1. The data wrangling steps to clean the dataset.

For this project I had the data in three different .csv files.

First, I created the oil production metric that I’ll building a model for. I chose to predict well cumulative oil production for the first year (I called it cum\_oil\_365). I had to engineer this parameter from the data provided in ‘monthly-production.csv’ file. The file contains multiple records (rows) for each well. Each row reports production volumes for different fluids for each month as well as number of days the well was producing during this month. The task was to find how much oil each well produces for the first 365 days.

I took the following steps to create cum\_oil\_365 metric:

* 1. Loaded data to a data frame
  2. Assigned datetime types to dates
  3. Indexed by unique well ID followed by Reported production date (month and year)
  4. Calculated a new column of production per day for each record
  5. Summed product of production\_per\_day and number of days over the rows till there was no more rows to sum or days on production exceeded 365.

There were 8939 unique wells in the database originally. After dropping wells that do not have oil production reported I’m left with 6307 wells.

The next step was to prepare feature data for the wells that have production data. Well parameters and measurement are contained in two different datasets: ‘completion.csv’ and ‘well-index.csv’.

First, I loaded ‘completion.csv’ file into a pandas data frame. I dropped columns that I believe not important (e.g. file number) and assigned correct types to dates and categorical data.

Data from ‘well-index.csv’ file was also loaded into a data frame. Some categorical data (e.g. Well Status data) had “Confidential” values that I treaded as missing data and replaced with empty string for further processing.

After merging all three datasets on the API (unique well id) key I have 46 feature variables and 6440 records of which 5987 are unique wells. This implies that some wells have multiple records.

Removing duplicate rows resulted in 6433 records.

Next, I split the dataset into two. The first one has only wells with single record per well. There are 5727 of such wells. The rest of the data must be worked on more carefully. I cannot apply the same aggregation methods to all the features in the dataset with multiple rows per well. Some columns it makes sense to sum, some to average, others to make extreme values (min or max). The aggregation method also depends on the particulars of a well treatment (e.g., the treatment repeated for the same interval in the same zones/stages by pumping more fluids, more stages were created in the same interval, the new interval treated and opened for production that does not overlap with an old one, the new interval treated and opened for production that overlaps with an old one.). After carefully analyzing each scenario and applying appropriate aggregation methods I reduced 706 record to 260 and merged them back to the single record data frame.

There are three columns with dates in the dataset: Well\_Status\_Date, treatment\_date, Spud\_Date. Each column was split into 3 different columns of year, month, and day.

Next, I removed outliers for some of the variables if the value was different from the mean by three standard deviations. Data points with infinite or none values were replaced by the averages.

The next step in data prep is to encode categorical data. For majority of variables I chose simple encoding scheme by replacing a string in a category with a number. This is the case for any variable that has too many categories or if values in the variable are dominated by one category. The only categorical variable I chose to do one hot encoding is Produced\_Pools (geological zone from which a well is producing).