Hourly Energy Consumption in The U.S.

By: Daniel Cwynar, Peter Stoermer, Sarahana Shrestha and Brian Buckley

Which Dataset Did We Choose?

- https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption?select=pjm_hourly_est.csv
- The attributes are energy companies around the U.S. and how much they consume hourly

```
df = pd.read_csv("pjm_hourly_est.csv")

df
```

✓ 0.3s

	Datetime	AEP	COMED	DAYTON	DEOK	DOM	DUQ	EKPC	FE	NI	PJME	PJMW	PJM_Load
0	1998-12-31 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	29309.0
1	1998-12-31 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28236.0
2	1998-12-31 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27692.0
3	1998-12-31 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27596.0
4	1998-12-31 05:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27888.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
178257	2018-01-01 20:00:00	21089.0	13858.0	2732.0	4426.0	18418.0	1962.0	2866.0	9378.0	NaN	44284.0	8401.0	NaN
178258	2018-01-01 21:00:00	20999.0	13758.0	2724.0	4419.0	18567.0	1940.0	2846.0	9255.0	NaN	43751.0	8373.0	NaN
178259	2018-01-01 22:00:00	20820.0	13627.0	2664.0	4355.0	18307.0	1891.0	2883.0	9044.0	NaN	42402.0	8238.0	NaN
178260	2018-01-01 23:00:00	20415.0	13336.0	2614.0	4224.0	17814.0	1820.0	2880.0	8676.0	NaN	40164.0	7958.0	NaN
178261	2018-01-02 00:00:00	19993.0	12816.0	2552.0	4100.0	17428.0	1721.0	2846.0	8393.0	NaN	38608.0	7691.0	NaN

178262 rows × 13 columns

Preprocessing:

```
df = pd.read_csv("pjm_hourly_est.csv")

df["Datetime"] = pd.to_datetime(df["Datetime"], format = "%Y-%m-%d %H:%M:%S")
```

```
df2["Season"] = 0

for i in range(0,len(df2)):
    if df2["Datetime"][i].month == 12 or df2["Datetime"][i].month == 1 or df2["Datetime"][i].month == 2:
        df2["Season"][i] = "Winter"
    elif df2["Datetime"][i].month == 3 or df2["Datetime"][i].month == 4 or df2["Datetime"][i].month == 5:
        df2["Season"][i] = "Spring"
    elif df2["Datetime"][i].month == 6 or df2["Datetime"][i].month == 7 or df2["Datetime"][i].month == 8:
        df2["Season"][i] = "Summer"
    else:
        df2["Season"][i] = "Fall"

df2
```

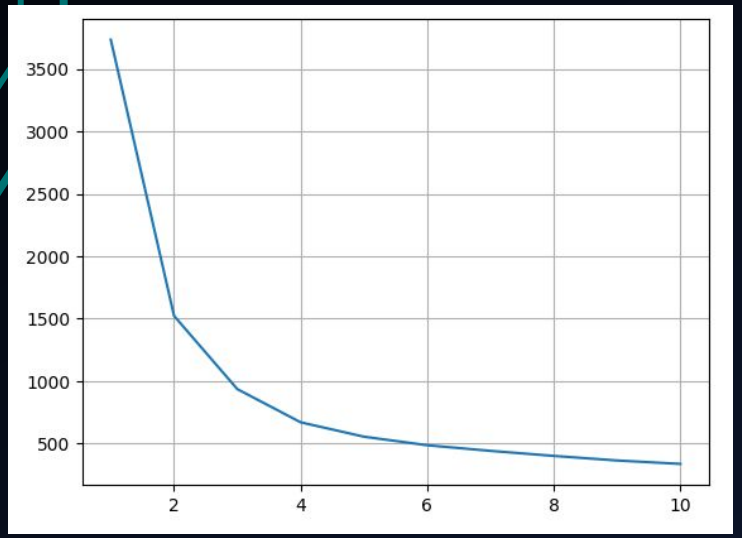
```
df2 = pd.get_dummies(df2, columns = ["Season"], prefix = "", prefix_sep= "")

df2
```

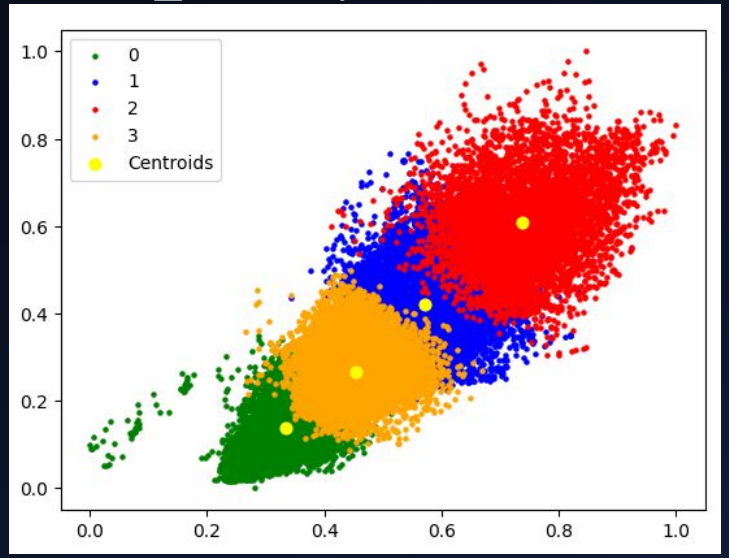
```
# df3 is for the second cluster
# includes companies aep dayton pjmw duq ekpc
df3 = df.copy()
df3 = df3[["Datetime","AEP","DAYTON","PJMW", "DUQ", "EKPC"]][(df3["AEP"].isnull() == False) &  (df["DAYTON"].isnull() == False) &
& (df["DUQ"].isnull() == False) & (df["EKPC"].isnull() == False)]
df3 = df3.reset_index(drop = True)

df3
```

Cluster #1

```
kmeans = KMeans(n_clusters = 4, init = "k-means++", max_iter = 300, n_init = 50, random_state =100)
# n_init is number of times the algorithim will run with centroid different seeds
# random state is random number for making centroid
```

- Separated data to have three main companies to remove most null values, chose (DEOK, DOM, DUQ) years from 2012-2018
- Standardized data (0-1), Used K-means
- Used elbow method to find best number of clusters in df
- Number of clusters: 4, n_init and random_state optimal

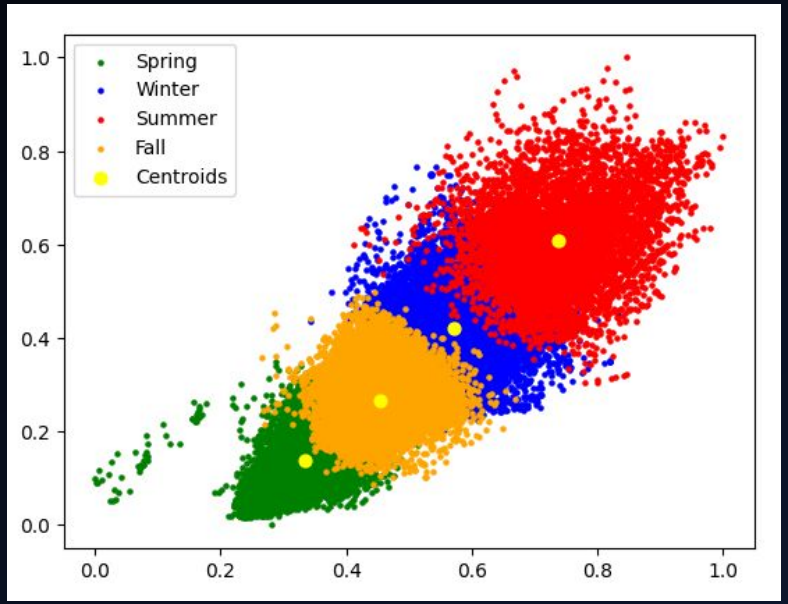


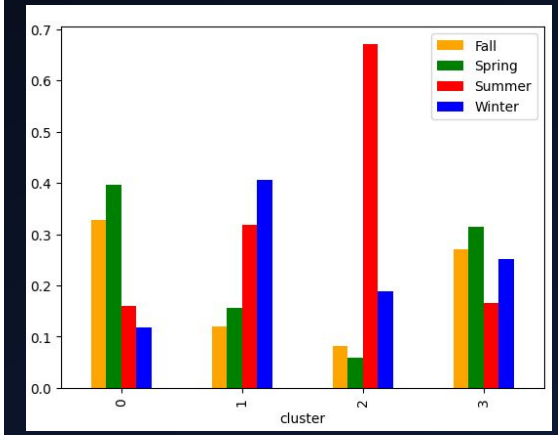


How to Get Meaning From Cluster #1

- A possibility is that it represents seasons

cluster	Fall	Spring	Summer	Winter
0	0.326964	0.396249	0.159165	0.117622
1	0.120504	0.155909	0.317660	0.405927
2	0.081975	0.058093	0.671131	0.188801
3	0.270028	0.313510	0.165482	0.250980

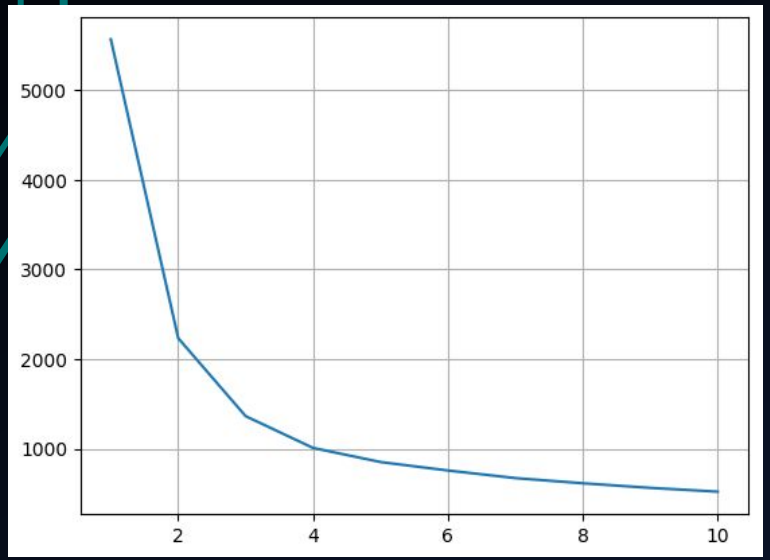


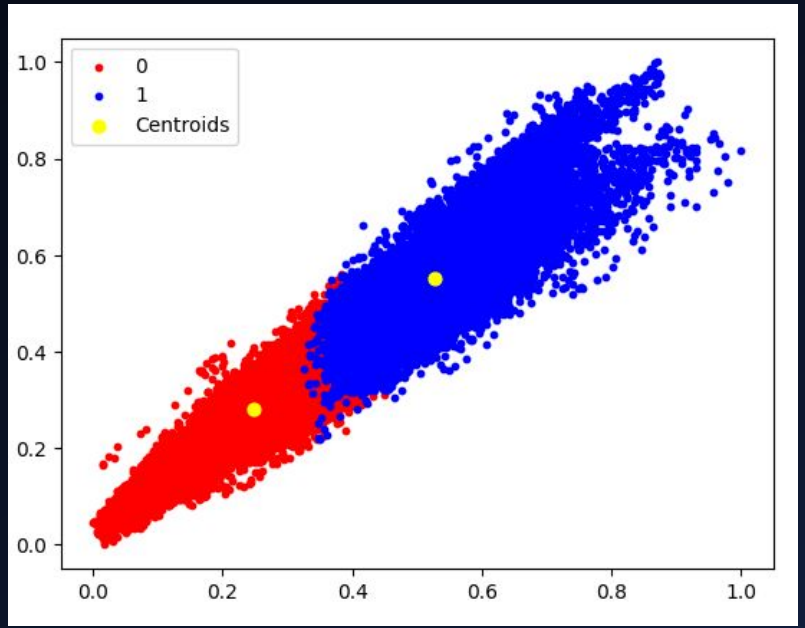


Cluster #2

```
kmeans = KMeans(n_clusters = 2, init = "k-means++", max_iter = 300, n_init = 50, random_state =100)
```

- Separated data to have five main companies, chose (AEP, DAYTON, PJMW, DUQ, EKPC) years from 2013-2018
- Standardized data (0-1), used K-means
- Used elbow method to find best number of clusters in df
- Number of clusters: 2

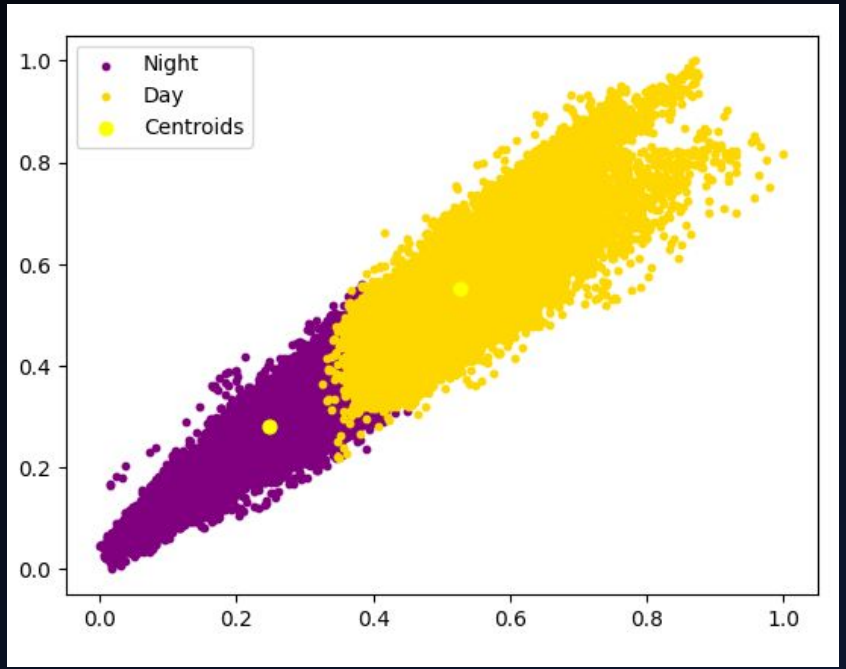


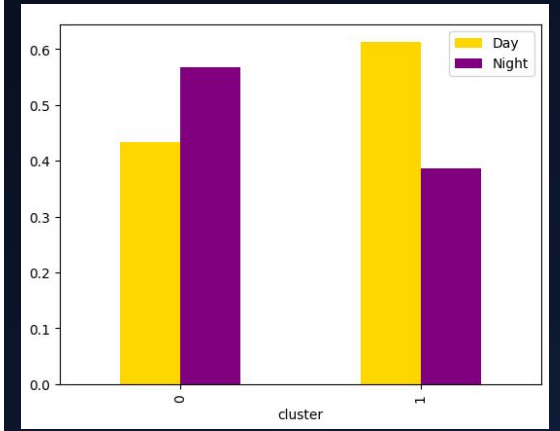


How to Get Meaning From Cluster #2

- It can represent night and day or working hours vs non working hours
- Day is considered from 7am to 6pm and night from 7pm to 7am

cluster	Day	Night
0	0.433263	0.566737
1	0.613022	0.386978

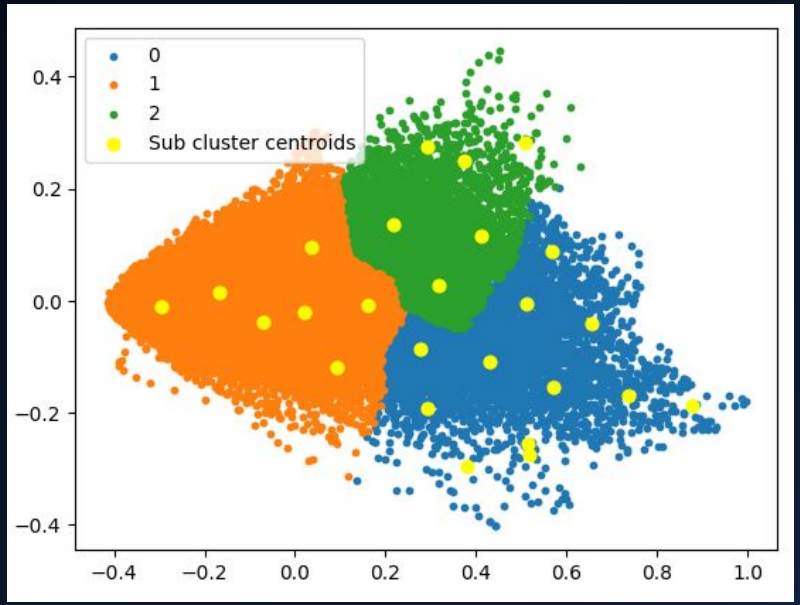




Cluster #3

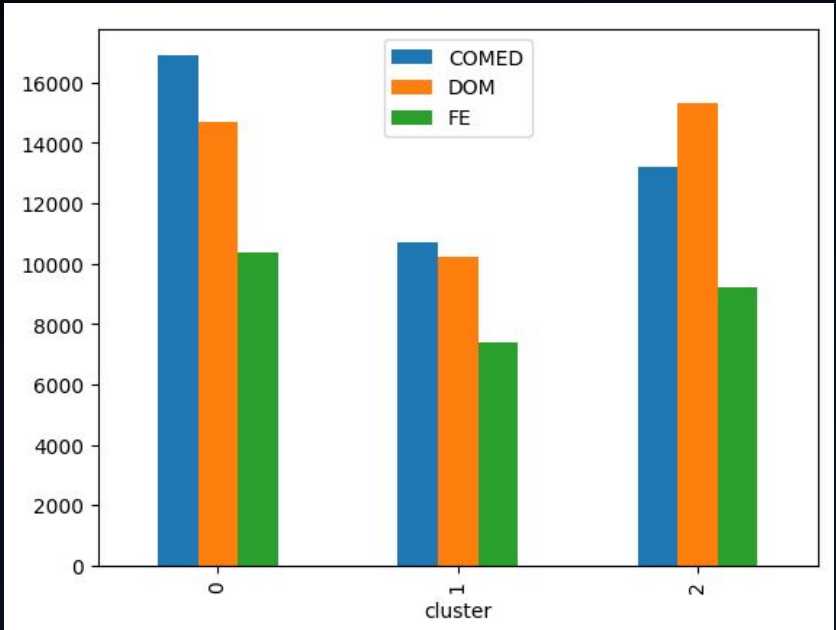
```
br = Birch(threshold = 0.1, branching_factor = 50, n_clusters= 3)
#threshold: radius of the subcluster
#branching_factor is the maximum # of subclusters in each node
```

- Chose three companies from around the U.S. (COMED, FE, DOM) years from 2013-2018 (Illinois, Pennsylvania/others, Virginia), respectively
- Standardized data (0-1), Used PCA, and BIRCH clustering
- Number of clusters: 3



How to Get Meaning From Cluster #3

- Could the algorithm have influences from the companies and their locations?



- Most likely not
- The computer is not biased, based on where the company is located