LIBRARY

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [0]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

READING THE DATA

In [0]:
```
#https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
df=pd.read_csv('/content/drive/My Drive/MiningProcess_Flotation_Plant_Database.csv',parse_dates=False,decimal=',')
```

In [0]:
```
df[:5]
```

Out[0]:

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotati Colu 03 Fl
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	250.5

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotati Colu 03 Fl
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	250.8
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	250.3
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	250.0
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	249.8

EXPLANATION OF PARAMETERS

1. pd.read_csv--- used to read the csv files
2. decimal (,)--- parameter since our dataset has misformatted readings so by using the decimal parameter to recongize the decimal parts 3.parse_dates---is used to parse the date and used to recongise the date format and orients the data according to given format and parse it

HIGH LEVEL STATISTICS

In [0]:
```
#HABERMAN ASSIGNMENT--https://classroom.appliedcourse.com/classrooms/jEARG7xb/assignment/4VAjo7jL/user/BmAM4101/

print("SHAPE OF DATAFRAME",df.shape)
print("*"*1000)
print("COLUMNS IN THE DATAFRAME",df.columns)
```

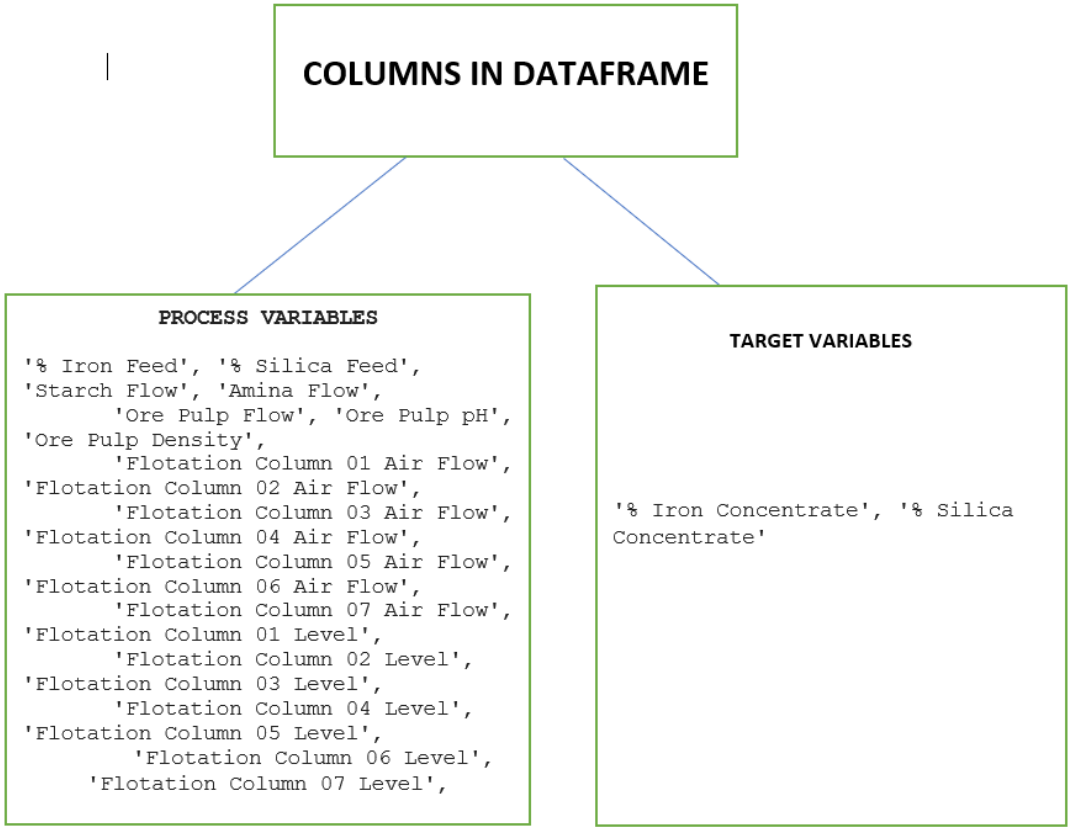
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
SHAPE OF DATAFRAME (737453, 24)
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
******
COLUMNS IN THE DATAFRAME Index(['date', '% Iron Feed', '% Silica Feed',
'Starch Flow', 'Amina Flow',
       'Ore Pulp Flow', 'Ore Pulp pH', 'Ore Pulp Density',
       'Flotation Column 01 Air Flow', 'Flotation Column 02 Air Flow',
       'Flotation Column 03 Air Flow', 'Flotation Column 04 Air Flow',
       'Flotation Column 05 Air Flow', 'Flotation Column 06 Air Flow',
       'Flotation Column 07 Air Flow', 'Flotation Column 01 Level',
       'Flotation Column 02 Level', 'Flotation Column 03 Level',
       'Flotation Column 04 Level', 'Flotation Column 05 Level',
       'Flotation Column 06 Level', 'Flotation Column 07 Level',
       '% Iron Concentrate', '% Silica Concentrate'],
      dtype='object')
```

INFERENCES

1. THE SHAPE OF DATASET IS (737453,24) WHICH MEANS THE DATASET HAS TOTAL 737453 DATA SAMPLES GENERATED EVERY 20 SECS FROM MARCH 2017 TO SEPTEMBER 2017 AND THERE ARE 24 FEATURES TO DETERMINE THE PERCENTAGE OF SILICA CONCENTERATE .
2. THE COLUMNS CAN BE STUDIED AS TWO PARTS:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



DATA TYPE OF ALL THE COLUMNS

In [0]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 737453 entries, 0 to 737452
Data columns (total 24 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   date                          737453 non-null  datetime64[ns]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
 1   % Iron Feed                  737453 non-null  float64
 2   % Silica Feed                737453 non-null  float64
 3   Starch Flow                  737453 non-null  float64
 4   Amina Flow                   737453 non-null  float64
 5   Ore Pulp Flow                737453 non-null  float64
 6   Ore Pulp pH                  737453 non-null  float64
 7   Ore Pulp Density             737453 non-null  float64
 8   Flotation Column 01 Air Flow 737453 non-null  float64
 9   Flotation Column 02 Air Flow 737453 non-null  float64
 10  Flotation Column 03 Air Flow 737453 non-null  float64
 11  Flotation Column 04 Air Flow 737453 non-null  float64
 12  Flotation Column 05 Air Flow 737453 non-null  float64
 13  Flotation Column 06 Air Flow 737453 non-null  float64
 14  Flotation Column 07 Air Flow 737453 non-null  float64
 15  Flotation Column 01 Level    737453 non-null  float64
 16  Flotation Column 02 Level    737453 non-null  float64
 17  Flotation Column 03 Level    737453 non-null  float64
 18  Flotation Column 04 Level    737453 non-null  float64
 19  Flotation Column 05 Level    737453 non-null  float64
 20  Flotation Column 06 Level    737453 non-null  float64
 21  Flotation Column 07 Level    737453 non-null  float64
 22  % Iron Concentrate           737453 non-null  float64
 23  % Silica Concentrate         737453 non-null  float64
dtypes: datetime64[ns](1), float64(23)
memory usage: 135.0 MB
```

DESCRIPTION OF THE DATAFRAME

In [0]:
```
df.describe()
```

Out[0]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH
count	737453.000000	737453.000000	737453.000000	737453.000000	737453.000000	737453.00000
mean	56.294739	14.651716	2869.140569	488.144697	397.578372	9.76763

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH
std	5.157744	6.807439	1215.203734	91.230534	9.699785	0.38700
min	42.740000	1.310000	0.002026	241.669000	376.249000	8.75334
25%	52.670000	8.940000	2076.320000	431.796000	394.264000	9.52736
50%	56.080000	13.850000	3018.430000	504.393000	399.249000	9.79810
75%	59.720000	19.600000	3727.730000	553.257000	402.968000	10.03800
max	65.780000	33.400000	6300.230000	739.538000	418.641000	10.80810

DESCRIPTION OF THE DATAFRAME

1. THE COUNT,MEAN STD,MIN,DIFFERENT PERCENTILES AMD MAX IN EACH COLUMNS IS DESCRIBED

   COUNT : INDICATES THE NUMBER OF VARIABLES IN EACH COLUMN.

   MEAN : THE MEAN FROM FLOATION OF AIR FLOW AND LEVEL 01-07 IS ALMOST SAME MEANS ONLY SMALL CHANGES ARE MADE FOR EVERY 20 SECONDS.

   % silica concenteerate MEAN IS LESS FROM % IRON CONCENTRATE WHICH MEANS IRON FROM ORE OBTAINED IS MORE AND IMPURITIES LEVEL IS LESS FROM THESE TYPE OF ORE.

   25 % : THE 25 % OF SILICA CONCENTRATE LIES BETWEEN 1.44 WHICH MEANS OUT OF 737453 SAMPLES WE ARE GETTING 300,000 SAMPLES OF SILICA CONCENTRATE LESS THAN 1.12 WHEN % IRON FEED AND % SILICA FEED IS LESS THAN 53% AND 9%.

   50% : THE 50 % OF SILICA CONCENTRATE LIES BETWEEN 2.00 WHICH MEANS OUT OF 737453 SAMPLES WE ARE GETTING 400,000 SAMPLES OF SILICA CONCENTRATE

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

LESS THAN 2.00 WHEN % IRON FEED AND % SILICA FEED IS LESS THAN 56% AND 14%.

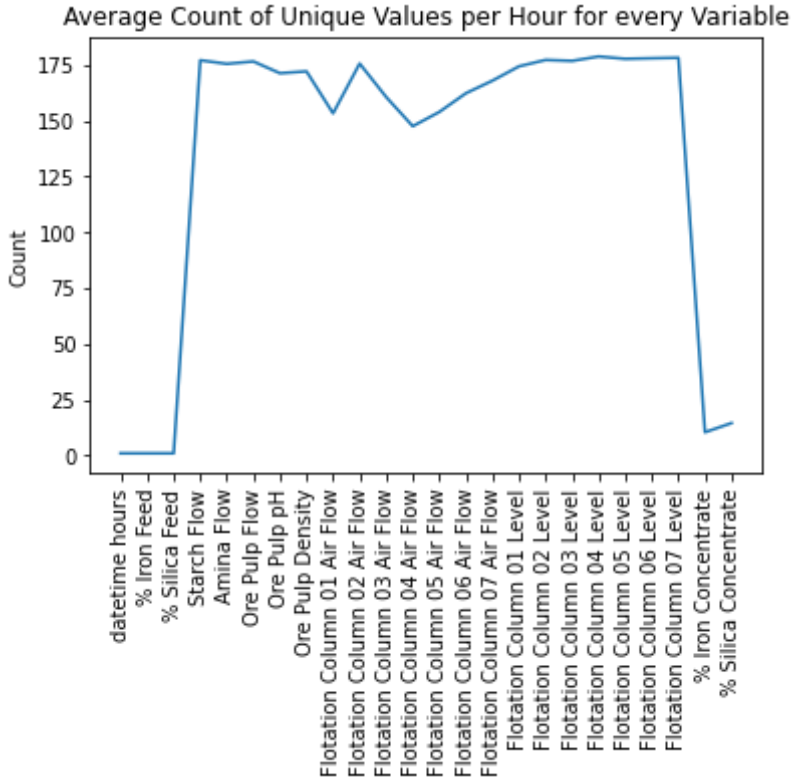
75% : THE 75 % OF SILICA CNCENTRATE LIES BETWEEN 3 WHICH MEANS OUT OF 737453 SAMPLES WE ARE GETTING 553089 SAMPLES OF SILICA CONCENTRATE LESS THAN 3.00 WHEN % IRON FEED AND % SILICA FEED IS LESS THAN 60% AND 20%.

WITH THE HELP OF THESE PERCENTILES WE CAN DRAW TO CONCLUSION FOR INCREASE IN THE FEED RATES WE CAN SEE THE INCREASE IN % SILICA CONCENTRATE AND IRON CONCENTRATE

Checking which variables have hourly vs 20-sec frequency

In [0]:
```
#https://towardsdatascience.com/a-basic-guide-into-time-series-analysis
-2ad1979c7438
#https://www.cognoesis.com/time-series-data-exploratory-data-analysis-f
orecast/
unique_avg = []
for col in df.columns:
    unique_avg.append(df.groupby('datetime hours').apply(lambda x: len(
x[col].unique())).mean())
plt.plot(np.arange(len(unique_avg)), unique_avg)
plt.title('Average Count of Unique Values per Hour for every Variable')
plt.ylabel('Count')
plt.xticks(list(range(len(unique_avg))), list(df.columns), rotation='ve
rtical')
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



EXPLANATION

NEED

SINCE, THE DATA ARE SAMPLED FOR EVERY 20 SECS AND WHICH ARE MAIN PROCESS VARIABLES THAT ARE CHANGED FOR EVERY 20 SEC AND THE VARIABLES CHANGED FOR EVERY ONE HOUR .

INFORMATION OBTAINED

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

THE PLOT EXPLAINS ABOUT THE CHANGE OF VARIABLES IN ONE HOUR AND THE VARIABLES LIKE IRON FEED SILICA FEED DO NOT CHANGE MUCH IN THE HOUR.

THE SCALE IS FROM 0 TO 180 RECORDS BECAUSE FOR EVERY 20 SECS WE ARE SAMPLING HENCE FOR ONE HOUR WE GET 180 RECORDS IN TOTAL SO THE SCALE ARE FROM 0 - 180 .

THE VARIABLES LIKE FLOATATION AIR FLOW AND LEVEL CHANGES IN ONE HOUR . HENCE, THE MPOST IMPORTANT VARIABLES ARE AIR-FLOW AND LEVEL. CONTROLLING THESE VARIABLES CAN BE USED TO CONTROL TYE YIELD RATE OF SILICA CONCENTRATE.

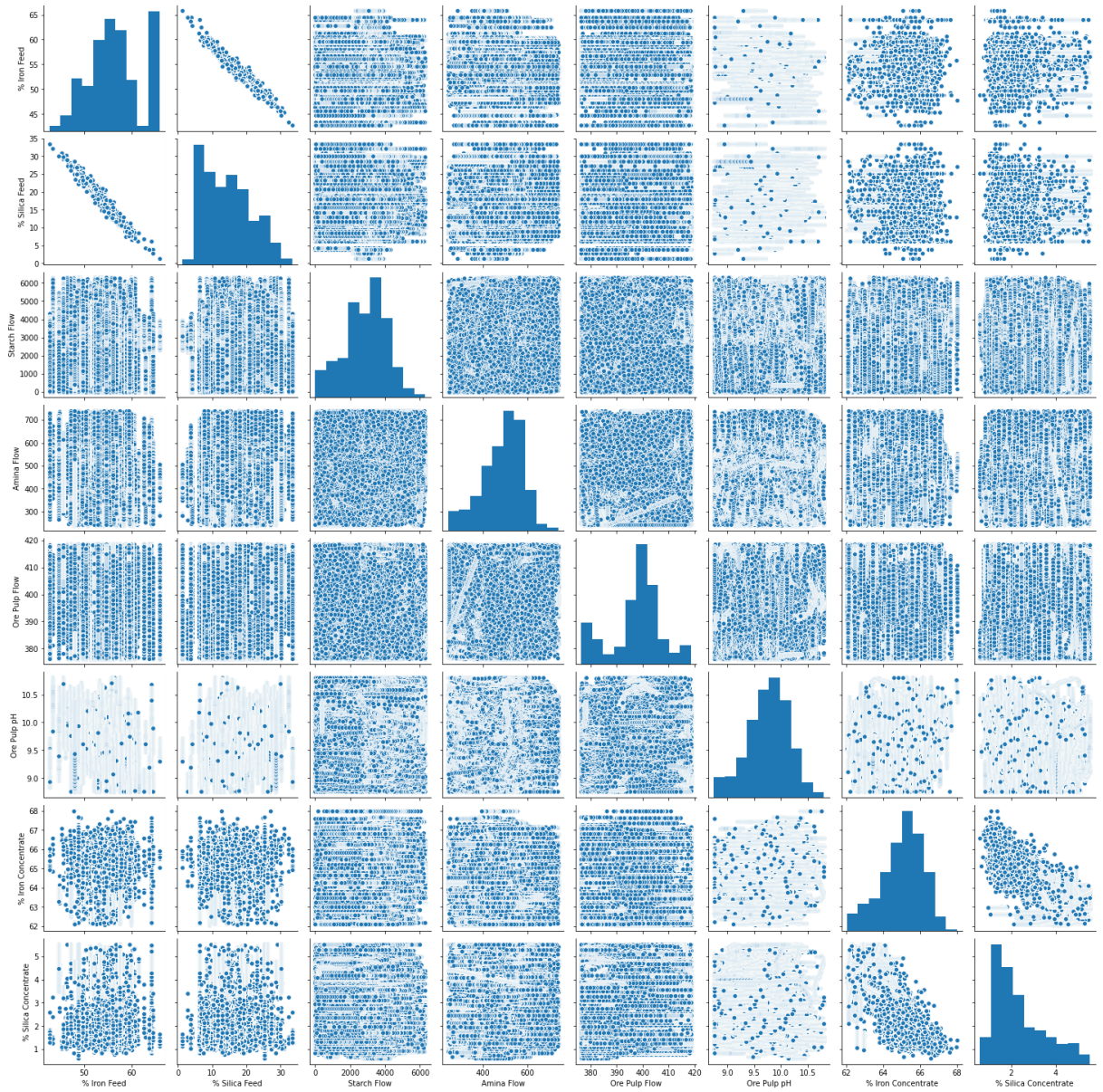
PAIR PLOTS

In [0]:
```
pair_cols = list(df.columns[2:8])
pair_cols.extend(df.columns[-2:])
print(pair_cols)
#smol_df = clean_df.loc[:,pair_cols]
```

```
['% Iron Feed', '% Silica Feed', 'Starch Flow', 'Amina Flow', 'Ore Pulp Flow', 'Ore Pulp pH', '% Iron Concentrate', '% Silica Concentrate']
```

In [0]:
```
sns.pairplot(df.loc[:,pair_cols])
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

NEED

THE PLOT IS USEFUL TO GIVE OVERVIEW ABOUT THE CORRELATION OF FEATURES . IN THIS PLOT WE ARE NOT CONSIDERING THE FLOATATION OF AIRFLOW AND LEVEL BCAUSE THEY ARE HIGHLY CORRELATED SO WE ARE MAINLY TAKING CERTAIN FEATURES INTO CONSIDERATION AND TRY TO DRAW CONCLUSIONS FROM IT.

INFORMATION OBTAINED:

1. NO APPARENT MEANINGFUL PATTERNS BESIDES BETWEEN IRON AND SILICA CONCENTRATE AND IRON AND SILICA FEED (WHICH ARE TO BE EXPECTED).

   WE ALSO DECIDED TO CHECK WHICH MINUTE OF THE HOUR SHOWED THE HIGHEST CORRELATION WITH % SILICA CONCENTRATE FOR EACH VARIABLE. OUR HYPOTHESIS WAS THAT THEY SHOULD PEAK AROUND WHEN THE MEASUREMENTS WHERE USUALLY TAKEN.

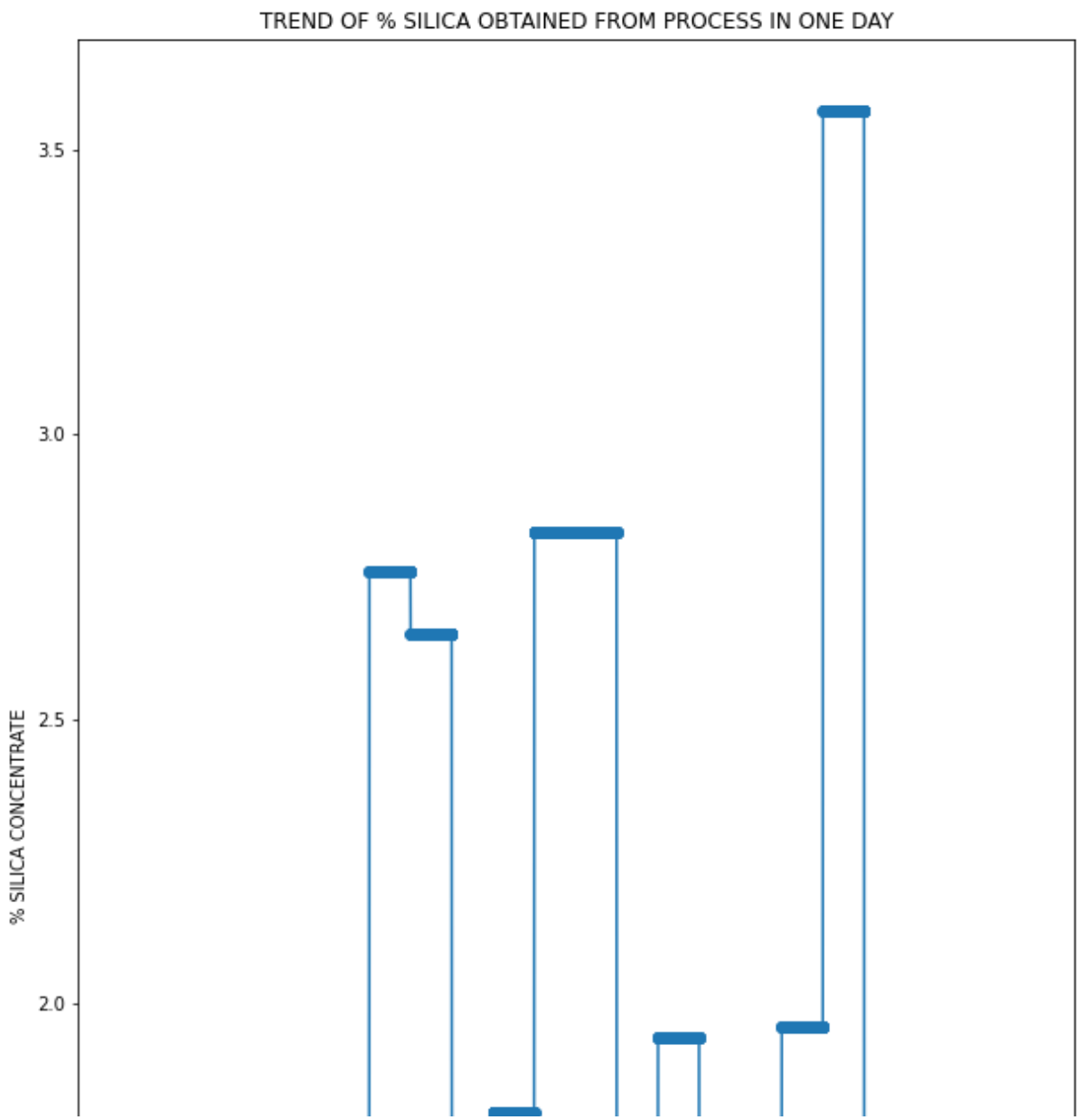
1. WE OBSERVE A NEGATIVE CORRELATION BETWEEN THE IRON AND SILICA CONCENTRATE AND WE OBSERVE THAT WHEN THERE IS LARGE AMOUNT OF IRON CONCENTRATE IN IRON ORE WE GET MINIMUM IMPURITIES (SILICA CONCENTRATE) WHICH CONCLUDES THAT IRON ORE WE USE IT FOR OUR PROCESS IS OF HIGH QUALITY .

TREND OF SILICA CONCENTRATE IN ONE DAY

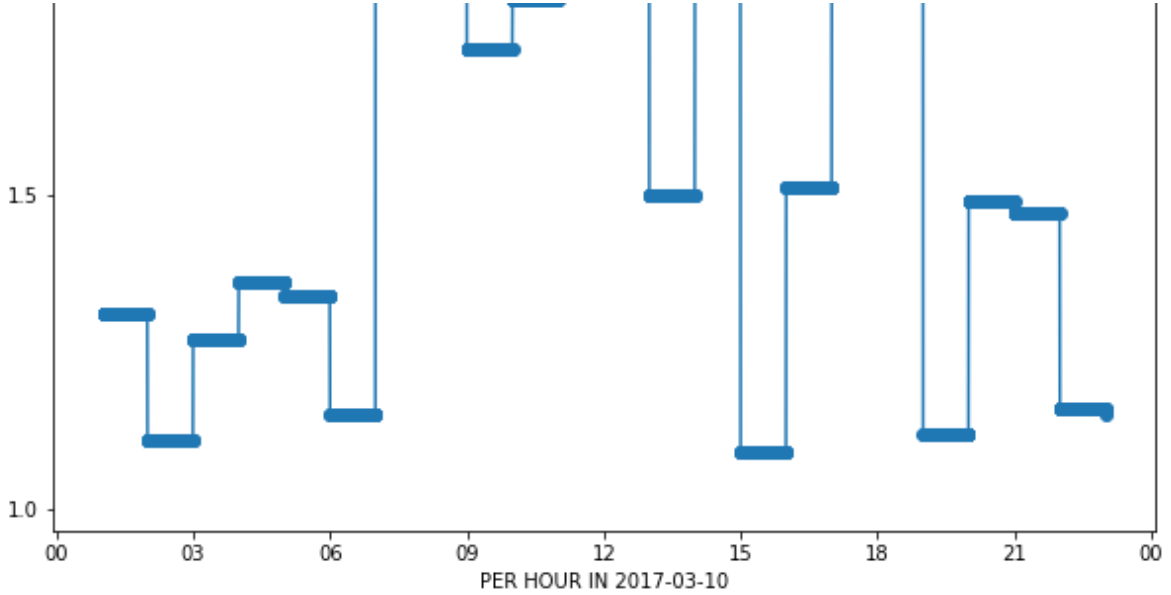
In [0]:
```
#https://matplotlib.org/3.1.1/api/dates_api.html
#https://www.programcreek.com/python/example/61483/matplotlib.dates.DateFormatter
# blog about petrol consumption in one day graph
import matplotlib.dates as mdates
fig, ax = plt.subplots(figsize=(10,16))
ax.plot(df.loc['2017-03-10 1:00:00':'2017-03-10 23:00:00', '% Silica Concentrate'], marker='o', linestyle='-')
ax.set_ylabel('% SILICA CONCENTRATE')
ax.set_title('TREND OF % SILICA OBTAINED FROM PROCESS IN ONE DAY')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ax.set_xlabel('PER HOUR IN 2017-03-10')
# Format x-tick labels as 3-letter month name and day number
ax.xaxis.set_major_formatter(mdates.DateFormatter('%H'))
```



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

1 . **NEED**

SINCE THE ABOVE PLOTS ARE NOT CLEAR TO DRAW IMPORTANT INFORMATION WE GO FOR ONE DAY ANALYSIS INORDER TO GET ANY IMPORTANT CONCLUSIONS.

INFORMATION OBTAINED

THE PLOT MAINLY FOCUS THE TREND OF % SILCCA CONCENTRATE IN ONE DAY(2017-03-10) AND WE GET THE IDEA THT THE SILICA CONCENTRATE IS NOT SAME FOR ONE DAY IT IS HIGHLY CHANGING FOR EVERY HOUR BASED ON SEVERAL FACTORS.

THE HIGHEST PEAK IS REACHED AT THE LATE HOURS IN ONE DAY AND BEFORE REACHING THE MAXIMUN VALUE IT ALSO REACHED ITS MINMUM VALUE AT 15:00 AND THE % SILICA CONCENTRATE IS NOT MONOTONIC GRAPH OR LINEAR GRAPH IT S HIGHLY NON-LINEAR GRAPH.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

PARAMETERS OF HIGHEST AND LOWEST PEAK IN THE ABOVE GRAPH

In [0]:
```
s=df.loc['2017-03-10 18:00:00']
m=df.loc['2017-03-10 15:00:00']
print(s)
print("-----------------------------------------------------")
print(m)
```

```
datetime hours                  2017-03-10 18:00:00
% Iron Feed                                   55.99
% Silica Feed                                 17.16
Starch Flow                                 5611.82
Amina Flow                                   535.553
Ore Pulp Flow                                397.676
Ore Pulp pH                                  10.2855
Ore Pulp Density                             1.75575
Flotation Column 01 Air Flow                 252.235
Flotation Column 02 Air Flow                 249.598
Flotation Column 03 Air Flow                 248.687
Flotation Column 04 Air Flow                 295.096
Flotation Column 05 Air Flow                   306.4
Flotation Column 06 Air Flow                 249.906
Flotation Column 07 Air Flow                   251.4
Flotation Column 01 Level                    559.771
Flotation Column 02 Level                    538.144
Flotation Column 03 Level                     552.23
Flotation Column 04 Level                    541.228
Flotation Column 05 Level                    547.838
Flotation Column 06 Level                    533.877
Flotation Column 07 Level                     536.31
% Iron Concentrate                             66.21
% Silica Concentrate                            3.57
Name: 2017-03-10 18:00:00, dtype: object
-----------------------------------------------------
datetime hours                  2017-03-10 15:00:00
% Iron Feed                                   54.95
% Silica Feed                                  17.4
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Starch Flow                                4588.87
Amina Flow                                 514.771
Ore Pulp Flow                              392.461
Ore Pulp pH                                10.3162
Ore Pulp Density                              1.78
Flotation Column 01 Air Flow               248.643
Flotation Column 02 Air Flow               252.334
Flotation Column 03 Air Flow               249.741
Flotation Column 04 Air Flow               295.096
Flotation Column 05 Air Flow                 306.4
Flotation Column 06 Air Flow               250.488
Flotation Column 07 Air Flow                 250.4
Flotation Column 01 Level                  565.352
Flotation Column 02 Level                  532.648
Flotation Column 03 Level                  555.092
Flotation Column 04 Level                  568.126
Flotation Column 05 Level                   572.15
Flotation Column 06 Level                  533.514
Flotation Column 07 Level                  563.566
% Iron Concentrate                            67.6
% Silica Concentrate                          1.09
Name: 2017-03-10 15:00:00, dtype: object
```

INFERENCES

1. THE ABOVE DATA IS ABLE TO DRAW CONCLUSION ABOUT THE BEHAVIOUR OF THE VARIABLES TO REACH THE MAXIMUM SILICA CONCENTRATE

|**COMPARISION OF PROCESS VALUES**

THE IRON CONCENTRATE IN THE HIGHEST VALUE IS MINIMUM AND IN THE LOWEST VALUE FOR SILICA CONCENTRATE THE VALUE OF IRON CONCENTRATE IS HIGHER WHICH STRONGLY SUPPORT OUR POINT OF NEGATIVE CORRELATION BETWEEN THE ABOVE VARIABLES.

FOR LESS IRON FEED WE ARE GETTING LESS SILICA CONCENTRATE AND TO FIND THE RELATION BETWEEN THE VARIABLES WE GO FOR THE FUTHER BREAK DOWN THE

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

ANALYSIS.

THE REAGENT FLOW IS IMPORTANT IN ANY INDUSTRIAL PROCESS ISNCE IT ACTS AS CATALYST FOR THE PROCESS HENCE , THE STUDY OF ITS IMPACT ON SILICA IS ALOS IMPORANT AND WE QUANTITVELY CONCLUDE THAT THE AMINA FLOW VALUES AFFECT THE SILICA CONCENTARTE

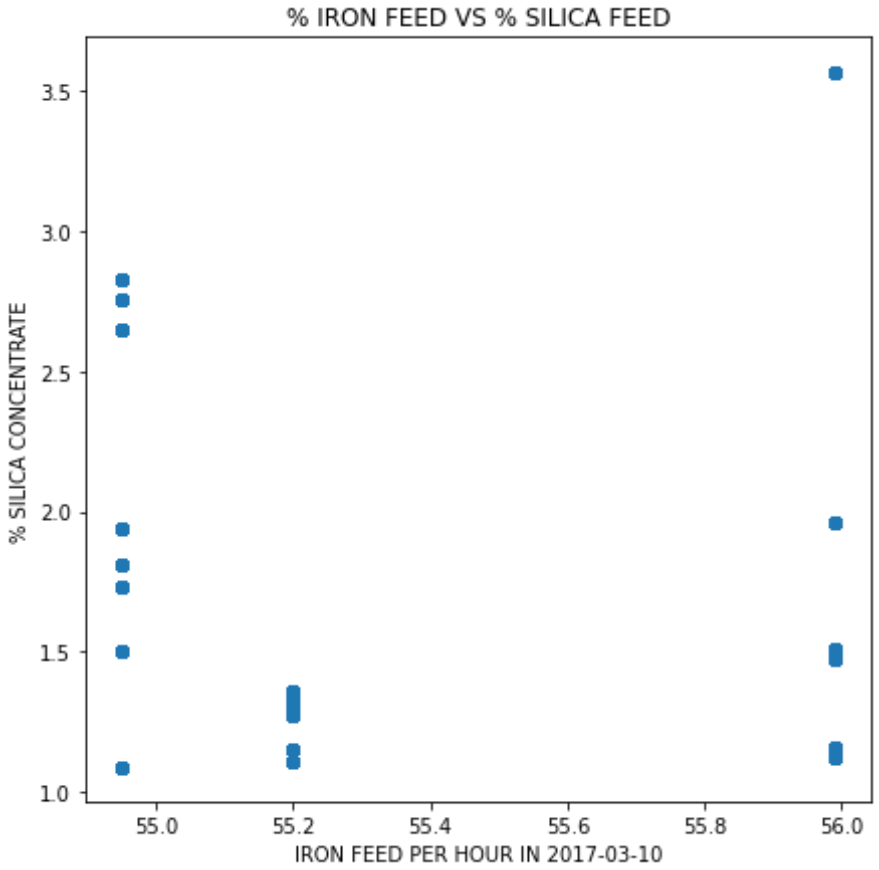
IRON FEED VS SILICA CONCENTRATE

In [0]:
```
fig, ax = plt.subplots(figsize=(7,7))
amina_A=df.loc[:3994, '% Iron Feed']
amina_s=df.loc[:3994, '% Silica Concentrate']
ax.scatter(amina_A,amina_s, marker='o', linestyle='-')

ax.set_ylabel('% SILICA CONCENTRATE')
ax.set_title('% IRON FEED VS % SILICA FEED')
ax.set_xlabel('IRON FEED PER HOUR IN 2017-03-10')
```

Out[0]:
```
Text(0.5, 0, 'IRON FEED PER HOUR IN 2017-03-10')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

NEED

SINCE FROM TREND GRAPH WE OBTAINED CERTAIN REALTION SO FOR DEEP ANALYSIS WE HAVE PLOTTED THE IRON VS SILICA CONCENTRATE TO GET PATTERN.

INFORMATION

WE OBSERVE THAT THERE IS NO DATAPOINTS IN RANGE 55.2 TO 55.8 WHICH TELLS MOST OF THE POINTS LIES EITHER LESSTHAN 55.2 AND GREATER THN 55.8.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

FOR RANGES LESS THAN 55.2 THE SILICA CONCENTRATE ARE MAXIMUN UPTO 2.8 AND MINIMUM CVALUES IS GRETER THAN 1

FOR RANGES GRATER THAN 55.8 THE SILICA CONCENTRATE IS ACHIEVING THE MAXIMUM OF 3.5 AND MIMIMUM OF 1.5

FROM ANALYSISING THE TWO RANGES , ONE CAN CONCLUDE THAT WE MORE THE IRON FEED GRETER IS THE SILICA CONCENTRATE.

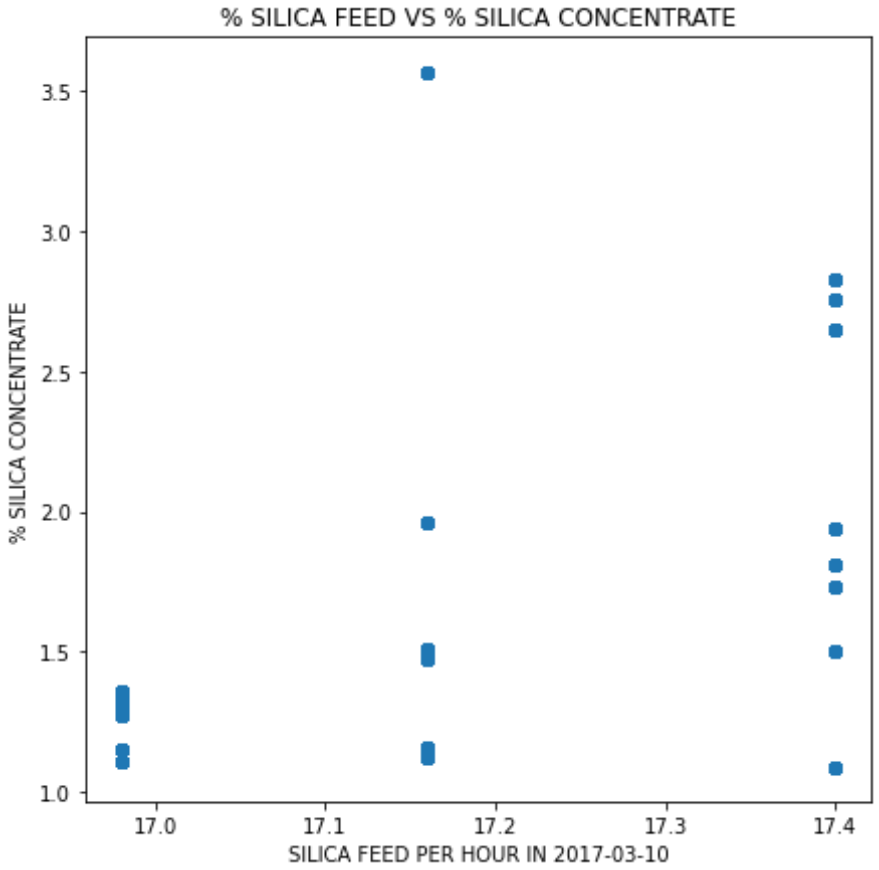
SILICA FEED VS SILICA CONCENTRATE

In [0]:
```
import matplotlib.dates as mdates
fig, ax = plt.subplots(figsize=(7,7))
amina_A=df.loc[:3994, '% Silica Feed']
amina_s=df.loc[:3994, '% Silica Concentrate']
ax.scatter(amina_A,amina_s, marker='o', linestyle='-')

ax.set_ylabel('% SILICA CONCENTRATE')
ax.set_title('% SILICA FEED VS % SILICA CONCENTRATE')
ax.set_xlabel('SILICA FEED PER HOUR IN 2017-03-10')
```

Out[0]:
```
Text(0.5, 0, 'SILICA FEED PER HOUR IN 2017-03-10')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

NEED

SINCE FROM TREND GRAPH WE OBTAINED CERTAIN REALTION SO FOR DEEP ANALYSIS WE HAVE PLOTTED THE SILICA FEED VS SILICA CONCENTRATE TO GET PATTERN.

INFORMATION

WE ANALYSE THE GRAPH INTO THREE SECTION BASED ON RANGES:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

FOR RANGES LESS THAN 17.15 THE SILICA CONCENTRATE ARE MAXIMUN UPTO 1.4

FOR RANGES GRATER THAN 17.15 TO 17.4 THE SILICA CONCENTRATE IS ACHIEVING THE MAXIMUM OF 3.5 AND MIMIMUM OF 1.5

AT 17.4 THE SILICA CONCENTRATE IS ACHIEVING THE MAXIMUM OF 3 AND MIMIMUM OF 1.10

FROM ANALYSISING THE THREE RANGES , ONE CAN CONCLUDE THAT WE MORE THE SILICA FEED IS NOT IMPACTING SILICA CONCENTRATE AS WE THOUGHT.

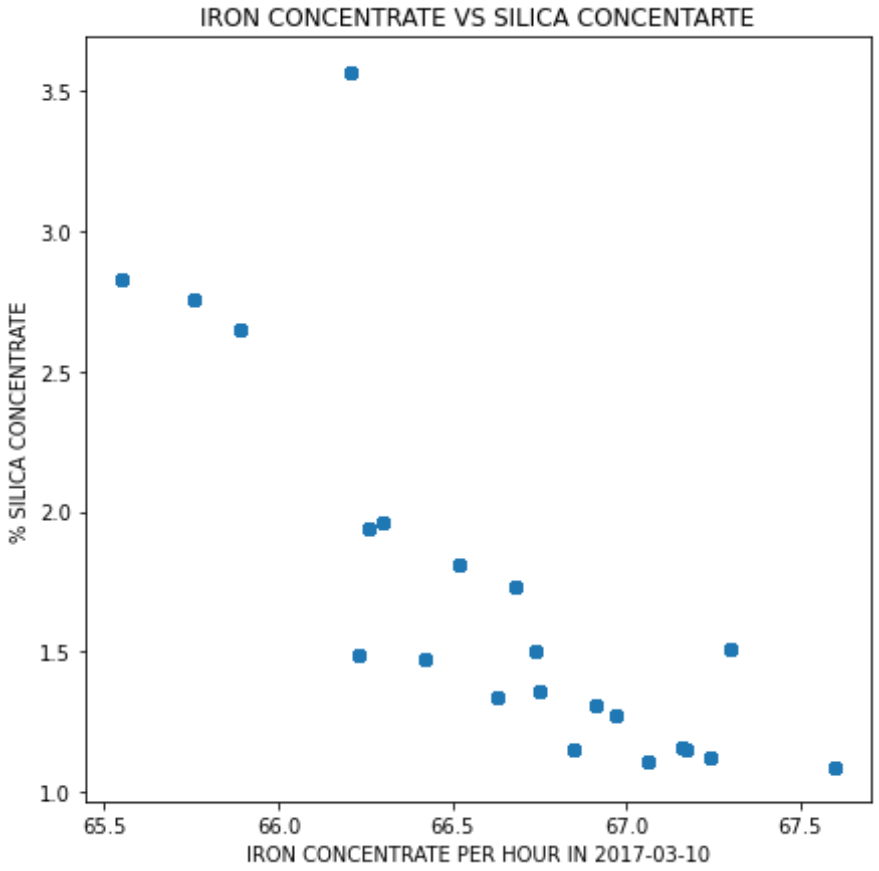
IRON CONCENTRATE VS SILICA CONCENTARTE

In [0]:
```
#https://www.kaggle.com/schummel/froth-floating-process-with-random-forests
fig, ax = plt.subplots(figsize=(7,7))
amina_A=df.loc[:3994, '% Iron Concentrate']
amina_s=df.loc[:3994, '% Silica Concentrate']
ax.scatter(amina_A,amina_s, marker='o', linestyle='-')

ax.set_ylabel('% SILICA CONCENTRATE')
ax.set_title('IRON CONCENTRATE VS SILICA CONCENTARTE')
ax.set_xlabel('IRON CONCENTRATE PER HOUR IN 2017-03-10')
```

Out[0]:
```
Text(0.5, 0, 'IRON CONCENTRATE PER HOUR IN 2017-03-10')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

NEED

SINCE FROM TREND GRAPH WE OBTAINED CERTAIN REALTION SO FOR DEEP ANALYSIS WE HAVE PLOTTED IRON CONCENTRATE VS SILICA CONCENTRATE TO GET PATTERN.

INFORMATION

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

THE MOST IMPORTANT PLOT FOR OUR TASK IS IRON CONCENTRATE VS SILICA CONCENTRATE AND WE SEE LOT OF OBSERVATIONS

1. THE IRON AND SILICA CONCENTRATE ARE HIGHLY CORRELATED IN A NEGATIVE MANNER.
2. EXCEPT FOR ONE POINT , WE CAN FORM A DECREASING CLOUD PATTERN IN GRAPH AND WHEN THERE IS INCREASE IN THE IRON CONCENTRATE THERE IS DECRAESE IN THE SILICA CONCENTRATE WHICH INDIRECTLY INDICATES THAT THE AMOUNT OF IRON CONCENTRATE IN THAT PARTIULAR BATCH FEED IS HIGH AND LESS WITH IMPURITIES

WE ARE MAING VERY IMPORTANT CONCLUSION FROM ABOVE PLOT THAT BOTH THE VARUIBALES ARE NEGATIVELY CORRELATED

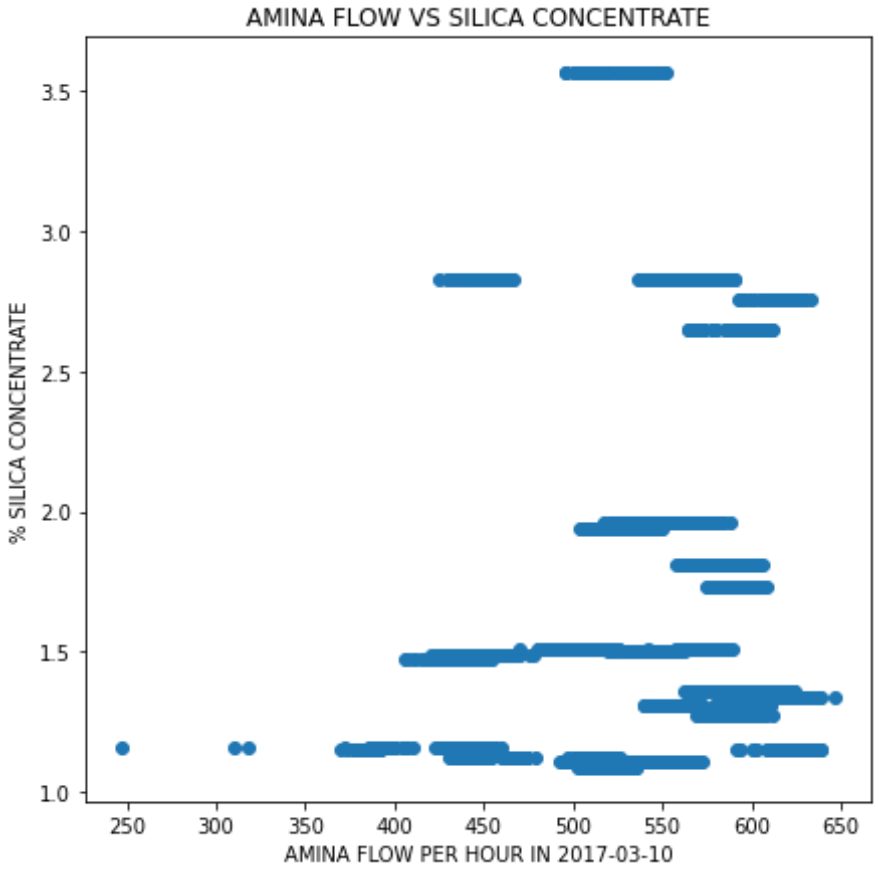
AMINA FLOW VS SILICA CONCENTRATE

In [0]:
```
#https://en.wikipedia.org/wiki/Reagent
#https://www.kaggle.com/tastycanofmalk/predict-silica-concentration
fig, ax = plt.subplots(figsize=(7,7))
amina_A=df.loc[:3994, 'Amina Flow']
amina_s=df.loc[:3994, '% Silica Concentrate']
ax.scatter(amina_A,amina_s, marker='o', linestyle='-')

ax.set_ylabel('% SILICA CONCENTRATE')
ax.set_title('AMINA FLOW VS SILICA CONCENTRATE')
ax.set_xlabel('AMINA FLOW PER HOUR IN 2017-03-10')
```

Out[0]:
```
Text(0.5, 0, 'AMINA FLOW PER HOUR IN 2017-03-10')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

NEED

SINCE FROM TREND GRAPH WE OBTAINED CERTAIN REALTION SO FOR DEEP ANALYSIS WE HAVE PLOTTED THE AMINA FLOW VS SILICA CONCENTRATE TO GET PATTERN.

INFORMATION

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

A REAGENT IS A SUBSTANCE OR COMPOUND ADDED TO A SYSTEM TO CAUSE A CHEMICAL REACTION, OR ADDED TO TEST IF A REACTION OCCURS. THE TERMS REACTANT AND REAGENT ARE OFTEN USED INTERCHANGEABLY—HOWEVER, A REACTANT IS MORE SPECIFICALLY A SUBSTANCE CONSUMED IN THE COURSE OF A CHEMICAL REACTION.

THE REAGENT INCREASES THERE IS VARIATION IN THE SILICA CONCENTRATE BUT FOR SMALL VALUES WE CAN OBSERVE VERY LITTLE SILICA CONCENTRATE.

BOX-PLOTS

NEED

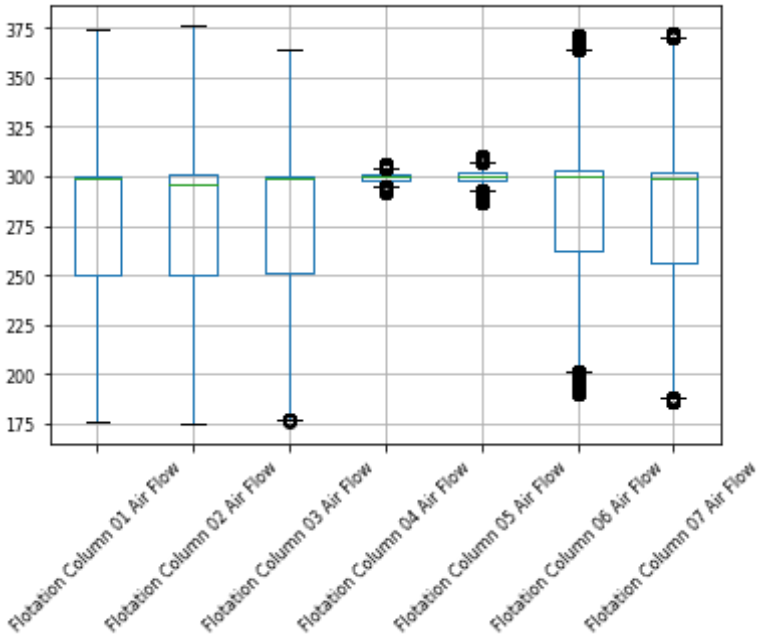
TO FIND THE OUTLIERS IN DATASET

WE BREAK DOWN THE ANALYSIS INTO THREE BOX PLOTS

1. FIRST PLOT FOCUS ON THE FLOATION AIR FLOW
2. SECOND PLOT FOCUS ON FLOATTION LEVEL

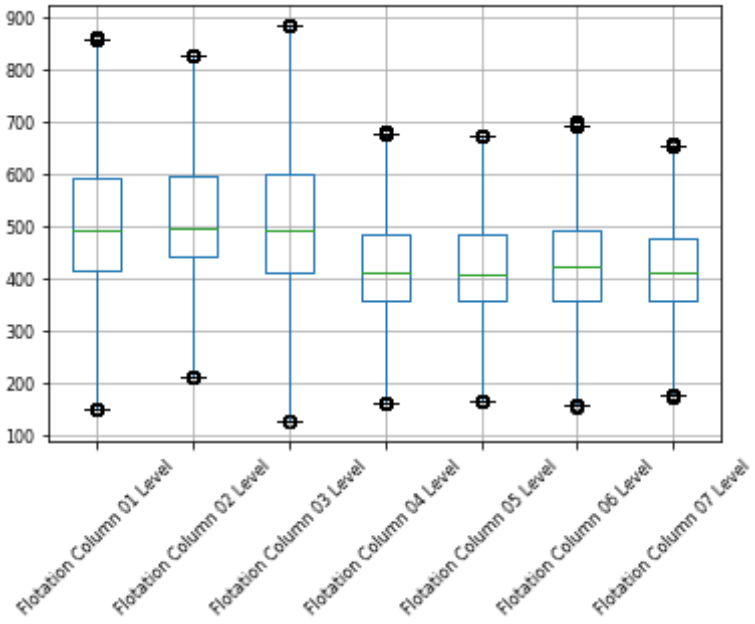
In [0]:
```
#https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
boxplot = df.boxplot(column=['Flotation Column 01 Air Flow','Flotation Column 02 Air Flow','Flotation Column 03 Air Flow','Flotation Column 04 Air Flow','Flotation Column 05 Air Flow','Flotation Column 06 Air Flow','Flotation Column 07 Air Flow'], rot=45, fontsize=8)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



In [0]:
```
boxplot = df.boxplot(column=['Flotation Column 01 Level','Flotation Col
umn 02 Level','Flotation Column 03 Level','Flotation Column 04 Level',
'Flotation Column 05 Level','Flotation Column 06 Level','Flotation Colu
mn 07 Level'], rot=45, fontsize=8)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



INFERENCES

THE FIRST AND SECOND PLOT HAS LOT OF OUTLIERS AND REMOVING OF OUTLIERS CAN LEAD TO BAD MODEL HENCE, OULIERS SHOULD NOT BE REMOVED FROM DATASET.

THE BOX-PLOTS ARE DRWAN ONLY FOR FLOATION AIRFLOW AND LEVEL BECAUSE THESE ARE MOST VARIABLES WHICH VARIED IN THE ONE HOUR AND MAY INCLUDE OUTLIERS .

BUT, DROPPING THESE OUTLIER WILL LEAD TO BAD MODEL .

SUMMARY

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

1. IN HIGH LVEL STATISTCS,1. THE SHAPE OF DATASET IS (737453,24) WHICH MEANS THE DATASET HAS TOTAL 737453 DATA SAMPLES GENERATED EVERY 20 SECS FROM MARCH 2017 TO SEPTEMBER 2017 AND THERE ARE 24 FEATURES TO DETERMINE THE PERCENTAGE OF SILICA CONCENTERATE .
2. WITH THE HELP OF THESE PERCENTILES WE CAN DRAW TO CONCLUSION FOR INCREASE IN THE FEED RATES WE CAN SEE THE INCREASE IN % SILICA CONCENTRATE AND IRON CONCENTRATE
3. FROM THE TREND OF % SILICA CONCENTRATE GRAPH WE GET THE MOST IMPORTANT VARIABLES ARE AIR-FLOW AND LEVEL. CONTROLLING THESE VARIABLES CAN BE USED TO CONTROL TYE YIELD RATE OF SILICA CONCENTRATE.
4. WE CAN CONCLUDE FROM IRON FEED GRAPH THAT WE MORE THE IRON FEED GRETER IS THE SILICA CONCENTRATE.
5. WE ARE MAING VERY IMPORTANT CONCLUSION FROM IRON CONCENTRATE VS SILICA CONCENTRATE PLOT THAT BOTH THE VARUIBALES ARE NEGATIVELY CORRELATED.
6. THE REAGENT INCREASES THERE IS VARIATION IN THE SILICA CONCENTRATE BUT FOR SMALL VALUES WE CAN OBSERVE VERY LITTLE SILICA CONCENTRATE.

PRE-PROCESSING OF DATA

MISSING VALUES

In [0]:
```
#https://towardsdatascience.com/data-pre-processing-techniques-you-should-know-8954662716d6
#https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/
counts = df.groupby('date').count()
counts
```

Out[0]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotati Colu 04 Fl

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotati Colu 04 Fl
date											
2017-03-10 01:00:00	174	174	174	174	174	174	174	174	174	174	1
2017-03-10 02:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-03-10 03:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-03-10 04:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-03-10 05:00:00	180	180	180	180	180	180	180	180	180	180	1
...	...	...	...	...	...	...	...	...	...	...	
2017-09-09 19:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-09-09 20:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-09-09 21:00:00	180	180	180	180	180	180	180	180	180	180	1
2017-09-09 22:00:00	180	180	180	180	180	180	180	180	180	180	1

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotati Colu 04 / Fl
2017-09-09 23:00:00	180	180	180	180	180	180	180	180	180	180	1

4097 rows × 23 columns

CALCULATION

ONE HR = 3600 SECS

SAMPLES AT 20 SECONDS

LET US HAVE ONE RECORD AT END OF 20 SECS

SO, 3600/20 = 180

WHICH MEANS WE GET 180 RECORDS FOR ONE HOUR SO WE NEED TO WE FIND THE NUMBER OF RECORDS = 180 AND IF IT EQUAL WE ARE LUCKY TO COCNLUDE THERE IS NO MISSING VALUE ELSE WE NEED TO FILL THOSE MISSIN VALUES

In [0]:
```
counts_less=counts<180
counts_less
```

Out[0]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flota Colu 04 F

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flota Colu 04 F
date											
2017-03-10 01:00:00	True	True	True	True	True	True	True	True	True	True	
2017-03-10 02:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-03-10 03:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-03-10 04:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-03-10 05:00:00	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	
2017-09-09 19:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-09-09 20:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-09-09 21:00:00	False	False	False	False	False	False	False	False	False	False	F
2017-09-09 22:00:00	False	False	False	False	False	False	False	False	False	False	F

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flota Colu 04 F
2017-09-09 23:00:00	False	False	False	False	False	False	False	False	False	False	F

4097 rows × 23 columns

In [0]:
```
df.iloc[:,0]
```

Out[0]:
```
0         2017-03-10 01:00:00
1         2017-03-10 01:00:00
2         2017-03-10 01:00:00
3         2017-03-10 01:00:00
4         2017-03-10 01:00:00
                  ...
737448    2017-09-09 23:00:00
737449    2017-09-09 23:00:00
737450    2017-09-09 23:00:00
737451    2017-09-09 23:00:00
737452    2017-09-09 23:00:00
Name: date, Length: 737453, dtype: datetime64[ns]
```

In [0]:
```
len(df)
```

Out[0]:
```
737453
```

WE ARE LUCKY ENOUGH TO HAVE ONLY SIX MISSING RECORD FROM THE DATASET AND WE CAN FILL THOSE MISSING VALUES BY CONVERSION METHODS

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

MISSING VALUES TREATMENT

In [0]:
```
#https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/
#https://www.kaggle.com/juejuewang/handle-missing-values-in-time-series-for-beginners
date_range=pd.date_range(start='2017-03-10 1:00:00',end='2017-09-09 23:59:40',freq='20S')
date_range = date_range[6:]
date_range[-5:]
```

Out[0]:
```
DatetimeIndex(['2017-09-09 23:58:20', '2017-09-09 23:58:40',
               '2017-09-09 23:59:00', '2017-09-09 23:59:20',
               '2017-09-09 23:59:40'],
              dtype='datetime64[ns]', freq='20S')
```

In [0]:
```
hours=pd.Series(df['date'].unique())
hours.index = hours
len(hours)
```

Out[0]:
```
4097
```

In [0]:
```
hours_list = hours.index.format()
print(hours_list[:5])
seconds_list = date_range.format()
print(seconds_list[:5])
```

```
['2017-03-10 01:00:00', '2017-03-10 02:00:00', '2017-03-10 03:00:00',
'2017-03-10 04:00:00', '2017-03-10 05:00:00']
['2017-03-10 01:02:00', '2017-03-10 01:02:20', '2017-03-10 01:02:40',
'2017-03-10 01:03:00', '2017-03-10 01:03:20']
```

In [0]:
```
len(seconds_list),len(hours_list)
```

Out[0]:
```
(794694, 4097)
```

In [0]:
```
new_index = []
for idx in seconds_list:
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
    if (idx[:13] + ':00:00') in hours_list:
        new_index.append(idx)
```

In [0]:
```
#remove the one missing interval within the hour which we found earlier
 using the counts
new_index.remove('2017-04-10 00:00:00')
```

In [0]:
```
print(len(new_index))
print(len(df))
```

```
737453
737453
```

In [0]:
```
df['index'] = new_index
df['index'] = pd.to_datetime(df['index'])
df.index = df['index']
df = df.loc[:, df.columns[:-1]]
df.rename(columns={'date': 'datetime hours'}, inplace=True)
df.head()
```

Out[0]:

index	datetime hours	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow
2017-03-10 01:02:00	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235
2017-03-10 01:02:20	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532
2017-03-10 01:02:40	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874
2017-03-10 01:03:00	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

index	datetime hours	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow
2017-03-10 01:03:20	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136

In [0]:
```
df.to_csv('/content/drive/My Drive/preprocessed_time')
```

EXPLANATION

THE ABOVE STEPS TREAT THE MISSING VALUES BY CONVERTING TO THEIR RESPECTIVE 20 SEC SAMPLE INTERVAL AND FOR ACCORDINGLY FILLING THOSE SIX MISSING VALUE RECORDS

NULL VALUE DETECTION

In [0]:
```
df.isnull().values.any()
```

Out[0]:
```
False
```

THERE IS NO NULL VALUES

CORRELATION

In [0]:
```
 #https://stackoverflow.com/questions/29294983/how-to-calculate-correla
tion-between-all-columns-and-remove-highly-correlated-on
 #https://benalexkeen.com/correlation-in-python/
 corr=df.corr(method ='pearson')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [0]:
```
corr_triu = corr.where(~np.tril(np.ones(corr.shape)).astype(np.bool))
corr_triu = corr_triu.stack()
corr_triu=corr_triu[corr_triu > 0.6].to_frame()
corr_triu
```

Out[0]:

		0
Amina Flow	**Ore Pulp Density**	0.655788
Flotation Column 01 Air Flow	**Flotation Column 02 Air Flow**	0.848277
Flotation Column 03 Air Flow	0.954528	
Flotation Column 06 Air Flow	0.662963	
Flotation Column 07 Air Flow	0.646768	
Flotation Column 02 Air Flow	**Flotation Column 03 Air Flow**	0.860438
Flotation Column 03 Air Flow	**Flotation Column 06 Air Flow**	0.659506
Flotation Column 07 Air Flow	0.649435	
Flotation Column 06 Air Flow	**Flotation Column 07 Air Flow**	0.854067
Flotation Column 01 Level	**Flotation Column 02 Level**	0.717500
Flotation Column 03 Level	0.725294	
Flotation Column 02 Level	**Flotation Column 03 Level**	0.654044
Flotation Column 04 Level	**Flotation Column 05 Level**	0.682411
Flotation Column 07 Level	0.619252	
Flotation Column 05 Level	**Flotation Column 07 Level**	0.710699
Flotation Column 06 Level	**Flotation Column 07 Level**	0.606954

In [0]:
```
corr_triu = corr.where(~np.tril(np.ones(corr.shape)).astype(np.bool))
corr_triu = corr_triu.stack()
corr_triu_neg=corr_triu[corr_triu < -0.6].to_frame()
corr_triu_neg
```

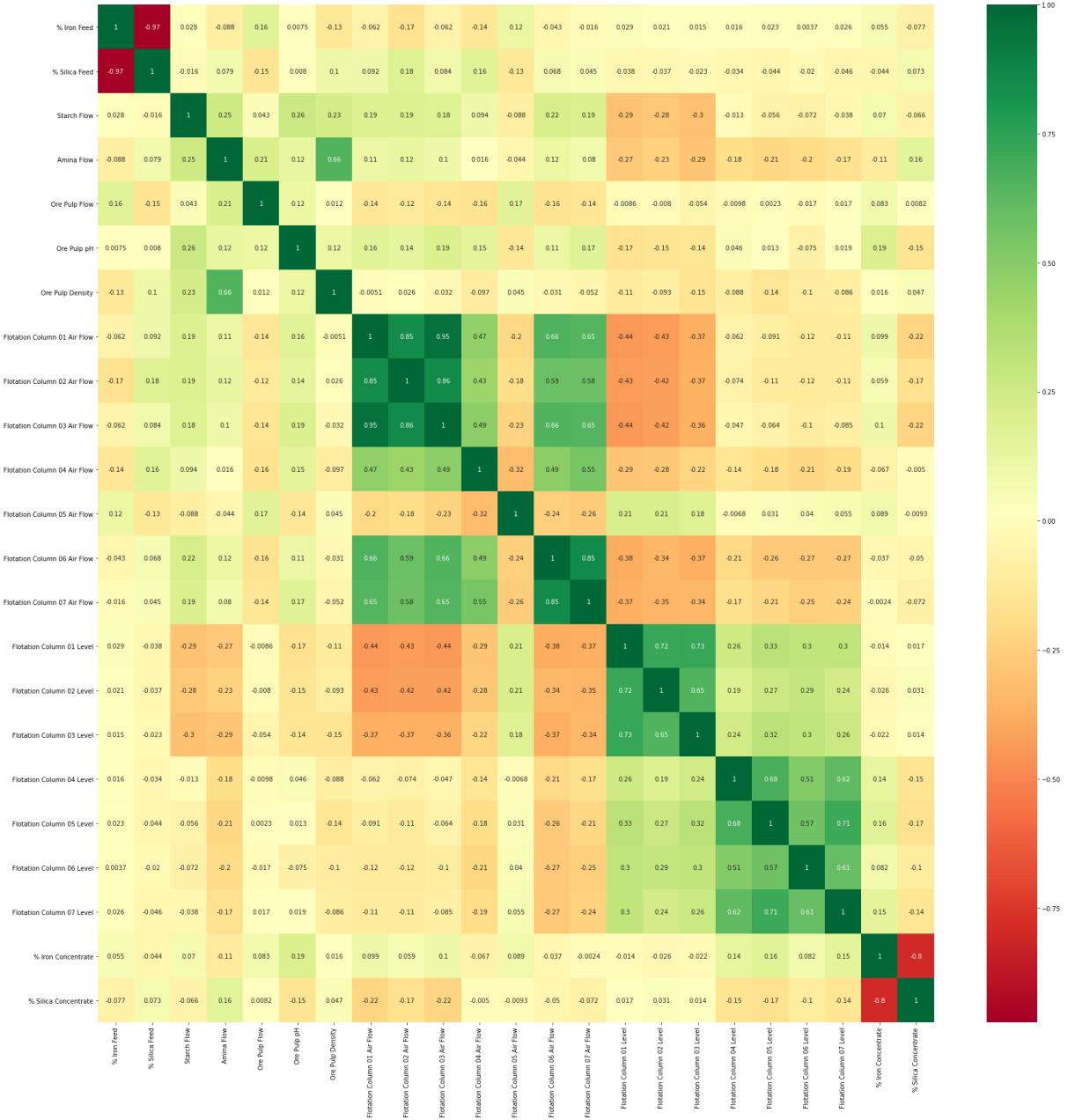
Out[0]:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

		0
% Iron Feed	% Silica Feed	-0.971833
% Iron Concentrate	% Silica Concentrate	-0.800560

In [0]:
```
plt.figure(figsize=(30, 30))
cor= df.corr()
corelation = sns.heatmap(cor, annot=True, cmap="RdYlGn")
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD

WE DONT TREAT THE CORRELATION FEATURES SINCE OUR TASK IS BASED ON CORRELATION ON FEATURES WE DONT ELIMINATE THEM SIMPLY .

WE TREAT THEM WHILE DOING MODELING OF DATA

Interactions

In [17]:
```
from sklearn.preprocessing import PolynomialFeatures
pf = PolynomialFeatures(degree=2, interaction_only=False,
                        include_bias=False)
res = pf.fit_transform(df.iloc[:,1:])
res
```

Out[17]:
```
array([[5.5200000e+01, 1.6980000e+01, 3.0195300e+03, ..., 4.4769481e+0
3,
        8.7652100e+01, 1.7161000e+00],
       [5.5200000e+01, 1.6980000e+01, 3.0244100e+03, ..., 4.4769481e+0
3,
        8.7652100e+01, 1.7161000e+00],
       [5.5200000e+01, 1.6980000e+01, 3.0434600e+03, ..., 4.4769481e+0
3,
        8.7652100e+01, 1.7161000e+00],
       ...,
       [4.9750000e+01, 2.3200000e+01, 2.6922000e+03, ..., 4.1306329e+0
3,
        1.0990170e+02, 2.9241000e+00],
       [4.9750000e+01, 2.3200000e+01, 1.1641200e+03, ..., 4.1306329e+0
3,
        1.0990170e+02, 2.9241000e+00],
       [4.9750000e+01, 2.3200000e+01, 1.1641200e+03, ..., 4.1306329e+0
3,
        1.0990170e+02, 2.9241000e+00]])
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [19]:
```
pd.DataFrame(pf.powers_, columns=['% Iron Feed', '% Silica Feed', 'Starch Flow', 'Amina Flow',
       'Ore Pulp Flow', 'Ore Pulp pH', 'Ore Pulp Density',
       'Flotation Column 01 Air Flow', 'Flotation Column 02 Air Flow',
       'Flotation Column 03 Air Flow', 'Flotation Column 04 Air Flow',
       'Flotation Column 05 Air Flow', 'Flotation Column 06 Air Flow',
       'Flotation Column 07 Air Flow', 'Flotation Column 01 Level',
       'Flotation Column 02 Level', 'Flotation Column 03 Level',
       'Flotation Column 04 Level', 'Flotation Column 05 Level',
       'Flotation Column 06 Level', 'Flotation Column 07 Level',
       '% Iron Concentrate', '% Silica Concentrate'])
```

Out[19]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotation Column 04 Air Flow
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
294	0	0	0	0	0	0	0	0	0	0	0
295	0	0	0	0	0	0	0	0	0	0	0
296	0	0	0	0	0	0	0	0	0	0	0
297	0	0	0	0	0	0	0	0	0	0	0
298	0	0	0	0	0	0	0	0	0	0	0

299 rows × 23 columns

DATA SPLITTING

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [0]:
```
y = df['% Silica Concentrate']
X = df.drop(['% Silica Concentrate'], axis=1)
```

In [0]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=30)
```

FEATURE SCALING

In [0]:
```
#assignment donors dataset-preprocessing
from sklearn.preprocessing import StandardScaler
scale_features_std = StandardScaler()
features_train = scale_features_std.fit_transform(X_train)
features_test = scale_features_std.transform(X_test)
```

In [0]:
```
print(X_train.shape,features_train.shape,X_test.shape,features_test.sha
pe)
```

```
(516217, 22) (516217, 22) (221236, 22) (221236, 22)
```

FEATURE ENGINEERING

Rounding

Often when dealing with continuous numeric attributes like proportions or percentages, we may not need the raw values having a high amount of precision. Hence it often makes sense to round off these high precision percentages into numeric integers.The data is being converted to proper percentages

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

2017-03-10 01:00:00	55,2	16,98	3019,53	557,434
2017-03-10 01:00:00	55,2	16,98	3024,41	563,965
2017-03-10 01:00:00	55,2	16,98	3043,46	568,054
2017-03-10 01:00:00	55,2	16,98	3047,36	568,665
2017-03-10 01:00:00	55,2	16,98	3033,69	558,167

THE RAW DATA IS SHOWN ABOVE

THE ROUND OFF DATA IS SHOWN BELOW

	index	datetime hours	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotation Column 04 Air Flow	Flotation Column 05 Air Flow	Flotation Column 06 Air Flow	Flotation Column 07 Air Flow	Flotation Column 01 Level	Flotation Column 02 Level	Flotat Column Le
0	2017-03-10 01:02:00	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	250.576	295.096	306.4	250.225	250.884	457.396	432.962	424.
1	2017-03-10 01:02:20	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	250.862	295.096	306.4	250.137	248.994	451.891	429.560	432.

INTERACTIONS

Machine learning models usually try to model the output responses (discrete classes or continuous values) as a function of the input feature variables.

Our interaction dataframe is obtained and we look upon the main columns and we are able to see most of the columns are having high interactions

IMPUTATION OF VALUES

ONE HR = 3600 SECS

SAMPLES AT 20 SECONDS

LET US HAVE ONE RECORD AT END OF 20 SECS

SO, 3600/20 = 180

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

WHICH MEANS WE GET 180 RECORDS FOR ONE HOUR SO WE NEED TO WE FIND THE NUMBER OF RECORDS = 180 AND IF IT EQUAL WE ARE LUCKY TO COCNLUDE THERE IS NO VALUE ELSE WE NEED TO FILL THOSE VALUES

HENCE, WE ARE IMPUTED THE VALUES.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD