**Excel CARwash: Cleaning dirty data**

**Description**: Dirty data lurk everywhere. We'll walk through some integrity checks to help diagnose problems with your data and learn how to deal with some of the most common problems, including standardizing misspelled names. This session is good for: People with some experience working with data in columns and rows, in spreadsheets or database managers.

**Objectives**: Practice using a typical data cleaning workflow in Excel, including filtering and pivoting to perform integrity checks to find misspellings, nonstandard values, etc.

**Data used in this session**:

* *ildebtor.xlsx* -- Child support orders from Illinois
* *tngives.xlsx* -- A cut of FEC federal campaign finance data with donors from Knoxville, Tennessee

Talk about best practices for cleaning data: Create a new copy of data and leave originals untouched, don’t clean original columns -- leave them as a reference. Etc. OpenRefine is a good tool for data cleaning, but lots of the time you can get what you need done in Excel.

First, open *Ildebtor.xlsx*.This is a file of child support orders in Illinois.

Pivot table to group by city to find misspellings, e.g., of *Chicago* -- talk about running integrity checks on any text field that will be part of your analysis. Filters can also be a way to get a quick read of data cleanliness, which is useful when estimating the time needed to clean and analyze a data set.

Create a new *clean\_city* column, and copy+paste city column into it. Filter to find Chicago misspellings, fix manually (bulk find-replace). Have them look for other cities and fix a couple others for practice.

Pivot on names to find similar names, point out that can’t group these people together using names alone. Need to look at address or sometimes do other reporting while you’re cleaning names.

Sort on numeric and date columns to check for outliers.

Talk about the purpose of writing a good data diary. At this stage, as you’re running integrity checks, you’re fixing obvious problems and noting anything that looks weird to ask your data source about later.

Next, open *tngives.xlsx*. This is a cut of some FEC campaign finance data with donors from Knoxville.

Discussion: Do you need to clean up every value in every column of data before you can start extracting answers from your data? Nope! Tailor your cleaning and analysis to the questions you need to ask your data.

Let’s say an initial goal is to identify the top family donors. Run a pivot table to group by *last* and sum *amount*, then sort by the amount column. The Haslams! (Politically connected family, owner of Pilot/Flying J, Bill Haslam was governor.)

Next, have students sort by last name and look for misspellings/nonstandard entries for Haslam, Pilot,. Flying J, etc. Make a list of entries to standardize. Talk about how you would need to do additional reporting to make sure that each person/entity is, in fact, connected to the Haslam family.

Make a *contributor\_name\_clean* column and copy in the contributor name values. Sort data by last name and start looking for inconsistent names, then fixing them.

**Excel: Using string functions to manipulate data**

**Description**: Maybe you converted a PDF or imported a table into Excel -- or maybe an agency gave you a poorly formatted file. You can use string functions to reformat your data and get your spreadsheets working for you. This session is good for: Anyone comfortable with using formulas in Excel.

**Objectives**: Learn a variety of Excel functions (string, date, IF) to clean and reshape your data.

**Data used in this session**:

* *Excel\_Cleaning.xlsx* → A series of worksheets with cleaning problems to work through.

Overview: No matter what your goal is (cleaning, reshaping, reformatting, etc.), stress that you need to look for patterns in the data to find the correct solution. When you’re cleaning data, start a list of things that are wrong, or look wrong -- document as part of your data diary.

Also, when you’re cleaning, don’t touch the data in the original columns. Having the original data on hand to reference will save you time when bulletproofing your analysis. Typically I clean in several passes: E.g., fix whitespace, then you notice punctuation typos, incorrectly spelled cities, etc.

Mention [OpenRefine](http://openrefine.org/) as the best power tool for cleaning data -- a common workflow involves examining the data in Excel, pulling into OpenRefine for cleaning, then exporting a clean file for analysis in Excel or other tools.

Open *Excel\_cleaning.xlsx* and start working through the worksheets.

**Worksheet: Trim**

This is a couple of columns from a data set of lottery winners -- but the game names need cleaning.

Create a pivot table with the *Game* column in Rows to show the problem: leading and trailing whitespace means that Excel is interpreting some as different, incorrectly -- pivot tables group items literally and exactly. Our first step will be to use the *=TRIM()* function to strip whitespace.

Back in your main data sheet, create a new column: *Game-Cleaned*

In that column, use a trim function on the original *Game* column -- *=TRIM(A2)* -- and fill down.

Select your new column by clicking on the column letter label. Copy, then Paste Special in place to remove the formula -- show before/after in the formula bar.

Run a new pivot table to show how it worked. Scroll down the list to see what *else* is wrong and needs fixing.

**Worksheet: Dates**

The *Approval\_date* column (F) was imported as text, so need to pull apart the date into its constituent parts -- year, month, day -- and then put it back together as a date. We’ll also exploding the date we create into year/month/day pieces, which can be useful for aggregating data by year, month, etc.

The new columns we’re going to populate -- *Year, Month, Day, New\_appr\_dt, Year, Month, Day* -- exist already. Here are the formulas we’re going to use:

* *=LEFT()*, to extract the leftmost *n* characters from a string.
  + In G2, under the first *Year* column, enter *=LEFT(F2, 4)* and fill down.
* *=MID()*, to get *n* characters from the middle of a string.
  + In H2, under the first *Month* column, enter *=MID(F2, 5, 2)* and fill down. The first argument is the string to pull from; the second argument is the starting point; the third is how many characters to take.
* *=RIGHT()*, to extract the rightmost *n* characters from a string.
  + In I2, under the first *Day* column, enter *=RIGHT(F2, 2)* and fill down.
* *=DATE()*, to construct a date object out of year, month, day parts.
  + In J2, under the *New\_appr\_date* column, enter *=DATE(G2, H2, I2)* and fill down.
  + Select column J by clicking on the letter J
  + Copy and paste special values to remove the formula
* *=YEAR()* to extract the year from a date.
  + In K2, enter *=YEAR(J2)* and fill down.
* *=MONTH()*, to extract the month from a date.
  + In L2, enter *=MONTH(J2)* and fill down.
* *=DAY()*, to extract the day from a date.
  + In M2, enter *=DAY(J2)* and fill down.

**Worksheet: Split1**

We need to split the city and state so that they we can group by state. In general, you want to format your data to be as specific as the questions you have for it are.

* In G2, enter *=MID(D2, 5, 100)* and fill down. (Explain why 100)
* In H2, enter *=LEFT(D2, 2)* and fill down.

There are different ways to accomplish the same task -- you might insert columns and use the “Text to Columns” command.

**Worksheet: Split2**

Problem here is, the number of characters to the left of the comma changes each time. The pattern we need to encode in a formula is: Take all of the characters to the left of the comma, wherever that is. To do that we’ll use a function called *=FIND()*.

* Insert 3 empty columns (B, C, D) to the right of the *contributor* column
* In B2, enter *=FIND(",", A2)* and fill down. This formula gives the numeric position of the comma in the string.
* In C2, enter *=LEFT(A2, B2-1)* and fill down to get the last name. Explain why you’re subtracting one -- to leave out the comma -- and show what happens if you don’t.
* In D2, enter *=MID(A2, B2+2,100)* to get the rest of the name. Explain why you’re adding two -- to leave out the comma and the space -- and why you’re using 100.

**Worksheet: BasicIF**

What if you want to calculate a value, or return a string, based on the results of some logic? You’ll use something called an *=IF()* statement. Here, we want to write “WIN” in a new column if the Cards won the game and “LOSS” if they lost.

* In G3, add a new column header: OUTCOME
* In G4, enter =IF(E4>D4, "WIN", "LOSS") and fill down. Step through the function logic one piece at a time and explain what each thing is doing -- you might need to introduce the concept of TRUE and FALSE booleans first by showing what the result of E4>D4 is.

**Worksheet: NextIF**

The scenario: Newsroom employees are all getting a 1% raise unless that amount is less than $350, in which case they’ll get $350 -- that’s the floor that our union negotiated. We can figure that using an *=IF()* statement.

* In G6, enter =IF(F6\*0.01>350, F6\*0.01, 350) and fill down. Step through each part of the formula, slowly, explaining what each thing is doing.
* In H6, enter *=F6+G6* to get the new salary value and fill down.

**Worksheet: Teamname**

This kind of untidy data is common (“summary cell” that applies to multiple subsequent cells). Our goal here is to add the team name to each player row so that we can sort, filter, pivot, etc.

Look for the pattern. What happens when a new team name appears? There’s a blank cell in column B. That’s going to be our hook.

* In column C, create a column header: *Team*
* In C2, enter this formula: *=IF(B2="", A2, C1)* and fill down.
* Hard to explain, but start by using the “false” case from C3 as a starting point. Here, we’re using contextual knowledge (“I am going to fill this formula down”) to work out the correct logic.

**Worksheet: Blanks**

Here, we want to fill in the blanks in column A with the correct year. Lots of new stuff on this one, so don’t try to anticipate commands -- we’ll walk through each step together.

* Highlight Column A, then select all the blanks:  Edit > Find > Go To > Special > Blanks
* With the blanks highlighted -- don’t touch anything else! -- type *=* and then hit the up arrow, which will put *=A1* into cell A2.
* Don’t hit enter! Instead, hit Ctrl+Enter, which will populate the formula in each of the blank cells we have selected, and each cell will refer to the one *above* it.
* Select the column and copy/paste special values to get rid of the formulas
* Next, we need to get the rep’s name copied into Column C:
  + In C2, enter *=IF(LEFT(B2,3)="Rep", B2, C1)* and fill down. Walk through each piece of this formula, slowly, and explain what each one is doing

**Worksheet: Reshape**

We want to reshape this data to get the address on the same line. What’s the pattern? Three blank rows in between.

* In F8, enter *=B9*
* In G8, enter *=B10*
* Highlight from F8:G11 (the two cells of data we just created and three blank lines) and drag down *all the way* -- will fill in every four lines correctly

**Excel 3: Filtering and pivot tables**

**Description**: A look at the awesome power of pivot — and how to use it to analyze your dataset in minutes rather than hours. We'll work up to using a pivot table by first sorting and filtering a dataset, learning how to find story ideas along the way. This session is good for: Anyone familiar with formulas and navigating Excel or another spreadsheet program.

**Objectives:** Practice using filters and pivot tables to find stories by comparing aggregates across groups of data.

**Data used in this session**:

* *Mlb2018.xlsx* → Opening-day 2018 salaries for MLB players

Open *mlb2018.xlsx*. This is MLB opening-day player salary data from USA Today -- explain why it’s important to understand your data. See source notes worksheet.

Any baseball fans? Who’s our favorite team? (Go with whoever yells first.)

Sorting and filtering

* Sort: Let’s look at teams alphabetically, not players. Select all → Data → Sort by Team
* Filter: Turn filters on, find a Team to focus on
* Create a new worksheet. Select all (including header rows), then copy+paste special values into a new tab
* Reformat salary column as currency
* Rename your worksheet (“Tigers” or whatever)
* In your new worksheet, after a blank row, =SUM() all the salaries. Make sure that you didn’t accidentally bring over any hidden players you don’t see with the filter. Also do =AVERAGE() and =MEDIAN()
* Talk about the language you’d use in a story -- e.g., a median value is the “typical” value

Pivot tables!

* *They will change your life*
* Pivot tables are good for comparing aggregates -- e.g., grouping all the players from each team into a “bucket” and adding up the salaries inside to compare it with totals from other buckets
* Begin by visualizing the output you want, e.g., a column with team names, a column with salary totals
* Select ALL (control A instead of top arrow)
* Insert → Pivot Table
* In the wizard, the data being pivoted is what you selected
* Quick tour of the pivot table builder
* What is the total salary by team?
  + The thing that you want to group by goes in *rows*
  + The thing that you want to do math on goes in *values*
* Drag *team* into *rows* (caution: If you just click the box, it will guess!)
* How many MLB teams? Go to bottom, subtract 3
* The thing we want to know, to do math on, goes into values, so drag *salary* into *value*
* It summed (probably). How do we know? Check the labels at the top
* Shows you grand total at the bottom
* Reformat cells to currency
* How much in salaries for all baseball? $3.6B
* Sort: You can break the select all rule!
  + Put cursor in column, can use quick buttons
  + Could also right-click and use the *sort* option in the context menu
  + Who are the top payroll teams?
  + Where does our team rank?
* Let’s change the math: click on Sum of Salary, then click on Value Field Settings
* Click Count
* Reformat column to *number*
* What does *count* do? Counting the number of salaries you put into the group (how many people on the team)
* Why these numbers? Includes injured -- whenever you find a pitfall like this, write it down and go back to documentation or call someone
* What happens if your builder window disappears? Click back into your pivot table and it should reappear.
* Now drag *salary* to values again and select “average” from Value Field Settings
* Resort Z-A
* Pivot table takes the mean -- doesn’t do medians, so don’t use on data sets with huge outliers
* Total at the bottom of that column is the *average* of the total
* Add *salary* to values again and select *max* from Value field settings -- the highest single number in that group
* Who is that? Double click on the value -- creates a new sheet up at the bottom with the individual records that make up the calculation
* Rename your sheets so you know what they are
* Subgroups: Drag *position* into rows under *team*
  + Breaks it out by position (subgroups)
  + What happens if you change the order?
* Columns: Add a new dimension to your data -- set up your pivot table so *team* is in rows, *salary* (sum) is in values, and *position* is in columns
* Filters: Look at pivots for a subset of your data -- e.g., team in rows, salary in values, filter to just include pitchers
* How you shape your data will always depend on the questions you’re trying to answer
* Pivot tables are also good for integrity checks and getting to know data.
  + Run a Pivot Table just to see a list of teams. How many am I expecting (30). How many are there?
  + Look at what’s in each column -- compare what you’re seeing with what you’d expect to see (use common sense)

Exercises on your own (or in small groups) -- answer these and other questions using any of the tools we’ve worked with so far (sorting, filtering, pivot tables):

* The team with the highest *average* pay for its pitching staff is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
* How many players make the league minimum?
* Collectively, MLB catchers were paid \_\_\_\_\_\_\_\_\_ in 2018.
* What else?