

# An Exploratory Analysis of the German EEX Power Market

*Jacob S Townson*

*May 20, 2018*

## Introduction

This paper covers my observations and analysis of the data given to me by Josef Spalenka of Genscape in order to determine my qualifications for a position at Genscape. The data used in this document is used in the German day-ahead power market called “EEX”. It contains the power prices, fuel prices, emission prices, heating degrees, and wind and solar forecasts for the next day.

My goals with this data are to first and foremost, understand what exactly is this data presenting us with. Then, I will make some code to read in this data to make it usable in R. Then, I will look to see if there are any interesting correlations and relationships to be found in the data, as well as make a model to predict future outcomes. I will present my findings here and make conclusions based off of them.

## The Data

The data given to me was split into 7 files with some of the excel files containing more sheets than others. My goal here is to explain precisely what this data is telling us, then to explore what possible correlations and relationships we may be able to find.

### Coal Future Data

In this section we will be exploring the file “coal\_futures\_historie\_2013.xls”. This excel file has 8 sheets. The first tells us most of what we need to know: coal seems to be phasing out from the German market (if it already hasn’t phased out now that we are well past the timeframe, 2013, of the data). The second sheet indicates similarly with its page full of zeros.

The rest of the sheets in this excel file show the 2013 data for specific contracts and their trade unit along with prices. These bits of data make me curious, as the previously mentioned sheets seem to indicate that there were no trades through the year, however these sheets show the monthly, quarterly, and yearly trade units with their settlement prices. I can only assume this means that these prices are the prices that the trade units cost, however no contracts for these units were made.

Since we can see that most of this data is just zeros, we will ignore it from here. However, I would like to understand more of what exactly these data sets are saying, especially the differences between the “Coal-Futures Total” sheets and the contract sheets afterwards. If offered this position, I would be interested in learning more about this.

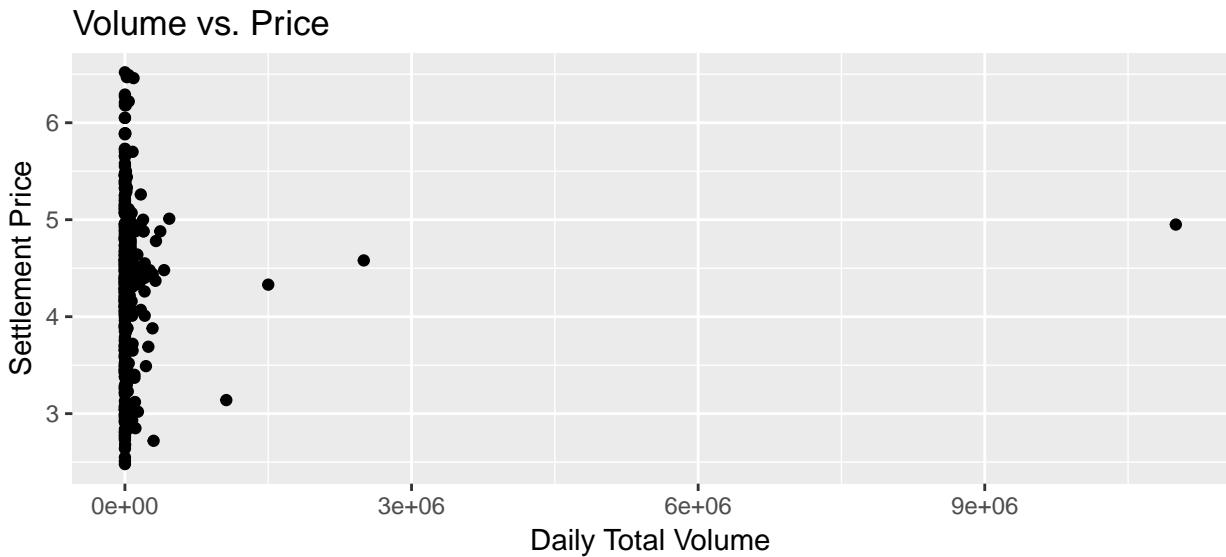
### Emission History

This data set (in the file “emission\_spot\_historie\_2013.xls”) gives the emission prices based on specific contracts and their emission volumes. This excel file has 3 sheets, first explaining the contracts’ emission allowance, the next showing reductions, and the final showing the market auction information.

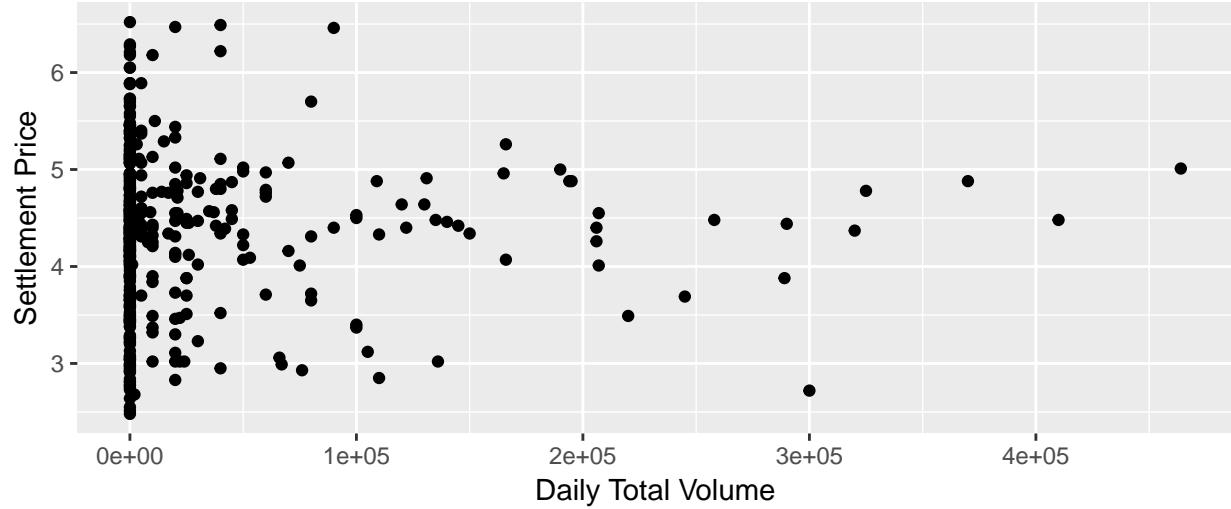
For the most part, this data set is relatively clear in what it shows, however in the market auction sheet, we are given a column labeled “auction details”. For my purposes, this information will be ignored as without more former knowledge of the system, I will not be able to accurately try to analyze anything about it.

For curiosity's sake, let's quickly look at the emission allowance data, and compare the settlement price with the total volume of emissions. We will use the “emission\_allowance” dataframe I created in R to do this efficiently.

```
em_price = emission_allowance$Settlement.Price.EUR.EUA
em_vol = emission_allowance$Daily.Total.Volume..incl..OTC.
em_cont = emission_allowance$Contract
em_trades = emission_allowance$Trades
qplot(em_vol,em_price, xlab = 'Daily Total Volume', ylab = 'Settlement Price',
      main = 'Volume vs. Price')
```



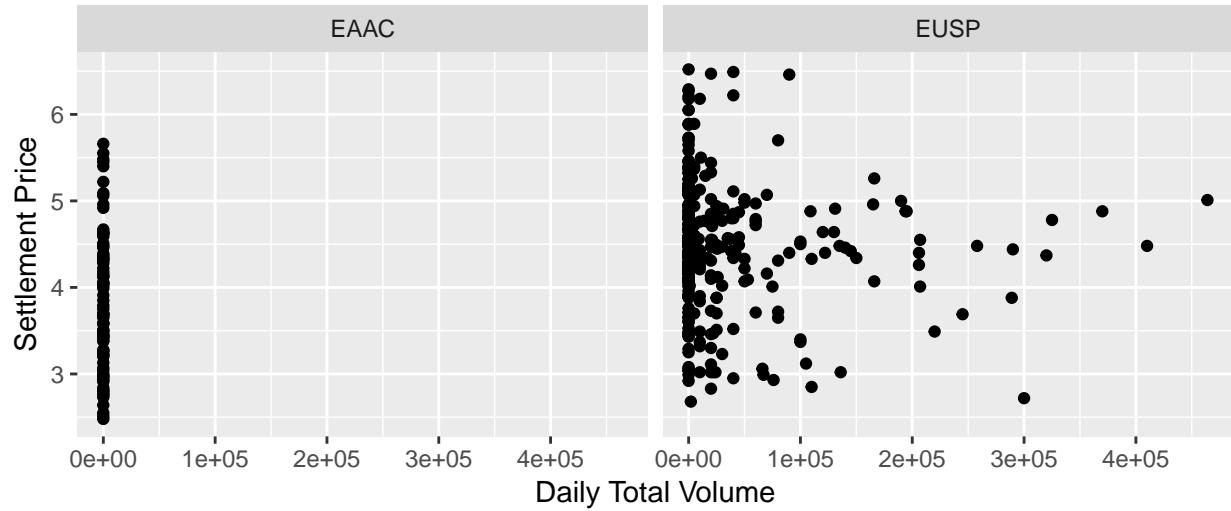
## Volume vs. Price



This plot still seems fairly noisy. Perhaps we should divide the information by contract to see if that has something to do with the noise in price. There are two types of contracts in this data set, EAAC, and EUSP.

```
qplot(em_vol,em_price, data = allow, xlab = 'Daily Total Volume', ylab = 'Settlement Price',
      main = 'Volume vs. Price', facets = .~em_cont)
```

## Volume vs. Price



This seems to show just more noise, no correlation seems to show for this information. I think we can then conclude that the daily volume of emissions doesn't affect the settlement price.

As a quick aside, note that emission volume seems to stay mostly constant over time, excluding a few outliers. Although no plot was included to show this, it can be seen fairly easily through a quick glance at the data as well. So let's move on!

## Gas History

This data set (found in the gas\_spot\_historie\_2013.xls) is by far the most dense. With 10 sheets, it has a lot of information to offer. The first of these sheets gives us the reference price for each market area for each day.

The strange thing about this collection of data is all of the missing values. I have to wonder if this is possibly because there were no trades on these given days for these market areas, or if whoever was collecting this data was unable to get it. Either way, for time's purpose, we will simply ignore the empty values for now.

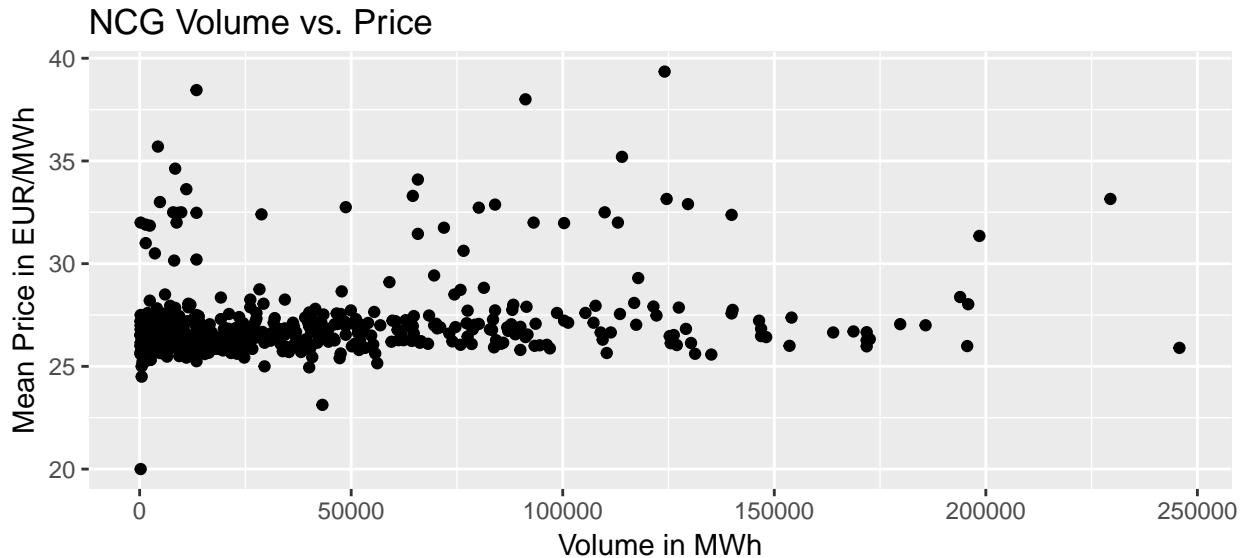
The following sheets give us data on what I have understood to be 3 European gas companies/traders, NCG (NetConnect Germany), GASPOOL, and TTF (Title Transfer Facility). For each one, we get the market information for the whole 2013 year. This has their number of trades, price information, and volume traded for every day.

Let's look and see if we can find any trends in this data. To keep things simple, let's start by looking at the NCG data, and if we find some interesting things, we'll dive down the rabbit hole. Otherwise, we'll move on. First things first, let's write a bit of code to simplify the low vs. high price problem in the data.

```
ncg.high = gas_market.ncg$High.Price.EUR.MWh
ncg.low = gas_market.ncg$Low.Price.EUR.MWh
ncg.mean_price = rowMeans(cbind(ncg.low,ncg.high))
```

Now that we have this midpoint price for each day, let's see if the volume vs. price plot shows us anything interesting.

```
ncg.trades = gas_market.ncg$Trades
ncg.vol = gas_market.ncg$Volume.MWh
ncg.day = gas_market.ncg$Trading.Date
NCG = data.frame(ncg.day,ncg.mean_price,ncg.vol,ncg.trades)
qplot(ncg.vol,ncg.mean_price, data = NCG, xlab = 'Volume in MWh', ylab = 'Mean Price in EUR/MWh',
      main = 'NCG Volume vs. Price')
```

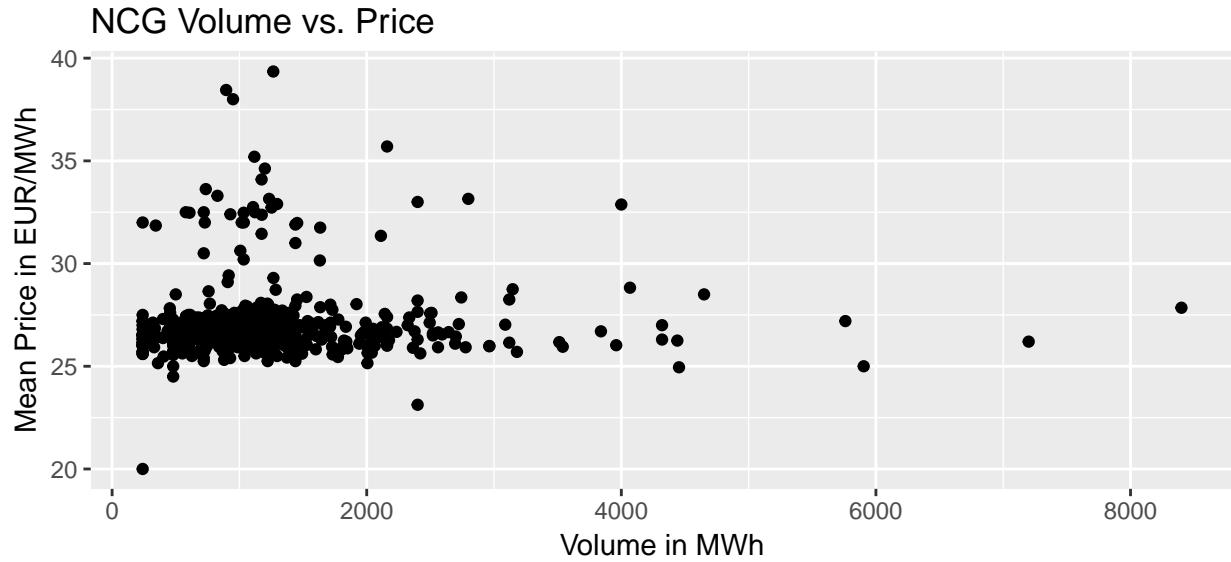


We seem to get a strange-looking plot here. It almost looks as if no matter the volume, the price stays between 25 and 30 euros. This makes me wonder if the number of trades has something to do with it. Let's take a look.

```
NCG = filter(NCG, ncg.trades > 0)
NCG = mutate(NCG, vol_trade = ncg.vol/ncg.trades)
```

What I have done in this code chunk is to first remove any zero trade values in the data, then make a new variable to show roughly the amount of volume per trade.

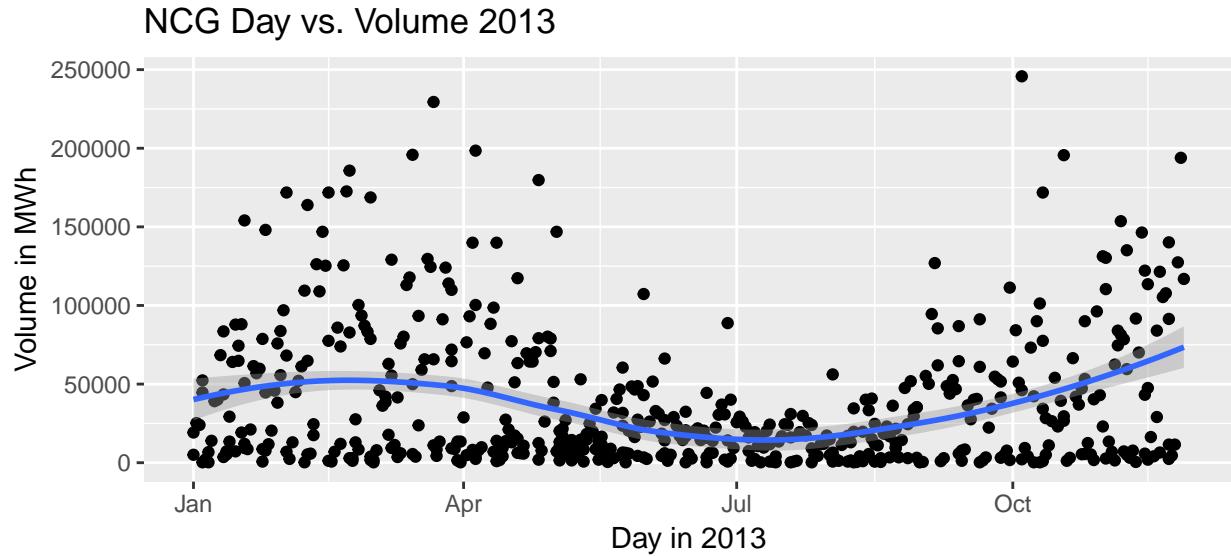
```
qplot(vol_trade,ncg.mean_price, data = NCG, xlab = 'Volume in MWh', ylab = 'Mean Price in EUR/MWh',
      main = 'NCG Volume vs. Price')
```



This didn't really clear much up for us. So seemingly the price in euros per MWh is not affected by the volume. Before we move on, let's take a quick look and see how it changes throughout the year though.

```
qplot(ncg.day, ncg.vol, data = NCG, xlab = 'Day in 2013', ylab = 'Volume in MWh',
      main = 'NCG Day vs. Volume 2013', geom = c("point", "smooth"))
```

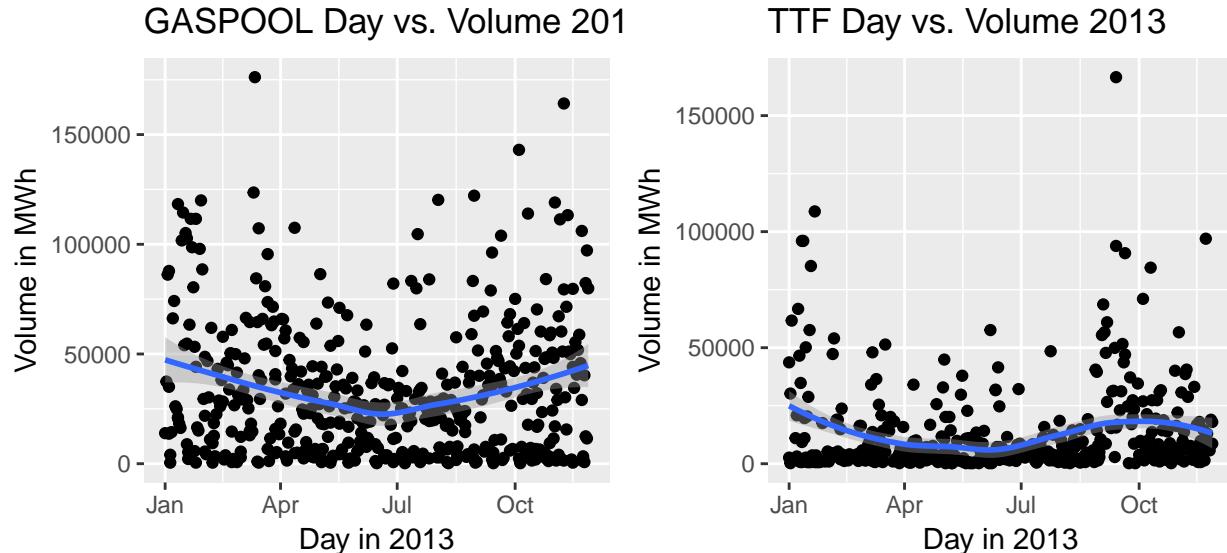
```
## `geom_smooth()` using method = 'loess'
```



Here we see an interesting bit of information. There was a surge of volume traded right before April and again going into December. This could mean a number of things, from people needing to heat their houses in the winter months, to people driving and using fuel to travel during spring. Without more data, we can't be sure, but we can make some interesting assumptions.

If we take this work and apply it to the GASPOOL and TTF data, and look at the time vs. volume plot, this is what we find:

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```



It's strange, the GASPOOL data doesn't seem to tell us much when it comes to the time of year other than there is a dip in average volume leading into July and rises thereafter. The TTF data shows us a little more, however it strangely looks almost like the exact opposite of the NCG data. Why this is, I can't say from this data alone, but it is interesting to note nonetheless.

### Heating Degrees

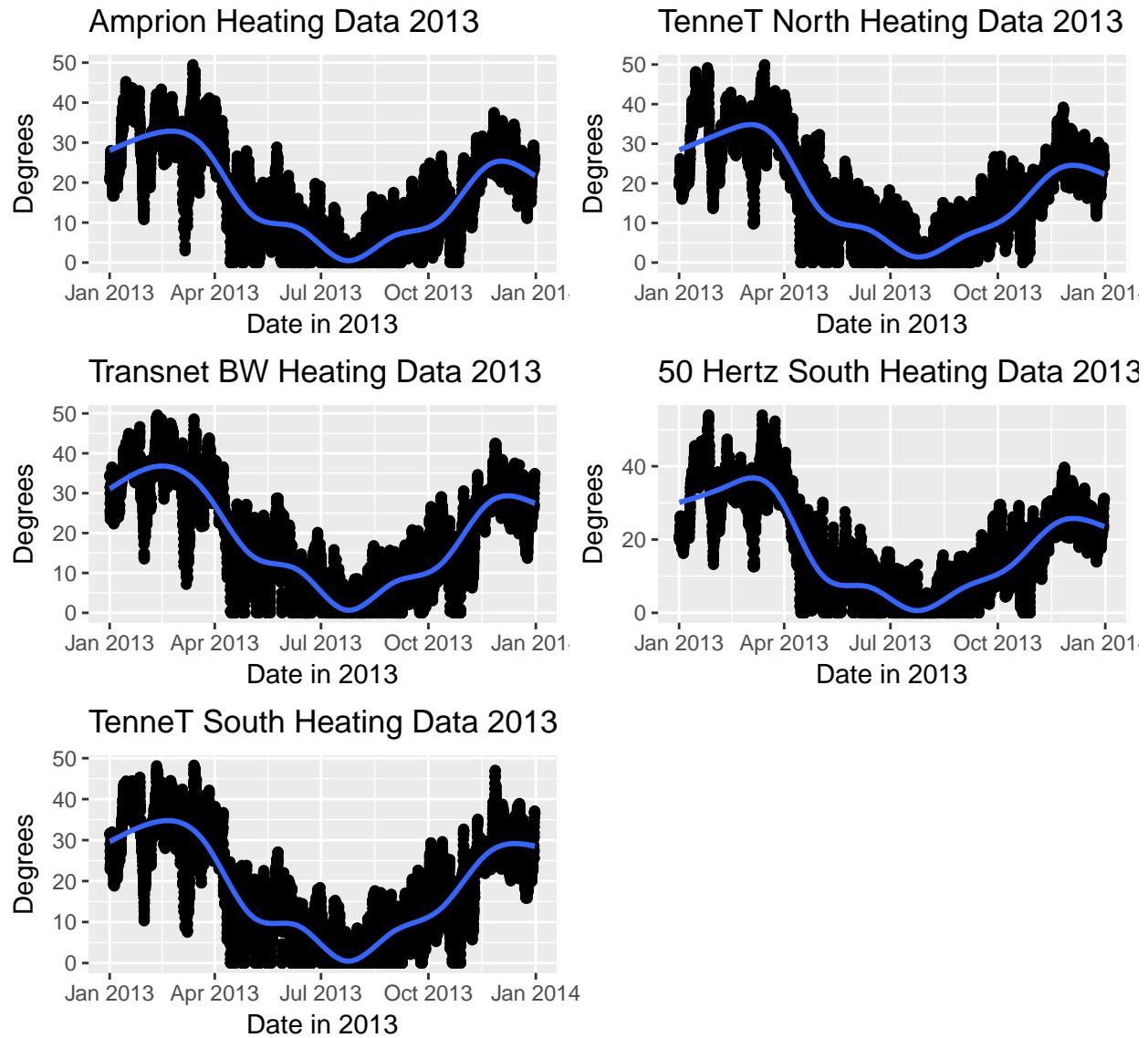
This data set (found in HeatingDegrees.xlsx) contains data for heating energy. The first sheet covers from 12/30/2009 - 07/31/2013, and the second sheet covers from 08/01/2013 - 01/06/2014. After a quick Google search, we can understand that Amprion, Transnet BW, TenneT South, TenneT North, and 50 Hertz are all companies. Based off of the title of the data set, I believe we can assume that the values given for each then are degrees heated on that given day and time.

Since our focus is on the year 2013, let's shrink this data down a bit to get it usable for our purposes. Then we'll take a look at how much these companies heat things up.

```
colnames(heating2) = c('Date..GMT.', 'Amprion.EDDL', 'Transnet.BW', 'TenneT.South', 'TenneT.North',
                      'X50.Hertz')
heating = bind_rows(heating1, heating2)
heating13 = filter(heating, Date..GMT. >= "2013-01-01" & Date..GMT. <= "2013-12-31")
```

And now we have a complete dataframe with all of the data from 2013! So let's check it out some plots and see how things look for each company.

```
## `geom_smooth()` using method = 'gam'
```

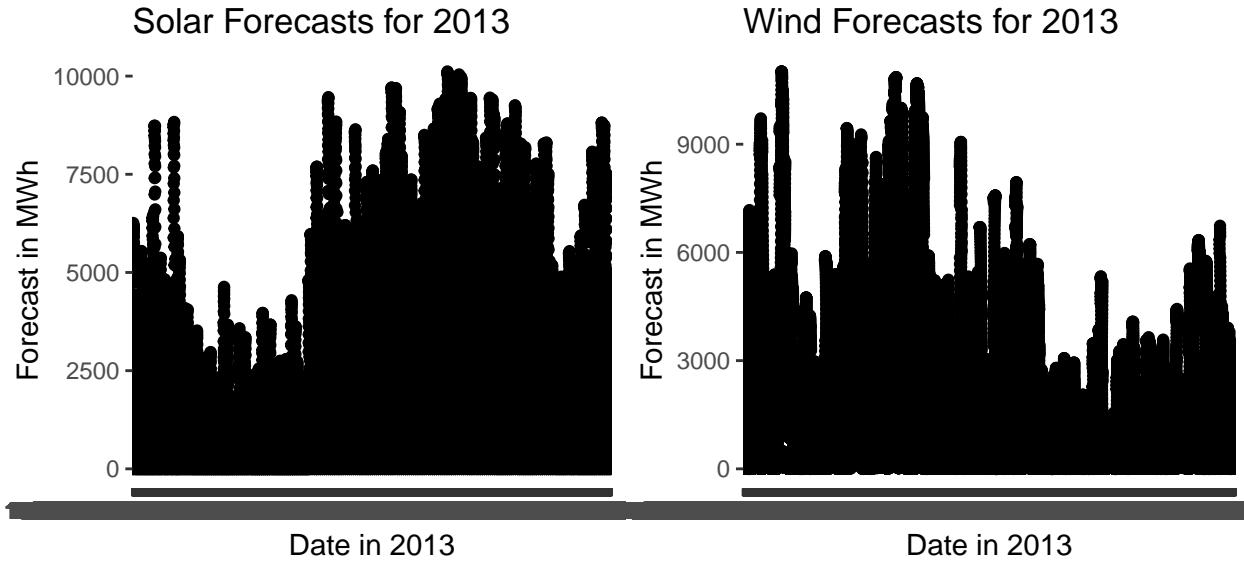


These plots are very interesting! So here, notice how similar all of these companies' heating data are! It's easier to see in the blue lines here how the growth is happening, and you can see how strikingly similar all of these are. From this we can assume that the time of year affects all of the companies the same temperature-wise. While this isn't all that surprising, it is interesting to see exactly how similar these plots are to each other.

### Wind and Solar Forecasts

These data sets are simple enough in explanation, we have dates and the forecast for how many megawatt hours can be produced on given days at given times. Now let's take a look at these to see what this data looks like exactly.

```
## `geom_smooth()` using method = 'loess'  
## `geom_smooth()` using method = 'loess'
```



These are both pretty noisy, so I'm not so sure we'll get too much out of these for a quick exploratory analysis like this. So let's move on for now.

### Power Prices

Finally we've made it to the real meat and potatoes of the data given to us. This data set has two sheets (contained in the file "energy\_spot\_historie\_2013.xls"). The first contains the delivery date, information on the euro per megawatt hour for each date, and even each hour of each day. The second sheet contains the strict volumes in megawatt hours (not euro per megawatt hours) for each hour of each day. There is a lot of information in this set of data, but by the end of this paper I hope to find a relatively simple model that we can use to help make predictions for data such as this.

We will touch more on the aforementioned model in the next section, but for now let's talk in more detail what's going on in this data. In sheet 1, the first thing we are given is the delivery date. This concept is simple enough, so we won't elaborate anymore on it. But next is the *Phelix Day Base* and *Peak* variables. Well, a quick Google search will tell us that the *Phelix Base* is the average price of all of the hours in a day for the electricity traded on the spot market. And the *Phelix Peak* is the average price of during the hours 9-20 for electricity traded on the spot market. One can assume it gets its name because these are the active hours of the day. This method of calculation is used for the the *Phelix* day and month values, just for days and months respectively. Another quick Google search will tell us that the meaning of *Off Peak* is the time when demand is at its lowest. So our off peak values are the lowest values given the times explained for each off peak; off peak 1 is during hours 1-8, and off peak 2 is during hours 21-24. Then the rest of the data explains the price for each hour in a given day.

Sheet 2 is almost identical to sheet 1, except for 2 columns of difference, those being the EPEX Spot Volume columns. These just seem to give the market totals for the traded energy.

At this point, we now understand the data to the best of our ability in a short exploratory analysis. But what if we could discover something deeper in the data? This would be a lot of fun, so let's give it a shot!

## Making a Model

I would like to see if using the data offered to us, if we can make an interesting model that may be helpful in predicting energy totals as the *Phelix Day Base* in MWh given the *Phelix Day Base* of prices and gas prices

for the given day. Let's see what we can find!

So the first thing I will do is gather our required variables into one nice organized dataframe. As I mentioned earlier, I already read all of the data into R to make it easy and usable for coding. Once we do this, we can make our model and see how it turns out.

```
temp_gas = gas_ref.price %>% group_by(Trading.Date) %>%
  summarise(avg = mean(Price.EUR.MWh,na.rm = TRUE))
colnames(temp_gas) = c('day','avg.gasprice')
temp_en = select(energy_volumes, Liefertag.Delivery.Date, Phelix.Day.Base.MWh)

model_data = select(energy_prices, Liefertag.Delivery.Date, Phelix.Day.Base.EUR.MWh)
model_data = left_join(model_data, temp_en, by = 'Liefertag.Delivery.Date')
colnames(model_data) = c('day', 'phelix.eur.p.mwh', 'phelix.mwh')
model_data = left_join(model_data, temp_gas, by = 'day')
kable(head(model_data))
```

day	phelix.eur.p.mwh	phelix.mwh	avg.gasprice
2013-12-31	25.98	640218.7	28.17200
2013-12-30	30.27	656609.9	27.02267
2013-12-29	27.77	610882.3	27.39333
2013-12-28	22.65	695486.6	27.18333
2013-12-27	23.09	745866.5	27.17033
2013-12-26	33.70	658471.2	27.47900

The dataframe made in the above code (with a few rows presented) lays out organized by day the Phelix day base price information, Phelix day base volume information, and the average referenced gas price. Using the handy power of dplyr, this code was relatively simple, but made a dataframe to make our desired model easy to create.

For the sake of simplicity, and since this paper is already fairly lengthy, I will only make a linear model to represent this for now. To do this, R will yet again be a great help!

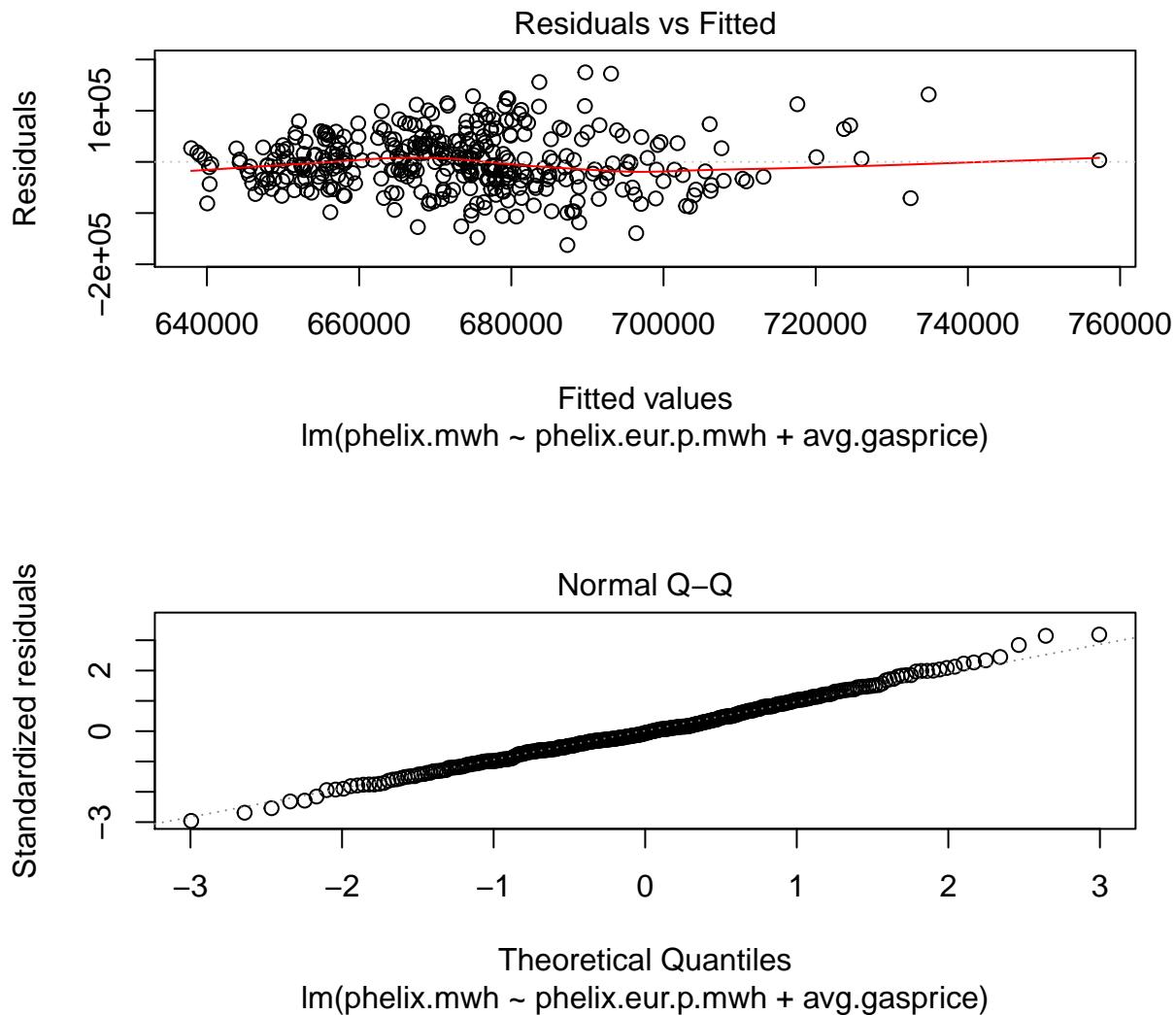
```
en.vol_mod = lm(phelix.mwh~phelix.eur.p.mwh+avg.gasprice, data = model_data)
summary(en.vol_mod)
```

```
##
## Call:
## lm(formula = phelix.mwh ~ phelix.eur.p.mwh + avg.gasprice, data = model_data)
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -162445   -34253   -2934   35957   174882
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 612495.4    42592.7 14.380 < 2e-16 ***
## phelix.eur.p.mwh -1466.4      252.6 -5.806 1.4e-08 ***
## avg.gasprice    4268.5     1565.5   2.727 0.00671 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55030 on 362 degrees of freedom
## Multiple R-squared:  0.09551,    Adjusted R-squared:  0.09051
## F-statistic: 19.11 on 2 and 362 DF,  p-value: 1.286e-08
```

First, we use the `lm()` function in R to create our linear model. Then we will finally present our summary. This shows us our linear model, with coefficients and other information laid out nicely. Notice that the *p*-values for all of the coefficients are very small. This is a very good sign, indicating that the likelihood of rejecting this model is very low in its current state.

The last things we will look at here are the residual plots. These will tell us some information on the accuracy of our model.

```
plot(en.vol_mod, which=1:2, labels.id = '')
```



The first plot seems to indicate that we have a relatively good correlation found here in our model. The Q-Q plot also looks very nice, showing us that we do indeed have a good correlation following a normal distribution with few outliers. All of this is very promising for our model!

## Conclusions

After a great deal of exploratory work, I feel as though I have gotten to know this data quite well. It is entirely possible that I may have misunderstood some of the more foreign energy concepts presented, but I did my best to make sense of them and put them into everyday terms. Through this exploration, I was able to find interesting trends that I'm sure some people may be interested to know or discover.

After finally getting to know this large assortment of data, I was able to make a nice model that seems to be fairly accurate, even though it may not take too many factors into account. A model such as this could be used for energy companies to get a better idea on how to charge for certain amounts of energy; or for companies being charged different energy prices, to give them a better understanding of why their costs are the amount that they received.

If I was to work on this model more, I would like to add more variables to it to find if there are any more attributing factors. I would also make more models like this for different bits of data, and to see how exactly they turned out. I am very excited to see that this model turned out as well as it did, but I also realize that because there isn't much to the model, it still may not be the most accurate that it could be for a real world setting. Given time constraints and the scope of this project though, I felt best to keep it relatively simple.

Finally, thank you, the reader, for taking the time to read my exploratory analysis of this data and information. I would love to do more formal projects like this for you in a job setting. Thank you!

## Appendix

### Code: Reading Data into R

```
# Reading the Data into R
require(xlsx)
require(dplyr)
require(lubridate)

# Note that we do not read in the coal_futures file as it seems to
# be incomplete at the time. There's not much to be done with a
# dataset full of zeros.

solar = read.csv("./Datafiles/Solar.csv", header = TRUE)

wind = read.csv("./Datafiles/Wind.csv", header = TRUE)

emission_allowance = read.xlsx("./Datafiles/emission_spot_historie_2013.xls",
                                sheetIndex=1, header=TRUE, startRow=2)

emission_reductions = read.xlsx("./Datafiles/emission_spot_historie_2013.xls",
                                 sheetIndex=2, header=TRUE, startRow=2)

emission_auction = read.xlsx("./Datafiles/emission_spot_historie_2013.xls",
                             sheetIndex=3, header=TRUE, startRow=2)
emission_auction$Auction.Time = format(emission_auction$Auction.Time,
                                         "%H:%M:%S")
```

```

energy_prices = read.xlsx("./Datafiles/energy_spot_historie_2013.xls",
                         sheetIndex=1, header=TRUE, startRow=3)

energy_volumes = read.xlsx("./Datafiles/energy_spot_historie_2013.xls",
                           sheetIndex=2, header=TRUE, startRow=2)

gas_ref.price = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                         sheetIndex=1, header=TRUE, startRow=2)

gas_market.ncg = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                           sheetIndex=2, header=TRUE, startRow=2)

gas_market.ncg.1mw = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                               sheetIndex=3, header=TRUE, startRow=2)

gas_market.ncg.day = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                               sheetIndex=4, header=TRUE, startRow=2)

gas_market.gaspool = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                               sheetIndex=5, header=TRUE, startRow=2)

gas_market.gaspool.1mw = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                                    sheetIndex=6, header=TRUE, startRow=2)

gas_market.gaspool.day = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                                    sheetIndex=7, header=TRUE, startRow=2)

gas_market.ttf = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                          sheetIndex=8, header=TRUE, startRow=2)

gas_market.ttf.1mw = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                               sheetIndex=9, header=TRUE, startRow=2)

gas_market.ttf.day = read.xlsx("./Datafiles/gas_spot_historie_2013.xls",
                               sheetIndex=10, header=TRUE, startRow=2)

heating1 = read.xlsx("./Datafiles/HeatingDegrees.xlsx",
                     sheetIndex=1, header=TRUE, startRow=NULL)

heating2 = read.xlsx("./Datafiles/HeatingDegrees.xlsx",
                     sheetIndex=2, header=TRUE, startRow=NULL)

```

## Code: Function; multiplot

```

# Multiple plot function

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

```

```

# Make a list from the ... arguments and plotlist
plots <- c(list(...), plotlist)

numPlots = length(plots)

# If layout is NULL, then use 'cols' to determine layout
if (is.null(layout)) {
  # Make the panel
  # ncol: Number of columns of plots
  # nrow: Number of rows needed, calculated from # of cols
  layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                  ncol = cols, nrow = ceiling(numPlots/cols))
}

if (numPlots==1) {
  print(plots[[1]])

} else {
  # Set up the page
  grid.newpage()
  pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout)))

  # Make each plot, in the correct location
  for (i in 1:numPlots) {
    # Get the i,j matrix positions of the regions that contain this subplot
    matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

    print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                    layout.pos.col = matchidx$col))
  }
}
}

```

## Code: Plots for Day vs. Vol. Gas

```

#NCG
ncg.vol = gas_market.ncg$Volume.MWh
ncg.day = gas_market.ncg$Trading.Date
NCG = data.frame(ncg.day,ncg.mean_price,ncg.vol,ncg.trades)
qplot(ncg.day, ncg.vol, data = NCG, xlab = 'Day in 2013',
      ylab = 'Volume in MWh', main = 'NCG Day vs. Volume 2013',
      geom = c("point", "smooth"))

#GASPOOL
gas.vol = gas_market.gaspool$Volume.MWh
gas.day = gas_market.gaspool$Trading.Date

qplot(gas.day,gas.vol, xlab = 'Day in 2013',
      ylab = 'Volume in MWh', main = 'GASPOOL Day vs. Volume 2013',
      geom = c("point", "smooth"))

```

```
#TTF
ttf.vol = gas_market.ttf$Volume.MWh
ttf.day = gas_market.ttf$Trading.Date

qplot(ttf.day, ttf.vol, xlab = 'Day in 2013',
      ylab = 'Volume in MWh', main = 'TTF Day vs. Volume 2013',
      geom = c("point", "smooth"))
```