses are not null. The majority of our data is in two responses, with 38% of the respondents having a Will and no Trust, and 56% having neither a Will or a Trust. It will be interesting to see the potential correlations between these answers and other queries. Several of our other queries assume an individual has a Will in the first place and we hope that 42% of our respondents having something in place is enough to create and confirm useful estate planning hypotheses. The latter response of having no Will/Trust is ripe for our business but may need some convincing. We also have a curious category of "Web non-response". From HRS:

In the web version of the questionnaire, respondents are allowed to leave any given question that was presented on the screen unanswered to comply with the voluntary nature of the survey. This was distinctly different from the interviewer-administered version of the questionnaire which did not allow questions to be "empty." If the respondent could not or refused to answer, the interviewer could select "Don't Know" or "Refused" as appropriate.

The above adds to our curiosity of the number of blanks we have seen so far. We will likely commingle these non-answers as we are primarily focused on those respondents who we know their estate planning status.

3. What is the breakdown of beneficiaries for respondents' estates?
    a. RT003 (Does that include any of your children or step-children?)
        i. 1 -8. Web non-response
        ii. 5213 1. YES
        iii. 152 5. NO
        iv. 5 8. DK (Don't Know); NA (Not Ascertained)
        v. 8 9. RF (Refused)
        vi. 10344 Blank. INAP (Inapplicable); Partial Interview
    b. RT004M1 (Which first child is included in Will?)
        i. 2344 3-39. Person Index Number
        ii. 10 98. DK (Don't Know); NA (Not Ascertained)
        iii. 52 99. RF (Refused)
        iv. 10521 Blank. INAP (Inapplicable); Partial Interview

c. RT004M2 (Which second child is included in Will?)

i. 1377 3-39. Person Index Number
ii. 0 98. DK (Don't Know); NA (Not Ascertained)
iii. 0 99. RF (Refused)
iv. 14321 Blank. INAP (Inapplicable); Partial Interview

d. RT004M3 (Which third child is included in Will?)

i. 491 3-39. Person Index Number
ii. 0 98. DK (Don't Know); NA (Not Ascertained)
iii. 0 99. RF (Refused)
iv. 15173 Blank. INAP (Inapplicable); Partial Interview

e. RT004M4 (Which fourth child is included in Will?)

i. 178 3-39. Person Index Number
ii. 0 98. DK (Don't Know); NA (Not Ascertained)
iii. 0 99. RF (Refused)
iv. 15173 Blank. INAP (Inapplicable); Partial Interview

f. RT005 (Does your Will provide for all your children equally?)

i. 6 -8. Web non-response
ii. 2938 1. YES
iii. 280 5. NO
iv. 10 8. DK (Don't Know); NA (Not Ascertained)
v. 5 9. RF (Refused)
vi. 12484 Blank. INAP (Inapplicable); Partial Interview

g. RT006 (Does your Will include grandchildren?)

i. 8 -8. Web non-response
ii. 750 1. YES
iii. 235 2. ONLY THROUGH THEIR PARENTS
iv. 3124 5. NO
v. 25 8. DK (Don't Know); NA (Not Ascertained)
vi. 19 9. RF (Refused)
vii. 11562 Blank. INAP (Inapplicable); Partial Interview

Initial quantitative/qualitative analysis of responses for Query 3

It is expected that, for most people, if you have children, you will include them in your estate planning, unless they did something terrible to make you cut them out. Our answers above majority null (66%), but nearly all of those responding did include their children. We assume that null responses include those without a Will, although it might be beneficial to add this option to paint a fuller picture. Alternatively, and in combination with the next four questions regarding number of children in a Will, it might be best just to have the respondent answer "how many children are listed in your Will?" with a ratio valued response. A directive could be used to list "0" if you do not have a Will or do not list any children in your Will.

Instead, we get four questions of the number of children listed in the Will. These inherently decrease in response as number increases, as few families have that many children. We have

excluded children numbers 5-10 as they decrease towards 0 responses quickly. It is interesting to note that the first child is not included by respondents as often as they responded affirmatively to including any children. This could either be confusion with the survey or possibly a noticeable amount of first children being skipped in a Will. Also, "Person Index Number" is not clear from the surveyors notes as to why it has a range of 3-39. We might recode these with the assumption that they are binary.

In terms of distribution among children, the results mirror an affirmative hypothesis that yes, children are highly likely to receive the same benefits in the eyes of their parents. We can assume that respondents would not answer if they have one or no children (based on our results). This explains why we have approximately 2000 less responses to this question than the first question asking if a Will includes children at all.

Finally, we review a binary question regarding grandchildren included in the Will. With only approximately 1000 positive responses, its clearer that grandchildren are less likely (about one fifth) to be included than a person's children. However, this also assumes that not having grandchildren means you would answer as null. The impact of having a grandchild on our clients may or may not be negligible, however, this reminds us that our relatively elderly demographics (which really begs us to ask about respondent age!)

In conclusion, we are again disappointed in the many blank responses in this section. However, we have far fewer null responses in this section which will hopefully lead us to stronger conclusions. We will be interested in how the inclusion of these beneficiaries and how much they will receive relate to other demographics of our prospective clients. We thought about additional questions here regarding "how much you are giving to your child", but we often see wills leave tangible property or residual amounts as the gift. These are hard to quantify, especially when the client/respondent is still alive.

4. What amount of respondents include some form of charitable giving?
    a. RT008 (Have you made provisions for any charities in your will or trust?)
        i. 14 -8. Web non-response
        ii. 692 1. YES
        iii. 5807 5. NO
        iv. 35 8. DK (Don't Know); NA (Not Ascertained)
        v. 26 9. RF (Refused)
        vi. 9149 Blank. INAP (Inapplicable); Partial Interview
    b. RQ454 (Total amount donated to Charity)
        i. N: 3797
        ii. Min: $0
        iii. Max: $201,000
        iv. Mean: $3,968.31
        v. SD: $8,732.86
        vi. Missing: 7150

Initial quantitative/qualitative analysis of responses for Query 4
The breakdown of charity giving in a Will or Trust might be lower than expected, but we can be happy that more than a third of respondents answered the binary question. The totals given to charity appear to be quite low, but one might imagine that really wealthy individuals make the bulk of donations (although we might then expect a higher standard deviation from a low mean).

Finally, we are reminded of our above quandary, as its often hard to quantify how much will be donated to beneficiaries. The amounts above (should) illustrate giving during a person's life. While it might be expected these individuals might also include further giving in their Will, it will be interesting to see the connection between these individuals and their demographic beneficiary breakdown. Will the rich give away more of their funds?

5. What amount of respondents have some form of life insurance as part of their estate planning?
    a. RT011 (Do you have any life insurance, including individual or group policies?)
        i. 39 -8. Web non-response
        ii. 8069 1. YES
        iii. 7271 5. NO
        iv. 86 8. DK (Don't Know); NA (Not Ascertained)
        v. 88 9. RF (Refused)
        vi. 170 Blank. INAP (Inapplicable); Partial Interview
    b. RT013 (What is the total face value of your life insurance policies?)
        i. N: 6762
        ii. Min: $0
        iii. Max: $150,000,000
        iv. Mean: $198,187.25
        v. SD: $2,650,299.68
        vi. Missing: 7654

Initial quantitative/qualitative analysis of responses for Query 5
Life insurance is a common safety net for our clients. It's generally not taxed like other assets and pays out to beneficiaries like other low-risk investments. When it comes to life insurance, we generally don't see these policies listed in estate planning documents. Thus, you could hypothetically give your children nothing in a Will, but list them on a massive life insurance policy. We want to see how these respondents above interact the previous queries, and if these are the same people with extensive estate planning giving measures. We can then decide how to recruit and advise clients who are thinking about or already have life insurance policies.

In terms of numerical patterns seen above, we have a striking range of policy valuations. Our mean appears to be low, in that the amount can't buy a house (in our area anyway), but would still be a nice gift to a child. Our amount of responses for these two questions are extremely positive, although we might wonder where the large delta went between answering Yes to having life insurance and not answering their value.

With all of the data attributes in our reduced set above, no invalid values or faulty data types exist. This might indicate that this dataset has already been pre-processed by the researchers. However, missing values and outliers are abundant. We will review these further for distribution and possible solutions in our following sections.

INVESTIGATING FOR ISSUES

Removing Duplicates

One of our first orders of business is to review our merged dataset (reduced above) for wholly duplicate values across tuples. We will use python to review the compiled set and remove all duplicates if necessary (keeping the first iteration). Please see Appendix for all code.

Columns: [HHID, RSUBHH, QSUBHH, RPN_SP, RT001, RT003, RT004M1, RT004M2, RT004M3, RT004M4, RT005, RT006, RT008, RT011, RT013, RQ020, RB017M, RB020, RB022, RB089M1M, RQ454]

We will also review our subset of our identification columns for duplicates and then see if these need to be further explained. Repeating our step above, we find none. This confirms our inner merges were successful and no tuples are wholly, or partially based on id, duplicate.

Reviewing Irrelevant Inputs

As discussed above, we have a few non-null values coded with numerical content that we might wish to remove. "Web non-response" (always coded as "-8"), "DK (Don't Know);NA (Not Ascertained)" (always coded as "8", or "98" in attributes with "Person Index Number"), and "RF (Refused)" (always coded as "9", or "99" in attributes with "Person Index Number") are all minimally different from null values to our analysis. These will be wholly replaced with null values in the next stage.

Attribute counts of "-8":
RT001 41
RT003 1
RT005 6
RT006 8
RT008 14
RT011 39

Attribute counts of "8":
RT001 56
RT003 5
RT004M1 52 (includes "Person Index Number")
RT004M2 49 (includes "Person Index Number")
RT004M3 31 (includes "Person Index Number")
RT004M4 34 (includes "Person Index Number")
RT005 10
RT006 25
RT008 37
RT011 86
RB017M 1
RB020 4
RB022 7

Attribute counts of "9":

RT001 67
RT003 8
RT004M1 41 (includes “Person Index Number”)
RT004M2 28 (includes “Person Index Number”)
RT004M3 23 (includes “Person Index Number”)
RT004M4 36 (includes “Person Index Number”)
RT005 5
RT006 20
RT008 27
RT011 90
RQ020 2
RB020 2

Attribute counts of “98”:
RT004M1 13

Attribute counts of “99”:
RT004M1 54

We can also review our subset of non-id attributes for their unique in python:
RT001 7
RT003 5
RT004M1 20
RT004M2 18
RT004M3 19
RT004M4 18
RT005 5
RT006 6
RT008 5
RT011 5
RT013 364 (continuous)
RQ020 411 (continuous)
RB017M 5
RB020 5
RB022 3
RB089M1M 3
RQ454 231 (continuous)

All of our values above match our expected output from our previous section. All are numeric float values and no unwanted strings exist. RT004M1-M4 include unknown personal identifiers, which we will reduce to a single value in our next section.

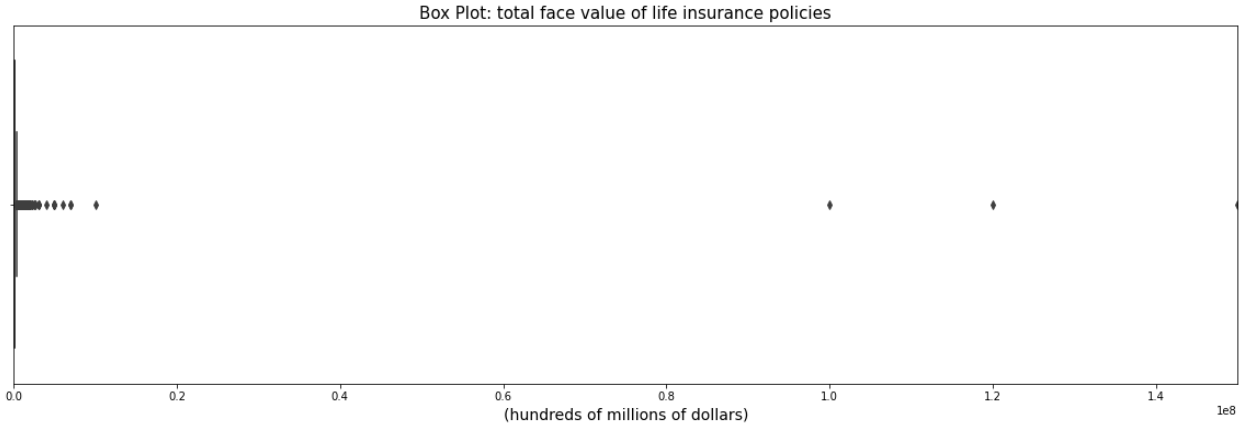
Reviewing Outliers

We will review our three continuous variables for outliers, and then decide what to do with them in our dataset based on likely causes and effects. We will use Seaborn’s boxplot function to visualize those values outside of the whiskers (ignoring null values for this evaluation).

Our first is RT013 (total face value of life insurance policies). We expected to find values between 0 and 150,000,000, per the published findings. However, we have many values off the chart:

count 1.139000e+03
mean 1.000000e+09
std 4.900053e-01
min 0
25% 1.000000e+09
50% 1.000000e+09
75% 1.000000e+09
max 1.000000e+09

These values appear to be an error in coding (uniformly 1.000000e+09, which we will label as an integrity constraint violation) and will need to be removed at our later stage. For now, let's review our boxplot without them:



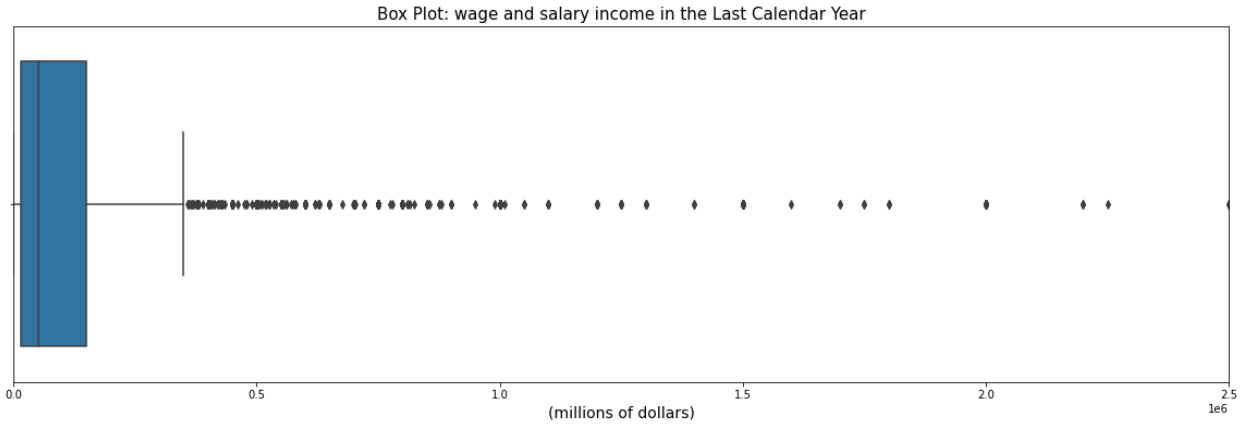
We now have four values in the hundreds of millions range, which are clearly many orders outside of the IQR. We may drop these values in our next stage, but these are at least realistic values that a (wealthy) individual could have for insurance.

Next is RQ020 (wage and salary income in the Last Calendar Year). We expected to find values between 0 and 2,500,000, per the published findings. However, we have many values off the chart:

count 6.810000e+02
mean 9.999998e+06
std 4.990706e-01
min 0
25% 9.999998e+06
50% 9.999998e+06
75% 9.999999e+06
max 9.999999e+06

While a salary of 10,000,000 isn't impossible, these values appear to be an error in coding (uniformly 9.999999e+06, which we will label as an integrity constraint violation) and will need

to be removed at our later stage. Even if these indicate the individual made more than the recordable maximum, we can't compare these values against our continuous in-range values. For now, let's review our boxplot without them:

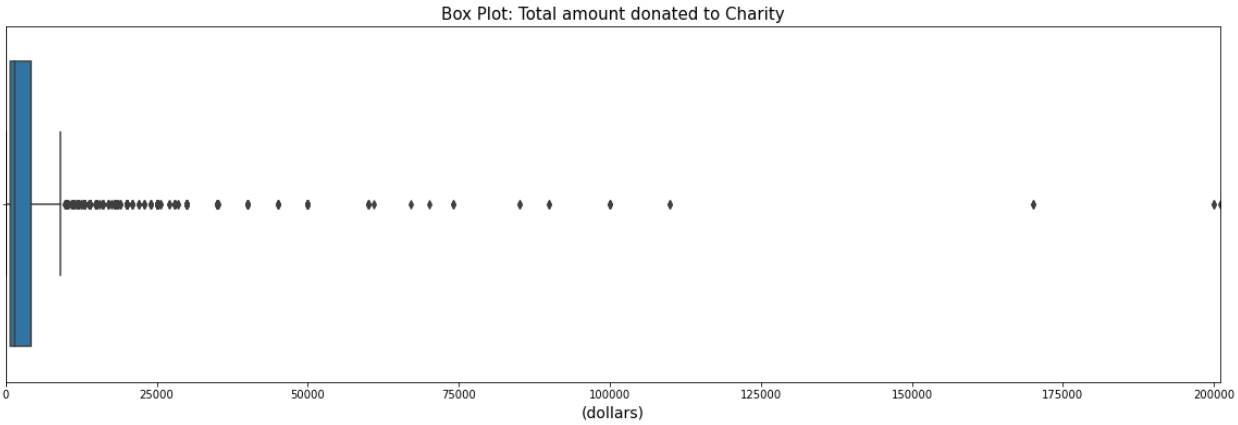


We now have many values outside of our whiskers. We may drop these values in our next stage, but these are at least realistic values that a (wealthy) individual could have for income.

Our final attribute is RQ454 (Total amount donated to Charity). We expected to find values between 0 and 201,000, per the published findings. However, we have many values off the chart:

count	643.000000
mean	999998.292379
std	0.455210
min	999998.000000
25%	999998.000000
50%	999998.000000
75%	999999.000000
max	999999.000000

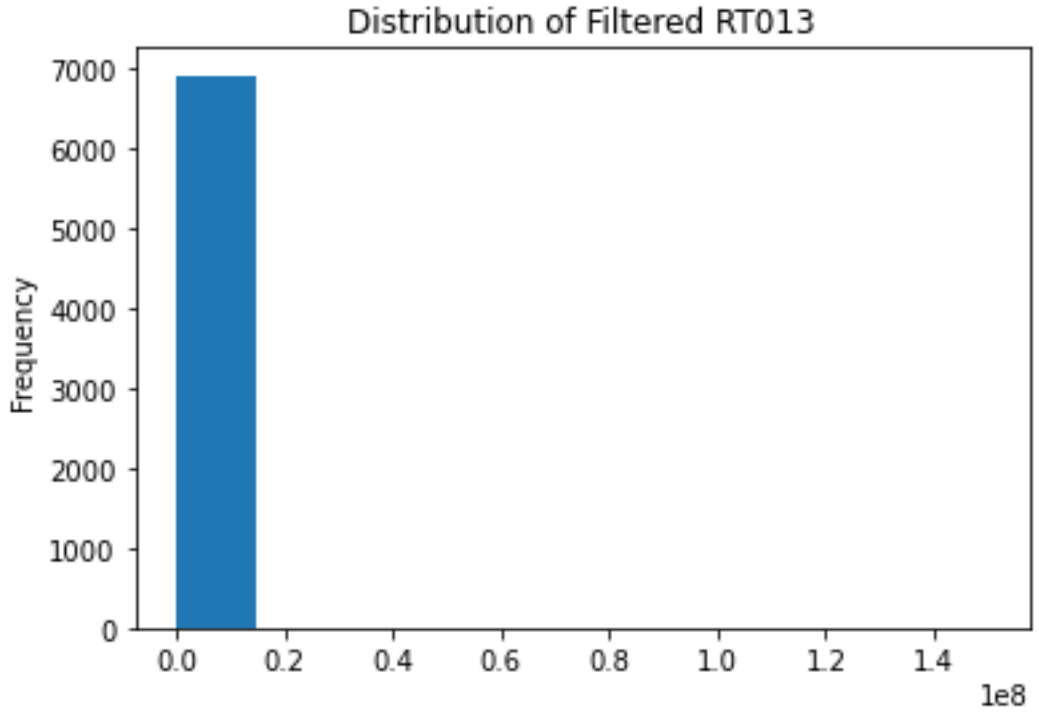
While a donation of $999,999 isn't impossible, these values appear to be an error in coding (uniformly ~999999, which we will label as an integrity constraint violation) and will need to be removed at our later stage. Even if these indicate the individual donated more than the recordable maximum, we can't compare these values against our continuous in-range values. For now, let's review our boxplot without them:



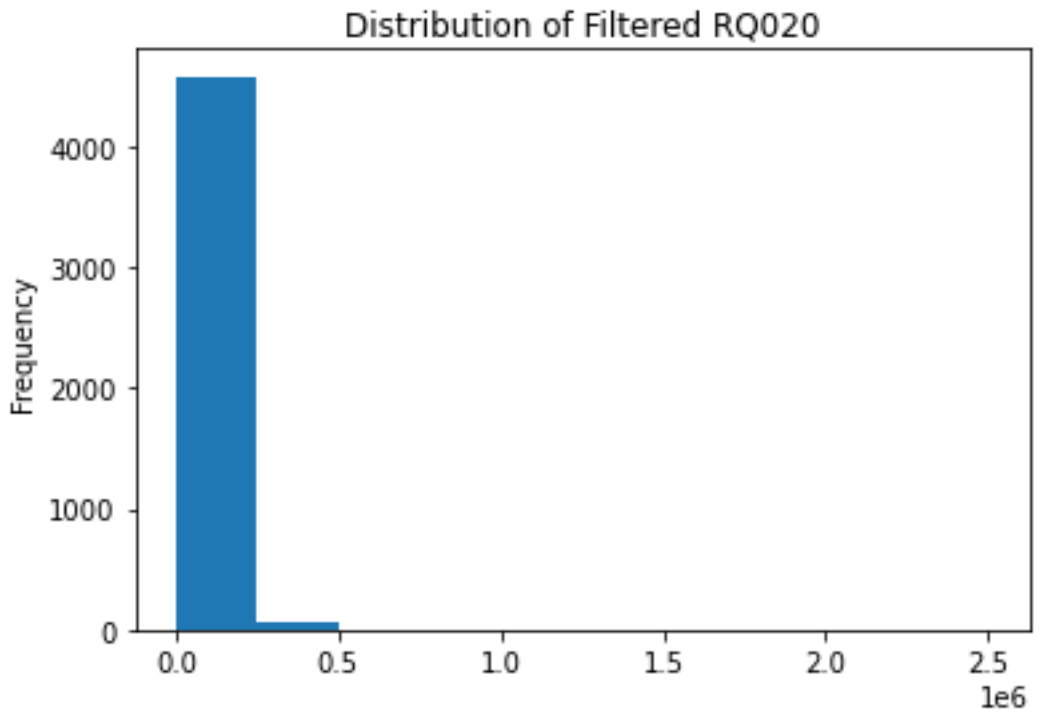
We again have many values outside of our whiskers. We may drop these values in our next stage, but these are at least realistic values that a (wealthy) individual could have for donations.

Numerical Distribution

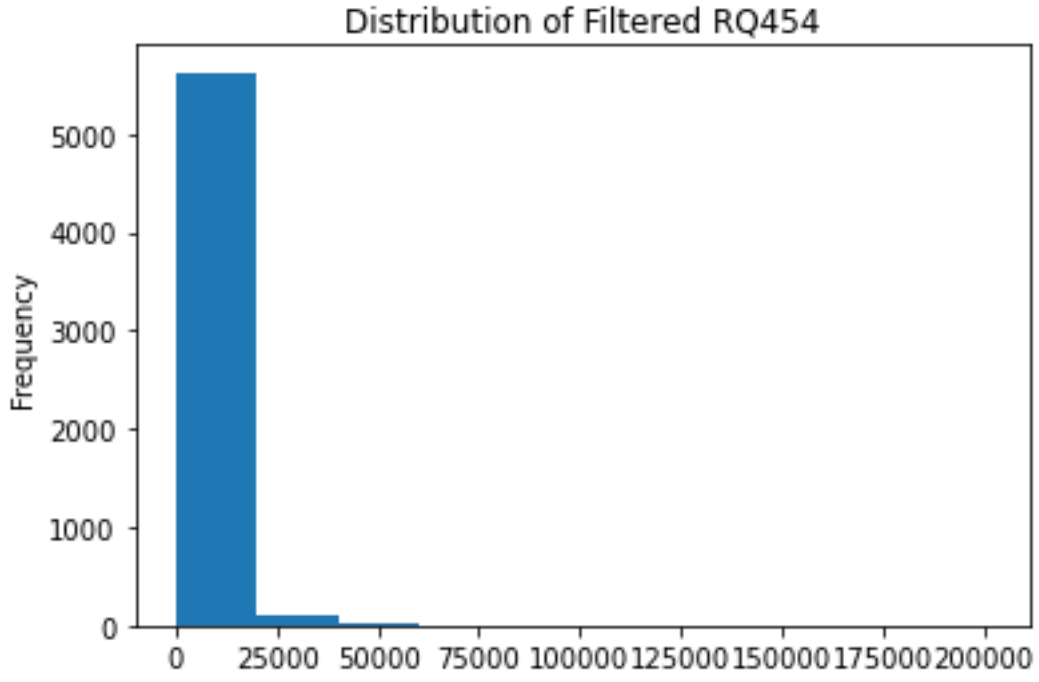
Reviewing the distribution of our four numerical attributes, RT013 (filtered for the integrity constraint violations above) has the below distribution curve and a skewness score of 48.86. Despite being filtered for the extreme outliers, the data is skewed extremely to the right.



RQ020 (filtered for the integrity constraint violations above) has the below distribution curve and a skewness score of 16.01. Despite being filtered for the extreme outliers, the data is also skewed extremely to the right.



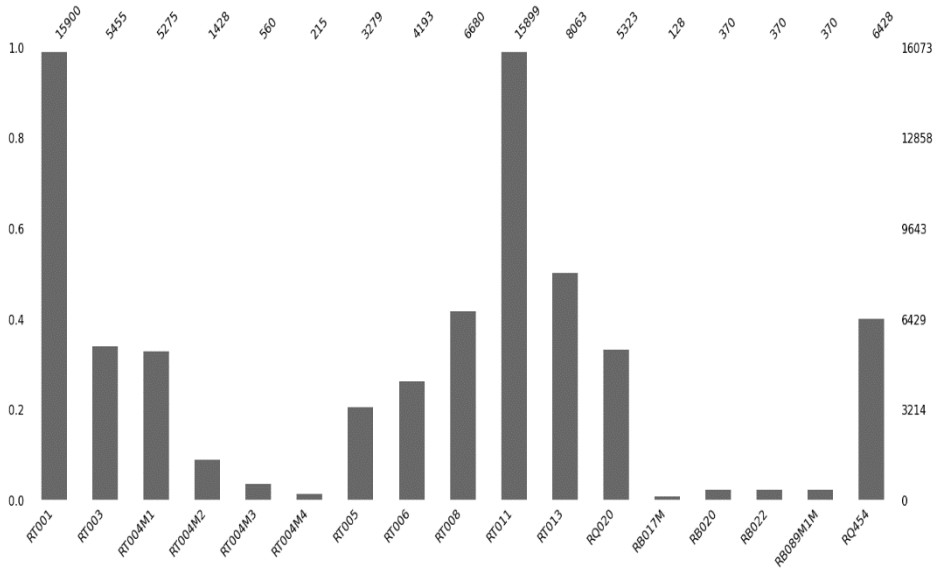
RQ454 (filtered for the integrity constraint violations above) has the below distribution curve and a skewness score of 10.36. Despite being filtered for the extreme outliers, the data is also skewed extremely to the right.



Null Values

Finally, we review our data for missing null values, which we know are prevalent. Using the missingno library in python for visualization of the non-id attributes:

RT001 173
RT003 10618
RT004M1 10798
RT004M2 14645
RT004M3 15513
RT004M4 15858
RT005 12794
RT006 11880
RT008 9393
RT011 174
RT013 8010
RQ020 10750
RB017M 15945
RB020 15703
RB022 15703
RB089M1M 15703
RQ454 9645



Missingno's bar chart uses a y-axis range from 0.0 to 1.0, where 1.0 represents 100% data completeness (few null values). The majority of our attributes have few actual values to work with, however some stick out as barely existing for our analysis. RT004M1-4 represent our children included in a Will, which naturally decrease as n children included (less families have 4 children than at least 1 to include in a Will). RT005 and RT006 are also about the distribution of children and grandchildren in the Will, and are again lacking values. Our demographic values on the right side are also lacking. Assuming we don't drop these, this calls into question how we might include them in our analysis. For example, it is hard for us to correlate those who donate a lot to charity by their education status since the values are generally missing.

Also, we have 155 tuples where the entire row of non-id attributes are null (which may increase after cleaning). These will be dropped.

As always, we are hesitant to drop those respondents who are missing some or even most of their inputs. Even a percentage of responses can be sliced to focus on their impacts among one or a few attributes.

Conclusion

HRS as an academic study naturally involves human interviewers and reviewers (even for those web-based responses), who make natural mistakes we can't control for. Additionally, participants are not forced to answer questions and the result is the massive amount of null values above. We might attempt to entice interviewees or control for these issues if we collected this data in the future. A high-level review of past HRS studies shows similar issues among our three sub-datasets, so this is not a temporary problem. However, this is an early release of the coded responses and some null values may be manually fixed in later releases:

> *"Other Specify" and "Open End", or questions that are answered with text (e.g., vocabulary words, industry and occupation) are not included in Early data releases. In many instances, these appear as blanks in the data but sometimes they are designated with "Data Not Available" text. Data from these types of questions should be available, where possible, in the Final data release.*

HRS Codebook January 2022, Version 1.0, p. 15.

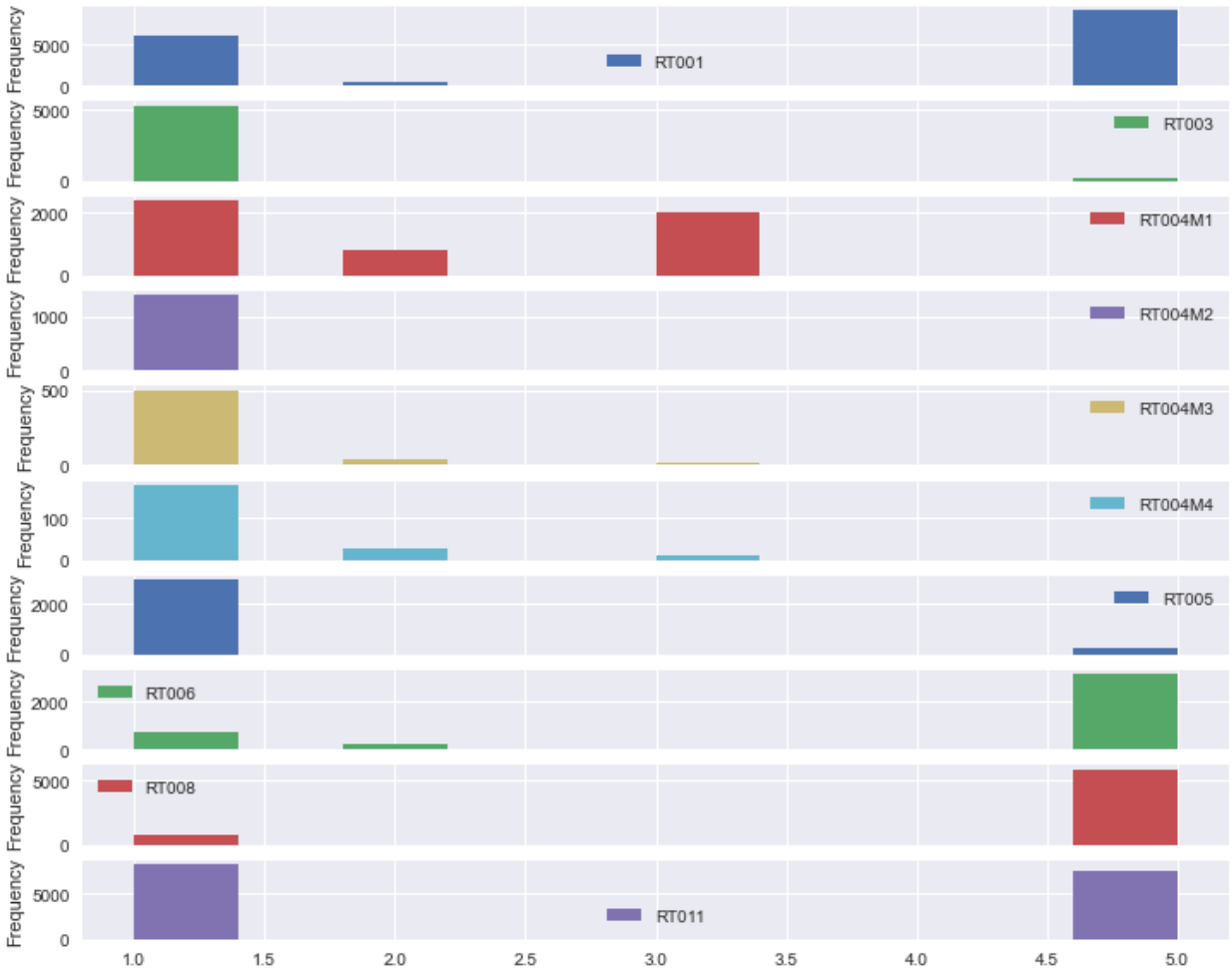
Manual review and cleaning is at the heart of fixing *garbage in, garbage out*. Thus, we might revisit this study at a later date to review error corrections in the data to hopefully make our analysis more robust.

CLEANING DATA

Removing Irrelevant Inputs

As discussed above, we have a few non-null values coded with numerical content that we might wish to remove. "Web non-response" (always coded as "-8"), "DK (Don't Know);NA (Not Ascertained)" (always coded as "8", or "98" in attributes with "Person Index Number"), and "RF (Refused)" (always coded as "9", or "99" in attributes with "Person Index Number") are all minimally different from null values to our analysis. Before we replace these with null values, we will substitute all non-null Person Index Numbers to "1" for easier batch processing. We will also replace "93" for "ALL CHILDREN EQUALLY" and "96" for "ALL CHILDREN - "EQUALLY" NOT MENTIONED" to "2" and "3" respectively in our attributes for children mentioned in Will.

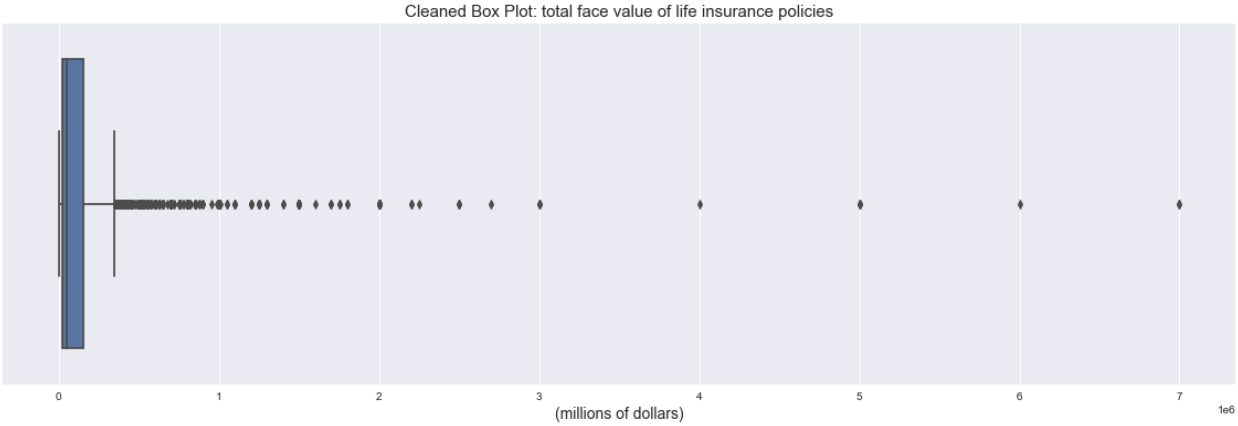
Once this is accomplished, we will replace "-8", "8", "9", "98", and "99" across the dataframe with np.nan in python. While we are at it, we can replace the irrelevant inputs in our continuous variables: "9999998" and "9999999" from RQ020, "999999998" and "999999999" from RT013, and "999998" and "999999" from RQ454 for the same reasons. See Appendix for code and resulting 0 counts post-cleaning. Our histograms of our ordinal attributes post-cleaning:



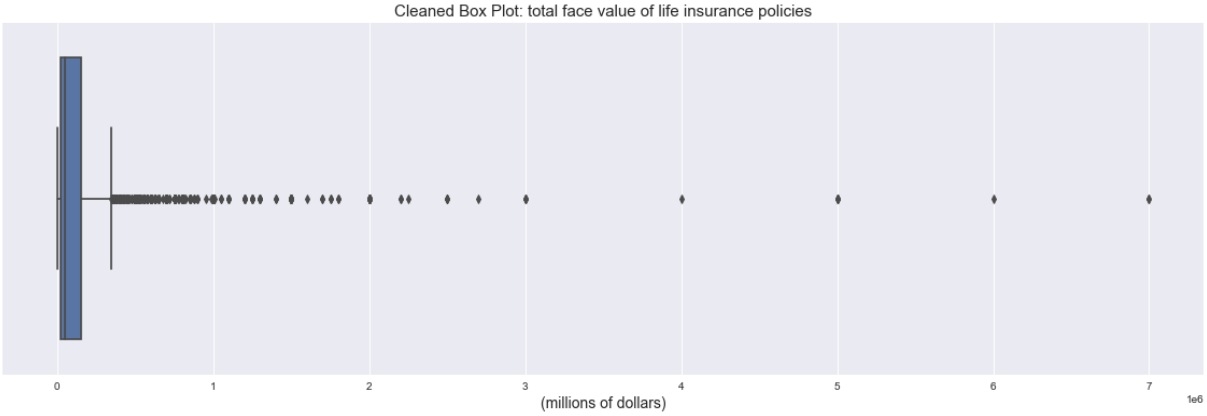
Dealing with Outliers

Our first continuous values for potential outlier cleaning is RT013 (total face value of life insurance policies). We now have four values in the hundreds of millions range, which are many orders outside of the IQR. These are at least realistic values that a (wealthy) individual could have for insurance. We do not know the underlying distribution of the observations (but they are heavily skewed, per above) and thus used the nonparametric outlier detection method by means of the IQR (InterQuartile Range). If the outlier creates a strong association, we should drop the outlier and not report any association from our analysis.

With only four tuples, we can easily review their values (or lack thereof) to see their impact in further analysis. Only five attributes have values (including RT013) and thus this variable's impact on other variables is minimal. Also, regression analysis, or other machine learning algorithms will likely be unnecessary to our decision to drop these outliers. All four do not have Wills (RT001), and they logically have life insurance policies (RT011). However, our most influential logical attribute is that all four wage and salary income in the Last Calendar Year (RQ020) average $67,333, with a maximum of $150,000. Thus, its logical to drop these outlier life insurance values as being incongruent to their income. Our resulting boxplot (still skewed heavily to the right with a skewness score of 9.35):

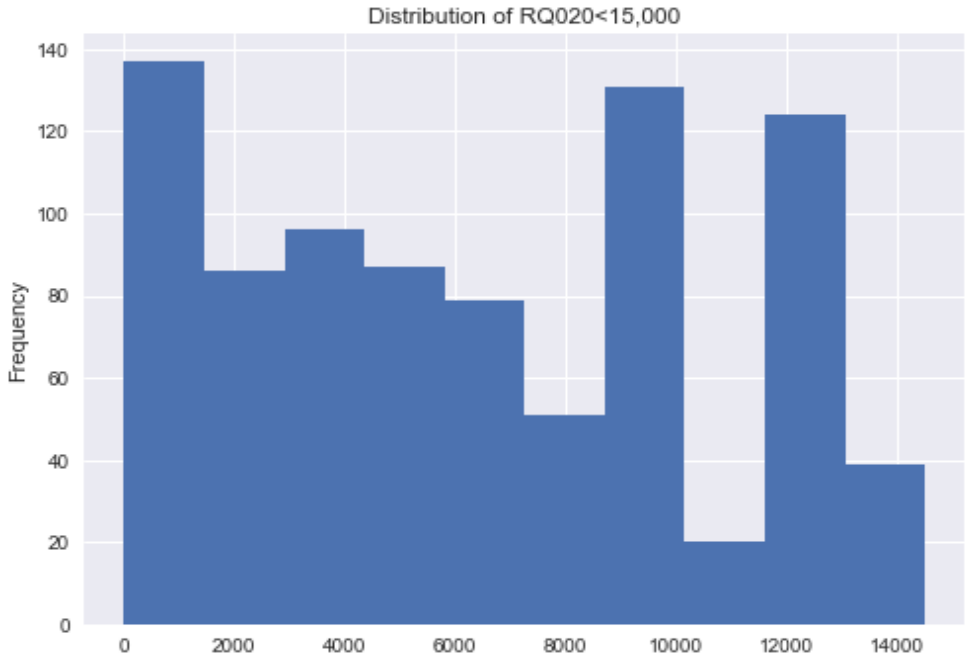


After our first pass of RT013, we will turn our attention to the potential of semantic anomalies in small value life insurance policies. An aggregate search of minimum policy values on the market place gives us a baseline of $5,000[1]. To be safe, we will choose any values less than $1,000 to remove (but keeping $0). We then replace the resulting 26 values with null values:

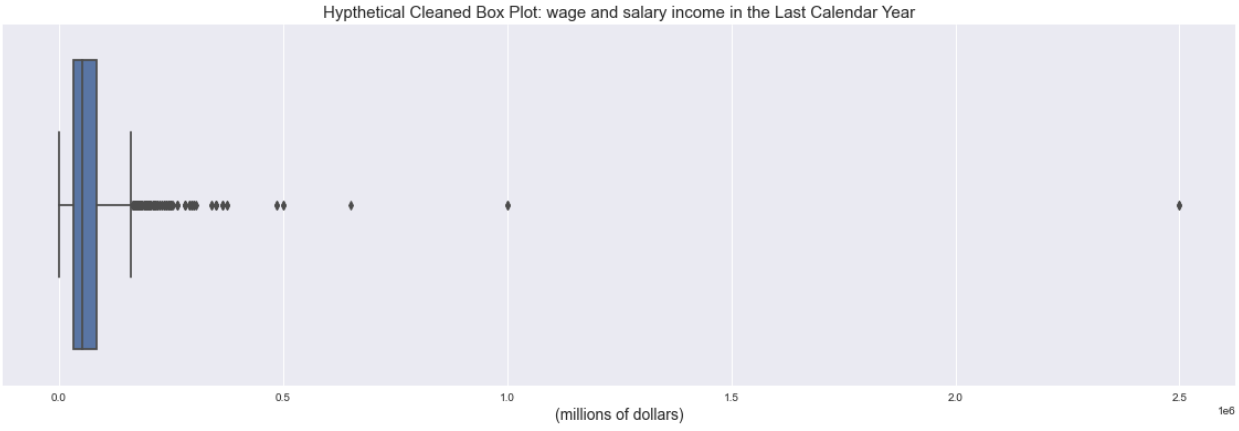


[1] https://www.usnews.com/insurance/life-insurance/cheap-life-insurance

Next is RQ020 (wage and salary income in the Last Calendar Year). We do not have the egregious outliers as above, but we might still be able to remove outliers from the lower end, even if they are realistically possible (i.e. a senior citizen making a few hundred dollars in extra income). Assuming a minimum wage salary of $15,000 per year, we can set our baseline using the same cleaning as before. However, this baseline produced 839 attributes with potential semantic anomalies:



Replacing these values with $0 will only increase skewness and decrease the real-world accuracy of our data. We might hypothesize that the age attribute (which we don't have) is skewed towards those in retirement. With an N of only 3118, we will leave these outliers alone with a skewness score of 16.01 to prevent loss of potential influential conclusions. Replacing these with null values will drastically alter our analysis, but we have provided a hypothetical boxplot below to visualize this (even more skewed at 16.58).



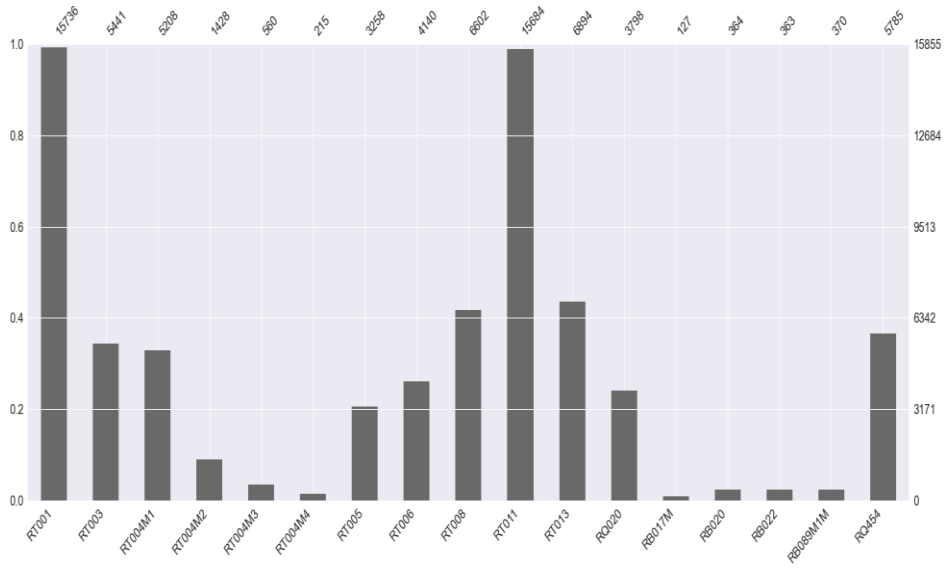
Our final numeric attribute is RQ454 (Total amount donated to Charity). Like with wage and salary income, all of the numbers currently in the dataset are logical and valid. Individuals can

give any amount to charity they feel like. Everything said above is even more true for this variable, and we will choose not to remove outliers to maintain data integrity.

Null Values

Since we have made additional records null, we will re-review our dataset for missing values. Prior to analysis, we can drop tuples that are now completely null and get the resulting summary:

RT001	119
RT003	10414
RT004M1	10647
RT004M2	14427
RT004M3	15295
RT004M4	15640
RT005	12597
RT006	11715
RT008	9253
RT011	171
RT013	8961
RQ020	12057
RB017M	15728
RB020	15491
RB022	15492
RB089M1M	15485
RQ454	10070



Again, Missingno's bar chart uses a y-axis range from 0.0 to 1.0, where 1.0 represents 100% data completeness (few null values). The ratios from initial investigation till now remain mostly the same across all attributes, but with some slight reductions based on our dropping values. We will review each attribute for its likely reason(s) for containing null values below.

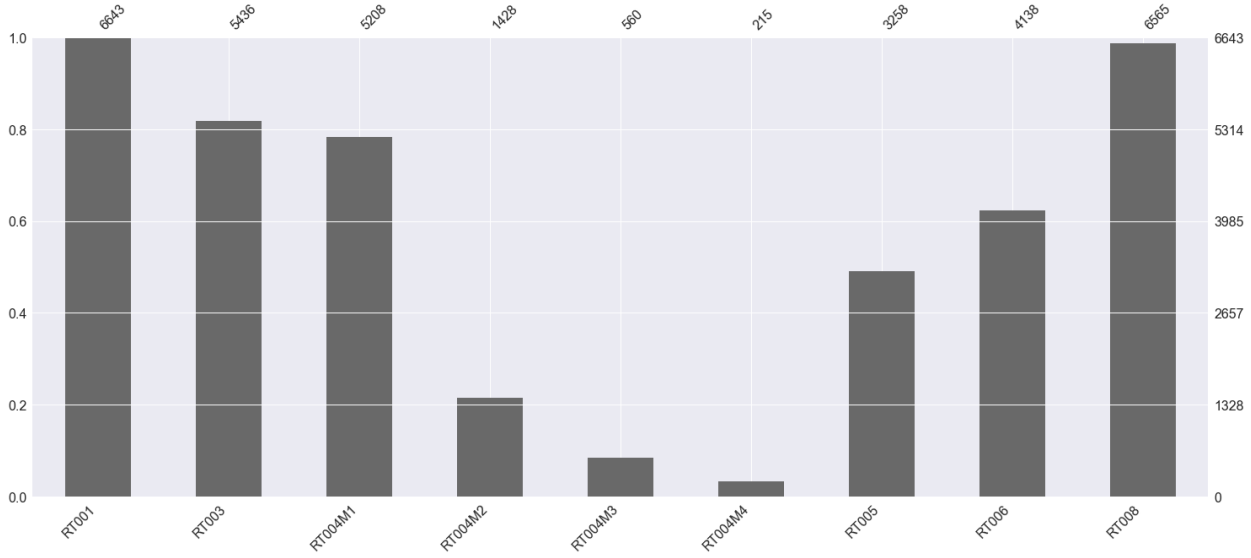
Structurally Missing Data

The majority of our data has a structural dependency on the respondent's answer to RT001 (Do you currently have a will that is written and witnessed?). With only 119 out of 15852 tuples as now null, RT001 provides us with a nearly complete picture in our dataset. If 'no' (coded as 3 or 5, with or without Trust), then the following attributes are moot:

1. RT003 (Does that include any of your children or step-children?) – 9088 out of 10414 missing values are structurally missing due to having no Will (87.27%)
2. RT004M1 (Which first child is included in Will?) 9093 out of 10647 (85.40%)
3. RT004M2 (Which second child is included in Will?) 9165 out of 14427(63.53%)
4. RT004M3 (Which third child is included in Will?) 9188 out of 15295(60.07%)
5. RT004M4 (Which fourth child is included in Will?) 9198 out of 15640(58.81%)

6. RT005 (Does your Will provide for all your children equally?) 9135 out of 12597 (72.52%)
7. RT006 (Does your Will include grandchildren?) 9117 out of 11715 (77.82%)
8. RT008 (Have you made provisions for any charities in your will?) 9056 out of 9253 (97.87%)

Additionally, RT004M1-4 represent our children included in a Will, which naturally decrease as n children included (less families have 4 children than at least 1 to include in a Will). When the above missing values for RT004M2-M4 are combined with when a first child is not included in a Will (and likely no additional child exists), we have increases to 74.30% (RT004M2), 70.23% (RT004M2), and 68.75% (RT004M2). These ratios when sliced look quite a bit more appealing in terms of remaining missing values not explained by these structurally missing data:



A breakdown of the resulting sliced values where a Will is present:

	RT001	RT003	RT004M1	RT004M2	RT004M3	RT004M4	RT005	RT006	RT008
1	93.13%	97.17%	45.70%	98.20%	89.93%	82.94%	91.41%	18.13%	10.66%
2	6.87%	N/A	15.81%	1.37%	7.51%	12.32%	N/A	5.72%	N/A
3	N/A	N/A	38.50%	0.43%	2.56%	4.74%	N/A	N/A	N/A
5	N/A	2.83%	N/A	N/A	N/A	N/A	8.59%	76.15%	89.34%

We can also apply this structural dependency assumption to RT011 (Do you have any life insurance, including individual or group policies?) and RT013 (What is the total face value of your life insurance policies?). 7594 (84.75%) of the missing values from RT013 can be explained as structurally missing because the respondent doesn't have life insurance in the first place.

Multiple linear regression was attempted on these fields with missing values dropped, but each pass fails due to not having at least 2 classes in the data with this assumption. The remaining missing values are thus likely caused by NMAR, as the reason for these null values based on respondent answers could be due to some reason we just do not know. We will not delete,

impute, or replace these as they should be handled by going back to the survey and obtaining more information. As above, this dataset is still in early release and will likely have more complete data at a later date.

Turning to our remaining five demographic values, we have a bleak picture of missing values. They are missing values an average nearly 89% among tuples and dropping any rows is out of the question for this reason. Like above, these are likely caused NMAR and we need more information from those who participated in the survey. For example, it is hard for us to correlate those who donate a lot to charity by their education status since the values are generally missing.

As always, we are hesitant to drop those respondents who are missing some or even most of their inputs. Even a percentage of responses can be sliced to focus on their impacts among one or a few attributes.

CONCLUSION

Because this dataset is still in early release and will likely have more complete data at a later date, it is hard to draw conclusions from this current dataset. We would love more complete frames with a fraction of the null values to more accurately derive conclusions. However, we note that having estate planning documents in place is actually a dominant theme among Americans. More regional and socio-economic demographic attributes would be tremendously helpful to tie into these respondents. We have a tough time currently describing our respondents by their demographics and applying that knowledge to an estate practice in Pennsylvania.

We can tell that children are logically included in many people's Wills and Trusts, but the breakdown of what each child gets as a percentage of the total estate is missing. It is recommended that these questions be added to the survey. Also, non-responses to all questions should be focused on for elimination based on empirical survey taking ideals. They might require that a respondent answer most/all questions or not be recorded at all. Even a "no" response (if accurate) to any of our attributes is valuable to our results.

REFERENCES

UNIV. OF MICH., HEALTH AND RETIREMENT STUDY: HEALTH AND RETIREMENT STUDY, 2020 Core, Early, Version 1.0, January 2022 (2022). Retrieved February 3, 2022, from https://hrsdata.isr.umich.edu/data-products/2020-hrs-core.

James III, Russell N., *The New Statistics of Estate Planning: Lifetime and Post-Mortem Wills, Trusts, and Charitable Planning*, 8 EST. PLAN. & COMMUNITY PROP. L.J. 1, 5 (2015)

APPENDIX

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import seaborn as sns
import missingno as msno
from sklearn.model_selection import train_test_split

#COLLECTING & MERGING DATA
h20b_r = pd.read_sas('h20b_r.sas7bdat') #convert Demographics sas file to csv
h20b_r.to_csv('h20b_r.csv', index=False)
h20q_h = pd.read_sas('h20q_h.sas7bdat') #convert Assets and Income sas file to csv
h20q_h.to_csv('h20q_h.csv', index=False)
h20t_r = pd.read_sas('h20t_r.sas7bdat') #convert Wills and Life Insurance sas file to csv
h20t_r.to_csv('h20t_r.csv', index=False)

#outer join Demographics and Assets and Income
merged_df = h20q_h.merge(h20b_r, how='outer', left_on=["HHID", "RSUBHH", "QSUBHH"],
right_on=["HHID", "RSUBHH", "QSUBHH"])

#outer join previous merge and Wills and Life Insurance
merged_df2 = h20t_r.merge(merged_df, how='outer', left_on=["HHID"], right_on=["HHID"])
merged_df2.to_csv('joinedall.csv', index = False)

df=merged_df2[['HHID', 'RSUBHH', 'QSUBHH', 'RPN_SP', 'RT001', 'RT003', 'RT004M1', 'RT004M2',
'RT004M3', 'RT004M4', 'RT005', 'RT006', 'RT008', 'RT011', 'RT013', 'RQ020', 'RB017M', 'RB020',
'RB022', 'RB089M1M', 'RQ454']] #reduced dataset

df["RT001"].value_counts()

df["RT003"].value_counts()

df["RT004M1"].value_counts()

df["RT004M2"].value_counts()

df["RT004M3"].value_counts()

df["RT004M4"].value_counts()

df["RT005"].value_counts()

df["RT006"].value_counts()

df["RT008"].value_counts()

df["RT011"].value_counts()
```

```
df["RT013"].describe()

df["RQ020"].describe()

df["RB017M"].value_counts()

df["RB020"].value_counts()

df["RB022"].value_counts()

df["RB089M1M"].value_counts()

df["RQ454"].describe()


df.isna().sum().sum() #19995268 missing values across the merged dataframe.

print(df[df.duplicated()])

#INVESTIGATING FOR ISSUES
df.drop_duplicates(keep="first",inplace=True)

print(df[df.duplicated()])

#number of irrelevent inputs
df[df == -8].count()

df[df == 8].count()

df[df == 9].count()

df[df == 98].count()

df[df == 99].count()

subset = df.iloc[:, 4:21]

subset.nunique(dropna=True)

subset.dtypes.value_counts()

# Box Plot RT013
filtered_RT013 = subset["RT013"][~np.isnan(subset["RT013"])]
fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RT013)
plt.title('Box Plot: total face value of life insurance policies', fontsize=15)
plt.xlabel('(hundreds of millions of dollars)', fontsize=14)
plt.show()

filtered_RT013.describe()
filtered_RT013[filtered_RT013 > 150000000].describe()
```

```
fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RT013[filtered_RT013 < 150000001])
plt.title('Box Plot: total face value of life insurance policies', fontsize=15)
plt.xlabel('(hundreds of millions of dollars)', fontsize=14)
ax.set_xlim(0, 150000001)
plt.show()

# Box Plot RQ020
filtered_RQ020 = subset["RQ020"][~np.isnan(subset["RQ020"])]
filtered_RQ020.describe()

filtered_RQ020[filtered_RQ020 > 2500000].describe()

fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RT013[filtered_RT013 < 2500001])
plt.title('Box Plot: wage and salary income in the Last Calendar Year', fontsize=15)
plt.xlabel('(millions of dollars)', fontsize=14)
ax.set_xlim(0, 2500001)
plt.show()

# Box Plot RQ454
filtered_RQ454 = subset["RQ454"][~np.isnan(subset["RQ454"])]
filtered_RQ454.describe()

filtered_RQ454[filtered_RQ454 > 201000].describe()

fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RQ454[filtered_RQ454 < 201001])
plt.title('Box Plot: Total amount donated to Charity', fontsize=15)
plt.xlabel('(dollars)', fontsize=14)
ax.set_xlim(0, 201001)
plt.show()

#Distribution of RT013
histRT013 = filtered_RT013[filtered_RT013 < 150000001].plot(kind='hist', title='Distribution of Filtered RT013')
filtered_RT013[filtered_RT013 < 150000001].skew() #extremely skewed

histRQ020 = filtered_RQ020[filtered_RQ020 < 2500001].plot(kind='hist', title='Distribution of Filtered RQ020')
filtered_RQ020[filtered_RQ020 < 2500001].skew() #extremely skewed

histRQ454 = filtered_RQ454[filtered_RQ454 < 201001].plot(kind='hist', title='Distribution of Filtered RQ454')
filtered_RQ454[filtered_RQ454 < 201001].skew() #extremely skewed


# Count of missing values of each column
subset.isna().sum()
```

```
msno.bar(subset)

subset.isnull().values.ravel().sum()

len(subset) - len(subset.dropna(how='all'))#entire rows of non-id attributes are null


#ACTUAL CLEANING
#replacing Person Index Number with "1"
clean = df
clean['RT004M1'] = np.where(clean['RT004M1'].between(3,39), 1, clean['RT004M1'])
clean['RT004M2'] = np.where(clean['RT004M2'].between(3,39), 1, clean['RT004M2'])
clean['RT004M3'] = np.where(clean['RT004M3'].between(3,39), 1, clean['RT004M3'])
clean['RT004M4'] = np.where(clean['RT004M4'].between(3,39), 1, clean['RT004M4'])
clean['RT004M1'] = np.where(clean['RT004M1'] == 93, 2, clean['RT004M1'])
clean['RT004M2'] = np.where(clean['RT004M2'] == 93, 2, clean['RT004M2'])
clean['RT004M3'] = np.where(clean['RT004M3'] == 93, 2, clean['RT004M3'])
clean['RT004M4'] = np.where(clean['RT004M4'] == 93, 2, clean['RT004M4'])
clean['RT004M1'] = np.where(clean['RT004M1'] == 96, 3, clean['RT004M1'])
clean['RT004M2'] = np.where(clean['RT004M2'] == 96, 3, clean['RT004M2'])
clean['RT004M3'] = np.where(clean['RT004M3'] == 96, 3, clean['RT004M3'])
clean['RT004M4'] = np.where(clean['RT004M4'] == 96, 3, clean['RT004M4'])
clean.iloc[:, 6:10].value_counts()

#clean Irrelevant Inputs
clean = clean.replace(-8, np.nan)
clean[clean == -8].count()

clean = clean.replace(8, np.nan)
clean[clean == 8].count()

clean = clean.replace(9, np.nan)
clean[clean == 9].count()

clean = clean.replace(98, np.nan)
clean[clean == 98].count()

clean = clean.replace(99, np.nan)
clean[clean == 99].count()

clean = clean.replace(9999998, np.nan)
clean[clean == 9999998].count()

clean = clean.replace(9999999, np.nan)
clean[clean == 9999999].count()

clean = clean.replace(999999998, np.nan)
clean[clean == 999999998].count()

clean = clean.replace(999999999, np.nan)
clean[clean == 999999999].count()
```

```
clean = clean.replace(999998, np.nan)
clean[clean == 999998].count()

clean = clean.replace(999999, np.nan)
clean[clean == 999999].count()

plt.style.use('seaborn')
clean.iloc[:, 4:14].plot.hist(subplots=True, legend=True, figsize=(12, 10))

#Dealing with Outliers - RT013
clean[clean['RT013'] > 7000000].describe() #5 attributes have values
clean.loc[clean["RT013"] > 7000000, "RT013"] = np.nan
clean[clean['RT013'] > 7000000].describe() #cleaned

#Distribution of Cleaned RT013
filtered_RT013 = clean["RT013"][~np.isnan(clean["RT013"])]
fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RT013)
plt.title('Cleaned Box Plot: total face value of life insurance policies', fontsize=15)
plt.xlabel('(millions of dollars)', fontsize=14)
plt.show()

histRT013 = clean["RT013"].plot(kind='hist', title='Distribution of Cleaned RT013')
clean["RT013"].skew() #still extremely skewed

clean[clean['RT013'].between(1,999)].describe() #26 attributes have semantic anomalies in small value life insurance policiesvalues
clean.loc[clean["RT013"].between(1,999), "RT013"] = np.nan
clean[clean['RT013'].between(1,999)].describe() #cleaned

#Distribution of Second Cleaned RT013
filtered_RT013 = clean["RT013"][~np.isnan(clean["RT013"])]
fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RT013)
plt.title('Cleaned Box Plot: total face value of life insurance policies', fontsize=15)
plt.xlabel('(millions of dollars)', fontsize=14)
plt.show()

histRT013 = clean["RT013"].plot(kind='hist', title='Distribution of Cleaned RT013')
clean["RT013"].skew() #still extremely skewed

#Dealing with Outliers - RQ020
clean[clean["RQ020"].between(1,14999)].describe() #839 attributes have potential semantic anomalies
MinWage = clean["RQ020"][clean["RQ020"] < 15000]
histRQ020 = MinWage.plot(kind='hist', title='Distribution of RQ020<15,000')

cleanHypo = clean
cleanHypo.loc[cleanHypo["RQ020"].between(1,14999), "RQ020"] = np.nan

#Distribution of Hypthetical Cleaned RQ020
```

```
filtered_RQ020 = cleanHypo["RQ020"][~np.isnan(cleanHypo["RQ020"])]
fig, ax = plt.subplots(figsize=(20,6))
sns.boxplot(x=filtered_RQ020)
plt.title('Hypthetical Cleaned Box Plot: wage and salary income in the Last Calendar Year', fontsize=15)
plt.xlabel('(millions of dollars)', fontsize=14)
plt.show()

cleanHypo["RQ020"].skew() #still extremely skewed

#Cleaning missing values
clean.dropna(how='all', subset = ['RT001', 'RT003', 'RT004M1', 'RT004M2','RT004M2','RT004M3','RT004M4','RT005', 'RT006', 'RT008', 'RT011', 'RT013', 'RQ020', 'RB017M', 'RB020', 'RB022', 'RB089M1M', 'RQ454', ], inplace = True)#remove rows of non-id attributes with null

subset = clean.iloc[:, 4:21]

subset.isna().sum()

msno.bar(subset)

#RT001
subsetNoWill = subset[subset["RT001"].isin([3, 5])]
subsetNoWill.isna().sum()

#RT004M1-M4
subsetNoKid = subset[(subset["RT004M1"] != 1) & (subset["RT004M1"] != 2) & (subset["RT004M1"] != 3) & (subset["RT001"] != 3) & (subset["RT001"] != 5)]
subsetNoKid.isna().sum()

subsetNoKidNoWill = subset[(subset["RT004M1"] == 1) | (subset["RT004M1"] == 2) | (subset["RT004M1"] == 3) | (subset["RT001"] == 1) | (subset["RT001"] == 2)]
subsetNoKidNoWill.isna().sum()

subsetNoKidNoWill = subsetNoKidNoWill.iloc[:, 0:9]
msno.bar(subsetNoKidNoWill)

#Will Relationships
subsetWill = subset[subset["RT001"].isin([1, 2])]
subsetWill = subsetWill.iloc[:, 0:9]

subsetWill.groupby("RT001")["RT001"].count()

subsetWill.groupby("RT003")["RT003"].count()

subsetWill.groupby("RT004M1")["RT004M1"].count()

subsetWill.groupby("RT004M2")["RT004M2"].count()

subsetWill.groupby("RT004M3")["RT004M3"].count()
```

```
subsetWill.groupby("RT005")["RT005"].count()

subsetWill.groupby("RT006")["RT006"].count()

subsetWill.groupby("RT008")["RT008"].count()

subsetWill.iloc[:,  2: 8]

# Logistic regression RT003
from sklearn import linear_model
#split the data into train and test sets with a ratio of 0.7:0.3.
cleanWill = subsetWill.dropna()

X = cleanWill[:,  2: 8]
y = cleanWill.RT003 #RT003 is our dependent variable
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42)

lr = linear_model.LogisticRegression()
lr.fit(X_train, y_train)
lr.coef_ #ValueError: This solver needs samples of at least 2 classes in the data, but the data contains only
one class: 1.0


#life insurance
subsetInsurance = clean[["RT011", "RT013"]]
subsetInsurance = subsetInsurance[(subsetInsurance["RT011"] != 1)]
subsetInsurance.isna().sum()

subsetNoKidNoWill = subsetNoKidNoWill.iloc[:, 0:9]
msno.bar(subsetNoKidNoWill)


clean.to_csv('clean.csv', index=False)


# dropping duplicate values
data = pd.read_csv("compressed data.csv")
len(data)

data.drop_duplicates(keep="first",inplace=True)

# length after removing duplicates
len(data)
data.to_csv('compressed data2.csv', index=False)


#Removing
```