bccl_analysis

January 15, 2019

```
In [166]: import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import seaborn as sns
          import scipy
          import datetime
          import time
          import random
          import json
          import pickle
```

```
In [18]: cd ../backup/mongodb_backup/
```

```
[Errno 2] No such file or directory: '../backup/mongodb_backup/'
/home/revant/Desktop/scraping2/bccl/backup/mongodb_backup
```

```
In [167]: ls -al
```

```
total 648128
drwxrwxr-x 2 revant revant      4096 Jan  9 22:50 ./
drwxr-xr-x 6 revant revant      4096 Jan  5 16:24 ../
-rw-rw-r-- 1 revant revant 215488704 Jan  6 17:44 daily_dispatch_summary_04-01-18_2.json
-rw-rw-r-- 1 revant revant 100980713 Jan  5 16:34 daily_dispatch_summary_04-01-18.json
-rw-r--r-- 1 revant revant 308886730 Jan  6 18:21 daily_dispatch_summary_manual_with_dates_04-0
-rw-rw-r-- 1 revant revant         0 Jan  7 01:44 dispatch_against_daily_sales_04-01.json
-rw-rw-r-- 1 revant revant   1208285 Jan  5 16:26 loading_schedules_04-01-18.json
-rw-r--r-- 1 revant revant   2367006 Jan  6 18:23 loading_schedules_manual_with_dates_04-01-18
-rw-r--r-- 1 revant revant     40316 Jan  9 22:51 required2.json
-rw-r--r-- 1 revant revant   1063942 Jan 10 18:57 required.json
-rw-r--r-- 1 revant revant  13383713 Jan  6 16:52 sales_order_data_manual_with_dates_04-01.pic
-rw-rw-r-- 1 revant revant   9342489 Jan  5 16:25 sales_order_details_04-01-18.json
-rw-rw-r-- 1 revant revant  10653954 Jan  9 22:37 sales_order_details_08-01-18.json
-rw-r--r-- 1 revant revant    230895 Jan  6 20:01 sales_order_nums.pickle
```

```
In [168]: with open("sales_order_details_08-01-18.json") as file:
              f = file.readlines()
```

```
In [169]: json_data = [json.loads(i) for i in f]

In [170]: for i, data in enumerate(json_data):
              json_data[i]['Sales Order Date'] = datetime.datetime.fromtimestamp(int(json_data
              json_data[i]['Sales Order Expiry Date'] = datetime.datetime.fromtimestamp(int(js

In [171]: df = pd.DataFrame.from_dict(json_data)

In [172]: df.drop(columns = ['_id', 'Sales Order No'], inplace = True)

In [173]: df.head()
```

Out[173]:

	Consignee ID	Consignee Name
0	220917	LAXMI HARD COKE MFG.CO.
1	212968	AMBIKA HARD COKE MFG. CO.
2	213530	RAHUL COKE PRIVATE LTD
3	220376	SHREE DURGA HARD COKE MFG. CO.
4	220882	O S D COKE PVT.LTD.

	Grade&Size(Product Description)	Sales Order Date	Sales Order Expiry Date
0	Washery-3 STM/FSA	2018-06-29 05:30:00	2018-08-15 05:30:00
1	Washery-3 STM/FSA	2018-06-29 05:30:00	2018-08-15 05:30:00
2	Washery-3 STM/FSA	2018-06-29 05:30:00	2018-08-15 05:30:00
3	Washery-3 STM/FSA	2018-06-29 05:30:00	2018-08-15 05:30:00
4	Washery-3 STM/FSA	2018-06-29 05:30:00	2018-08-15 05:30:00

	Sales Order Quantity	Scheme(FSA/Auction etc)	area_id	area_name
0	52.46	COKERY	01	Barora Area
1	31.08	COKERY	01	Barora Area
2	72.84	COKERY	01	Barora Area
3	23.22	COKERY	01	Barora Area
4	71.84	COKERY	01	Barora Area

	colliery_id	colliery_name
0	0106	Phularitand
1	0106	Phularitand
2	0106	Phularitand
3	0106	Phularitand
4	0106	Phularitand

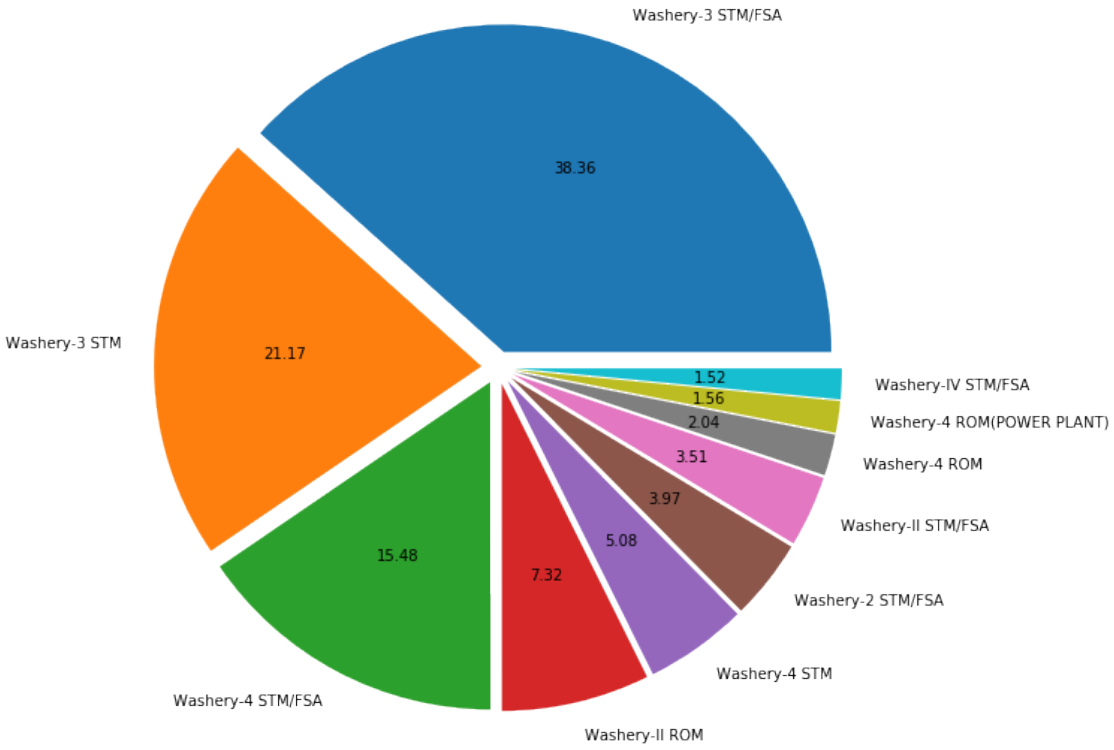
0.1 Top 10 Most Used Types Of Coal

```
In [35]: df['Grade&Size(Product Description)'].value_counts()[:10]
```

```
Out[35]: Washery-3 STM/FSA            7648
         Washery-3 STM                4222
         Washery-4 STM/FSA            3086
         Washery-II ROM               1459
         Washery-4 STM                1012
```

```
Washery-2 STM/FSA                791
Washery-II STM/FSA               700
Washery-4 ROM                    406
Washery-4 ROM(POWER PLANT)       312
Washery-IV STM/FSA               304
Name: Grade&Size(Product Description), dtype: int64
```

In [183]:
```
fig = plt.figure(figsize = (8, 8))
plt.pie(
            list(df['Grade&Size(Product Description)'].value_counts()[:10].values),
            labels = list(df['Grade&Size(Product Description)'].value_counts()[:10].
            explode = [0.05]*10,
            autopct = '%.2f'
        )
#         startangle = 90)
# plt.axis('equal')
# plt.
plt.tight_layout()
plt.show()
```

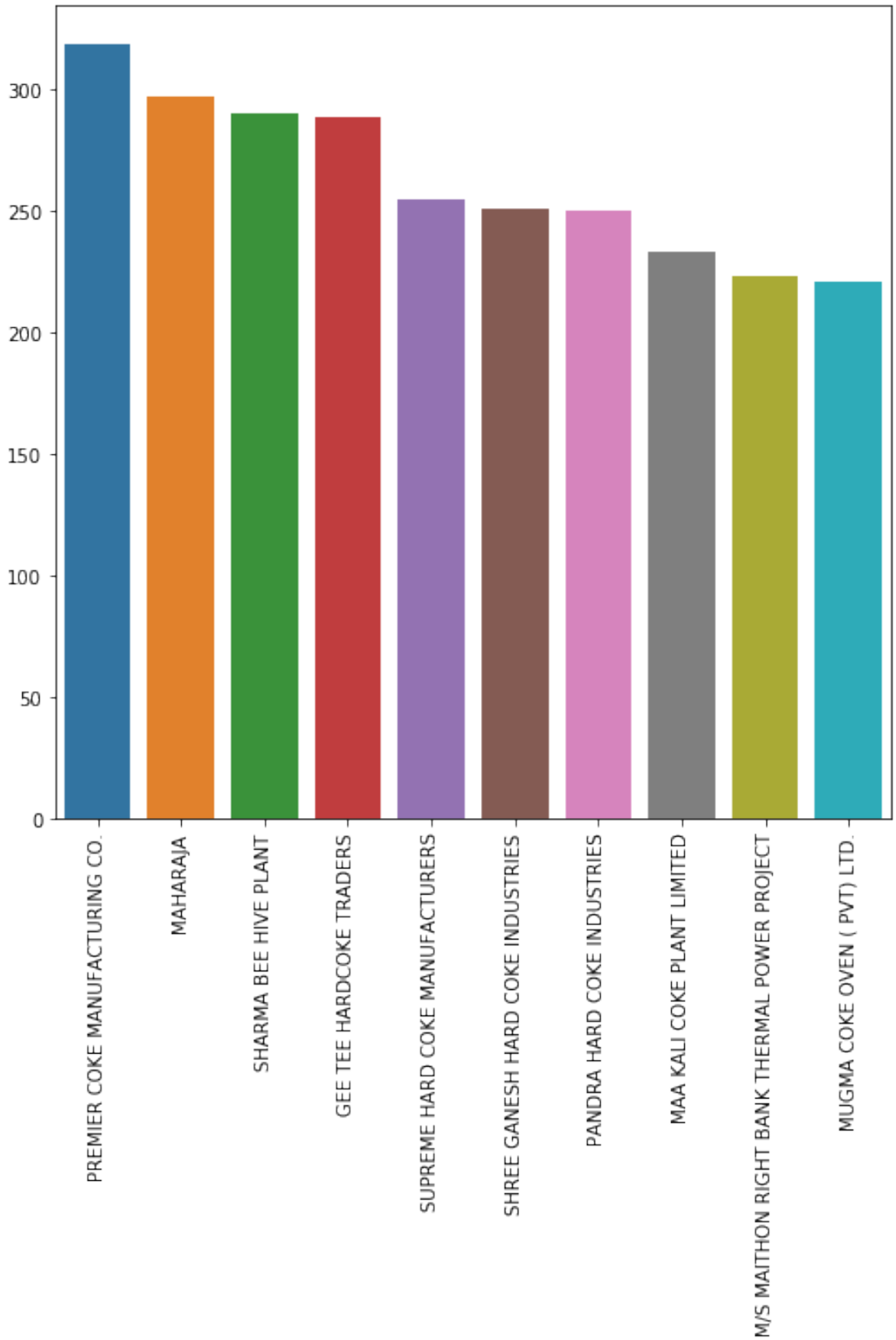


0.2 Top 10 Consignees

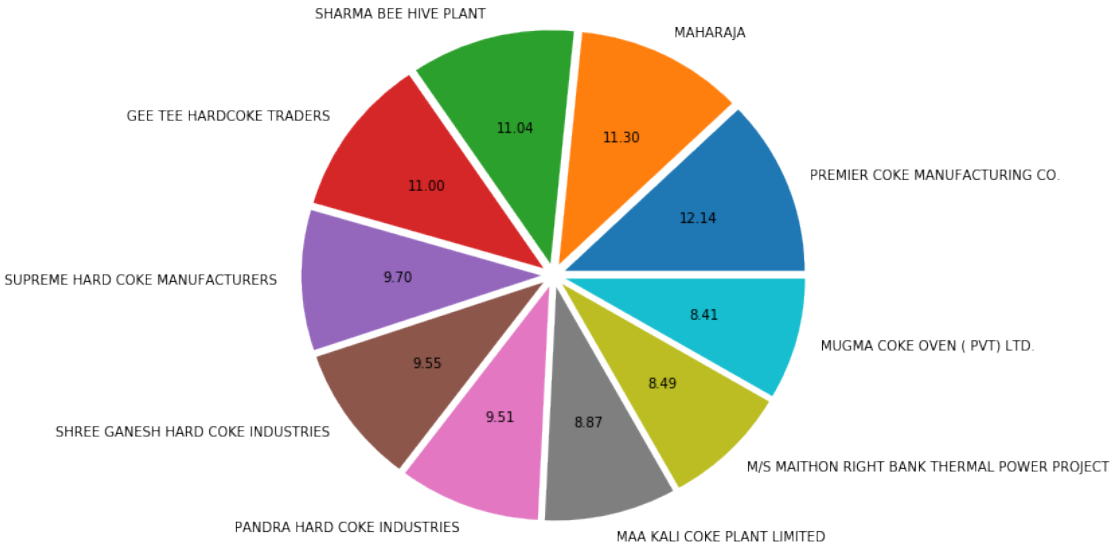
```
In [54]: df['Consignee Name'].value_counts()[:10]
```

```
Out[54]: PREMIER COKE MANUFACTURING CO.                  319
         MAHARAJA                                       297
         SHARMA BEE HIVE PLANT                          290
         GEE TEE HARDCOKE TRADERS                       289
         SUPREME HARD COKE MANUFACTURERS                255
         SHREE GANESH HARD COKE INDUSTRIES              251
         PANDRA HARD COKE INDUSTRIES                    250
         MAA KALI COKE PLANT LIMITED                    233
         M/S MAITHON RIGHT BANK THERMAL POWER PROJECT   223
         MUGMA COKE OVEN ( PVT) LTD.                    221
         Name: Consignee Name, dtype: int64
```

```
In [122]: fig = plt.figure(figsize = (8, 8))

          bar_plot = sns.barplot(
                      y = list(df['Consignee Name'].value_counts()[:10].values),
                      x = list(df['Consignee Name'].value_counts()[:10].index),
                 )

          bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation = 90)

          plt.show()
```


300
250
200
150
100
50
0
PREMIER COKE MANUFACTURING CO.
MAHARAJA
SHARMA BEE HIVE PLANT
GEE TEE HARDCOKE TRADERS
SUPREME HARD COKE MANUFACTURERS
SHREE GANESH HARD COKE INDUSTRIES
PANDRA HARD COKE INDUSTRIES
MAA KALI COKE PLANT LIMITED
M/S MAITHON RIGHT BANK THERMAL POWER PROJECT
MUGMA COKE OVEN (PVT) LTD.

```
In [109]: fig = plt.figure(figsize = (8, 8))
          plt.pie(
                      list(df['Consignee Name'].value_counts()[:10].values),
                      labels = list(df['Consignee Name'].value_counts()[:10].index),
                      explode = [0.05]*10,
                      autopct = '%.2f'
                  )

          plt.show()
```



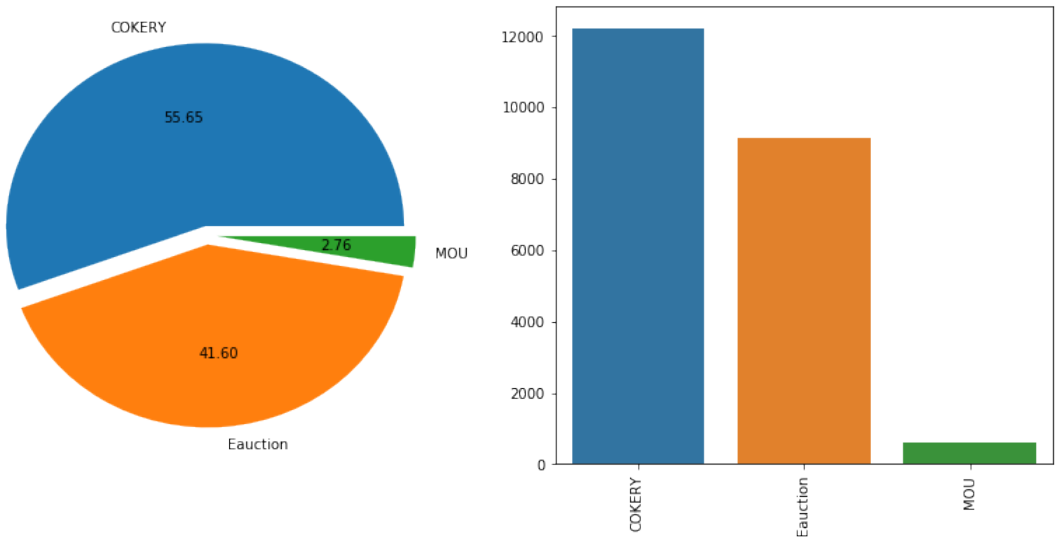
0.3 Scheme (FSA/Auction etc)

```
In [126]: df['Scheme(FSA/Auction etc)'].value_counts()
```

```
Out[126]: COKERY      12208
          Eauction     9126
          MOU           605
          Name: Scheme(FSA/Auction etc), dtype: int64
```

```
In [143]: fig = plt.figure(figsize = (14, 6))
          fig.add_subplot(1, 2, 1)
          plt.pie(
                      df['Scheme(FSA/Auction etc)'].value_counts().values,
                      labels = df['Scheme(FSA/Auction etc)'].value_counts().index,
                      explode = [0.05]*3,
                      autopct = '%.2f'
```

```
        )

fig.add_subplot(1, 2, 2)
bar_plot = sns.barplot(
            y = list(df['Scheme(FSA/Auction etc)'].value_counts().values),
            x = list(df['Scheme(FSA/Auction etc)'].value_counts().index),
       )

bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation = 90)

plt.show()
```



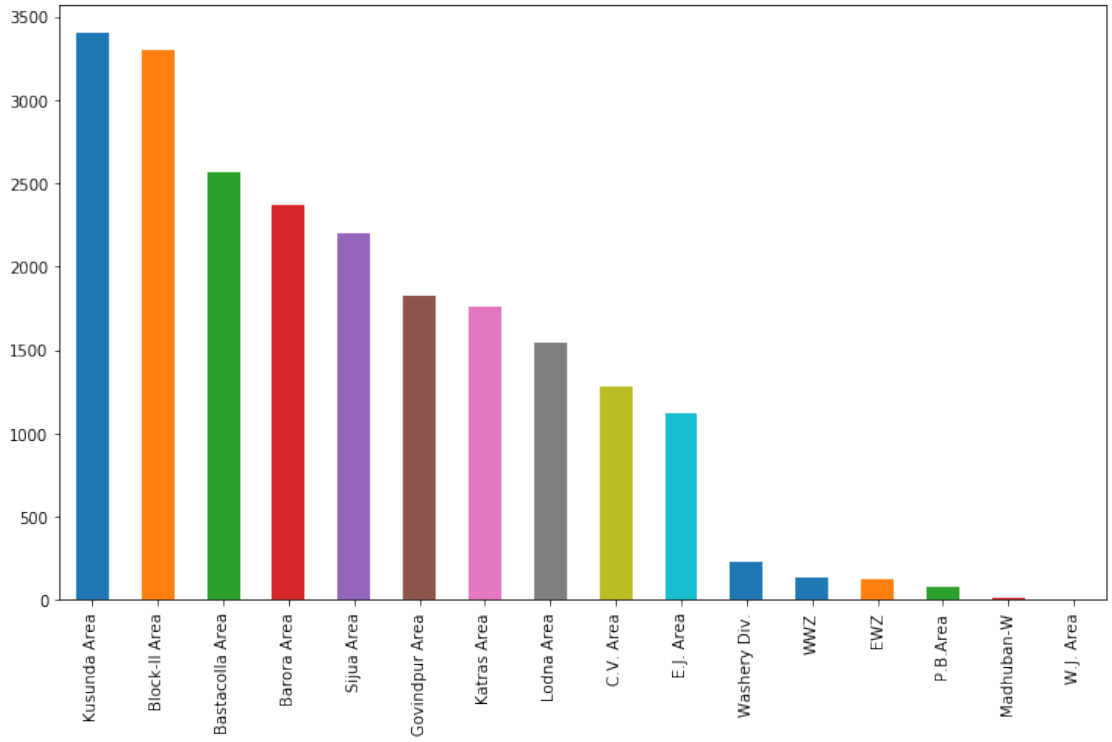
0.4 Top Areas

```
In [158]: df['area_name'].value_counts()
```

```
Out[158]: Kusunda Area       3402
          Block-II Area      3297
          Bastacolla Area    2568
          Barora Area        2373
          Sijua Area         2196
          Govindpur Area     1824
          Katras Area        1760
          Lodna Area         1539
          C.V. Area          1284
          E.J. Area          1121
          Washery Div.        225
          WWZ                 132
          EWZ                 123
```

```
P.B.Area              78
Madhuban-W            14
W.J. Area              3
Name: area_name, dtype: int64
```

```
In [157]: fig = plt.figure(figsize = (12, 7))
          df['area_name'].value_counts().plot(kind = 'bar')
          plt.show()
```

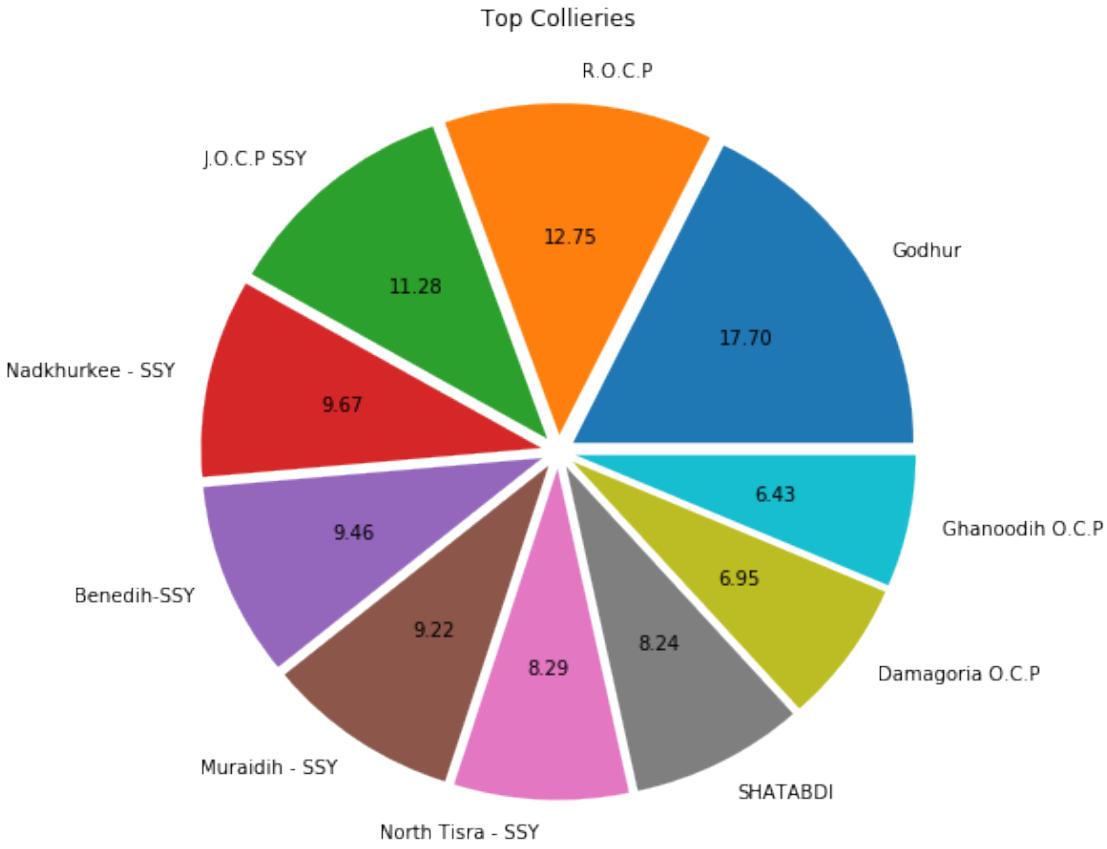


0.5 Top Collieries

```
In [160]: df['colliery_name'].value_counts()[:10]
```

```
Out[160]: Godhur              1919
          R.O.C.P             1383
          J.O.C.P SSY         1223
          Nadkhurkee - SSY    1048
          Benedih-SSY         1026
          Muraidih - SSY      1000
          North Tisra - SSY    899
          SHATABDI             894
          Damagoria O.C.P      754
          Ghanoodih O.C.P      697
          Name: colliery_name, dtype: int64
```

```
In [162]: fig = plt.figure(figsize = (8, 8))
          plt.pie(
                      list(df['colliery_name'].value_counts()[:10].values),
                      labels = list(df['colliery_name'].value_counts()[:10].index),
                      explode = [0.05]*10,
                      autopct = '%.2f'
                  )

          plt.title('Top Collieries')
          plt.show()
```



1 Analysing for all MOUs

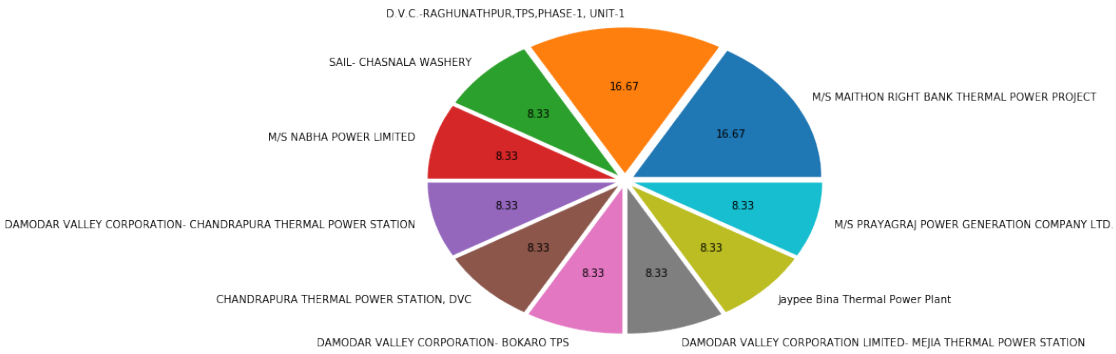
```
In [179]: df2 = df[df['Scheme(FSA/Auction etc)'] == "MOU"]
```

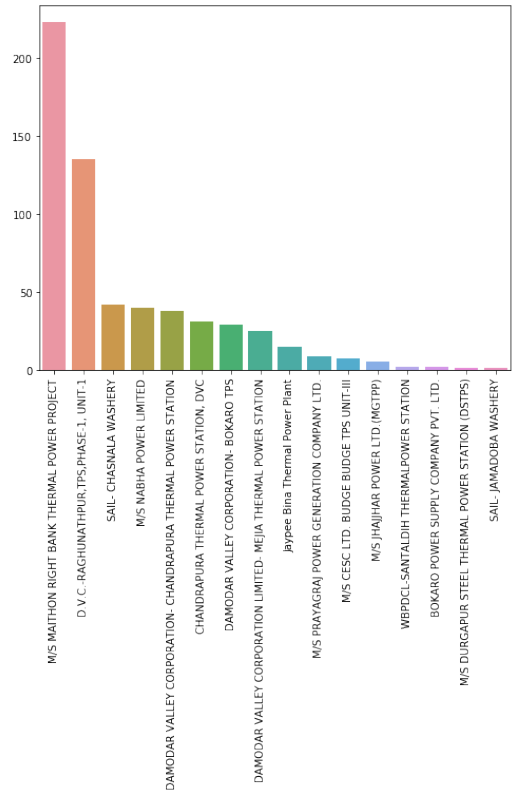
1.1 Top MOU Consignees

```
In [186]: df2['Consignee Name'].value_counts()
```

Out[186]:
```
M/S MAITHON RIGHT BANK THERMAL POWER PROJECT                          223
D.V.C.-RAGHUNATHPUR,TPS,PHASE-1, UNIT-1                              135
SAIL- CHASNALA WASHERY                                                42
M/S NABHA POWER LIMITED                                               40
DAMODAR VALLEY CORPORATION- CHANDRAPURA THERMAL POWER STATION          38
CHANDRAPURA THERMAL POWER STATION, DVC                                31
DAMODAR VALLEY CORPORATION- BOKARO TPS                                29
DAMODAR VALLEY CORPORATION LIMITED- MEJIA THERMAL POWER STATION       25
Jaypee Bina Thermal Power Plant                                       15
M/S PRAYAGRAJ POWER GENERATION COMPANY LTD.                           9
M/S CESC LTD. BUDGE BUDGE TPS UNIT-III                                7
M/S JHAJJHAR POWER LTD.(MGTPP)                                        5
WBPDCL-SANTALDIH THERMALPOWER STATION                                 2
BOKARO POWER SUPPLY COMPANY PVT. LTD.                                 2
M/S DURGAPUR STEEL THERMAL POWER STATION (DSTPS)                      1
SAIL- JAMADOBA WASHERY                                                1
Name: Consignee Name, dtype: int64
```

In [193]:
```
fig = plt.figure(figsize = (8, 14))
fig.add_subplot(2, 1, 1)
plt.pie(
            df2['Consignee Name'].value_counts().value_counts()[:10].values,
            labels = list(df2['Consignee Name'].value_counts()[:10].index),
            explode = [0.05]*10,
            autopct = '%.2f'
        )
# plt.

fig.add_subplot(2, 1, 2)
bar_plot = sns.barplot(
            y = list(df2['Consignee Name'].value_counts().values),
            x = list(df2['Consignee Name'].value_counts().index),
       )

bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation = 90)

plt.show()
```



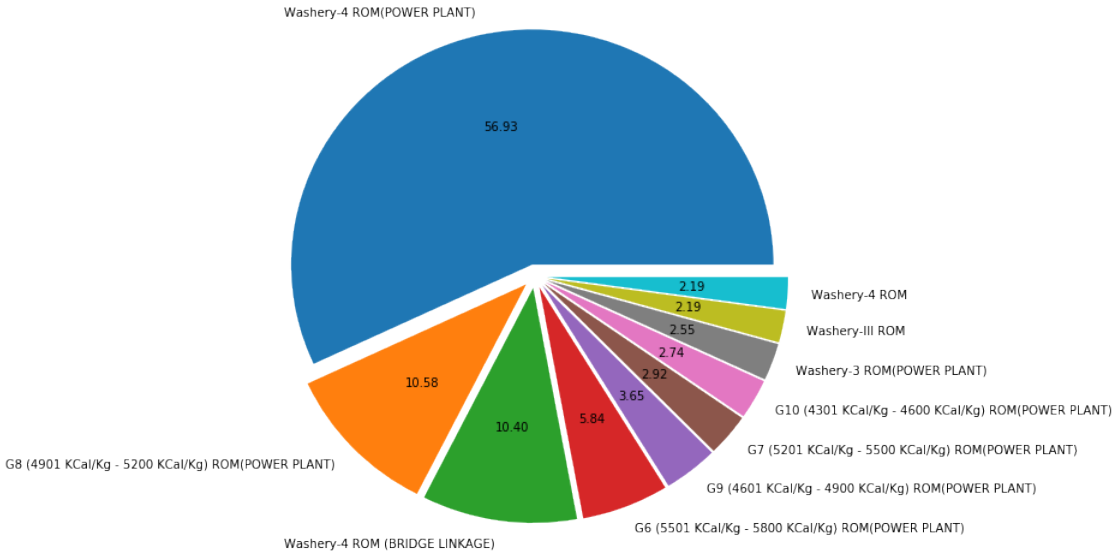


1.2 Grade&Size(Product Description) for all the MOUs

```
In [180]: df2['Grade&Size(Product Description)'].value_counts()

Out[180]: Washery-4 ROM(POWER PLANT)                              312
          G8 (4901 KCal/Kg - 5200 KCal/Kg) ROM(POWER PLANT)        58
          Washery-4 ROM (BRIDGE LINKAGE)                           57
          G6 (5501 KCal/Kg - 5800 KCal/Kg) ROM(POWER PLANT)        32
          G9 (4601 KCal/Kg - 4900 KCal/Kg) ROM(POWER PLANT)        20
          G7 (5201 KCal/Kg - 5500 KCal/Kg) ROM(POWER PLANT)        16
```

```
G10 (4301 KCal/Kg - 4600 KCal/Kg) ROM(POWER PLANT)      15
Washery-3 ROM(POWER PLANT)                              14
Washery-III ROM                                         12
Washery-4 ROM                                           12
Steel Grade II ROM                                      10
Washery-II ROM (PREIMUM15PER)                           10
Washery-4 ROM (PREIMUM15PER)                             8
Washery-4   ROM(POWER PLANT)                             7
AT 34% ASH WASHED POWER COAL(WPC)                        6
Washery-II ROM                                           5
Washery-4 ROM(CTPS)                                      4
G8 (4901 KCal/Kg - 5200 KCal/Kg)   ROM(POWER PLANT)      3
Washery-4   ROM (BRIDGE LINKAGE)                         2
G7 (5201 KCal/Kg - 5500 KCal/Kg)   ROM(POWER PLANT)      1
G6 (5501 KCal/Kg - 5800 KCal/Kg)   ROM(POWER PLANT)      1
Name: Grade&Size(Product Description), dtype: int64
```

In [182]:
```
fig = plt.figure(figsize = (9, 9))
plt.pie(
            list(df2['Grade&Size(Product Description)'].value_counts()[:10].values),
            labels = list(df2['Grade&Size(Product Description)'].value_counts()[:10]
            explode = [0.05]*10,
            autopct = '%.2f'
        )

plt.show()
```

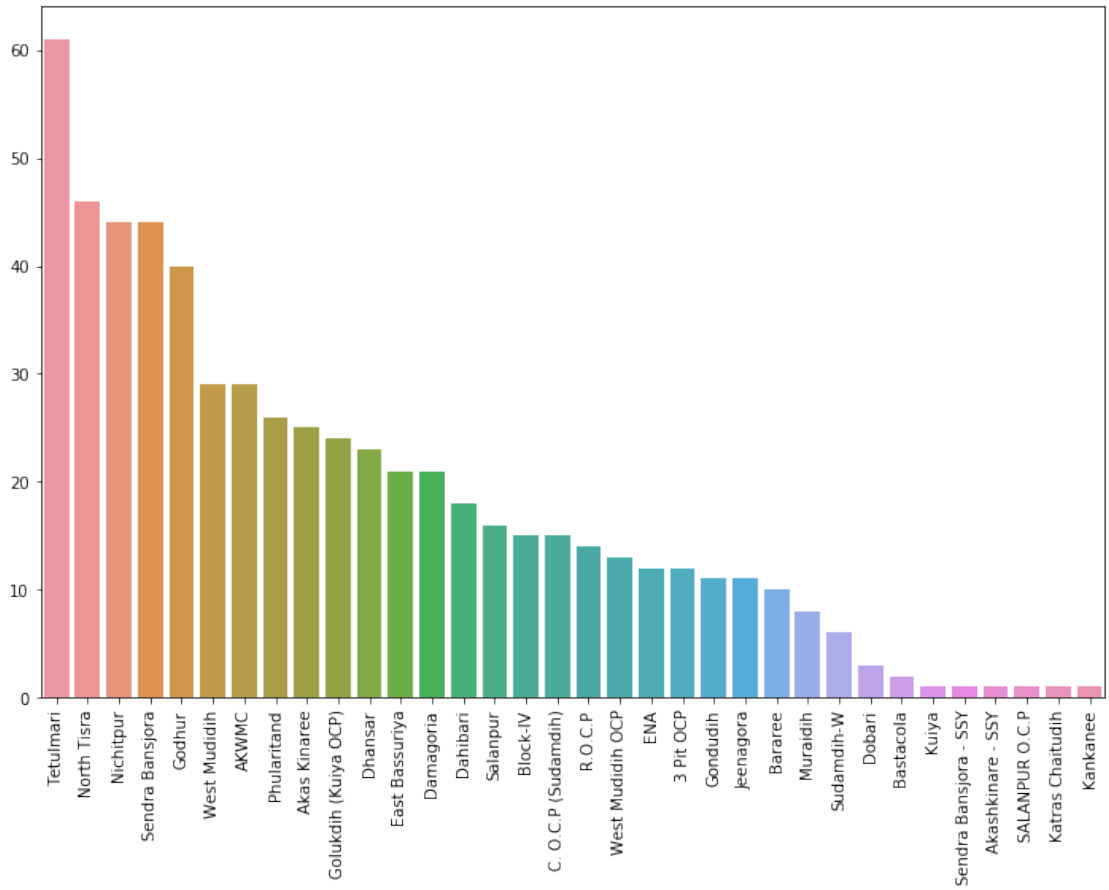


2 Top Areas of operation for MOUs

In [196]:
```
df2['colliery_name'].value_counts()
```

Out[196]:
```
Tetulmari                61
North Tisra              46
Nichitpur                44
Sendra Bansjora          44
Godhur                   40
West Mudidih             29
AKWMC                    29
Phularitand              26
Akas Kinaree             25
Golukdih (Kuiya OCP)     24
Dhansar                  23
East Bassuriya           21
Damagoria                21
Dahibari                 18
Salanpur                 16
Block-IV                 15
C. O.C.P (Sudamdih)      15
R.O.C.P                  14
West Mudidih OCP         13
ENA                      12
3 Pit OCP                12
Gondudih                 11
Jeenagora                11
Bararee                  10
Muraidih                  8
Sudamdih-W                6
Dobari                    3
Bastacola                 2
Kuiya                     1
Sendra Bansjora - SSY     1
Akashkinare - SSY         1
SALANPUR O.C.P            1
Katras Chaitudih          1
Kankanee                  1
Name: colliery_name, dtype: int64
```

In [200]:
```
fig = plt.figure(figsize = (12, 8))
bar_plot = sns.barplot(
            y = list(df2['colliery_name'].value_counts().values),
            x = list(df2['colliery_name'].value_counts().index),
       )

bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation = 90)

plt.show()
```




```
In [212]: df2['Sales Order Date Month'] = df2['Sales Order Date'].apply(lambda x : datetime.dat

/home/revant/.local/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  """Entry point for launching an IPython kernel.
```