Paper 125-2009

Wholesale Electric Energy Market Analysis with Electronic Quarterly Reports Data
Sarah Whittier, ISO New England, Holyoke, MA

Abstract
In the first six months of 2008, physical sales of wholesale electric energy in New England totaled approximately $13.4 billion. Suppose you want to get a better understanding of this market: you download the 40GB+, Visual FoxPro database of Electronic Quarterly Reports (EQR) data from the Federal Energy Regulatory Commission (FERC) website and open a SAS session. Where do you go from here? This paper explains how to bring EQR data into SAS, clean it up, and produce useful reports, using the example of a project analyzing New England transactions. The topic of this paper is how to analyze EQR data, not how to meet the FERC's EQR filing requirements. The SAS® products used were Base SAS and SAS/Access for OLEDB.

Introduction
A little bit about wholesale electric energy markets in New England
Wholesale electric energy can be purchased from ISO New England's (ISO-NE)[1] day-ahead and real-time markets, on power exchanges such as the Intercontinental Exchange (ICE), or through bilateral contracts between companies. Depending on the forum where the transactions occur, they may be purely financial or for physical delivery of power.

Energy bought through the ISO-NE's markets is priced the same for all buyers, with variations for location and hour of purchase. Prices of bilateral transactions, on the other hand, vary depending on the terms negotiated by the contracting parties. Contract lengths for these transactions can vary from hours to years. Parties to bilateral transactions do not provide the terms of those contracts to ISO-NE. However, analyzing bilateral contract activity is useful because it helps provide a more complete picture of the wholesale electricity market.

What is Electronic Quarterly Reports Data?
The Federal Energy Regulatory Commission (FERC) requires companies to make quarterly filings of information about sales of wholesale physical electric energy and related products. As explained in the FERC staff report, "*Draft Report to Congress on Competition in the Wholesale and Retail Markets for Electric Energy,*" the Electronic Quarterly Reports requirement applies to "all holders of market-based-rate authorizations.[2]" Sales that are reported may take place in a variety of forums. The report explains, "Sales reportable to EQR could have been transacted bilaterally, on RTO/ISO's, through ICE or through voice brokers, and credit cleared through ICE-LCH or NYMEX-ClearPort." The EQR filing requirement excludes transactions that are purely financial as well as retail sales to native load and wholesale sales by municipal power companies and some other types of companies. Virtually all New England transactions on ICE are financial, and there is very little New England activity on NYMEX.

The main challenges to working with EQR data are: the database is in Visual Foxpro; there is a very large amount of data; free-form text is allowed in some columns, including the buyer name column; it is difficult to determine the length of contracts; it is difficult to determine whether energy transaction prices might be for "all requirements"; and the data includes flat rate transactions (for these, only the total dollar amount is available, not the MW quantity or cost per MW). In addition, the analyst should keep in mind that while EQR data gives price and volume data for transactions, other contract terms and profits on transactions are unknown.

Goals of my project
Despite the limitations listed above, EQR data is an excellent resource for understanding the bilateral wholesale electricity market. I undertook this project to gain a better understanding of transactions that occur outside of ISO New England's markets.

My goal was to look at transaction volume and cost for different companies and groups of companies in New England. These groups include retail suppliers, marketers/traders, and generation owners

---

[1] ISO New England (ISO-NE) is a regional transmission organization, serving Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont. ISO-NE is responsible for the reliable operation of New England's bulk electric power system and the development, oversight, and fair administration of New England's wholesale electricity marketplace. For more information about the markets run by ISO-NE, visit http://www.iso-ne.com/nwsiss/grid_mkts/how_mkts_wrk/index.html.

[2] http://www.ferc.gov/legal/staff-reports/competition-rpt.pdf

Analysis
Getting started with EQR data
The first step is to download the EQR database from the FERC website: *http://www.ferc.gov/docs-filing/eqr/data/database.asp*. Download the "All Years" zip file and the transaction zip files for whatever quarters you wish to analyze and unzip the files. You will also need to make sure that you have Visual FoxPro drivers installed on your computer. On my PC, I navigate to Control Panel, Administrative Settings, Data Source (ODBC) to see the drivers that I have installed. Verify that the Visual FoxPro Database driver is listed. If it is not, you can download it from Microsoft.

Open a SAS session and define a libname to point to the location of the unzipped EQR data. You will need to have the SAS/Access for OLEDB module installed in order for this code to work:

```
LIBNAME eqr_db  OLEDB
   provider='VFPOLEDB.1'
   properties=("data source"="\\directory_name\EQR\EQR 2008 Q2");
```

The eqr_db library will appear in the Explorer pane, and if you open the library you will see many individual tables. Because the tables are extremely large, do not attempt to open them in browse mode.

The next step is to create SAS datasets from the files. Due to the data volume, I was not able to work with the files on my PC, so I uploaded them to a server and did all my analysis there. However, in order to simplify the code shown in this paper, I have removed the SAS/Connect, RSUBMIT, and PROC UPLOAD steps. You should evaluate your setup and determine the appropriate location to work with these files.

You do not need to create SAS datasets for every table from the EQR database: I found that I did not use all the tables. The EQR database schematic, available from the FERC website, is helpful. The tables used in my project were:

EQR_IDENT_ATTSTN

EQR_CONTRACTS: *Keep all records from the EQR_CONTRACTS table in your SAS dataset: limiting to just the dates of the quarters in the transaction table may result in dropping records for long-term contracts.*

EQR_TRANSACTIONS_200803 & EQR_TRANSACTIONS_200806: *EQR data is filed quarterly, and the report periods are designated as 03 (first quarter), 06 (second quarter), 09 (third quarter), and 12 (fourth quarter).*

Because I wanted to look only at New England transactions, I subsetted the transaction dataset on control area (the contracts table does not have a field for control area). I also dropped some of the larger columns from the transaction dataset. These were not needed for my project, and removing them helped reduce processing time. The two tables eqr_db.eqr_transaction_200803 and eqr_db.eqr_transaction_200806 have a combined total of 15,462,272 records. Selecting records for only New England energy transactions reduces this to 2,292,936.

```
DATA permserv.eqr_transactions_2008 (DROP=tr_other tr_revised sys_key);
        SET     eqr_db.eqr_transaction_200806
                eqr_db.eqr_transaction_200803 ;

                IF tr_delv_ctrl_area='ISNE'  and tr_prod_name='ENERGY';
        RUN;
```

Once the SAS datasets are created, the next step is to combine the energy and contract datasets and start cleaning up the data. The step shown below merges the energy and contract datasets, limiting the resulting dataset to observations for transactions that occurred in 2008. Contract_id is a key variable; however, in the transaction table it is called "tr_contract_id." To use this variable in a data step merge, it must be renamed, as shown below.

```
DATA permserv.eqr_transact_contract_2008
        (where=(report_yr=2008));

        MERGE permserv.eqr_transactions_2008 (rename=(tr_contract_id=contract_id) in=t)
                permserv.eqr_contracts  (drop=c_actual_termination c_other c_revised in=c);

        BY contract_id respondent_id report_yr report_prd;
```

```
        IF t and c;
RUN;
```

Convert transactions to consistent units

Companies can enter energy transactions in a variety of units, including $/MWh, $/KWh, cents/KWh, and flat rate. I converted $/KWh and cents/KWh to $/MWh.

Flat rate transactions show only a total dollar amount, so I excluded them: there is not enough information in these records to allow them to be converted to $/MWh. During the first two quarters of 2008, there were six records with units of "flat rate." These were all sales by New England Power Company to Constellation Power Source. The total dollar amount of these sales was about $20.5 million.

Determine contract length

There are a few columns that look like they will make determining the length of contracts easy. For example, the transactions table has a column called "tr_term_name." The data dictionary says this should have a value of LT for long term contracts with a duration of longer than one year, and ST for short term contracts for with a duration of less than one year. However, I have found this column to be unreliable. I believe FERC is working on correcting this problem.

There are columns called "c_commencement_date" and "c_termination_date" in the contract table. Unfortunately, the majority of records have junk in the "c_termination_date" column (many show a date of 30Dec1899). So I find the best way to determine the length of contracts is to use the contract commencement date with the transaction end date. Obviously, this will underestimate the length of contracts that extend into the future, but it serves to distinguish longer-term contracts from shorter-term contracts. And there are still some odd results, such as contracts with the buyer listed as ISO-NE that show up as multi-year.

Clean up company names

EQR data is provided by the companies that sold the energy: they are responsible for entering the buyer information. The buyer name column is free-form text, so if two different sellers have transactions with the same buyer, they can enter the buyer's information in two different ways. For example, in the first two quarters of 2008, there were entries for Bangor Hydro-Electric Company with the following four formats: BANGOR HYDRO ELECTRIC, BANGOR HYDRO ELECTRIC COMPANY, BANGOR HYDRO-ELECTRIC COMPANY, and BANGOR-HYDRO ELECTRIC COMPANY. Similarly, I have found more than 30 different labels for ISO-NE since 2006.

In these examples, I wanted to group all the records into "BANGOR HYDRO-ELECTRIC COMPANY" and "ISO-NE." But there are other examples where similarly named companies – usually affiliated companies – have different functions and should not be grouped together for the purposes of my project. For example, Dominion Retail, Inc. is an alternative retail supplier, while Dominion Energy Marketing, Inc. is a generation owner. While the companies are affiliated, I wanted to keep them separate. The way you decide to handle affiliated companies will depend on the type of analysis you are doing. However, the fact that companies with similar names are often separate entities makes using text-matching functions or routines impractical. I had to manually review the names and choose the appropriate mapping. Since there were only 373 unique names, this was not too time-consuming.

Buyer and seller names can be entered as mixed-case text, so the first step to standardizing them is to use the UPCASE function. Some names contain two or more consecutive blanks, which I removed using the COMPBL function. You might also choose to remove commas or periods using the COMPRESS function. Removing these characters would help in cases such as "CALPINE ENERGY SERVICES, L.P." (with periods) and "CALPINE ENERGY SERVICES, L P" (without periods) to reduce the number of unique names. However, the versions with commas and periods are usually the correct format, so I opted to leave these characters in.

I also replaced missing seller names with the respondent name – this is imperfect, as the respondent may be an agent acting for the seller, but I decided it was preferable to leaving blanks. I then put both buyer and seller names into one alphabetically sorted dataset.

The code shown below does the processing to convert transactions to consistent units, determine contract length, and clean up company names.

```
DATA permserv.eqr_energy_2008 (drop=c_seller_name c_buyer_name respondent_name)
        permserv.flat_rate_2008 (drop=c_seller_name c_buyer_name respondent_name)
        permserv.messy_2008 (drop=c_seller_name c_buyer_name respondent_name);
```

```
        MERGE permserv.eqr_transact_contract_2008 (in=a)
              permserv.eqr_ident_attsttn
              (keep=respondent_id report_yr report_prd respondent_name respondent_duns);

        BY respondent_id report_yr report_prd;

        IF a;

        /*      start to clean up buyer & seller names      */
        IF c_seller_name = ' ' then seller_name_2=compbl(upcase(respondent_name));
        ELSE seller_name_2=compbl(upcase(c_seller_name));

        buyer_name_2= compbl(upcase(c_buyer_name));


        /*convert KW to MW*/
        IF tr_units in ('CENTS/KWH', '$/KWH')then mw_quantity=tr_quantity/1000;
        ELSE mw_quantity=tr_quantity;

        IF tr_units = 'CENTS/KWH' then mw_price = tr_price * 10;
        ELSE IF tr_units = '$/KWH' then mw_price = tr_price * 1000;
        ELSE mw_price = tr_price;

        /*get number of days & years -- will use to determine if contract is long or short term*/
        number_of_days=intck('day',c_commencement_date, datepart(tr_end_date));
        number_of_years=number_of_days/365;
        format number_of_years comma6.2;


        /*remove flat rate records from main DATASET*/
        IF tr_units = 'FLAT RATE' then OUTPUT permserv.flat_rate_2008;
        ELSE IF mw_price < -10 or mw_price > 1000 then OUTPUT permserv.messy_2008;
        ELSE OUTPUT permserv.eqr_energy_2008;

RUN;


DATA temp_names2;
        SET permserv.eqr_energy_2008 (keep=buyer_name_2 rename=(buyer_name_2=name))
              permserv.eqr_energy_2008 (keep=seller_name_2 rename=(seller_name_2=name));
RUN;

PROC freq DATA=temp_names2 noprint;
        TABLES name  /OUT=eqr_name_list (keep=name);
RUN;
```
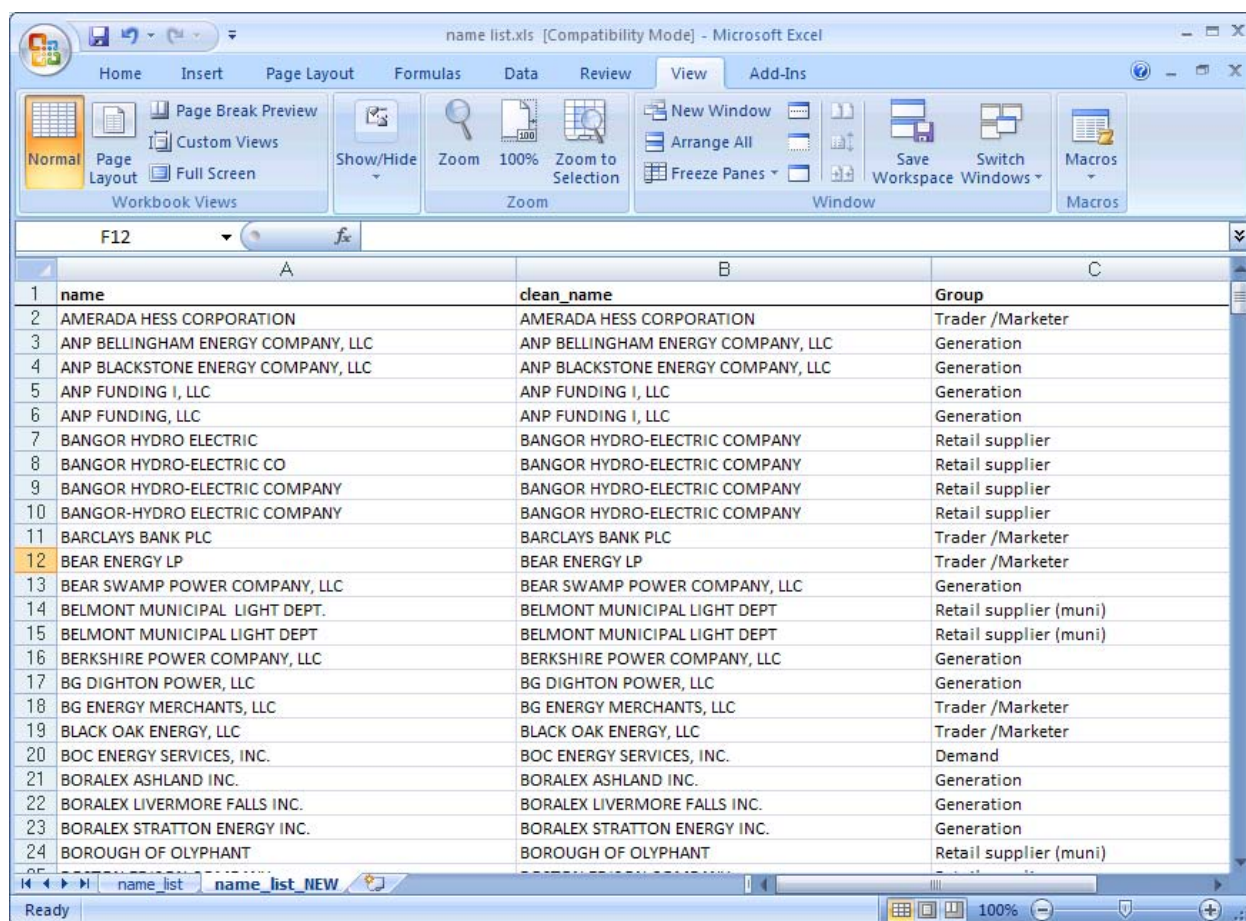
I opted to export the list of names from the dataset "temp_names2" into Excel and review them there. I then imported the results from Excel back into SAS.

Group companies

I classified companies according to their primary role in the New England markets, which may be different from their main business. The groups I used are: Retail supplier, Retail supplier (muni), Retail supplier (alternative), Retail supplier (out-of-state), Power authority, muni wholesale, etc., Generation, Trader /Marketer, Demand (*e.g.*, large manufacturers that buy power directly), ISO-NE, Other ISOs (Ontario, NYISO, PJM, MISO), and unknown.

In many cases, I knew what a company's business is. In other cases, I used Google to look up companies: I found that in general, the harder it is to find information on a company, the more likely it is to be a trader or speculator, rather than a retail supplier or generation owner. There were a few companies for which I was unable to determine the primary energy-related business: I classified these as unknown. Some of the categories I used were broad: for example, trader/marketers includes both companies that market energy management services to other companies and investment banks and hedge funds that trade in the energy market for profit. In retrospect, I should have included an "Energy Services Company" category, which would have been a better fit for some of these.

Regardless of whether you clean up the names in Excel or use some other tool, you will need to recombine the dataset with the clean names and company groups with your original dataset. I found that no sooner would I merge the name dataset back in than I would find an error or decide to make a change to a company grouping. I redid the merge several times. Because of this, finding a quick way to do the merge is useful.

I used hash tables: it was faster than using a sort and data step merge. The dataset with all the transactions is relatively large (2.3 million observations) and the dataset with the names is small (372 observations).

The traditional sort and data step merge code looks like this, and took a total of approximately 27 minutes (real time):

```
PROC SORT DATA=permserv.eqr_energy_2008;
        BY seller_name_2;
RUN;

DATA eqr_energy_2008_a (drop=seller_name_2);
        MERGE permserv.eqr_energy_2008 (in=a)
                name_list (rename=(name=seller_name_2 clean_name=seller_clean_name
group=seller_group));

        BY seller_name_2;
        IF a;
RUN;

PROC SORT DATA=eqr_energy_2008_a;
        BY buyer_name_2;
RUN;
```

5

```
DATA eqr_energy_2008_b (drop=buyer_name_2);
        MERGE eqr_energy_2008_a (in=a)
                name_list (rename=(name=buyer_name_2 clean_name=buyer_clean_name
group=buyer_group));

        BY buyer_name_2;
        IF a;

        LABEL seller_clean_name='seller_clean_name' buyer_clean_name='buyer_clean_name'
                seller_group='seller_group' buyer_group='buyer_group';
RUN;
```

The hash table code looks like this, and took a total of approximately 9 minutes (real time):

```
DATA name_list_temp
        (rename=(name=seller_name_2 clean_name=seller_clean_name group=seller_group));

        SET name_list;
RUN;

DATA eqr_energy_2008_a (drop = seller_name_2);
        if 0 then SET permserv.eqr_energy_2008
        name_list_temp;

        if _N_ = 1 then do;

                declare hash names(dataset:'name_list_temp');
                        names.defineKey('seller_name_2');
                        names.defineData('seller_clean_name', 'seller_group');
                        names.defineDone();
        end;

        SET permserv.eqr_energy_2008   ;

        if names.find(key:seller_name_2) = 0 then do;
                seller_clean_name = seller_clean_name;
                seller_group = seller_group;
        end;
        else do;
                seller_clean_name = ' ';
                seller_group = ' ';
        end;

        output;
RUN;

DATA name_list_temp2
        (rename=(name=buyer_name_2 clean_name=buyer_clean_name group=buyer_group));

        SET name_list;
RUN;

DATA eqr_energy_2008_b (drop = buyer_name_2);
        if 0 then SET eqr_energy_2008_a
        name_list_temp2;

        if _N_ = 1 then do;

                declare hash names(dataset:'name_list_temp2');
                        names.defineKey('buyer_name_2');
                        names.defineData('buyer_clean_name', 'buyer_group');
```

```
                              names.defineDone();
            end;

            SET eqr_energy_2008_a  ;

            if names.find(key:buyer_name_2) = 0 then do;
                    buyer_clean_name = buyer_clean_name;
                    buyer_group=buyer_group;
            end;
            else do;
                    buyer_clean_name = ' ';
                    buyer_group = ' ';
            end;

            output;
    RUN;
```

After merging the clean names back in with the other data, I created a final dataset called "eqr_energy_2008_clean" that I used for analysis. Here is the code used to create that dataset:

```
    PROC FORMAT ;
     VALUE yr_bkt low - 0.99999 = '<1'
                                1 - 2 = '1-2'
                                2.0001 - 5 = '2-5'
                                5.0001 - 10 = '5-10'
                                10.0001 - high = '10+'
                                ;
    RUN;

    DATA permserv.eqr_energy_2008_clean;
            SET eqr_energy_2008_b;

            LABEL seller_clean_name='seller_clean_name'
                    buyer_clean_name='buyer_clean_name'
                    seller_group='seller_group'
                    buyer_group='buyer_group'
                    tr_tot_transaction_chrg='tot_dollars' ;

            IF buyer_group='' THEN buyer_group='Other';
            IF seller_group='' THEN seller_group='Other';
            IF c_buy_sell_affil='' THEN c_buy_sell_affil='M';

            cost_per_mwh=tr_tot_transaction_chrg/mw_quantity;

            IF buyer_clean_name = 'ISO-NE' THEN number_of_years=0;
            year_bucket=number_of_years;
            FORMAT year_bucket yr_bkt.;
    RUN;
```

Results

EQR data showed transactions for 164,126 GWh in New England for January – June 2008. 71,706 GWh were sales to ISO-NE, while 92,420 GWh were sales to other parties.[3] The dollar value of these sales was around $13.9 billion[4], of which $6.4 billion was from sales to ISO-NE. The remaining $7.5 billion was from sales to other parties. The code used to calculate these numbers is below. It was faster to do this with PROC SQL than with a data step and proc summary:

---

[3] Actual New England system load for January – June 2008 totaled 64,657 GWH. Because of the way sales in the Day-Ahead and Real-Time ISO markets are reported, the figure for sales to the ISO in EQR data is higher.

[4] The abstract says $13.4 billion. This is incorrect.

```
PROC SQL;
        SELECT CASE buyer_clean_name
                            WHEN 'ISO-NE' THEN 'ISO-NE'
                            ELSE    'Other'
                        END AS iso_group,
                sum(tr_tot_transaction_chrg) AS tot_dollars format dollar15.,
                sum(mw_quantity/1000) AS tot_gw format comma12.

        FROM permserv.eqr_energy_2008_clean
        WHERE report_prd IN (3, 6)
        GROUP BY iso_group;


        SELECT sum(tr_tot_transaction_chrg) AS tot_dollars format dollar15.,
                sum(mw_quantity/1000) AS tot_gw format comma12.

        FROM permserv.eqr_energy_2008_clean
        WHERE report_prd IN (3, 6);
QUIT;
```

I was interested in finding out whether different types of buyers paid different prices for power. I calculated overall cost per MWh for various categories of buyers, using the code below.

```
PROC SUMMARY DATA=permserv.eqr_energy_2008_clean (where=(report_prd IN (3, 6))) nway;
        CLASS buyer_group ;
        OUTPUT OUT=eqr_buyer_group (drop=_:)
                    SUM(tr_tot_transaction_chrg mw_quantity)= ;
RUN;

DATA eqr_buyer_group_2;
        SET eqr_buyer_group;
        cost_per_mwh=tr_tot_transaction_chrg/mw_quantity;

        FORMAT cost_per_mwh dollar8.2 tr_tot_transaction_chrg dollar15. mw_quantity comma15.;
RUN;

PROC SORT DATA= eqr_buyer_group_2;
        by descending cost_per_mwh ;
RUN;

PROC PRINT DATA= eqr_buyer_group_2 noobs;
RUN;
```

buyer_group	tr_tot_ transaction_ chrg	mw_quantity	cost_ per_mwh
Retail supplier	$2,792,440,789	29,738,314	$93.90
ISO-NE	$6,419,072,603	71,706,024	$89.52
Generation /Marketer	$1,586,767,152	19,914,759	$79.68
Municipal	$127,603,879	1,616,850	$78.92
Other	$1,824,000	24,000	$76.00
Generation	$1,924,703,118	25,832,149	$74.51
Trader /Marketer	$595,582,161	8,024,053	$74.22
Power authority, muni wholesale, etc.	$117,448,028	1,615,562	$72.70
Demand	$13,142,480	215,530	$60.98
Retail supplier (alternative)	$322,443,242	5,366,724	$60.08
Other ISOs (Ontario, NYISO, PJM, MISO)	$4,056,493	72,371	$56.05

The total volume of power purchased by retail suppliers, alternative retail suppliers, and municipals during the first two quarters of 2008 was 36,721 GWh. This is 57% of New England load.

Notice that traditional retail suppliers are paying more than any other group: $4.38 higher than prices in the ISO-NE spot market, and $14.22 higher than the next highest buyer group. There are a variety of things that may contribute to this disparity: since contract terms are not known to us, we can only speculate. Retail suppliers may be buying contracts for all requirements while other purchasers are trading just the commodity. Retail suppliers may be buying longer term contracts, with a higher risk premium for the seller due to the term. It may also be that since they are able to pass the cost of procuring power on to their load, they have less incentive to negotiate lower contract prices.

There are differences in average price depending on whether the buyer and seller are affiliated companies or not. Adding the variable "c_buy_sell_affil" to the proc summary step shown above will generate results at the buyer type/affiliated level.

It is also interesting to look at MWh volume by contract length. I calculated this using the code below, which excludes sales to ISO-NE.

```
        PROC SUMMARY DATA=permserv.eqr_energy_2008_clean
                (where=(buyer_clean_name ne 'ISO-NE' and report_prd IN (3, 6))) nway;
                CLASS year_bucket;
                OUTPUT OUT=year_mwh_sum (drop=_:)
                SUM(mw_quantity)=;
                FORMAT mw_quantity comma12.;
        RUN;

        PROC PRINT DATA= year_mwh_sum noobs;
        RUN;
```

```
          year_
          bucket      mw_quantity

          <1          17,654,147
          1-2         19,292,589
          2-5         31,879,154
          5-10        22,858,925
          10+            735,498
```

Earlier in this paper, I noted that I find that the best way to determine the length of contracts is to use the contract commencement date with the transaction end date. This means that the number of years shown in the proc print results above is essentially the number of years from the contract start to the date of power delivery, not the actual length of the contract, which may extend beyond the date of power delivery.

We can also look at which companies are the most active sellers for power in the bilateral market using the code below. A screenshot of the first page of the resulting report follows the code.

```
        PROC SUMMARY DATA=permserv.eqr_energy_2008_clean (where=(report_prd IN (3, 6))) nway;
                CLASS seller_clean_name;
                OUTPUT OUT=seller_totals (drop=_:)
                        SUM(tr_tot_transaction_chrg mw_quantity)= ;
                        format tr_tot_transaction_chrg dollar20. mw_quantity comma20.;
        RUN;

        proc sort data=seller_totals;
                by descending mw_quantity;
        run;

        ODS LISTING CLOSE;
        OPTIONS nodate nonumber papersize="letter" orientation=portrait;
        ODS PDF file="C:\Documents and Settings\swhittier\My Documents\SUGI Paper\EQR Sellers.pdf"
        notoc ;
```

```
title1 justify=center font=Arial height=4 "EQR Sellers";

PROC REPORT DATA = seller_totals nowd style(REPORT) = {just=center};
column seller_clean_name tr_tot_transaction_chrg mw_quantity;
define seller_clean_name  / display 'Seller Name';
define tr_tot_transaction_chrg     / analysis  "Total Dollars" format = dollar20. ;
define mw_quantity     / analysis  "Total MW" format = comma12. ;
run;

ODS PDF CLOSE;
ODS LISTING;
```

Although I left data at the individual company level when looking at sellers in this example, it can also be useful to combine all data for affiliated companies.

Seller Name	Total Dollars	Total MW
CONSTELLATION ENERGY COMMODITIES GROUP	$1,236,783,843	17,000,934
DOMINION ENERGY MARKETING, INC.	$1,330,986,287	14,986,082
FPL ENERGY POWER MARKETING, INC	$1,081,562,217	12,107,989
SEMPRA ENERGY TRADING CORP	$643,154,100	7,702,275
AMERADA HESS CORPORATION	$624,791,424	6,280,486
H.Q. ENERGY SERVICES (US) INC.	$511,529,204	5,908,054
BOSTON GENERATING, LLC	$515,652,050	5,641,185
ENTERGY NUCLEAR POWER MARKETING, LLC	$323,892,223	5,455,853
ENTERGY NUCLEAR VERMONT YANKEE LLC	$183,275,832	4,876,469
CONNECTICUT LIGHT AND POWER COMPANY	$423,037,036	4,811,744
MYSTIC DEVELOPMENT, LLC	$371,770,851	4,096,860
PUBLIC SERVICE COMPANY OF NEW HAMPSHIRE	$359,696,968	3,997,705
DOMINION ENERGY BRAYTON POINT, LLC	$328,312,715	3,877,796
FPL ENERGY SEABROOK, LLC	$283,385,364	3,400,915
ENTERGY NUCLEAR GENERATING COMPANY	$143,788,045	2,940,451
PSEG ENERGY RESOURCES & TRADE LLC	$256,250,842	2,772,038
CONSOLIDATED EDISON ENERGY, INC.	$244,671,953	2,649,665
CONECTIV ENERGY SUPPLY INC.	$250,454,782	2,414,006
ANP FUNDING I, LLC	$227,600,127	2,334,986
GRANITE RIDGE ENERGY, LLC	$166,921,238	2,051,487
BP ENERGY COMPANY	$178,108,309	2,050,466

Conclusions

Working with EQR data can be challenging, but the information that can be gained makes the effort worthwhile. I found that spending time to prepare the data and finding efficient ways to do steps that would be repeated was important. The results included in this paper are just a few examples of the analysis you can do using EQR data.

References

- Deng, Shi-Jie, et al. *Electricity derivatives and risk management.* February 2005. Available online at http://www.pserc.wisc.edu/ecow/get/publicatio/2005public/deng_electricity_derivative.pdf (accessed February 3, 2009).

- Hausman, Ezra, et al. *Bilateral Contracting in Deregulated Electricity Markets, a Report to the American Public Power Association.* Synapse Energy Economics, Inc., April 18, 2008. Available online at http://www.appanet.org/files/PDFs/EMRISynapseBilateralsReport2008.pdf (accessed February 3, 2009).

- Federal Energy Regulatory Commission. *Draft Report to Congress on Competition in the Wholesale and Retail Markets for Electric Energy.* Available online at http://www.ferc.gov/legal/staff-reports.asp?new=sc2 (accessed October 8, 2008).

- Warner-Freeman, Jennifer. *I cut my processing time by 90% using hash tables – You can do it too!* NESUG 2007 Proceedings.

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Sarah Whittier
ISO New England
One Sullivan Road
Holyoke, MA 01040
(413) 540-4282
swhittier@iso-ne.com
http://www.iso-ne.com/