NPCA Salaries Clean-Up Exercise

February 1, 2023

Daina Bouquin

[127]:
```
import pandas as pd
import numpy as np
import nbconvert
```

convert xlsx to dataframe

[2]:
```
df = pd.read_excel(r'SalarySurveyExercise.xlsx')
df.head(1)
```

[2]:
```
            Timestamp How old are you?  What industry do you work in?  \
0 2021-04-27 11:03:01           35-44  Accounting, Banking & Finance

           Job title  \
0  Senior Accountant

  If your job title needs additional context, please clarify here:  \
0                                                NaN

   What is your annual salary? (You'll indicate the currency in a later
question. If you are part-time or hourly, please enter an   \
0                                              45000

   How much additional monetary compensation do you get, if any (for example,
bonuses or overtime in an average year)? Please only  \
0                                                0.0

  Please indicate the currency If "Other," please indicate the currency here:  \
0                          USD                                             NaN

  If your income needs additional context, please provide it here:  \
0                        I work for a Charter School

  What country do you work in?  \
0                United States

  If you're in the U.S., what state do you work in? What city do you work in?  \
```

```
0                                          Florida                 Palm Coast

  How many years of professional work experience do you have overall?  \
0                                    21 - 30 years

  How many years of professional work experience do you have in your field?  \
0                                    21 - 30 years

  What is your highest level of education completed? What is your gender?  \
0                                  College degree                  Woman

  What is your race? (Choose all that apply.)
0  Hispanic, Latino, or Spanish origin, White
```

0.0.1 Create unique IDs

[3]:
```
df["id"] = df.index + 1 # add ID column
cols = df.columns.tolist() # columns to list to make rearranging them easier
cols = cols[-1:] + cols[:-1] # move ID column to the front
df = df[cols]
```

0.0.2 Rename columns

[4]:
```
# rename method 1
df.rename(columns={'Timestamp': 'timestamp',
                   'How old are you?': 'age_range',
                   'What industry do you work in?': 'industry',
                   'Job title': 'job',
                   'If your job title needs additional context, please clarify␣
 ↪here:': 'job_context',
                   'Please indicate the currency': 'currency',
                   'If "Other," please indicate the currency here:':␣
 ↪'other_currency',
                   'If your income needs additional context, please provide it␣
 ↪here:': 'income_context',
                   'What country do you work in?': 'country',
                   'What city do you work in?': 'city',
                   'How many years of professional work experience do you have␣
 ↪overall?': 'all_experience',
                  'How many years of professional work experience do you have in␣
 ↪your field?': 'field_experience',
                   'What is your highest level of education completed?':␣
 ↪'education-level',
                   'What is your gender?': 'gender'
```

```
                  }, inplace=True)

# rename method 2 (columns with problem characters)
df.columns.values[6] = 'annual_salary'
df.columns.values[7] = 'add_compensation'
df.columns.values[12] = 'state'
df.columns.values[18] = 'race'

df.head(1)
```

[4]:
```
   id           timestamp age_range                       industry  \
0   1 2021-04-27 11:03:01     35-44  Accounting, Banking & Finance

                job job_context  annual_salary  add_compensation currency  \
0  Senior Accountant         NaN          45000               0.0      USD

  other_currency               income_context        country    state  \
0            NaN  I work for a Charter School  United States  Florida

         city all_experience field_experience education-level gender  \
0  Palm Coast  21 - 30 years    21 - 30 years  College degree  Woman

                                        race
0  Hispanic, Latino, or Spanish origin, White
```

0.0.3 Clean up country names

[5]:
```
df.country.unique()
```

[5]:
```
array(['United States', 'USA', 'Canada', 'Spain', 'England', 'US',
       'United Kingdom', 'UK', 'United States of America', 'U.S.A.',
       'Netherlands', 'Uk', 'U.S.', 'usa', 'Germany', 'Us', 'Usa',
       'Belgium', 'South Africa', 'us', 'U.S.A', 'Sweden', 'England/UK',
       'France', 'Australia', 'united states',
       'Worldwide (based in US but short term trips aroudn the world)',
       'Denmark', 'Unted States', 'United State', 'Trinidad and Tobago',
       'United states', 'United kingdom', 'Scotland', 'America',
       'Finland', 'Unites States', 'Bangladesh', 'Ireland',
       'Currently finance', ' U.S.', 'U.S', 'Turkey', 'canada', 'Japan',
       'Hong Kong', 'India', 'Czech Republic', 'Switzerland',
       'New Zealand', 'Indonesia', 'Norway', 'The Netherlands', 'The US',
       'Singapore', 'Wales (United Kingdom)', 'UnitedStates', 'UAE',
       'Unite States', 'USAB', 'Unites states', 'Unites kingdom', 'U. S.',
       'SWITZERLAND', 'Malaysia',
       "I work for an US based company but I'm from Argentina.", 'uk',
       'Portugal', 'Israel', 'United states of America', 'Brazil',
```

```
       'South Korea', 'Austria', 'Latvia', 'Romania', 'UA', 'Lithuania',
       'united kingdom', 'Wales', 'Estonia', 'NZ',
       'England, United Kingdom', 'Bermuda', 'Aotearoa New Zealand',
       'new zealand', 'Thailand', 'Cyprus', 'NIGERIA', 'Poland'],
      dtype=object)
```

[6]:
```
# inspect anomalies
df.loc[df['country'] == 'Currently finance']
```

[6]:
```
      id           timestamp age_range                     industry  \
750  751 2021-04-27 14:44:02     45-54  Marketing, Advertising & PR

                    job job_context  annual_salary  add_compensation currency  \
750  Digital Specialist         NaN          90000               0.0      USD

    other_currency income_context            country   state      city  \
750            NaN            NaN  Currently finance  Oregon  Portland

    all_experience field_experience education-level gender   race
750  11 - 20 years    11 - 20 years  College degree    Man  White
```

[7]:
```
df['country'] = df['country'].replace(['Currently finance'], 'United States') #␣
 ↪code as USA
```

[8]:
```
# inspect anomalies
df.loc[df['country'] == 'UA']
```

[8]:
```
        id           timestamp age_range                       industry  \
2117  2118 2021-04-29 14:04:07     35-44  Education (Higher Education)

                      job job_context  annual_salary  add_compensation  \
2117  Associate Consultant         NaN         105000           18000.0

     currency other_currency income_context country      state         city  \
2117      USD            NaN            NaN      UA  Minnesota  Minneapolis

     all_experience field_experience education-level gender   race
2117  11 - 20 years    11 - 20 years  College degree  Woman  White
```

[9]:
```
df['country'] = df['country'].replace(['UA'], 'United States') # code as USA
```

[10]:
```
# inspect anomalies
df.loc[df['country'] == 'I work for an US based company but I\'m from Argentina.
 ↪']
```

[10]:
```
        id           timestamp age_range     industry                      job  \
1669  1670 2021-04-28 17:38:09     25-34  Translation  Audiovisual Translator
```

```
     job_context  annual_salary  add_compensation currency other_currency  \
1669         NaN         240000               NaN    Other            ARS

                                        income_context  \
1669  I'm a freelancer, so my work varies tremendous…

                                                country state  \
1669  I work for an US based company but I'm from Ar…   NaN

                           city all_experience field_experience  \
1669  San Nicolás de los Arroyos    2 - 4 years        5-7 years

     education-level gender                                  race
1669  College degree  Woman  Hispanic, Latino, or Spanish origin
```

[11]:
```
df['country'] = df['country'].replace(['I work for an US based company but I\'m␣
 ↪from Argentina.'], 'Argentina') # code as Argentina
```

[12]:
```
# inspect anomalies
df.loc[df['country'] == 'Worldwide (based in US but short term trips aroudn the␣
 ↪world)']
```

[12]:
```
      id           timestamp age_range                         industry  \
313  314 2021-04-27 11:56:49     35-44  Federal Government Contracting

                                        job  \
313  Senior Acquisition & Assistance Specialist

                                       job_context  annual_salary  \
313  I do the same job as a federal direct hire, bu…         125500

     add_compensation currency other_currency  \
313            2000.0      USD            NaN

                                     income_context  \
313  I have a base salary but I bill to my contract…

                                             country                 state  \
313  Worldwide (based in US but short term trips ar…  District of Columbia

               city all_experience field_experience  education-level gender  \
313  Washington, DC  11 - 20 years    11 - 20 years  Master's degree  Woman

                               race
313  Asian or Asian American, White
```

[13]:
```
df['country'] = df['country'].replace(['Worldwide (based in US but short term␣
↪trips aroudn the world)'], 'United States')  # code as USA
```

[14]:
```
# inspect anomalies
df.loc[df['country'] == 'USAB']
```

[14]:
```
        id           timestamp age_range                       industry  \
1432  1433 2021-04-28 13:43:11     35-44  Education (Primary/Secondary)

                           job job_context  annual_salary  add_compensation  \
1432  Special Education Teacher         NaN          65000            7500.0

     currency other_currency income_context country           state  \
1432      USD            NaN            NaN    USAB  South Carolina

            city all_experience field_experience  education-level gender  \
1432  Greenville  11 - 20 years    11 - 20 years  Master's degree  Woman

       race
1432  White
```

[15]:
```
df['country'] = df['country'].replace(['USAB'], 'United States') # code as USA
```

[16]:
```
# inspect anomalies
df.loc[df['country'] == 'UAE']
```

[16]:
```
        id           timestamp age_range                   industry  \
1257  1258 2021-04-28 08:49:40     25-34  Property or Construction

                              job job_context  annual_salary  \
1257  Proposals & Marketing Manager         NaN          98000

      add_compensation currency other_currency income_context country state  \
1257               0.0      USD            NaN            NaN     UAE   NaN

       city all_experience field_experience  education-level  \
1257  Dubai   8 - 10 years      2 - 4 years  Master's degree

                            gender  \
1257  Other or prefer not to answer

                                                  race
1257  Another option not listed here or prefer not t…
```

[17]:
```
df['country'] = df['country'].replace(['UAE'], 'United Arab Emirates') # United␣
↪Arab Emirates
```

[18]:
```
# clean up country names
df['country'] = df['country'].replace([
                    'United States',
                    'US',
                    'USA',
                    'United States of America',
                    'U.S.A.',
                    'U.S.A',
                    'U.S.',
                    ' U.S.',
                    'usa',
                    'Us',
                    'Usa',
                    'us',
                    'united states',
                    'Unted States',
                    'United State',
                    'United states',
                    'America',
                    'Unites States',
                    'U.S',
                    'The US',
                    'U. S.',
                    'UnitedStates',
                    'Unite States',
                    'Unites states',
                    'United states of America',
                    'Worldwide (based in US but short term trips aroudn the␣
 ↪world)',
                    'Currently finance',
                    'UA'],
                                    'United States')
```

[19]:
```
df['country'] = df['country'].replace([
                    'Canada',
                    'canada'],
                                  'Canada')
```

[20]:
```
df['country'] = df['country'].replace([
                    'England',
                    'United Kingdom',
                    'UK',
                    'Uk',
                    'England/UK',
                    'United kingdom',
                    'Scotland',
                    'Wales (United Kingdom)',
```

```
                    'Unites kingdom',
                    'uk',
                    'united kingdom',
                    'Wales',
                    'England, United Kingdom'],
                                    'United Kingdom')
```

[21]:
```
df['country'] = df['country'].replace([
                    'Netherlands',
                    'The Netherlands'],
                                  'Netherlands')
```

[22]:
```
df['country'] = df['country'].replace([
                    'Switzerland',
                    'SWITZERLAND'],
                                  'Switzerland')
```

[23]:
```
df['country'] = df['country'].replace([
                    'New Zealand',
                    'NZ',
                    'Aotearoa New Zealand',
                    'new zealand'],
                                  'New Zealand')
```

[24]:
```
df['country'] = df['country'].replace(['NIGERIA'], 'Nigeria')
```

[25]:
```
df.country.unique()
```

[25]:
```
array(['United States', 'Canada', 'Spain', 'United Kingdom',
       'Netherlands', 'Germany', 'Belgium', 'South Africa', 'Sweden',
       'France', 'Australia', 'Denmark', 'Trinidad and Tobago', 'Finland',
       'Bangladesh', 'Ireland', 'Turkey', 'Japan', 'Hong Kong', 'India',
       'Czech Republic', 'Switzerland', 'New Zealand', 'Indonesia',
       'Norway', 'Singapore', 'United Arab Emirates', 'Malaysia',
       'Argentina', 'Portugal', 'Israel', 'Brazil', 'South Korea',
       'Austria', 'Latvia', 'Romania', 'Lithuania', 'Estonia', 'Bermuda',
       'Thailand', 'Cyprus', 'Nigeria', 'Poland'], dtype=object)
```

0.0.4 Clean up race

[26]:
```
df.race.unique()
```

[26]:
```
array(['Hispanic, Latino, or Spanish origin, White',
       'Asian or Asian American', 'White',
       'Another option not listed here or prefer not to answer',
       'Asian or Asian American, White',
```

```
       'Hispanic, Latino, or Spanish origin', 'Black or African American',
       'Black or African American, White',
       'Native American or Alaska Native, White',
       'Middle Eastern or Northern African, White', nan,
       'Black or African American, Hispanic, Latino, or Spanish origin',
       'Hispanic, Latino, or Spanish origin, Native American or Alaska Native',
       'White, Another option not listed here or prefer not to answer',
       'Asian or Asian American, Hispanic, Latino, or Spanish origin',
       'Hispanic, Latino, or Spanish origin, Another option not listed here or
prefer not to answer',
       'Black or African American, Hispanic, Latino, or Spanish origin, Native
American or Alaska Native, White',
       'Native American or Alaska Native',
       'Middle Eastern or Northern African',
       'Asian or Asian American, Black or African American, White',
       'Black or African American, Hispanic, Latino, or Spanish origin, White',
       'Middle Eastern or Northern African, Native American or Alaska Native,
White',
       'Middle Eastern or Northern African, White, Another option not listed
here or prefer not to answer',
       'Asian or Asian American, Black or African American',
       'Asian or Asian American, Hispanic, Latino, or Spanish origin, White,
Another option not listed here or prefer not to answer'],
      dtype=object)
```

[27]:
```
# remove commas to enable split
df['race'] = df['race'].str.replace('Hispanic, Latino, or Spanish␣
 ↪origin','Hispanic Latino or Spanish origin')
```

[28]:
```
df["race"] = df["race"].str.split(",")
```

[29]:
```
df = df.explode("race")
```

[30]:
```
df = df.replace(r"^ +| +$", r"", regex=True) # fix issue with leading and␣
 ↪trailing white space again
```

[31]:
```
df.race.unique()
```

[31]:
```
array(['Hispanic Latino or Spanish origin', 'White',
       'Asian or Asian American',
       'Another option not listed here or prefer not to answer',
       'Black or African American', 'Native American or Alaska Native',
       'Middle Eastern or Northern African', nan], dtype=object)
```

[32]:
```
# add multiracial column

multiracial = df[df.duplicated('id', keep=False) == True]
```

```
multiracial_id = (multiracial.id.unique().tolist())
df["multiracial"] = np.where(df["id"].isin(multiracial_id), "Yes", "No")
df.head(5)
```

[32]:

	id	timestamp	age_range	industry	job	job_context	annual_salary	add_compensation	currency	other_currency	income_context	country	state	city	all_experience	field_experience	education-level	gender	race	multiracial
0	1	2021-04-27 11:03:01	35-44	Accounting, Banking & Finance	Senior Accountant	NaN	45000	0.0	USD	NaN	I work for a Charter School	United States	Florida	Palm Coast	21 - 30 years	21 - 30 years	College degree	Woman	Hispanic Latino or Spanish origin	Yes
0	1	2021-04-27 11:03:01	35-44	Accounting, Banking & Finance	Senior Accountant	NaN	45000	0.0	USD	NaN	I work for a Charter School	United States	Florida	Palm Coast	21 - 30 years	21 - 30 years	College degree	Woman	White	Yes
1	2	2021-04-27 11:03:28	35-44	Government and Public Administration	Researcher	NaN	96000	1000.0	USD	NaN	NaN	United States	Ohio	Dayton	8 - 10 years	2 - 4 years	PhD	Woman	Asian or Asian American	No
2	3	2021-04-27 11:03:41	35-44	Government and Public Administration	Economist	NaN	140000	NaN	USD	NaN	NaN	United States	District of Columbia	Washington	11 - 20 years	11 - 20 years	Master's degree	Woman	White	No
3	4	2021-04-27 11:04:06	35-44	Computing or Tech	Mobile developer	NaN	144600	2500.0	USD	NaN	NaN	United States	Massachusetts	Boston	5-7 years	5-7 years	PhD	Woman	White	No

0.0.5 Clean up States

[33]:
```
df.state.unique()
```

[33]:
```
array(['Florida', 'Ohio', 'District of Columbia', 'Massachusetts',
       'Illinois', 'Minnesota', 'New York', 'Maryland', 'Oregon',
       'North Carolina', 'Colorado', nan, 'Pennsylvania', 'New Jersey',
```

```
        'California', 'Virginia', 'South Carolina', 'North Dakota',
        'Washington', 'Kansas', 'Indiana', 'Texas', 'Missouri', 'Delaware',
        'Georgia', 'Michigan', 'Kentucky', 'Rhode Island', 'South Dakota',
        'New Hampshire', 'Louisiana', 'New Mexico', 'Connecticut',
        'Oklahoma', 'Arizona', 'Vermont', 'Utah', 'Idaho', 'Tennessee',
        'Nebraska', 'West Virginia', 'Wisconsin', 'Mississippi', 'Alabama',
        'California, Colorado', 'Maine', 'Alabama, District of Columbia',
        'Arkansas', 'Nevada', 'Iowa', 'Alaska', 'Hawaii',
        'New Jersey, New York', 'Montana', 'Wyoming',
        'Georgia, Massachusetts', 'California, Texas',
        'Indiana, Massachusetts', 'Mississippi, Missouri',
        'California, Illinois, Massachusetts, North Carolina, South Carolina,
Virginia'],
      dtype=object)
```

[34]:
```
df["state"] = df["state"].str.split(",")
```

[35]:
```
df = df.explode("state")
```

[36]:
```
df.state.unique()
```

[36]:
```
array(['Florida', 'Ohio', 'District of Columbia', 'Massachusetts',
       'Illinois', 'Minnesota', 'New York', 'Maryland', 'Oregon',
       'North Carolina', 'Colorado', nan, 'Pennsylvania', 'New Jersey',
       'California', 'Virginia', 'South Carolina', 'North Dakota',
       'Washington', 'Kansas', 'Indiana', 'Texas', 'Missouri', 'Delaware',
       'Georgia', 'Michigan', 'Kentucky', 'Rhode Island', 'South Dakota',
       'New Hampshire', 'Louisiana', 'New Mexico', 'Connecticut',
       'Oklahoma', 'Arizona', 'Vermont', 'Utah', 'Idaho', 'Tennessee',
       'Nebraska', 'West Virginia', 'Wisconsin', 'Mississippi', 'Alabama',
       ' Colorado', 'Maine', ' District of Columbia', 'Arkansas',
       'Nevada', 'Iowa', 'Alaska', 'Hawaii', ' New York', 'Montana',
       'Wyoming', ' Massachusetts', ' Texas', ' Missouri', ' Illinois',
       ' North Carolina', ' South Carolina', ' Virginia'], dtype=object)
```

[37]:
```
df = df.replace(r"^ +| +$", r"", regex=True) # fix issue with leading and␣
↪trailing white space
```

[38]:
```
df.state.unique()
```

[38]:
```
array(['Florida', 'Ohio', 'District of Columbia', 'Massachusetts',
       'Illinois', 'Minnesota', 'New York', 'Maryland', 'Oregon',
       'North Carolina', 'Colorado', nan, 'Pennsylvania', 'New Jersey',
       'California', 'Virginia', 'South Carolina', 'North Dakota',
       'Washington', 'Kansas', 'Indiana', 'Texas', 'Missouri', 'Delaware',
       'Georgia', 'Michigan', 'Kentucky', 'Rhode Island', 'South Dakota',
       'New Hampshire', 'Louisiana', 'New Mexico', 'Connecticut',
```

```
       'Oklahoma', 'Arizona', 'Vermont', 'Utah', 'Idaho', 'Tennessee',
       'Nebraska', 'West Virginia', 'Wisconsin', 'Mississippi', 'Alabama',
       'Maine', 'Arkansas', 'Nevada', 'Iowa', 'Alaska', 'Hawaii',
       'Montana', 'Wyoming'], dtype=object)
```

[39]:
```
# add multistate column

multistate = df[df["multiracial"] == 'No']
multistate = (multistate[multistate.duplicated('id', keep=False) == True])
multistate_id = (multistate.id.unique().tolist())
df["multistate"] = np.where(df["id"].isin(multistate_id), "Yes", "No")
```

0.0.6 Clean up add_compensation

[40]:
```
df['add_compensation'] = df['add_compensation'].fillna(0) # replace NaN with␣
↪zeros
```

0.0.7 Check out currencies

[41]:
```
df.currency.unique()
```

[41]:
```
array(['USD', 'CAD', 'EUR', 'GBP', 'ZAR', 'SEK', 'AUD/NZD', 'Other',
       'CHF', 'JPY'], dtype=object)
```

[42]:
```
df.other_currency.unique()
```

[42]:
```
array([nan, 'Dkk', 'TTD', 'GBP', 'Bdt', 'Additonal = Bonus plus stock',
       'Overtime (about 5 hours a week) and bonus', 'TRY', 'Canadian',
       'INR', 'Czk', 'IDR', 'NOK', 'SGD', 'AUD', 'MYR', 'ARS',
       'Israeli Shekels', 'BRL', 'KRW', 'None', 'Korean Won', 'NZD',
       '47000', 'THB', 'NGN', 'PLN'], dtype=object)
```

[43]:
```
# inspect anomalies
df.loc[df['other_currency'] == 'GBP']
```

[43]:
```
      id           timestamp age_range                     industry  \
541  542 2021-04-27 13:08:37     25-34  Education (Higher Education)

                                          job job_context  annual_salary  \
541  Senior Research Fellow/Assistant Professor         NaN          41000

     add_compensation currency other_currency  …         country state  \
541               0.0    Other            GBP  …  United Kingdom   NaN
```

```
        city all_experience field_experience education-level gender   race  \
541  Glasgow      5-7 years        5-7 years             PhD  Woman  White

    multiracial multistate
541          No         No

[1 rows x 21 columns]
```

[44]:
```
# recode as currency = GBP and other_currency = nan

df['other_currency'] = df['other_currency'].replace(['GBP'], 'NaN')
df.at[541,'currency']='GBP'
```

[45]:
```
df.loc[df['id'] == 542]
```

[45]:
```
      id           timestamp age_range                     industry  \
541  542 2021-04-27 13:08:37     25-34  Education (Higher Education)

                                          job job_context  annual_salary  \
541  Senior Research Fellow/Assistant Professor         NaN          41000

     add_compensation currency other_currency  …         country state  \
541               0.0      GBP            NaN  …  United Kingdom   NaN

        city all_experience field_experience education-level gender   race  \
541  Glasgow      5-7 years        5-7 years             PhD  Woman  White

    multiracial multistate
541          No         No

[1 rows x 21 columns]
```

[46]:
```
# inspect anomalies
df.loc[df['other_currency'] == 'Additonal = Bonus plus stock']
```

[46]:
```
      id           timestamp age_range           industry                job  \
739  740 2021-04-27 14:35:26     45-54  Computing or Tech  Content specialist

    job_context  annual_salary  add_compensation currency  \
739         NaN          62000           17000.0      EUR

                   other_currency  …  country state  \
739  Additonal = Bonus plus stock  …  Ireland   NaN

                                city all_experience field_experience  \
739  Small country, prefer not to say!  31 - 40 years     8 - 10 years
```

```
    education-level gender   race multiracial multistate
739  College degree  Woman  White          No         No

[1 rows x 21 columns]
```

[47]:
```
# recode as other_currency = NaN and income_context = 'Additonal = Bonus plus␣
 ↪stock'

df['other_currency'] = df['other_currency'].replace(['Additonal = Bonus plus␣
 ↪stock'], 'NaN')
df.at[739,'income_context']='Additonal = Bonus plus stock'
```

[48]:
```
df.loc[df['id'] == 740]
```

[48]:
```
      id           timestamp age_range           industry                 job  \
739  740 2021-04-27 14:35:26     45-54  Computing or Tech  Content specialist

    job_context  annual_salary  add_compensation currency other_currency  …  \
739         NaN          62000           17000.0      EUR            NaN  …

     country state                                 city all_experience  \
739  Ireland   NaN  Small country, prefer not to say!  31 - 40 years

    field_experience education-level gender   race multiracial multistate
739     8 - 10 years  College degree  Woman  White          No         No

[1 rows x 21 columns]
```

[49]:
```
# inspect anomalies
df.loc[df['other_currency'] == 'Overtime (about 5 hours a week) and bonus']
```

[49]:
```
      id           timestamp age_range           industry  \
803  804 2021-04-27 15:23:13     25-34  Computing or Tech

                       job job_context  annual_salary  add_compensation  \
803  Executive Assiatant II     Grade 6          86000           20000.0

    currency                             other_currency  …        country  \
803      USD  Overtime (about 5 hours a week) and bonus  …  United States

            state                                                 city  \
803  Massachusetts  HQ us in Cambridge, Ma but moving to the subur…

    all_experience field_experience education-level gender   race multiracial  \
803      5-7 years      2 - 4 years  College degree  Woman  White          No

    multistate
```

```
803         No

[1 rows x 21 columns]
```

[50]:
```
# recode as other_currency = NaN and income_context = 'Overtime (about 5 hours a␣
↪week) and bonus'

df['other_currency'] = df['other_currency'].replace(['Overtime (about 5 hours a␣
↪week) and bonus'], 'NaN')
df.at[803,'income_context']='Overtime (about 5 hours a week) and bonus'
```

[51]:
```
df.loc[df['id'] == 804]
```

[51]:
```
      id           timestamp age_range           industry  \
803  804 2021-04-27 15:23:13     25-34  Computing or Tech

                        job job_context  annual_salary  add_compensation  \
803  Executive Assiatant II     Grade 6          86000           20000.0

    currency other_currency  …        country          state  \
803      USD            NaN  …  United States  Massachusetts

                                                 city all_experience  \
803  HQ us in Cambridge, Ma but moving to the subur…      5-7 years

    field_experience education-level gender   race multiracial multistate
803      2 - 4 years  College degree  Woman  White          No         No

[1 rows x 21 columns]
```

[52]:
```
# inspect anomalies
df.loc[df['other_currency'] == '47000']
```

[52]:
```
        id           timestamp age_range    industry  \
2707  2708 2021-07-06 18:49:41     25-34  Nonprofits

                                  job job_context  annual_salary  \
2707  Districtwide Program Coordinator         NaN          47000

      add_compensation currency other_currency  …        country     state  \
2707               0.0      USD          47000  …  United States  Michigan

         city all_experience field_experience  education-level gender   race  \
2707  Decatur   8 - 10 years     8 - 10 years  Master's degree  Woman  White

     multiracial multistate
2707          No         No
```

```
[1 rows x 21 columns]
```

[53]:
```
# recode as other_currency = NaN

df['other_currency'] = df['other_currency'].replace(['47000'], 'NaN')
df.loc[df['id'] == 2708]
```

[53]:
```
        id           timestamp age_range    industry  \
2707  2708 2021-07-06 18:49:41     25-34  Nonprofits

                                  job job_context  annual_salary  \
2707  Districtwide Program Coordinator         NaN          47000

      add_compensation currency other_currency  …        country     state  \
2707               0.0      USD            NaN  …  United States  Michigan

         city all_experience field_experience  education-level gender   race  \
2707  Decatur   8 - 10 years     8 - 10 years  Master's degree  Woman  White

     multiracial multistate
2707          No         No

[1 rows x 21 columns]
```

[54]:
```
# fix all nan values
# df.fillna('NaN', inplace=True)
```

[55]:
```
df['other_currency'] = df['other_currency'].replace([
                    'Dkk',
                    'Bdt',
                    'Czk',
                    'Korean Won',
                    'Israeli Shekels',
                    'Canadian'],
                        ['DKK',
                         'BDT',
                         'CZK',
                         'KRW',
                         'ILS',
                         'CAD'])
```

[56]:
```
df.other_currency.unique()
```

[56]:
```
array([nan, 'DKK', 'TTD', 'NaN', 'BDT', 'TRY', 'CAD', 'INR', 'CZK', 'IDR',
       'NOK', 'SGD', 'AUD', 'MYR', 'ARS', 'ILS', 'BRL', 'KRW', 'None',
       'NZD', 'THB', 'NGN', 'PLN'], dtype=object)
```

0.0.8 Drop city data

It's such a mess and I'm not planning to use it. Could do more work to clean it up and try resolving problems with either OpenRefine or Google Maps API, but it's just not precise enough to be useful (e.g., "metro area").

[57]:
```
df = df.drop(['city'], axis=1)
df.head(1)
```

[57]:
```
   id           timestamp age_range                       industry  \
0   1 2021-04-27 11:03:01     35-44  Accounting, Banking & Finance

                job job_context  annual_salary  add_compensation currency  \
0  Senior Accountant         NaN          45000               0.0      USD

  other_currency               income_context        country    state  \
0            NaN  I work for a Charter School  United States  Florida

  all_experience field_experience education-level gender  \
0  21 - 30 years    21 - 30 years  College degree  Woman

                               race multiracial multistate
0  Hispanic Latino or Spanish origin         Yes         No
```

0.1 Clean up industry

[124]:
```
# df.industry.unique() # Used a text editor to quickly organize these
```

[91]:
```
# create new broader categories

df['industry'] = df['industry'].replace([
                    'Accounting, Banking & Finance',
                    'Mortgage',
                    'FinTech/Payment Processing',
                    'commodities trading'],
                                    'Financial')
```

[92]:
```
df['industry'] = df['industry'].replace([
                    'Government and Public Administration',
                    'Government Relation'],
                                    'Government')
```

[93]:
```
df['industry'] = df['industry'].replace([
                    'Computing or Tech',
                    'IT MSP',
                    'Virtual reality',
```

```
                    'Saas',
                    'I work for Indeed.com',
                    'Customer Service'],
                                    'Tech')
```

[94]:
```
df['industry'] = df['industry'].replace([
                    'Synthetic Chemical Manufacturing',
                    'Engineering or Manufacturing',
                    'Manufacturing',
                    'Manufacturing : corporate admin support'],
                                    'Manufacturing')
```

[95]:
```
df['industry'] = df['industry'].replace([
                    'Nonprofits',
                    'Nonprofit - legal department'],
                                    'Nonprofit')
```

[96]:
```
df['industry'] = df['industry'].replace([
                    'Consumer goods',
                    'Consumer Good (Toys)',
                    'Wholesale - Apparel',
                    'Retail',
                    'FMCG',
                    'Consumer Goods',
                    'FMCG development',
                    'Ecommerce',
                    'Ecommerce',
                    'Fashion/e-commerce'],
                                    'Consumer Goods')
```

[97]:
```
df['industry'] = df['industry'].replace([
                    'Sales',
                    'Sales operations'],
                                    'Sales')
```

[98]:
```
df['industry'] = df['industry'].replace([
                    'Real Estate',
                    'Real Estate',
                    'Property Management',
                    'Commercial Real Estate'],
                                    'Property or Construction')
```

[99]:
```
df['industry'] = df['industry'].replace([
                    'Instructional Design and Training',
                    'Educational technology',
                    'Educational publishing / ed tech',
                    'ESL Teacher'],
```

```
                                        'Other Education')
```

[100]:
```
df['industry'] = df['industry'].replace([
                    'Education (Higher Education)',
                    'Academic science',
                    'Science academia',
                    'Research - academic',
                    'Research and Development Academia',
                    'academic research',
                    'Academic science'],
                                        'Higher Education')
```

[101]:
```
df['industry'] = df['industry'].replace([
                    'Marketing and PR',
                    'market research',
                    'Market Research',
                    'Public affairs / PR'],
                                        'Marketing, Advertising & PR')
```

[102]:
```
df['industry'] = df['industry'].replace([
                    'Supply Chain',
                    'Coffee - Importing',
                    'Logistics'],
                                        'Transport or Logistics')
```

[103]:
```
df['industry'] = df['industry'].replace([
                    'Hospital',
                    'Public health',
                    'Healthcare IT'],
                                         'Health Care')
```

[104]:
```
df['industry'] = df['industry'].replace([
                    'clinical research',
                    'biomedical research',
                    'Medical Research',
                    'Biology/Research',
                    'Biomedical Research',
                    'Biologist'],
                                         'Biomedical Research')
```

[105]:
```
df['industry'] = df['industry'].replace([
                    'Bitech',
                    'Biotech/Pharma',
                    'Biotech',
                    'Biotechnology',
                    'Biotech/pharmaceuticals',
                    'Biotech/pharma',
```

```
                    'Biotech/Drug Development',
                    'Pharmaceutical',
                    'Pharmaceutical Research',
                    'Pharmaceutical research',
                    'Pharmaceuticals',
                    'Pharma',
                    'Pharmaceutical R&D',
                    'Drug development'],
                                    'Pharmaceuticals')
```

[106]:
```
df['industry'] = df['industry'].replace([
                    'Recruitment or HR',
                    'Human Resources',
                    'Benefits Administration'],
                                    'Human Resources')
```

[107]:
```
df['industry'] = df['industry'].replace([
                    'Defense contracting',
                    'Federal Contracting/Business Development',
                    'Federal Government Contracting'],
                                    'Government Contracting')
```

[108]:
```
df['industry'] = df['industry'].replace([
                    'apparel design/product development'],
                                    'Art & Design')
```

[109]:
```
df['industry'] = df['industry'].replace([
                    'Oil & Gas',
                    'Renewable Energy',
                    'Energy: oil & gas'],
                                    'Energy')
```

[110]:
```
df['industry'] = df['industry'].replace([
                    'Security'],
                                    'Law Enforcement & Security')
```

[111]:
```
df['industry'] = df['industry'].replace([
                    'Public Librarian',
                    'Public Library',
                    'Librarian and Assistant Manager of a library',
                    'Public library',
                    'Library',
                    'Librarian in legal setting',
                    'municipal (public) libraries',
                    'Libraries',
                    'Public Libraries',
                    'Library/Archive',
```

```
                    'Library science / part-time work/study',
                    'Library Tech for a school system',
                    'library',
                    'Librarian',
                    'Museums',
                    'Archives/Libraries',
                    'Education (Other)'], #checked title
                                        'Libraries & Museums')
```

[112]:
```
df['industry'] = df['industry'].replace([
                    'auto repair',
                    'Automotive technician',
                    'Automotive'],
                                        'Automtive Repair')
```

[113]:
```
df['industry'] = df['industry'].replace([
                    'Government Affairs/Lobbying',
                    'Politics',
                    'Union/political organizing'],
                                        'Politics')
```

[114]:
```
df['industry'] = df['industry'].replace([
                    'Veterinary medicine',
                    'Pet',
                    'Veterinary m&a'],
                                        'Veterinary')
```

[115]:
```
df['industry'] = df['industry'].replace([
                    'Environmental Consulting',
                    'Environmental consulting',
                    'Consulting',
                    'Consultant',
                    'Business or Consulting'],
                                        'Consulting')
```

[116]:
```
df['industry'] = df['industry'].replace([
                    'Restaurant',
                    'Food Manufacture',
                    'Food service',
                    'Craft Beer Industry',
                    'Beverage'],
                                        'Food & Beverage')
```

[117]:
```
df['industry'] = df['industry'].replace([
                    'Fundraising for a university'],
                                        'Fundraising')
```

[118]:
```
df['industry'] = df['industry'].replace([
                    'Faith/spirituality',
                    'Clergy'],
                                    'Faith & Spirituality')
```

[119]:
```
df['industry'] = df['industry'].replace([
                    'funeral services',
                    'Funeral services'],
                                    'Funeral Services')
```

[120]:
```
df['industry'] = df['industry'].replace([
                    'Environmental',
                    'Enviromental',
                    'Environment',
                    'Environmental Restoration'],
                                    'Environmental')
```

[121]:
```
df.industry.unique() # Used a text editor to quickly organize these
```

[121]:
```
array(['Financial', 'Government', 'Tech', 'Higher Education', 'Sales',
       'Consulting', 'Manufacturing', 'Media & Digital',
       'Hospitality & Events', 'Publishing', 'Nonprofit', 'Architecture',
       'Consumer Goods', 'Law', 'Property or Construction', 'Insurance',
       'Education (Primary/Secondary)', 'Utilities & Telecommunications',
       'Research and development', 'Entertainment',
       'Transport or Logistics', 'Health care', 'Health Care',
       'Social Work', 'Human Resources', 'Marketing, Advertising & PR',
       'Leisure, Sport & Tourism', 'Government Contracting',
       'Life Sciences', 'Art & Design', 'Fire protection', 'Energy',
       'Gambling', 'Law Enforcement & Security', 'Pharmaceuticals',
       'Gaming', 'labour/professional organization', 'Food & Beverage',
       'Biomedical Research', 'Libraries & Museums', 'Other Education',
       'Automtive Repair', 'Politics', 'Immigration', 'Veterinary',
       'Philanthropy', 'Fundraising', 'Research at a National Laboratory',
       'International development', 'Program management',
       'Faith & Spirituality', 'Translation', 'Scientific Research',
       'Science', 'Environmental Sciences', 'Shared office space',
       'National laboratory', 'Agriculture or Forestry', 'Communications',
       'Science - QC lab', 'Cannabis', 'Animal welfare', 'Environmental',
       'Research', 'Auction house', 'Scientific research',
       'Security and manufacturing company', nan, 'Scientific',
       'Funeral Services', 'Science Research', 'Earth sciences',
       'fitness', 'Cultural Resource Management',
       'Professional Association', 'Scientific research (industry)',
       'Cleaning', 'Mining', 'Social Research', 'Family office'],
      dtype=object)
```

0.1.1 Final clean up and export

[122]:
```
# fix all nan values
df.fillna('NaN', inplace=True)
```

[123]:
```
df.to_csv('clean_salaries.csv')
```