Sample Notebook (Visualization)

This is a sample notebook for visualization of complex data

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

In [2]:
```
# Client Request Description :
# I have data for the accuracy scores of 5 algorithms.
# However, these scores have been measured in regard to different groups for
# For example, Algorithm 1 has 10 different scores one for each group from 1
```

In [3]:
```
# create a Python dictionary with different array lengths
accuracies_dict = dict(algo1=np.array([random.random() for i in range(10)]),
                       algo2=random.random(),
                       algo3=np.array([random.random() for i in range(6)]),
                       algo4=np.array([random.random() for i in range(2)]),
                       algo5=np.array([random.random() for i in range(8)]),
                      )
```

In [4]:
```
# create a Python dictionary to help up structure the data and present it
df = pd.DataFrame(dict([(key, pd.Series(value)) for key, value in accuracies
```

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

Out[5]:

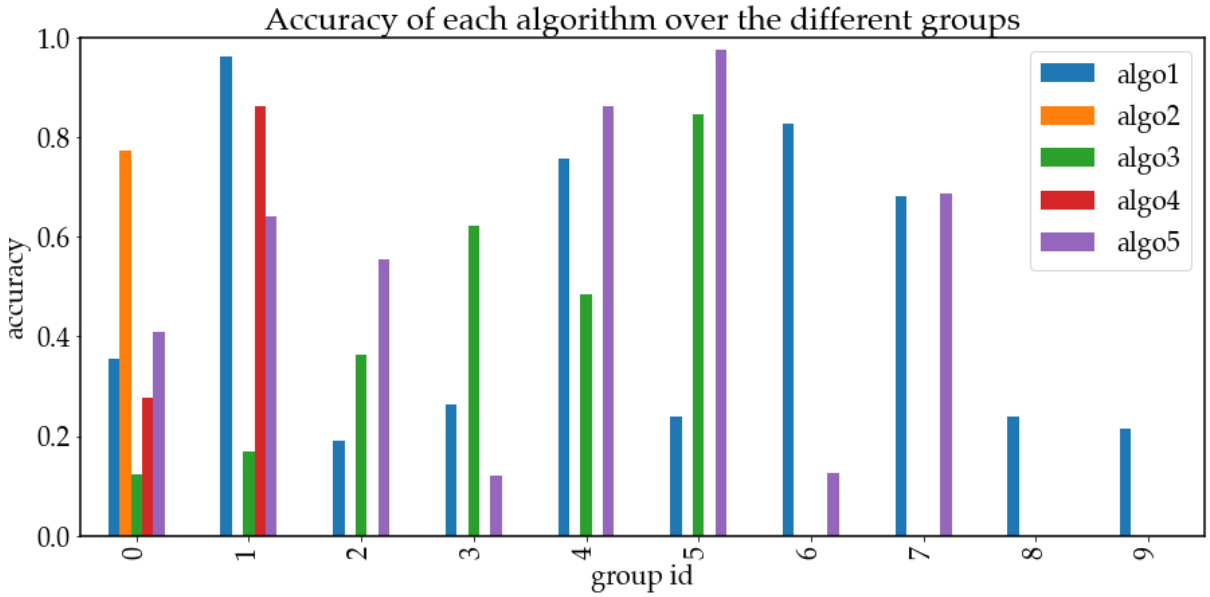
	algo1	algo2	algo3	algo4	algo5
0	0.355836	0.772782	0.121968	0.275737	0.407837
1	0.963359	NaN	0.170168	0.861072	0.639884
2	0.191565	NaN	0.364531	NaN	0.555246
3	0.262245	NaN	0.621989	NaN	0.121748
4	0.758031	NaN	0.484451	NaN	0.862486
5	0.238242	NaN	0.846191	NaN	0.976511
6	0.826730	NaN	NaN	NaN	0.126437
7	0.681022	NaN	NaN	NaN	0.686365
8	0.238887	NaN	NaN	NaN	NaN
9	0.214440	NaN	NaN	NaN	NaN

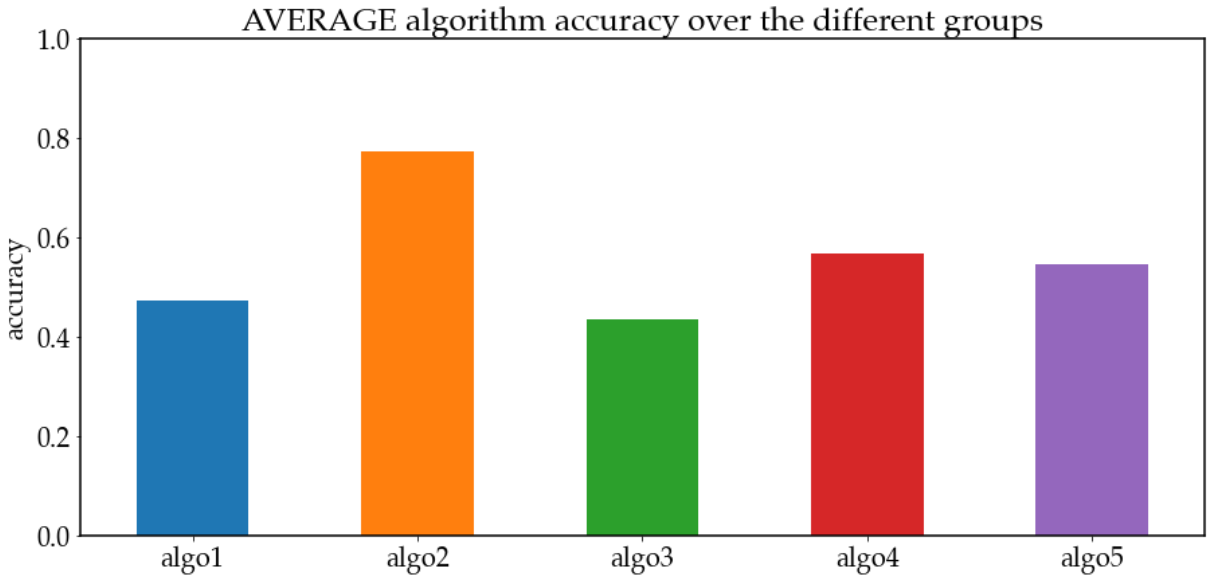
In [6]:
```
# font asthetics
def set_font():
    font = {'family' : 'Palatino',
```

```
            'weight' : 'bold',
            'size'   : 20}
    axes = {'linewidth':1.5}

    matplotlib.rc('font', **font)
    matplotlib.rc('axes', **axes)
    #matplotlib.rc('text', usetex=True
```

In [7]:
```
set_font()
#matplotlib.style.use('ggplot')
```

In [8]:
```
f,axs = plt.subplots(2,1,figsize=(15,15))

# matplotlib default colors
prop_cycle = plt.rcParams['axes.prop_cycle']
colors = prop_cycle.by_key()['color']

# Plot the accuracy of each algorithm over the different groups
df.plot(kind='bar',ax=axs[0])
axs[0].set_xlabel('group id')
axs[0].set_ylabel('accuracy')
axs[0].set_title('Accuracy of each algorithm over the different groups');
axs[0].set_ylim([0,1])

# Plot the AVERAGE algorithm accuracy over the different groups
df.apply(lambda row : row.mean()).plot(kind='bar',ax=axs[1],color=colors)
axs[1].set_ylabel('accuracy')
axs[1].set_title('AVERAGE algorithm accuracy over the different groups');
axs[1].set_ylim([0,1])

plt.xticks(rotation=0);
```


Accuracy of each algorithm over the different groups
accuracy
group id
algo1
algo2
algo3
algo4
algo5
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
7
8
9


AVERAGE algorithm accuracy over the different groups
accuracy
algo1
algo2
algo3
algo4
algo5
0.0
0.2
0.4
0.6
0.8
1.0