LJAC Wage Claims Data

September 18, 2019

1 LJAC Wage Claims Data

- Raf Alvarado, Data Science Institute, UVA
- Prepared for CodeForCharlottesville Meetup held at UVA's Data Science Institute on Tuesday, September 17th, 2019.

2 Prepare the data

2.1 Import libraries

```
In [1]: import pandas as pd
        import sqlite3
        import re
        import plotly_express as px
```

2.2 Pragmas

```
In [2]: %matplotlib inline
```

2.3 Import CSV

```
In [3]: df = pd.read_csv('../Raw data/lajc_wage_claim.csv')

In [4]: N = df.shape[0]

In [5]: N

Out[5]: 3948
```

2.4 Import codebook

We do this because we may want to use these details in visualizations and presentations later, e.g. when column names are used as labels, when can use descriptions as hover text.

```
In [6]: codebook_raw = """CLAIM NO         A unique ID for each official wage-theft complaint i
        COMPLAINT         Whether or not this row represents an official complaint         X i
        ROUTINE                          Missing everywhere, safe to ignore
        EMPLOYER NAME         The name of the employer against whom the wage-theft complaint is
```

```
EMPLOYER CITY         The employer's city         Character
ST         The employer's state         Character
ZIP         The employer's ZIP code         Character
gender         The predicted gender of the person issuing the complaint based on first
hispanic         Whether the complainant is predicted to be Hispanic based on the last
asian         Whether the complainant is predicted to be Asian based on the last name
CLAIM AMT         The amount of money claimed to have been illegally withheld         C
CASE OPEN/RE-OPEN         The date the case was opened         Character
CLAIM RECEIVED         The date the claim was received by DOLI         Character
Valid         Whether the claim was deemed by DOLI to be valid         Character: X if
Informal Resolution         Whether the claim was resulted in an informal resolution be
Bankrupt         Whether the claim is being made against a bankrupt employer         Ch
Invalid         Whether DOLI disqualified the claim for one of the following reasons:
Fringe Benefits         Whether DOLI disqualified the claim because it deals with contr
Independent Agent         Whether DOLI disqualified the claim because it involves a cor
Subcontractor         Whether DOLI disqualified the claim because it involves a subcont
False Claim         Whether DOLI determined that the claim is false         Character:
Other         Whether DOLI determined that the claim is invalid for another reason
CLAIM INVAL OTHER DESCRIPTION         The stated reason DOLI found a claim invalid if '
Claim Validity         Whether an undetermined claim is valid         Character: X if t
Employer left State         Whether a claim is undetermined because the employer left t
Employer Cannot be Located         Whether a claim is undetermined because the employer
Complainant Cannot be Located         Whether a claim is undetermined because the compl
Complainant Dropped Claim         Whether a claim is undetermined because the complaina
Paid Prior to Investigation         Whether a claim is undetermined because the employe
Business is Closed         Whether a claim is undetermined because the employer's busi
Other_1         Whether a claim is undetermined for another reason         Character: X
CLAIM UNDETERMINED OTHER DESCRIPTION         The stated reason for being undetermined i
VERIFIED CLAIM AMT         The total amount claimed after verification by DOLI
CASE CLOSE/RECLOSE DATE         Date case was officially closed         Character
Employer Contested Valid Determination         Whether the employer contested a claim t
1st Response Investigation         Whether only a 1st response was conducted         Ch
Formal Investigation         Whether a formal investigation was conducted         Chara
Request Settlement Conference         All missing         No such requests in t
Request Informal Fact Finding         All missing         No such requests in t
Request Formal Fact Finding         All missing         No such requests in the
Wage Order         Whether a legal order is given for the employer to pay lost wages
Informal Conference         All missing         No such actions in the data
Civil Action for Wages/Penalties         All missing         No such actions in
Other_2         All missing         No such actions in the data
OTHER DISPOSITION DESCRIPTION         All missing
CASE CLOSED for the REPRESENTATIVE         Date the DOLI labor law representative ended
JUDGMENT         All missing
DISMISSED         All missing
NON-SUITED         All missing
JUDGMENT_1         All missing
DISMISSED_1         All missing
NON-SUITED_1         All missing
```

```
CIVCOURTDTE FOR WAGES/PENALTY                  All missing
TOT AMT         Total amount recovered for the complainant         Character (but easi
WAGE AMT         Amount recovered in lost wages for the complainant         Character
INTEREST AMT         Amount recovered in interest on lost wages for the complainant
ATTY FEES         Amount recovered in reimbursed attorney fees for the complainant
DATE FOR WAGES         Date wages were returned to the complainant         Character
WAGES APPEALED                 All missing
WAGES APPEAL CIRCUIT COURT DATE                 All missing
ASSESSED         Whether the DOLI investigation concludes the employer owes a civil mo
AMT         Amount of the CMP owed by the employer         Character (but easily conve
ATTY FEES_1         Attorney fees owed by the employer as part of the CMP         Chara
ASSESSED DATE         Date CMP totals were assessed         Character         See above
TOTAL CMP AMT         CMP + attorney fees         Character (but easily converted to nu
CMP APPEALED                 All missing         See above
CMP APPEAL CIRCUIT COURT DATE                 All missing         See above
FINES & COURT COSTS         Other fines and court costs associated with the CMP
TOTAL WAGES         Total wages that the DOLI says have been actually collected
TOTAL INTEREST AMT         Total interest collected         Character (but easily conve
TOT WAGES & INTEREST         Wages + interest collected         Character (but easily
GARNISHED AMT                 All $0
JUDGMENT WAGE AMT         We think these are additional fees for repeat offenders
JUDGEMENT PENALTY AMT         We think these are additional fees for repeat offenders
DOCKETED/SENT FOR COLLECTION DATE                 All missing
ACTION TAKEN                 All missing
COURT DATE                 All missing
CONVICTED - DISPOSITION                 All missing
DISMISSED - DISPOSITION                 All missing
NOL Prossed                 All missing
CONFINEMENT                 All missing
SUSPENSION                 All missing
BOTH - C&F                 All missing
JUDGMENT_2                 All missing
DISMISSED_2                 All missing
NON-SUITED_2                 All missing
JUDGMENT_3                 All missing
DISMISSED_3                 All missing
NON-SUITED_3                 All missing         """.split('\n')
```

In [7]:
```
codebook = pd.DataFrame([row.split('\t') for row in codebook_raw], columns=['Variable'
```

In [8]:
```
codebook.head()
```

Out[8]:

	Variable	Description	\
0	CLAIM NO	A unique ID for each official wage-theft compl...	
1	COMPLAINT	Whether or not this row represents an official...	
2	ROUTINE		
3	EMPLOYER NAME	The name of the employer against whom the wage...	
4	EMPLOYER CITY	The employer's city	

```
        Values                                                  Notes
0
1   X if yes   All rows are complaints, so safe to ignore thi...
2                             Missing everywhere, safe to ignore
3  Character
4  Character
```

2.5 Normalize column names

It's always a good idea to remove spaces and special characters from column names, especially if we plan to import the CSV file into a database.

```
In [9]: df.columns = [re.sub(r'[/\s-]+', '_', col.lower().replace('&', 'and')) for col in df.co
        codebook['newcol'] = df.columns

In [10]: codebook = codebook.set_index('newcol')

In [11]: codebook

Out[11]:                                                          Variable  \
         newcol
         claim_no                                                 CLAIM NO
         complaint                                                COMPLAINT
         routine                                                    ROUTINE
         employer_name                                        EMPLOYER NAME
         employer_city                                        EMPLOYER CITY
         st                                                              ST
         zip                                                            ZIP
         gender                                                      gender
         hispanic                                                  hispanic
         asian                                                        asian
         claim_amt                                               CLAIM AMT
         case_open_re_open                                 CASE OPEN/RE-OPEN
         claim_received                                       CLAIM RECEIVED
         valid                                                         Valid
         informal_resolution                             Informal Resolution
         bankrupt                                                   Bankrupt
         invalid                                                     Invalid
         fringe_benefits                                    Fringe Benefits
         independent_agent                                Independent Agent
         subcontractor                                        Subcontractor
         false_claim                                            False Claim
         other                                                        Other
         claim_inval_other_description          CLAIM INVAL OTHER DESCRIPTION
         claim_validity                                       Claim Validity
         employer_left_state                            Employer left State
         employer_cannot_be_located                Employer Cannot be Located
         complainant_cannot_be_located          Complainant Cannot be Located
```

complainant_dropped_claim	Complainant Dropped Claim
paid_prior_to_investigation	Paid Prior to Investigation
business_is_closed	Business is Closed
...	...
wages_appeal_circuit_court_date	WAGES APPEAL CIRCUIT COURT DATE
assessed	ASSESSED
amt	AMT
atty_fees_1	ATTY FEES_1
assessed_date	ASSESSED DATE
total_cmp_amt	TOTAL CMP AMT
cmp_appealed	CMP APPEALED
cmp_appeal_circuit_court_date	CMP APPEAL CIRCUIT COURT DATE
fines_and_court_costs	FINES & COURT COSTS
total_wages	TOTAL WAGES
total_interest_amt	TOTAL INTEREST AMT
tot_wages_and_interest	TOT WAGES & INTEREST
garnished_amt	GARNISHED AMT
judgment_wage_amt	JUDGMENT WAGE AMT
judgement_penalty_amt	JUDGEMENT PENALTY AMT
docketed_sent_for_collection_date	DOCKETED/SENT FOR COLLECTION DATE
action_taken	ACTION TAKEN
court_date	COURT DATE
convicted_disposition	CONVICTED - DISPOSITION
dismissed_disposition	DISMISSED - DISPOSITION
nol_prossed	NOL Prossed
confinement	CONFINEMENT
suspension	SUSPENSION
both_candf	BOTH - C&F
judgment_2	JUDGMENT_2
dismissed_2	DISMISSED_2
non_suited_2	NON-SUITED_2
judgment_3	JUDGMENT_3
dismissed_3	DISMISSED_3
non_suited_3	NON-SUITED_3

	Description
newcol	
claim_no	A unique ID for each official wage-theft compl...
complaint	Whether or not this row represents an official...
routine	
employer_name	The name of the employer against whom the wage...
employer_city	The employer's city
st	The employer's state
zip	The employer's ZIP code
gender	The predicted gender of the person issuing the...
hispanic	Whether the complainant is predicted to be His...
asian	Whether the complainant is predicted to be Asi...
claim_amt	The amount of money claimed to have been illeg...

case_open_re_open	The date the case was opened
claim_received	The date the claim was received by DOLI
valid	Whether the claim was deemed by DOLI to be valid
informal_resolution	Whether the claim was resulted in an informal ...
bankrupt	Whether the claim is being made against a bank...
invalid	Whether DOLI disqualified the claim for one of...
fringe_benefits	Whether DOLI disqualified the claim because it...
independent_agent	Whether DOLI disqualified the claim because it...
subcontractor	Whether DOLI disqualified the claim because it...
false_claim	Whether DOLI determined that the claim is false
other	Whether DOLI determined that the claim is inva...
claim_inval_other_description	The stated reason DOLI found a claim invalid i...
claim_validity	Whether an undetermined claim is valid
employer_left_state	Whether a claim is undetermined because the em...
employer_cannot_be_located	Whether a claim is undetermined because the em...
complainant_cannot_be_located	Whether a claim is undetermined because the co...
complainant_dropped_claim	Whether a claim is undetermined because the co...
paid_prior_to_investigation	Whether a claim is undetermined because the em...
business_is_closed	Whether a claim is undetermined because the em...
...	...
wages_appeal_circuit_court_date	
assessed	Whether the DOLI investigation concludes the e...
amt	Amount of the CMP owed by the employer
atty_fees_1	Attorney fees owed by the employer as part of ...
assessed_date	Date CMP totals were assessed
total_cmp_amt	CMP + attorney fees
cmp_appealed	
cmp_appeal_circuit_court_date	
fines_and_court_costs	Other fines and court costs associated with th...
total_wages	Total wages that the DOLI says have been actua...
total_interest_amt	Total interest collected
tot_wages_and_interest	Wages + interest collected
garnished_amt	
judgment_wage_amt	We think these are additional fees for repeat ...
judgement_penalty_amt	We think these are additional fees for repeat ...
docketed_sent_for_collection_date	
action_taken	
court_date	
convicted_disposition	
dismissed_disposition	
nol_prossed	
confinement	
suspension	
both_candf	
judgment_2	
dismissed_2	
non_suited_2	
judgment_3	

```
dismissed_3
non_suited_3


                                                                                Values  \
newcol
claim_no
complaint                                                                        X if yes
routine
employer_name                                                                   Character
employer_city                                                                   Character
st                                                                              Character
zip                                                                             Character
gender                                                             Factor: (male, female)
hispanic                                                           Logical: (TRUE, FALSE)
asian                                                              Logical: (TRUE, FALSE)
claim_amt                                 Character (but easily converted to numeric)
case_open_re_open                                                               Character
claim_received                                                                  Character
valid                                      Character: X if valid, missing if not
informal_resolution                       Character: X if true, missing if false
bankrupt                                  Character: X if true, missing if false
invalid                                   Character: X if true, missing if false
fringe_benefits                           Character: X if true, missing if false
independent_agent                         Character: X if true, missing if false
subcontractor                             Character: X if true, missing if false
false_claim                               Character: X if true, missing if false
other                                     Character: X if true, missing if false
claim_inval_other_description                                                   Character
claim_validity                            Character: X if true, missing if false
employer_left_state                       Character: X if true, missing if false
employer_cannot_be_located                Character: X if true, missing if false
complainant_cannot_be_located             Character: X if true, missing if false
complainant_dropped_claim                 Character: X if true, missing if false
paid_prior_to_investigation               Character: X if true, missing if false
business_is_closed                        Character: X if true, missing if false
...                                                                               ...
wages_appeal_circuit_court_date                                          All missing
assessed                                  Character: X if true, missing if false
amt                                       Character (but easily converted to numeric)
atty_fees_1                               Character (but easily converted to numeric)
assessed_date                                                             Character
total_cmp_amt                             Character (but easily converted to numeric)
cmp_appealed                                                             All missing
cmp_appeal_circuit_court_date                                            All missing
fines_and_court_costs                     Character (but easily converted to numeric)
total_wages                               Character (but easily converted to numeric)
total_interest_amt                        Character (but easily converted to numeric)
tot_wages_and_interest                    Character (but easily converted to numeric)
```

garnished_amt	All $0
judgment_wage_amt	Character (but easily converted to numeric)
judgement_penalty_amt	Character (but easily converted to numeric)
docketed_sent_for_collection_date	All missing
action_taken	All missing
court_date	All missing
convicted_disposition	All missing
dismissed_disposition	All missing
nol_prossed	All missing
confinement	All missing
suspension	All missing
both_candf	All missing
judgment_2	All missing
dismissed_2	All missing
non_suited_2	All missing
judgment_3	All missing
dismissed_3	All missing
non_suited_3	All missing

	Notes
newcol	
claim_no	
complaint	All rows are complaints, so safe to ignore thi...
routine	Missing everywhere, safe to ignore
employer_name	
employer_city	
st	
zip	
gender	Gender is not included in the original data, b...
hispanic	Race/Ethnicity is not included in the original...
asian	See above
claim_amt	
case_open_re_open	
claim_received	
valid	
informal_resolution	
bankrupt	This is one of the reasons DOLI as outside the...
invalid	
fringe_benefits	This is one of the reasons DOLI as outside the...
independent_agent	This is one of the reasons DOLI as outside the...
subcontractor	This is one of the reasons DOLI as outside the...
false_claim	
other	This is one of the reasons DOLI as outside the...
claim_inval_other_description	
claim_validity	A claim is "undetermined" if DOLI finds that i...
employer_left_state	See above
employer_cannot_be_located	See above
complainant_cannot_be_located	See above

```
complainant_dropped_claim                                                See above
paid_prior_to_investigation                                              See above
business_is_closed                                                       See above
...                                                                            ...
wages_appeal_circuit_court_date
assessed                              A civil monentary penalty (CMP) is a fine an e...
amt                                                                      See above
atty_fees_1                                                              See above
assessed_date                                                            See above
total_cmp_amt                                                            See above
cmp_appealed                                                             See above
cmp_appeal_circuit_court_date                                            See above
fines_and_court_costs                                                    See above
total_wages                           We're not sure what to make of these "collecti...
total_interest_amt                                                       See above
tot_wages_and_interest                                                   See above
garnished_amt
judgment_wage_amt
judgement_penalty_amt
docketed_sent_for_collection_date
action_taken
court_date
convicted_disposition
dismissed_disposition
nol_prossed
confinement
suspension
both_candf
judgment_2
dismissed_2
non_suited_2
judgment_3
dismissed_3
non_suited_3

[89 rows x 4 columns]
```

2.6 Save codebook to CSV

In case someone wants it in this form, and not in the database.

```
In [12]: codebook.to_csv('codebook.csv', index=True)
```

2.7 Convert date columns

```
In [13]: date_cols = codebook[codebook.Description.str.contains('date', case=False)].index.toli

In [14]: date_cols
```

```
Out[14]: ['case_open_re_open',
          'claim_received',
          'case_close_reclose_date',
          'case_closed_for_the_representative',
          'date_for_wages',
          'assessed_date']

In [15]: for col in date_cols:
             df[col] = df[col].str.split('-')
             nas = df[col].isna()
             df.loc[~nas, col] = df.loc[~nas, col].apply(lambda x: "20{}-{}-{}".format(x[2], x
             df[col] = pd.to_datetime(df[col])

In [16]: df[date_cols].head()

Out[16]:   case_open_re_open claim_received case_close_reclose_date  \
         0        2014-08-22     2014-08-19              2018-03-21
         1        2014-09-08     2014-09-04              2015-01-22
         2        2015-03-06     2015-03-05              2017-03-17
         3        2015-07-31     2015-07-25              2016-05-23
         4        2015-08-04     2015-07-28              2016-01-26

           case_closed_for_the_representative date_for_wages assessed_date
         0                          2014-10-20            NaT           NaT
         1                          2014-10-17     2015-01-28    2015-01-28
         2                          2015-04-14            NaT           NaT
         3                          2015-10-08            NaT           NaT
         4                          2015-10-28            NaT           NaT
```

2.8 Convert money columns

```
In [17]: money_cols = codebook[codebook.Description.str.contains('(amount|fees|costs)', case=Fa

/Users/rca2t/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: UserWarning:

This pattern has match groups. To actually get the groups, use str.extract.



In [18]: money_cols = money_cols + "total_cmp_amt total_wages total_interest_amt tot_wages_and_

In [19]: df[money_cols].head()

Out[19]:    claim_amt verified_claim_amt    tot_amt   wage_amt interest_amt atty_fees  \
         0    $340.00           $340.00    $453.33    $340.00        $0.00   $113.33
         1  $1,275.50         $1,272.99  $1,644.67  $1,233.50        $0.00   $411.17
         2      $0.00           $547.88      $0.00      $0.00        $0.00     $0.00
         3     $50.00            $54.00      $0.00      $0.00        $0.00     $0.00
         4  $2,229.50         $1,286.13      $0.00      $0.00        $0.00     $0.00
```

```
       amt atty_fees_1 total_cmp_amt fines_and_court_costs judgment_wage_amt  \
0  $250.00      $83.33       $333.33                $0.00             $0.00
1  $700.00     $233.33       $933.33                $0.00             $0.00
2    $0.00       $0.00         $0.00                $0.00             $0.00
3    $0.00       $0.00         $0.00                $0.00             $0.00
4    $0.00       $0.00         $0.00                $0.00             $0.00

  judgement_penalty_amt total_cmp_amt total_wages total_interest_amt  \
0                 $0.00       $333.33       $0.00              $0.00
1                 $0.00       $933.33   $1,272.99              $0.00
2                 $0.00         $0.00     $547.88              $0.00
3                 $0.00         $0.00      $50.00              $4.00
4                 $0.00         $0.00   $1,235.00             $51.13

  tot_wages_and_interest garnished_amt
0                  $0.00         $0.00
1              $1,272.99         $0.00
2                $547.88         $0.00
3                 $54.00         $0.00
4              $1,286.13         $0.00
```

```
In [20]: for col in money_cols:
             df[col] = df[col].fillna('').astype('str')
             df[col] = df[col].str.replace('[$.,]', '')
             df[col] = df[col].astype('int')
```

```
In [21]: df[money_cols].head()
```

```
Out[21]:    claim_amt  verified_claim_amt  tot_amt  wage_amt  interest_amt  atty_fees  \
0      34000               34000    45333     34000             0      11333
1     127550              127299   164467    123350             0      41117
2          0               54788        0         0             0          0
3       5000                5400        0         0             0          0
4     222950              128613        0         0             0          0

     amt  atty_fees_1  total_cmp_amt  fines_and_court_costs  \
0  25000         8333          33333                      0
1  70000        23333          93333                      0
2      0            0              0                      0
3      0            0              0                      0
4      0            0              0                      0

   judgment_wage_amt  judgement_penalty_amt  total_cmp_amt  total_wages  \
0                  0                      0          33333            0
1                  0                      0          93333       127299
2                  0                      0              0        54788
3                  0                      0              0         5000
```

```
4                   0                       0               0       123500

   total_interest_amt  tot_wages_and_interest  garnished_amt
0                   0                       0              0
1                   0                  127299              0
2                   0                   54788              0
3                 400                    5400              0
4                5113                  128613              0
```

2.9 Convert boolean columns

```
In [22]: bool_cols =  codebook[codebook.Values.str.contains("X if")].index.tolist()

In [23]: bool_cols

Out[23]: ['complaint',
          'valid',
          'informal_resolution',
          'bankrupt',
          'invalid',
          'fringe_benefits',
          'independent_agent',
          'subcontractor',
          'false_claim',
          'other',
          'claim_validity',
          'employer_left_state',
          'employer_cannot_be_located',
          'complainant_cannot_be_located',
          'complainant_dropped_claim',
          'paid_prior_to_investigation',
          'business_is_closed',
          'other_1',
          'employer_contested_valid_determination',
          '1st_response_investigation',
          'formal_investigation',
          'wage_order',
          'assessed']

In [24]: df[bool_cols] = df[bool_cols].fillna('').astype('bool')

In [25]: df[bool_cols].head()

Out[25]:    complaint  valid  informal_resolution  bankrupt  invalid  fringe_benefits  \
         0       True   True                False     False    False            False
         1       True   True                False     False    False            False
         2       True  False                False     False    False            False
         3       True   True                False     False    False            False
         4       True  False                False     False    False            False
```

```
   independent_agent  subcontractor  false_claim  other    ...     \
0              False          False        False  False    ...
1              False          False        False  False    ...
2              False          False        False  False    ...
3              False          False        False  False    ...
4              False          False        False  False    ...

   complainant_cannot_be_located  complainant_dropped_claim  \
0                          False                      False
1                          False                      False
2                          False                      False
3                          False                      False
4                          False                      False

   paid_prior_to_investigation  business_is_closed  other_1  \
0                        False               False    False
1                        False               False    False
2                        False               False    False
3                        False               False    False
4                        False               False    False

   employer_contested_valid_determination  1st_response_investigation  \
0                                   False                       False
1                                   False                       False
2                                   False                       False
3                                   False                       False
4                                   False                       False

   formal_investigation  wage_order  assessed
0                 False        True     False
1                 False        True      True
2                  True       False     False
3                  True       False     False
4                  True       False     False

[5 rows x 23 columns]
```

2.10 Convert Zip Codes to integers

```
In [26]: df['zip']  = df['zip'].fillna(0).astype('int')

In [27]: df['zip'].value_counts().head()

Out[27]: 22192    99
         0        90
         23320    80
         23462    63
```

```
20166    62
Name: zip, dtype: int64
```

2.11 Remove empty columns

```
In [28]: nonempty_cols = codebook[~codebook['Values'].str.match('All missing')].index.tolist()

In [29]: df = df[nonempty_cols]

In [30]: df.head()
```

Out[30]:

	claim_no	complaint	routine	employer_name
0	LLVA56991	True	NaN	GREATER SCAPE'S GROUND MANAGEMENT, LLC
1	LLVA57047	True	NaN	TRANSFORMED LEADERSHIP, LLC.
2	LLVA57784	True	NaN	GIGI'S NEXT STEP CHILD CARE CENTER, LLC
3	LLVA58364	True	NaN	MERRYMAN GROUNDS MAINT INC.
4	LLVA58376	True	NaN	CONTINENTAL HOME HEALTH CARE, LLC,

	employer_city	st	zip	gender	hispanic	asian
0	HENRICO	VA	23075	male	True	False
1	ROANOKE	VA	24018	female	False	False
2	SOUTH CHESTERFIELD	VA	23803	female	False	False
3	HAMPTON	VA	23661	male	False	False
4	RICHMOND	VA	23294	female	False	False

	...	atty_fees_1	assessed_date	total_cmp_amt
0	...	8333	NaT	33333
1	...	23333	2015-01-28	93333
2	...	0	NaT	0
3	...	0	NaT	0
4	...	0	NaT	0

	fines_and_court_costs	total_wages	total_interest_amt
0	0	0	0
1	0	127299	0
2	0	54788	0
3	0	5000	400
4	0	123500	5113

	tot_wages_and_interest	garnished_amt	judgment_wage_amt
0	0	0	0
1	127299	0	0
2	54788	0	0
3	5400	0	0
4	128613	0	0

	judgement_penalty_amt
0	0
1	0

```
2                       0
3                       0
4                       0

[5 rows x 56 columns]
```

3 Look at things

3.1 Amounts of money

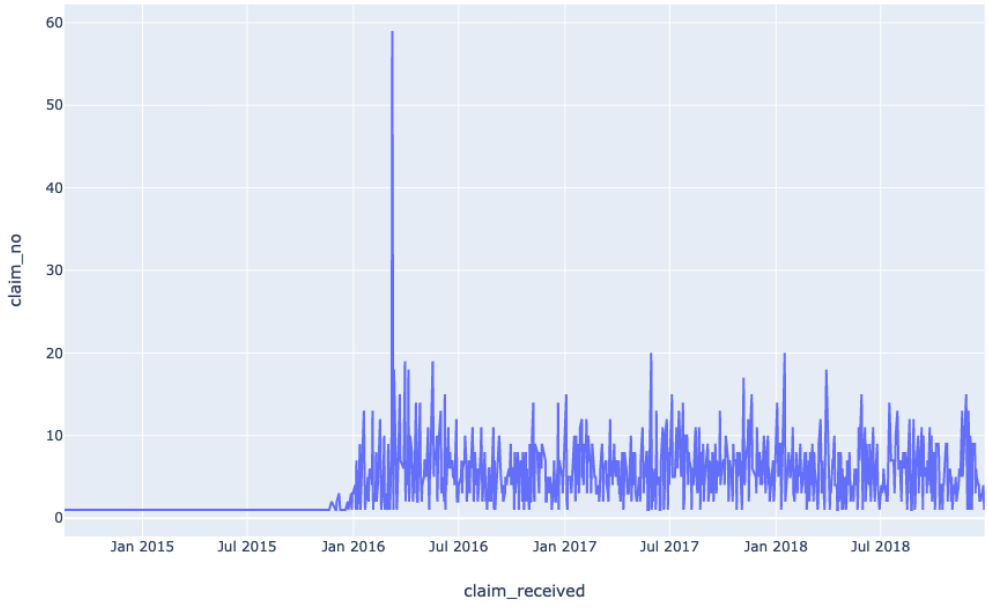
```
In [31]: df[money_cols].sum().to_frame().reset_index()\
             .rename(columns={'index':'item', 0:'amt_total'})\
             .sort_values('amt_total', ascending=False)
```

Out[31]:

	item	amt_total
0	claim_amt	1108830110
1	verified_claim_amt	64983670
15	tot_wages_and_interest	61616903
13	total_wages	60996554
2	tot_amt	1208029
3	wage_amt	906022
10	judgment_wage_amt	826575
12	total_cmp_amt	733332
8	total_cmp_amt	733332
14	total_interest_amt	620349
6	amt	550000
11	judgement_penalty_amt	446641
9	fines_and_court_costs	391304
5	atty_fees	302007
7	atty_fees_1	183332
4	interest_amt	0
16	garnished_amt	0

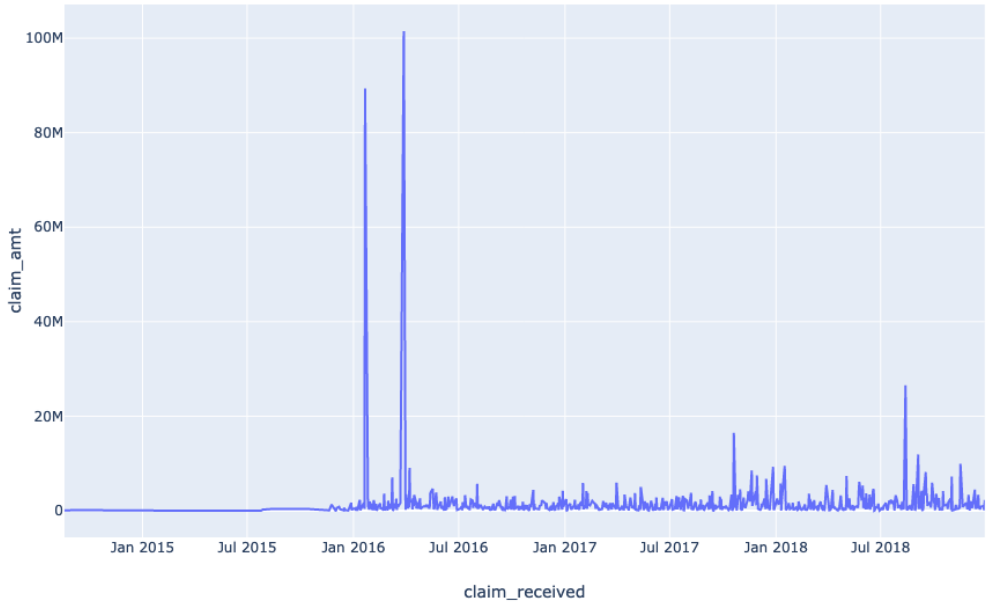
3.2 Number of claims over time

```
In [32]: px.line(df.groupby('claim_received')['claim_no'].count()\
                 .to_frame().reset_index(),
                 x='claim_received', y='claim_no')
```



3.3 Amount of claims over time (aggregate)

```
In [33]: px.line(df.groupby('claim_received')['claim_amt'].sum().reset_index(),
                 x='claim_received',  y='claim_amt')
```



3.4 Save prepared data to SQLite db

This database can be used later.

```
In [34]: with sqlite3.connect('lajc.db') as db:
             df.to_sql('wage_claim', db, if_exists='replace')
             codebook.to_sql('codebook', db, if_exists='replace')
```

4 Answer Questions

4.1 Question 1: Determine what % of claims get wages ordered to be returned

```
In [35]: codebook.loc['wage_order', 'Description']
```

```
Out[35]: 'Whether a legal order is given for the employer to pay lost wages '
```

```
In [36]: q1 =  round((df.wage_order.value_counts().values[0] / N) * 100, 2)
```

```
In [37]: q1
```

```
Out[37]: 99.77
```

4.2 Question 2: Determine what % of claims get 1st response and formal investigations opened

```
In [38]: q2v1 = '1st_response_investigation'
         q2v2 = 'formal_investigation'
```

```
In [39]: q2_a = round((df[q2v1].value_counts()[0] / N) * 100, 2)
```

```
In [40]: q2_a
```

```
Out[40]: 31.05
```

```
In [41]: q2_b = round((df[q2v2].value_counts()[0] / N) * 100, 2)
```

```
In [42]: q2_b
```

```
Out[42]: 80.45
```

```
In [43]: q2_c = df[(df[q2v1] == True)  & (df[q2v2]  == True)].values
```

```
In [44]: q2_c
```

```
Out[44]: array([], shape=(0, 56), dtype=object)
```

```
In [45]: len(q2_c)
```

```
Out[45]: 0
```

4.3 Question 3: Determine what % of claim are found to be valid, invalid, and undetermined

```
In [46]: q3v1 = 'valid'
         q3v2 = 'invalid'
         q3_a = df[df[q3v1] == True].shape[0]
         q3_b =  df[df[q3v2] == True].shape[0]
         q3_c = N - (q3_a + q3_b)

In [47]: q3_a, q3_b, q3_c

Out[47]: (173, 1038, 2737)
```

4.4 Question 4: How long does it take to conduct an investigation?

```
In [48]: start_date_col = 'case_open_re_open'
         end_date_col = 'case_close_reclose_date'
         df['how_long_investigation'] = df[end_date_col] - df[start_date_col]

In [49]: df.how_long_investigation.describe()

Out[49]: count                        3948
         mean     16 days 19:35:11.854103
         std      39 days 03:46:07.942834
         min             0 days 00:00:00
         25%             0 days 00:00:00
         50%             1 days 00:00:00
         75%            22 days 00:00:00
         max          1307 days 00:00:00
         Name: how_long_investigation, dtype: object
```

4.5 Question 5: How long does it take to get a wage order?

```
In [50]: start_date_col = 'claim_received'
         end_date_col = 'date_for_wages'
         df['how_long_get_wage_order'] = df[end_date_col] - df[start_date_col]

In [51]: df.how_long_get_wage_order.describe()

Out[51]: count                          4
         mean          118 days 06:00:00
         std      35 days 10:52:20.976187
         min            71 days 00:00:00
         25%           101 days 00:00:00
         50%           128 days 00:00:00
         75%           145 days 06:00:00
         max           146 days 00:00:00
         Name: how_long_get_wage_order, dtype: object
```

4.6 Question 6: If wages are ordered to be returned, what's the distribution of amounts?

```
In [52]: df['tot_amt'].value_counts().sort_index()
```

```
Out[52]: 0         3939
         7200         1
         8000         1
         45333        1
         79308        1
         140921       1
         164467       1
         216000       1
         228800       1
         318000       1
         Name: tot_amt, dtype: int64
```

4.7 Question 7: How do these outcomes listed above depend on ethnicity, gender, and industrial category?

4.7.1 Ethnicity

```
In [53]: round((df['hispanic'].sum() / N)  * 100, 2)
```

```
Out[53]: 8.56
```

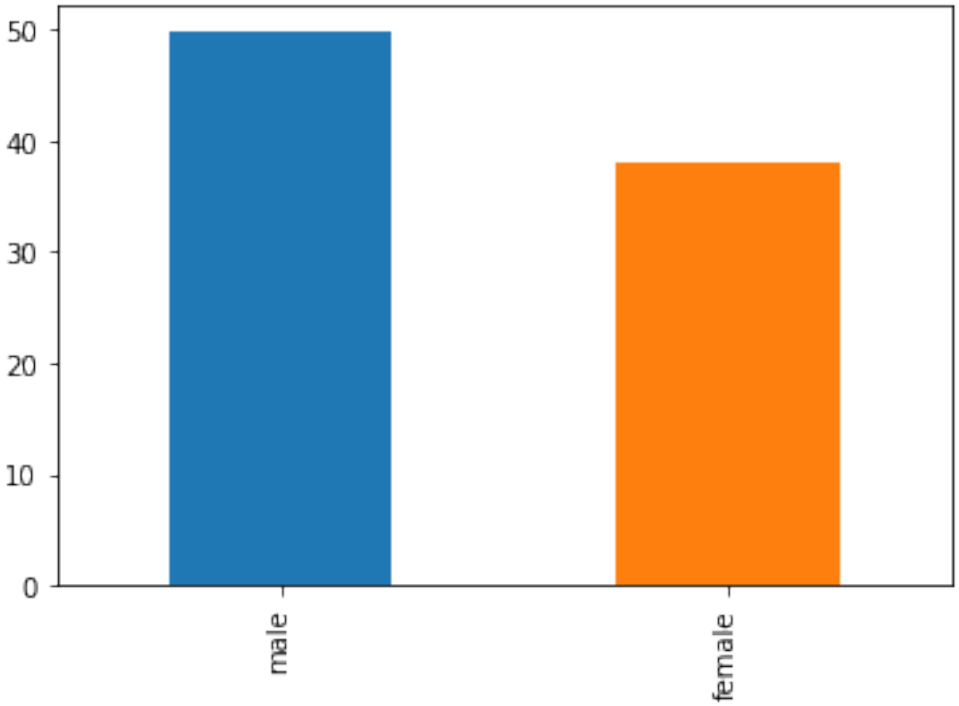
```
In [54]: round((df['asian'].sum() / N)  * 100, 2)
```

```
Out[54]: 2.89
```

4.7.2 Gender

```
In [55]: round((df['gender'].value_counts() / N) * 100, 2).plot(kind='bar')
```

```
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x1220724a8>
```



4.7.3 Industry

4.7.4 20 Most offending cities

```
In [56]: df.groupby(['employer_city', 'st']).claim_no.count()\
             .to_frame().sort_values('claim_no', ascending=False).head(20)
```

Out[56]:

employer_city	st	claim_no
RICHMOND	VA	368
VIRGINIA BEACH	VA	274
CHESAPEAKE	VA	161
WOODBRIDGE	VA	129
NORFOLK	VA	124
NEWPORT NEWS	VA	108
FREDERICKSBURG	VA	101
MANASSAS	VA	95
ALEXANDRIA	VA	81
HAMPTON	VA	78
STERLING	VA	73
ROANOKE	VA	70
CHANTILLY	VA	67
FAIRFAX	VA	58
ARLINGTON	VA	55

```
PORTSMOUTH      VA        54
HENRICO         VA        51
VIENNA          VA        48
FALLS CHURCH    VA        41
CHARLOTTESVILLE VA        41
```

4.7.5 20 Most offending zip codes

```
In [57]: df.groupby(['employer_city', 'st', 'zip']).claim_no.count()\
             .to_frame().sort_values('claim_no', ascending=False).head(20)
```

Out[57]:

employer_city	st	zip	claim_no
WOODBRIDGE	VA	22192	94
CHESAPEAKE	VA	23320	77
VIRGINIA BEACH	VA	23462	62
CHANTILLY	VA	20151	56
STERLING	VA	20166	53
VIRGINIA BEACH	VA	23452	50
MANASSAS	VA	20109	43
RICHMOND	VA	23230	39
HAMPTON	VA	23666	38
VIRGINIA BEACH	VA	23454	38
RICHMOND	VA	23235	37
		23224	36
MANASSAS	VA	20110	33
VIRGINIA BEACH	VA	23455	32
NEWPORT NEWS	VA	23606	32
VIENNA	VA	22182	32
NORFOLK	VA	23510	30
RICHMOND	VA	23220	29
MIDLOTHIAN	VA	23112	29
VIRGINIA BEACH	VA	23451	28

4.7.6 20 Most offending employers

```
In [58]: df.groupby(['employer_name','employer_city', 'st', 'zip']).claim_no.count()\
             .to_frame().sort_values('claim_no', ascending=False).head(20)
```

Out[58]:

employer_name	employer_city	st	zip	claim_no
FIRST TRANSIT	WOODBRIDGE	VA	22192	71
CHESAPEAKE SERVICE SYSTEMS	CHESAPEAKE	VA	23320	11
ANCHOR BAR	RICHMOND	VA	23235	11
A.C.I. DRYWALL CIA	SHADY SIDE	MD	20764	7
CHESAPEAKE SERVICE SYSTEMS, INC.	CHESAPEAKE	VA	23320	7
BRAVEN PAINTING LLC	VIRGINIA BEACH	VA	23453	6
EPC BUILDERS LLC	ROCKVILLE	VA	23146	5
PUBLIC PARTNERSHIP LLC	GLEN ALLEN	VA	23060	5

CARLISLE LIVING LLC	VIRGINIA BEACH	VA	23464	5
THOMAS SWANSTON	VIRGINIA BEACH	VA	23454	5
ROYAL CLEANING SERVICES	NORTH CHESTERFIELD	VA	23236	5
MINISTERING ANGELS LLC	WINCHESTER	VA	22601	5
TRUSTIFY	ARLINGTON	VA	22202	4
GREATER WASHINGTON ENDODONTICS	FAIRFAX	VA	22031	4
FCI ENTERPRISES LLC	CHANTILLY	VA	20151	4
INTEGRATIVE CENTERS FOR SCIENCE AND MEDICINE	MARTINSVILLE	VA	24112	4
BODY & SOL TANNING LLC	YORKTOWN	VA	23693	4
PUBLIC ELECTRONICS	SAUSALITO	CA	94965	4
SMYTH COUNTY AMBULANCE SERVICE	MARION	VA	24354	4
THE TREAT SHOP	RICHMOND	VA	23225	4

4.8 Fix time diff cols for storing in database

In [59]:
```
# how_long_cols =  [col for col in df.columns.tolist() if 'how_long' in col]
# for col in how_long_cols:
#     df[col] = df[col].astype('str').str.split()
#     df[col] = df[col].apply(lambda x: x[0])
#     print(df[col].head())
```