Data Diary

Monday, April 18, 2016
3:20 PM

4/17/2016

I used the database I created previously from the online WHISARD data to do this analysis. It's called cpi_whd20052015.

Commented [BT1]: I date my entries so I know where I last left off

Commented [BT2]: File names/locations highlighted in yellow

I used this query to find cases that were positive for violations under the Public Contracts Act, Service Contract Act or the Davis-Beacon Acts. All of those laws are only applicable to government contractors. It yielded 10680 results.

Commented [BT3]: Write your entries in a way so that if you step away and have to pick up where you left off, you can read the previous day's entry and clearly understand where you are in the analysis.

```
SELECT *
  FROM [Wage and Hour].[dbo].[cpi_whd20052015]
  WHERE sca_violtn_cnt>'0'
  OR dbra_cl_violtn_cnt>'0'
  OR pca_violtn_cnt>'0'
```

4/18/2016

I spent today trying to figure out how to do joins on the USASpending database that Alex imported into SQL and the WHISARD online database entered under cpi_whd20052015. I started with David Donald's syntax from the Old Data Methodology for doing a join between the contracts database and the company names in the WHISARD database:

```
SELECT DISTINCT a.CPI_act_specific_info_nohypens, b.reference_idv
FROM CPI_clean_case_act_eer_viol AS a INNER JOIN cpi_govtcontracts2 AS b
ON CPI_act_specific_info_nohypens = reference_idv
```

Commented [BT4]: Noting queries also ensures you can correct errors if they occur, and helps you keep track of what you're doing. Consider putting Queries in Red to make them easy to find.

I couldn't get the syntax to work, so I asked Alex to help me with it. He did some work and came up with this syntax -- he had to do some stuff to the cpi_whd20052015 table so that it could be compared to vendorname in USASpending. This is the ultimate syntax he came up with using the alias I was attempting:

```
SELECT a.vendorname, b.trade_nm_varc
FROM GSA.Contracts AS a
JOIN [Wage and Hour].dbo.cpi_whd20052015_notxt AS b
ON a.vendorname=b.trade_nm_varc
GROUP BY a.vendorname, b.trade_nm_varc
```

When I ran this query, matching the two databases by name, I came up with 6,707 matches. I need to narrow the universe I'm working in, just to make it more manageable. The WHD is from 2005 to 2015. First, I need to figure out what years the USASpending database is from. I tried this first:

Commented [BT5]: Logic. Writing it out helps make sure you are actually doing what you think you're doing. Tell yourself what you were trying to do. That way you can go back and figure out why it did or didn't work.

```
SELECT COUNT (DISTINCT fiscal_year)
FROM GSA.Contracts
```

After 7 minutes, I got 37. So...no. Maybe this?

```
SELECT fiscal_year, COUNT (DISTINCT fiscal_year)
```

```
FROM GSA.Contracts
GROUP BY fiscal_year
```

I don't think this got me exactly what I wanted -- which was the number of times each fiscal year occurred in that column -- but it got me what I needed, which is a list of the years that this dataset covers. The earliest is 1979; latest is 2015. How do I want to narrow this dataset? I'm going to look at contracts awarded between FY 2005 and FY 2015. The Fair Pay and Safe Workplaces Executive Order applies to contracts of more than $500,000 and any violations that occurred within three years of the contract. So, the contracts, I'll narrow to 2005 to 2015. **The violations should be narrowed to 2002 to 2015 (accounting for three years before and after? Or Just 2002 to 2015?**

Commented [BT6]: Thinking through logic; this can also be where you include decisions you made about the data. Explain your logic so that when you wonder why you made a particular choice, you have a record of what you were thinking and why. Also helpful when it comes to writing your methodology – you have a record of assumptions/choices/decisions you made throughout the analysis and can clearly explain why you made them.

Tried this first:

```
SELECT *
FROM GSA.Contracts
WHERE fiscal_year>=2005
GROUP BY fiscal_year
```

NO. Got this error in return:

"Msg 8120, Level 16, State 1, Line 1
Column 'GSA.Contracts.unique_transaction_id' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause."

Commented [BT7]: Note your errors. In the beginning, it will help you figure out where you went wrong, or how to fix it in the future. It's also helpful to note if you get an error code in case you need to call in someone more experienced for help – having that information helps them know exactly what went wrong.

<u>5/23/16</u>

Asked Ben for help with Trim function at IRE suggestion. He helped me craft this to take out all of the trailing and leading spaces in the CPI clean legal name column:

```
UPDATE dbo.WHD_subset
SET CPI_clean_legal_nm=LTRIM(RTRIM(CPI_clean_legal_nm))
```

I was going through and trimming some endings that I thought might make the cleaning process easier – DDS, P.A., etc. In the trimming, it seems I may have deleted some characters inside of words – not just those that are at the end of the string.

So, one of the suffixes was 'CO'. I ran this query:

```
UPDATE dbo.WHD_subset
SET CPI_clean_legal_nm = REPLACE(CPI_clean_legal_nm, 'CO', '')
```

(Also did this with 'LL', 'LC,' and 'NA' and a bunch of other characters, unfortunately. They were all really things that were at the end of a string, but my query syntax was off, I guess.)

Which, looking at it now, I see why that's not correct. Thankfully, I did this in the field I created, so my original data is still fine. I just honestly don't know what to do now. I'd already gone through 25,000 names and cleaned them in other ways before I got to this part, so I'm hesitant to just throw that away, make a new CPI clean column and start over. I know it's difficult, if not impossible, to undo in SQL. I'm just hoping you can help me figure out what I should do.

Commented [BT8]: THINGS WILL GO WRONG. If you've been detailing your logic, it'll be easier to see what you did incorrect. And possibly help you sort through how to fix it. Feel free to talk to yourself.

Ben said he'll help me figure things out some time tomorrow.