Python I - Group Assignment

IE MBD – Group B

Alan Haro, Jesús Fuster, Joao Ferris, Marilyn González, Muriel Vergara, Sergio Ospina, Vignesh Viswanathan, Zara Amer

Assignment Objectives

#01 /Concepts

Review and understand the **pandas concepts**

#02 /Exercise

Practice our **pandas coding skills** through a **practical exercise**

#03 /Group Work

Strengthen **abilities to work in group** in a coding context

#04 /Presentation Skills

Reinforce our **coding presentation skills**

We are working with an energy dataset

Date & Time

Specifies year, month, day, and hour of each data point

Energy Production

Hourly data of energy production (nuclear, gas, solar, hydro, coal, and wind) in Spain

Energy Demand

Hourly data of energy consumption in Spain

Spot Price

Hourly data for energy spot price in Spain

Q1 – Assigning strings to keys

ASK

Convert the **weekday column** in from a **number** to a **string**

Logic of approach

1) **Create dictionary** assigning **day name** to **corresponding integer** type key

2) **Map the dictionary** to get the **weekday names**

Conclusions

0 -----> Monday
1 -----> Tuesday
2 -----> Wednesday
3 -----> Thursday
4 -----> Friday
5 -----> Saturday
6 -----> Sunday

In [1]

```
energy_n["weekday"] = energy_n["weekday"].map(lambda x: weekday_name[x])
```

Q2 – Assigning strings to keys (2)

ASK

Convert **month column** from a **number to a string**

Logic of approach

1) **Create dictionary** assigning **month name** to **corresponding integer** type key

2) **Map the dictionary** to get the month names

Conclusions

```
0 -----> January
1 -----> February
2 -----> March
3 -----> April
4 -----> May
5 -----> June
6 -----> July
7 -----> August
8 -----> September
9 -----> October
10 -----> November
11 -----> December
```

In [2]

```
energy_n["month"] = energy_n["month"].map(lambda x: month_name[x])
```

Q3 – % unique days with at least 1 hr with P < € 10

ASK

Find **percentage of unique days** in the total period in which at **least one hour with a price < €10**

Logic of approach

1) Extract **unique dates** that **meet condition**
2) Find **total unique days**, then **required %**
3) Calculate **% of days** where **stock price < €10**

Conclusions

1) **16 days** with at least 1 hr with **stock price < €10**
2) **% of unique days** that meet condition: **4.38%**

In [3]

```
print ('percentage of unique days was', unique_days / total_days * 100, '%')
```

Q4 – # hours/month with P (<) & (>) Monthly P Avg

ASK

How many **hours per month**, in average, do we have a **price above** the **monthly average**? And **below**?

Logic of approach

1) **Group by month** to compute **monthly avg price**

2) Find for **how many hours** in **each month** the **spot price** is **above (below) monthly price avg.**

Conclusions

1) **11 / 12** months: more hours with **P above than** with P below monthly avg

2) Month with:
   - highest **# hours above –Mar** (66%)
   - highest **# hours below – Sep** (50%)

In [4]

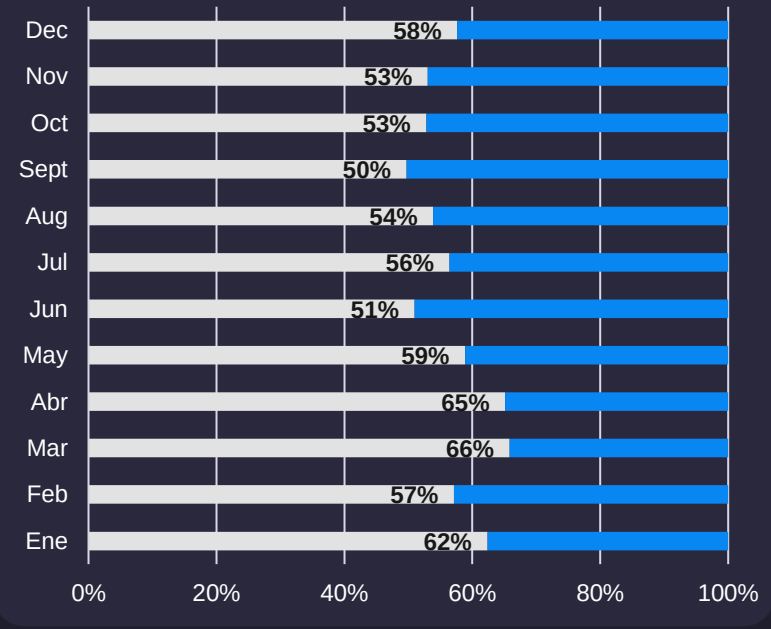
```
energy_final = pd.merge (energy_hours_above_avg,  energy_hours_below_avg,  left_on = 'month_year', right_on = 'month_year')
```

Q4 – # hours/month with P (<) & (>) Monthly P Avg

total hours above & below monthly price avg

month_year	count_hours_above_avg	count_hours_below_avg
2019-01	464	280
2019-02	384	288
2019-03	490	254
2019-04	469	251
2019-05	438	306
2019-06	367	353
2019-07	420	324
2019-08	401	343
2019-09	358	362
2019-10	393	351
2019-11	382	338
2019-12	415	305

% Hours above & below monthly price avg



Q5.1 – Gas generation vs Wind Generation

ASK

Is **gas generation higher or lower** that its **hr avg** when **wind generation > its hr avg.**

Logic of approach

1) Find **hr avg. of wind** and **gas production**
2) **Filter** using data points were **wind production > hr avg**
3) Find **data points: gas production higher / lower** than hr avg.

Conclusions

# Higher	748 hrs
# Lower	2864 hrs
% of higher gas generation	20.7 %

In [5]

```
energy_new.loc[:, 'H/L'] = np.where(energy_new.loc[:, 'gas'] > energy_new["gas_avg"], 'Higher', 'Lower')
```

Q5.2 – Spot Price vs Solar generation

ASK

Is the **spot_price higher or lower** than its **hr avg.** when **solar generation** is **above** its **hr avg?**

Logic of approach

1) Find **hr avg. of solar production and of spot_price**
2) **Filter** using data points were **solar production > hr avg**
3) Find **data points: spot_price higher / lower** than hr avg.

Conclusions

# Higher	1497 hrs
# Lower	1502 hrs
% of higher spot_price vs. hr mean	49.9 %

In [5]

```
energy_new.loc[:, 'H/L'] = np.where(energy_new.loc[:, 'spot_price'] > energy_new
["spot_price_avg"], 'Higher', 'Lower')
```

Q5.3 – Gas generation vs Wind Generation

ASK

Is the **spot_price higher or lower** than its **monthly avg.** when **power_demand** is **above** its **monthly avg?**

Logic of approach

1) Find **monthly avg. of: power and spot_price**
2) **Filter** using data points were **power demands > monthly avg**
3) Find **data points: spot_price higher / lower** than hr avg.

Conclusions

# Higher	**3521 hrs**
# Lower	**988 hrs**
% of higher spot_price vs. monthly mean	**78.1 %**

In [5]

```
energy_new.loc[:, 'H/L'] = np.where(energy_new.loc[:, 'spot_price'] > energy_new
["spot_price_avg"], 'Higher', 'Lower')
```

Q6.1 – Avg. Contribution of each energy type

ASK

Finding the **average contribution** of **each energy type** during the whole period?

Logic of approach

1) **Sum** the **mean of each energy type**

2) **Divide energy sums by whole energy mean**

Conclusions


27%


12%


25%


5%


4%


26%

In [6]

```
energy_mean_per = energy_sums/sum_energy_mean
```

Q6.2 – Avg. Contribution of each energy type in $ month

ASK

Finding the % **average contribution** of **each energy type** in most expensive month?

Logic of approach

1) We **index** the **month column** to **find** the **most expensive month**

2) Find **average usage** of **energy type** within **given month**

Conclusions

January 2019


25%


11%


16%


16%


2%


30%

In [6]

```
max_energy_sum =((energy_sum.groupby(["month_year"])["power_demand"].sum())).idxmax()
```

Q6.3 – Avg. Contribution of each energy type in $ month

ASK

Finding the % **average contribution** of **each energy type** in cheapest month?

Logic of approach

1) We **index** the **month column** to **find** the **cheapest month**

2) Find **average usage** of **energy type** within **given month**

Conclusions

December 2018


33%

18%

17%

10%

0%

21%

In [6]

```
min_energy_sum =((energy_sum.groupby(["month_year"])["power_demand"].sum())).idxmin()
```

Q8 – Weekend vs. Weekdays

ASK

How **much expensive** in average is a **weekend** day compared to a **weekday?**

Logic of approach

1) **Categorize** data as **weekday** or **weekend**
2) **Filter** and **sum means** of weekdays and weekends
3) **Find difference** between both means

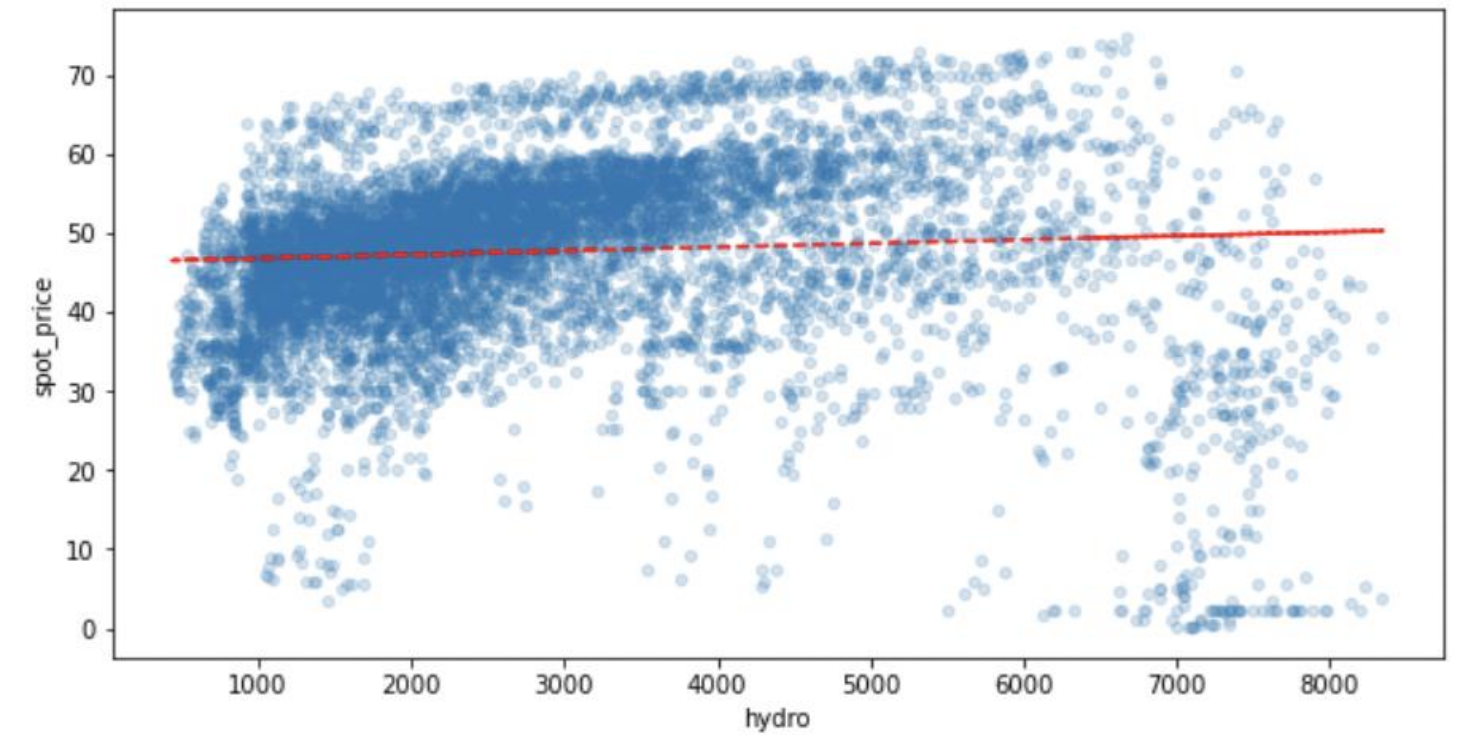
Conclusions

$ Weekdays	€ 49.17
$ Weekends	€ 44.04
Cost difference	€ -5.13

In [8]:

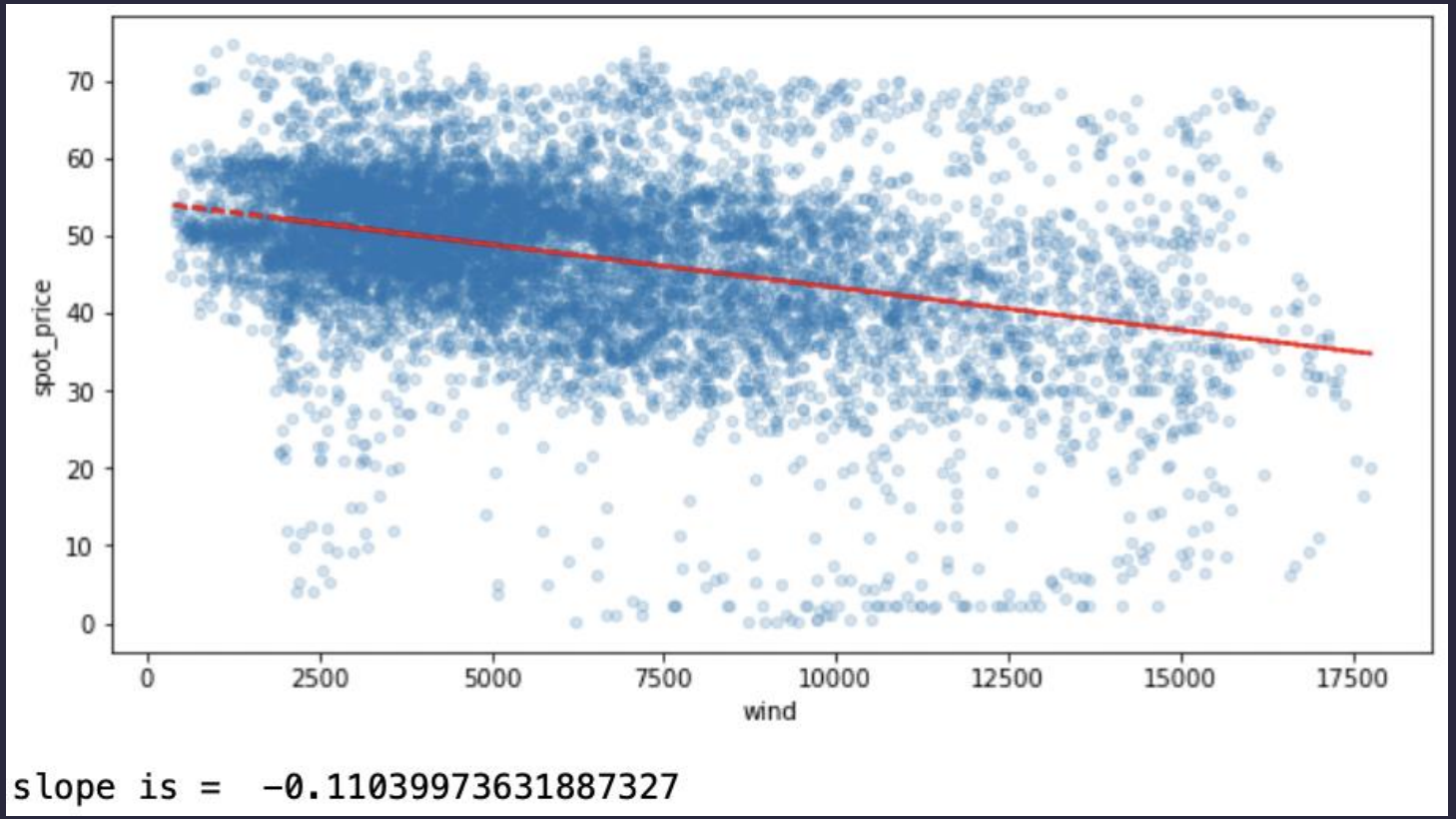
```
cost_difference = energy_weekend_filter_mean - energy_weekday_filter_mean
```

Q7.1 – Hydro Power vs Electricity Price



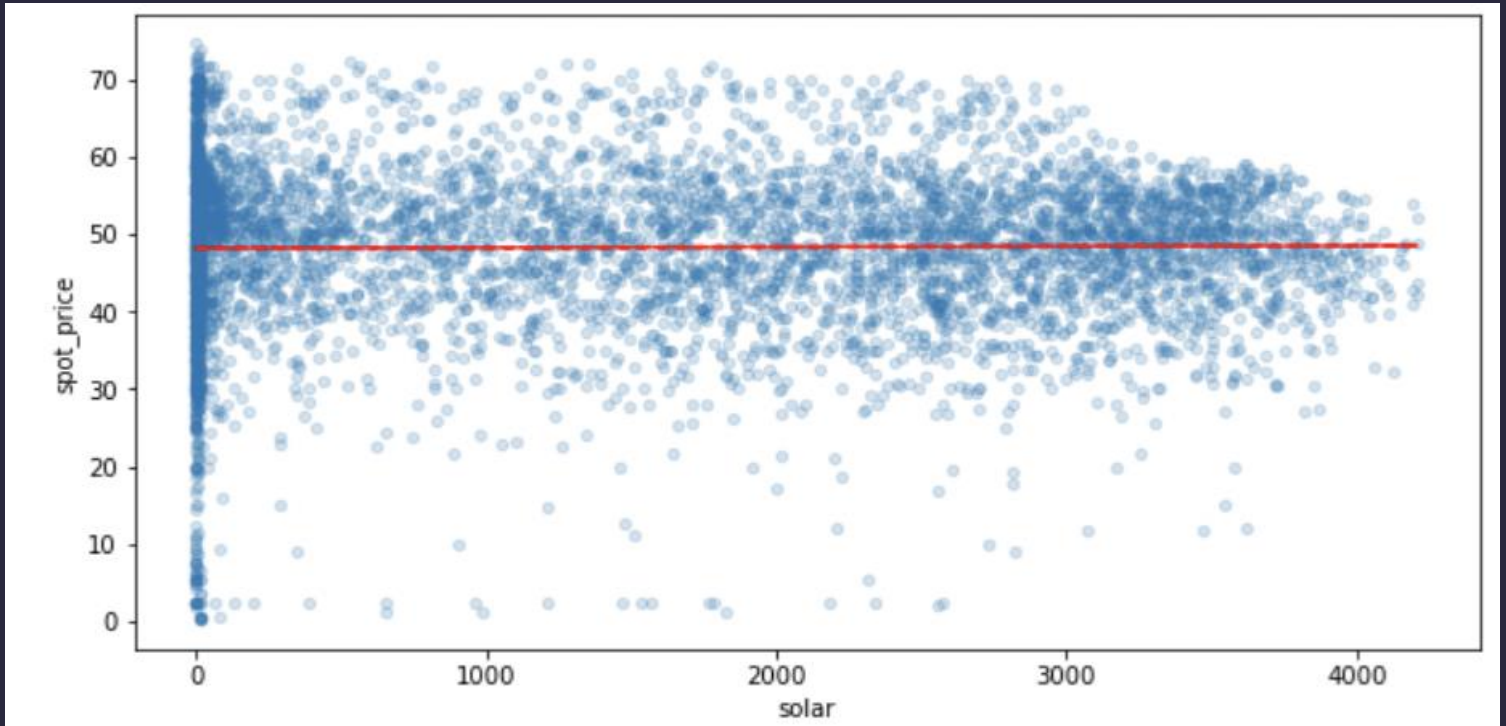

```
slope is =  0.047346492260768 58
```

Q7.2 – Wind Power vs Electricity Price



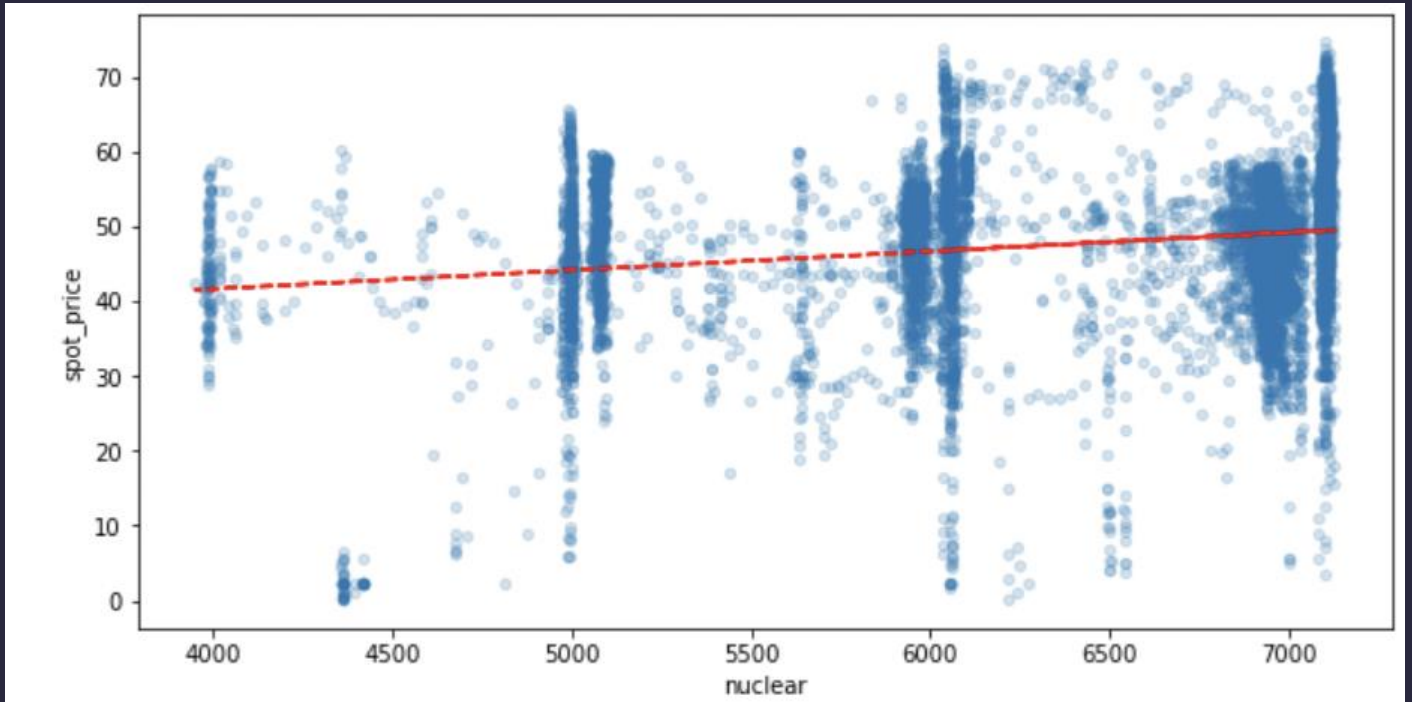

```
slope is =  -0.11039973631887327
```

Q7.3 – Solar Power vs Electricity Price



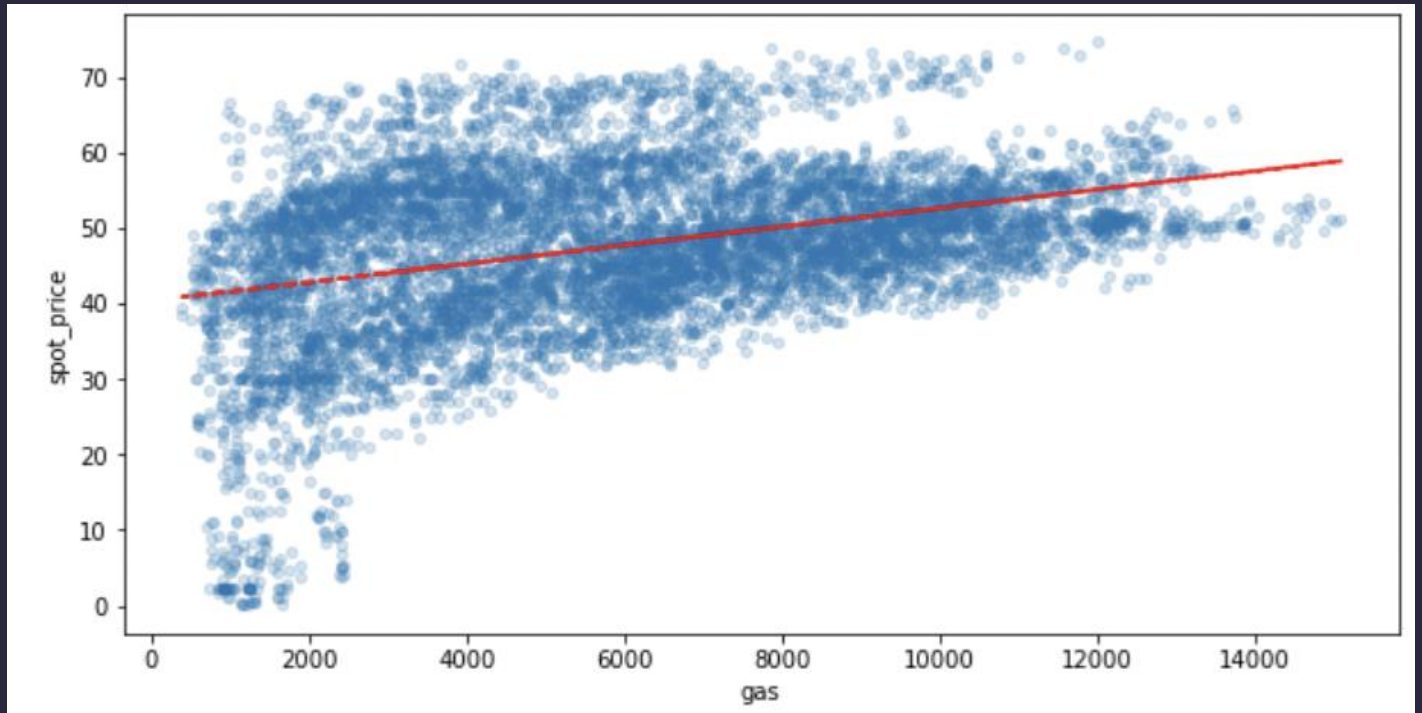

```
slope is =  0.0073946029674257 27
```

Q7.4 – Nuclear Power vs Electricity Price



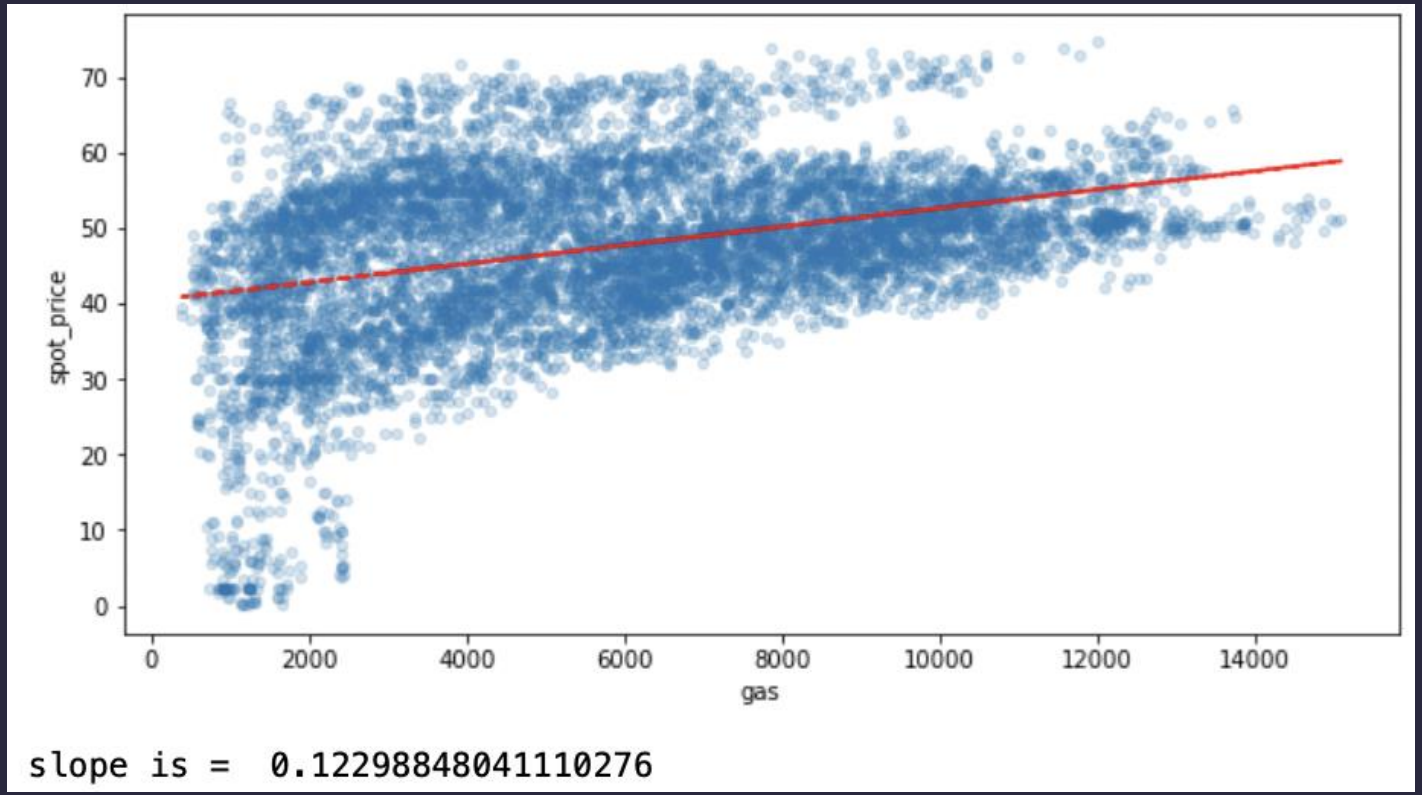

```
slope is =  0.25212678765476504
```

Q7.5 – Gas Power vs Electricity Price




```
slope is =  0.12298848041110276
```

Q7.6 – Coal Power vs Electricity Price



Print (“WHOA! What a great presentation from group B”)