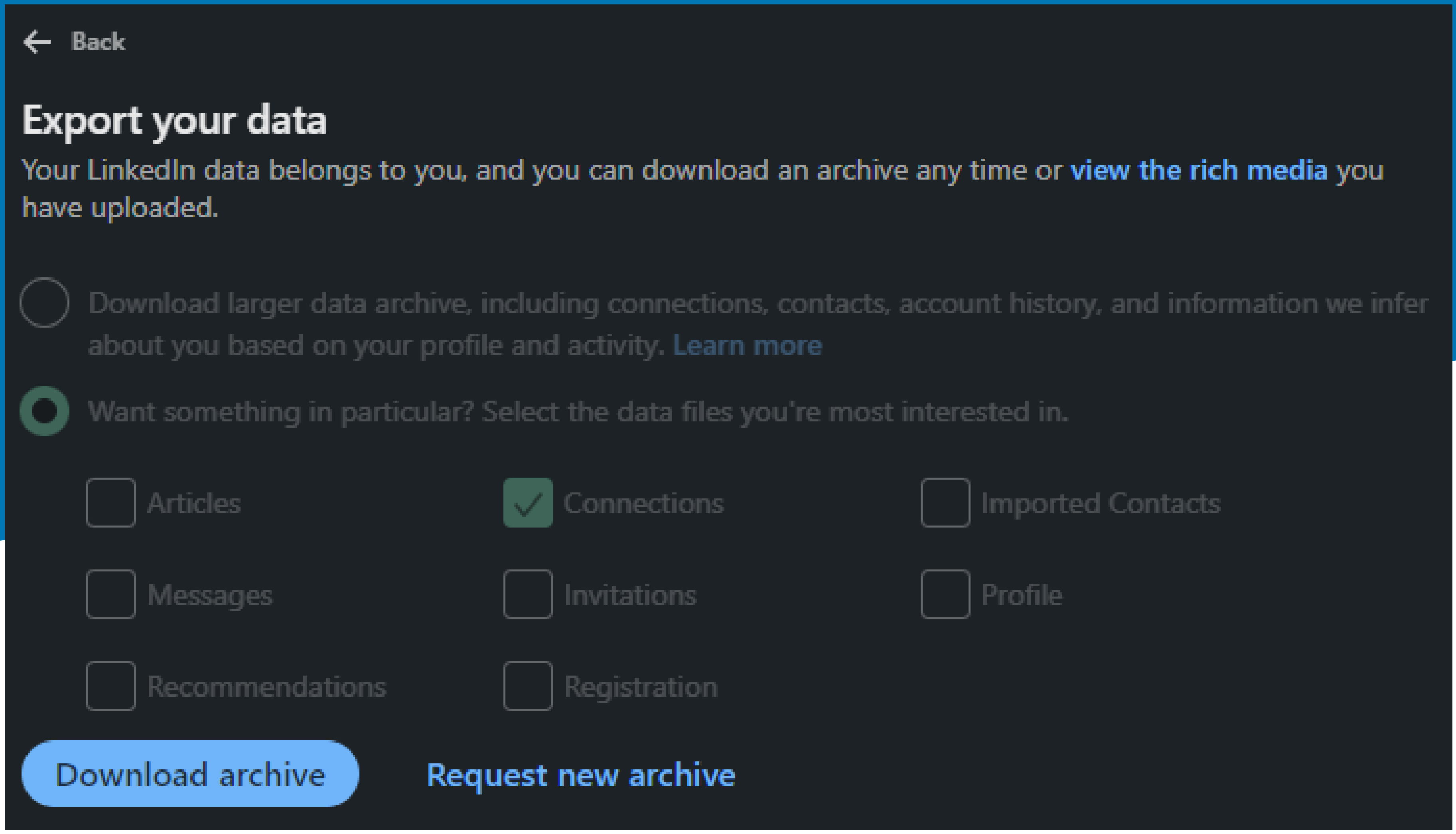
Analyse my LinkedIN network using Python

Aaya DINIA

You can request and download your data directly from LinkedIn :

Me → *Settings & Privacy → Data Privacy → Get a copy of your data*



← Back

Export your data

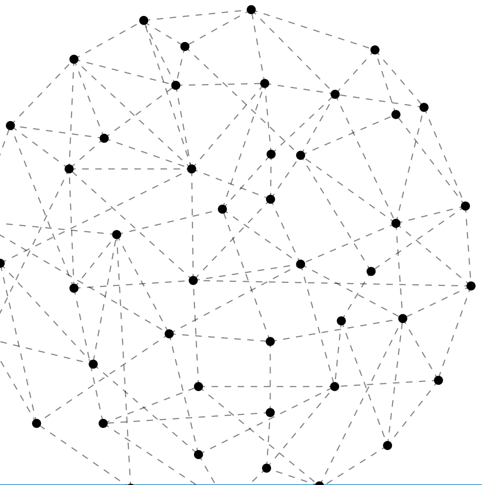
Your LinkedIn data belongs to you, and you can download an archive any time or **view the rich media** you have uploaded.

○ Download larger data archive, including connections, contacts, account history, and information we infer about you based on your profile and activity. Learn more

● Want something in particular? Select the data files you're most interested in.

□ Articles
□ Messages
□ Recommendations
☑ Connections
□ Invitations
□ Registration
□ Imported Contacts
□ Profile

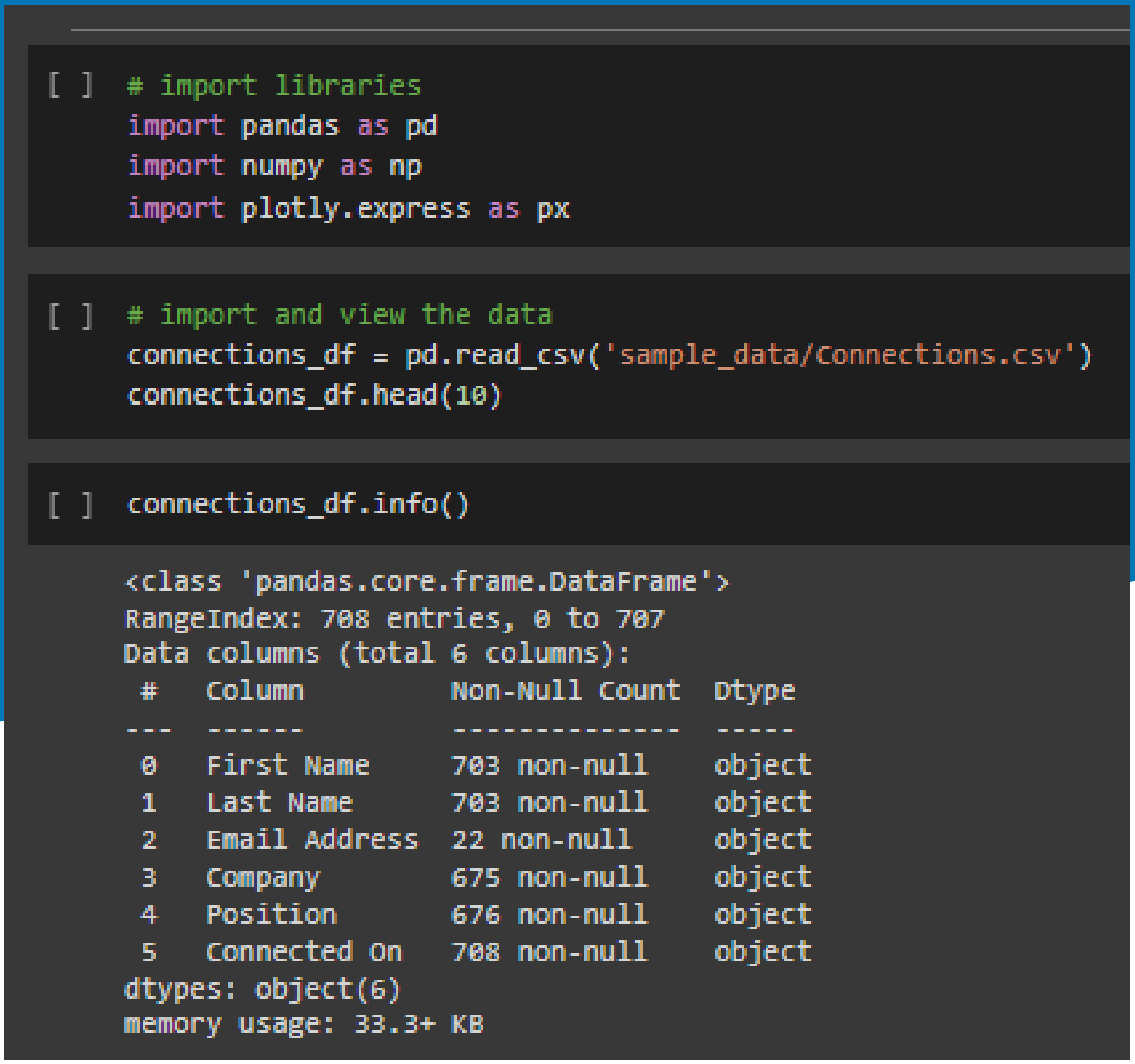
Data



Aaya DINIA

Specifically, I imported the connections data.

we will now import and check the data :



```
[ ]  # import libraries
     import pandas as pd
     import numpy as np
     import plotly.express as px
```

```
[ ]  # import and view the data
     connections_df = pd.read_csv('sample_data/Connections.csv')
     connections_df.head(10)
```

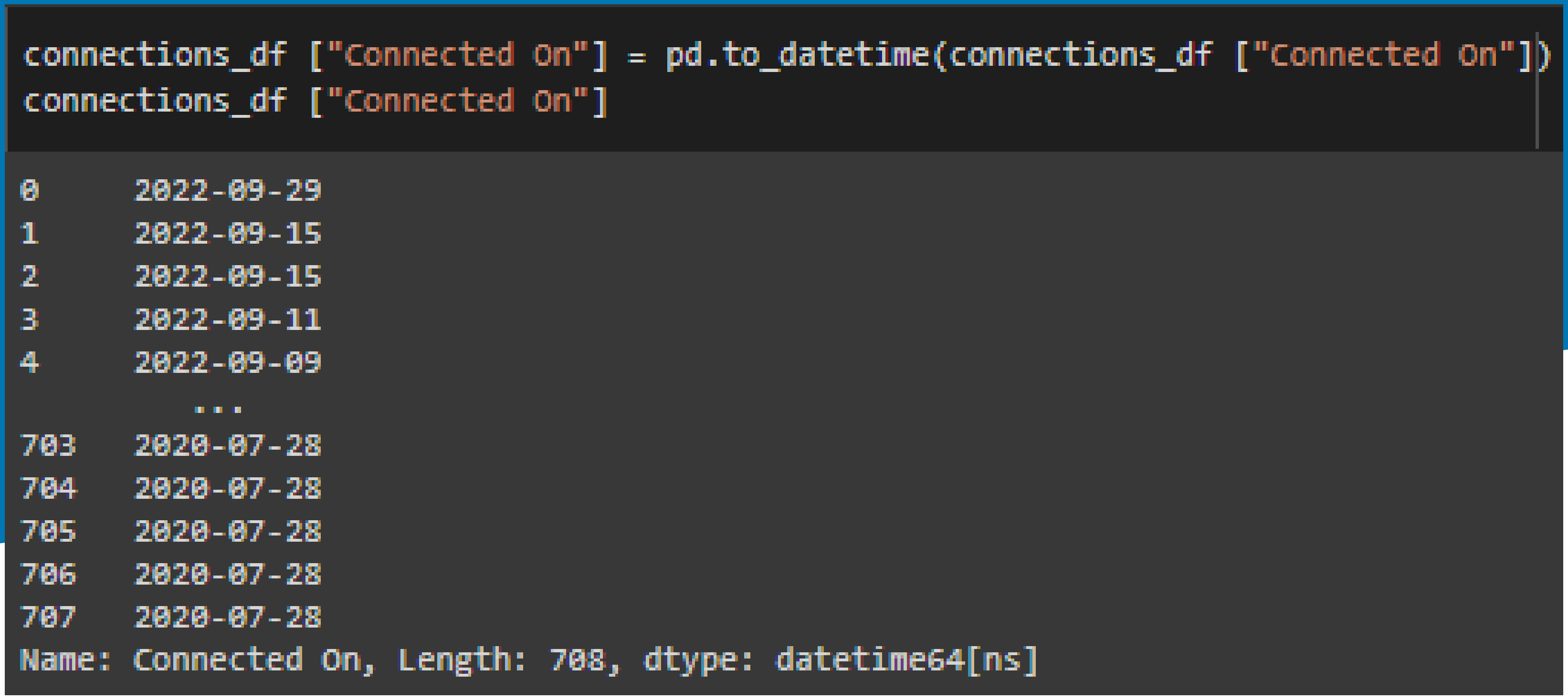
```
[ ]  connections_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 708 entries, 0 to 707
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   First Name      703 non-null    object
 1   Last Name       703 non-null    object
 2   Email Address   22 non-null     object
 3   Company         675 non-null    object
 4   Position        676 non-null    object
 5   Connected On    708 non-null    object
dtypes: object(6)
memory usage: 33.3+ KB
```

Data

Aaya DINIA

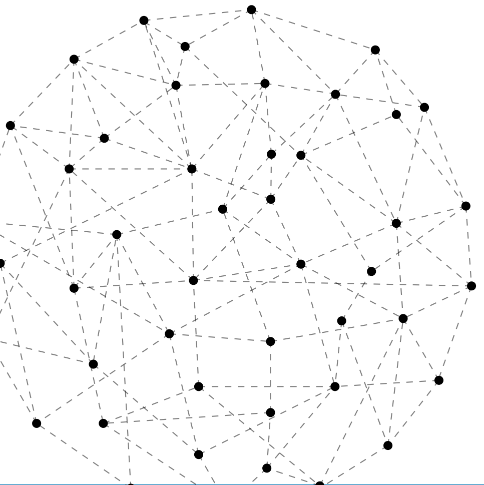
Connected On indicates the date I connected to that person, I will convert that column into a date-time and visualize it with Plotly



```
connections_df ["Connected On"] = pd.to_datetime(connections_df ["Connected On"])
connections_df ["Connected On"]
```

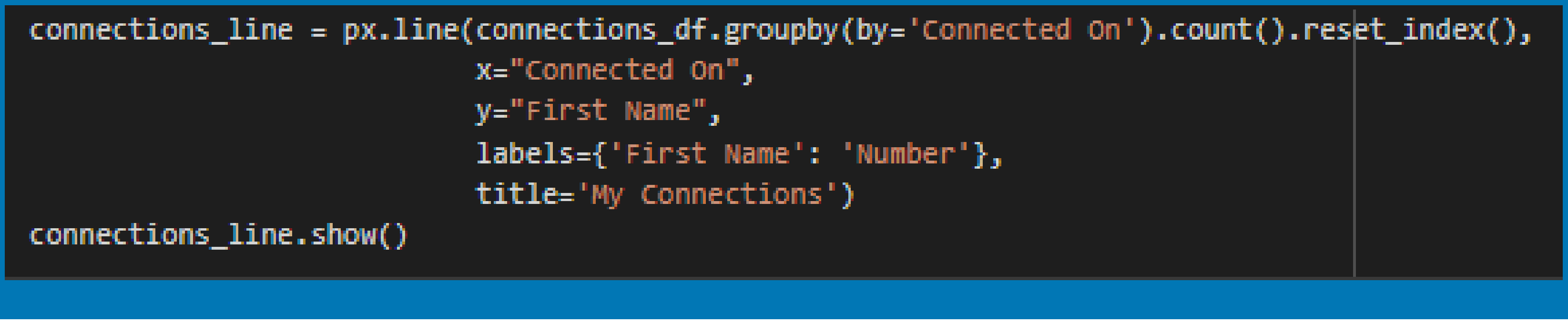
```
0      2022-09-29
1      2022-09-15
2      2022-09-15
3      2022-09-11
4      2022-09-09
          ...
703    2020-07-28
704    2020-07-28
705    2020-07-28
706    2020-07-28
707    2020-07-28
Name: Connected On, Length: 708, dtype: datetime64[ns]
```

Data

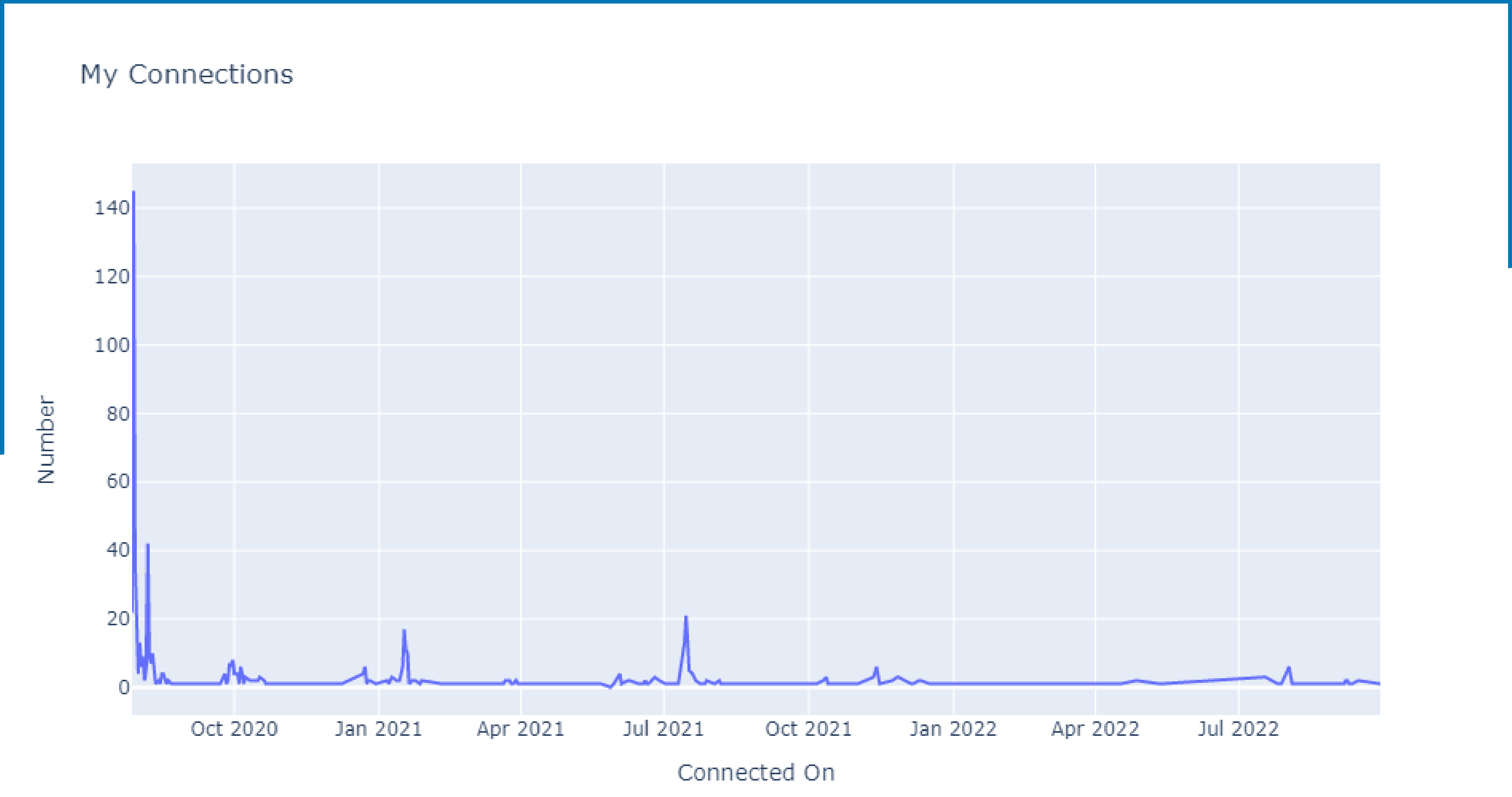


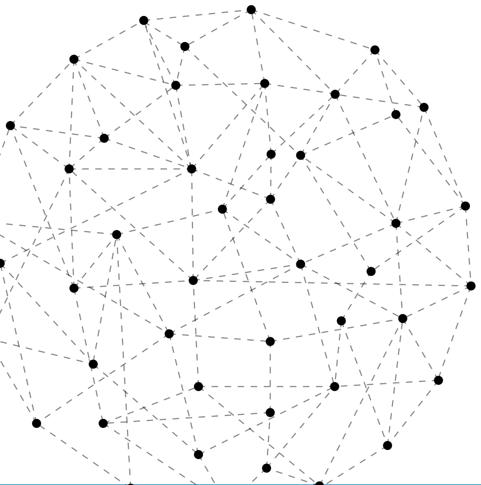
Aaya DINIA

Number of Connections



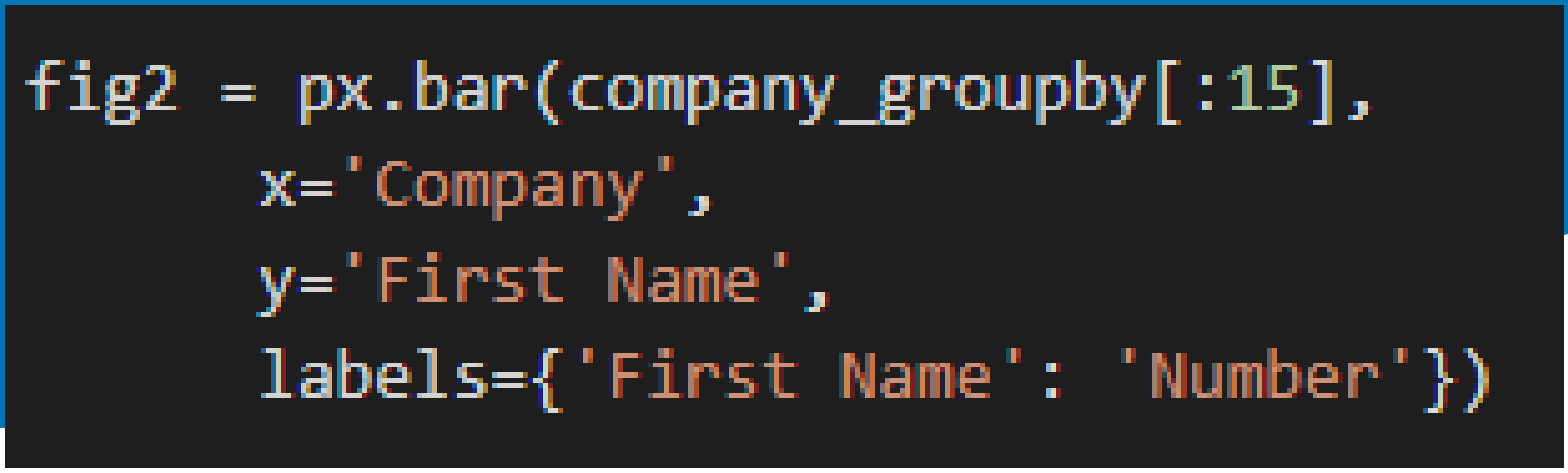
```
connections_line = px.line(connections_df.groupby(by='Connected On').count().reset_index(),
                           x="Connected On",
                           y="First Name",
                           labels={'First Name': 'Number'},
                           title='My Connections')
connections_line.show()
```





Aaya DINIA

Which organizations do the people in my network work at?

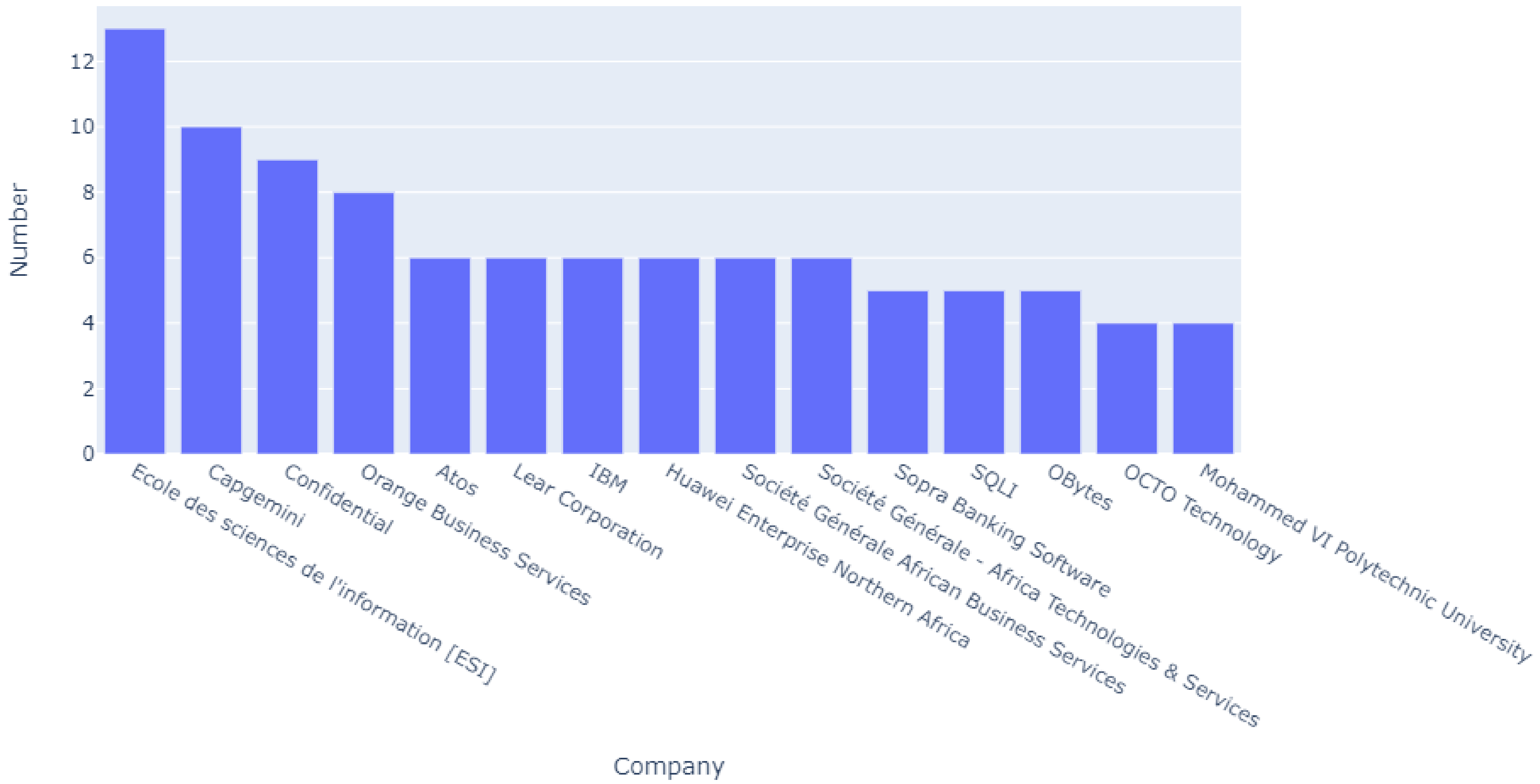


```
fig2 = px.bar(company_groupby[:15],
      x='Company',
      y='First Name',
      labels={'First Name': 'Number'})
```

Bar Graph

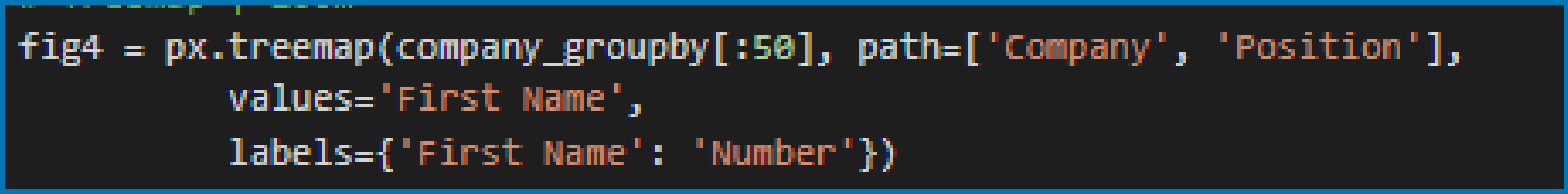
Aaya DINIA

Which organizations do the people in my network work at?



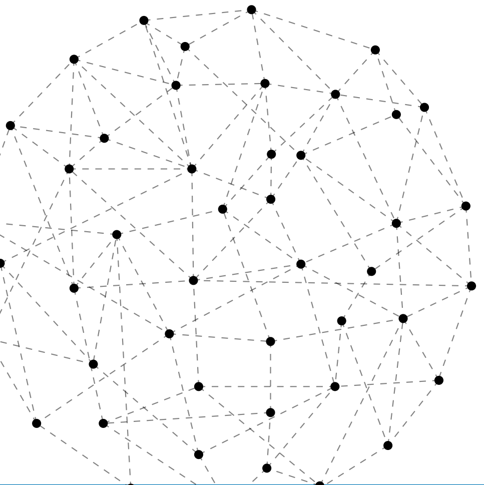
Aaya DINIA

Which organizations do the people in my network work at?



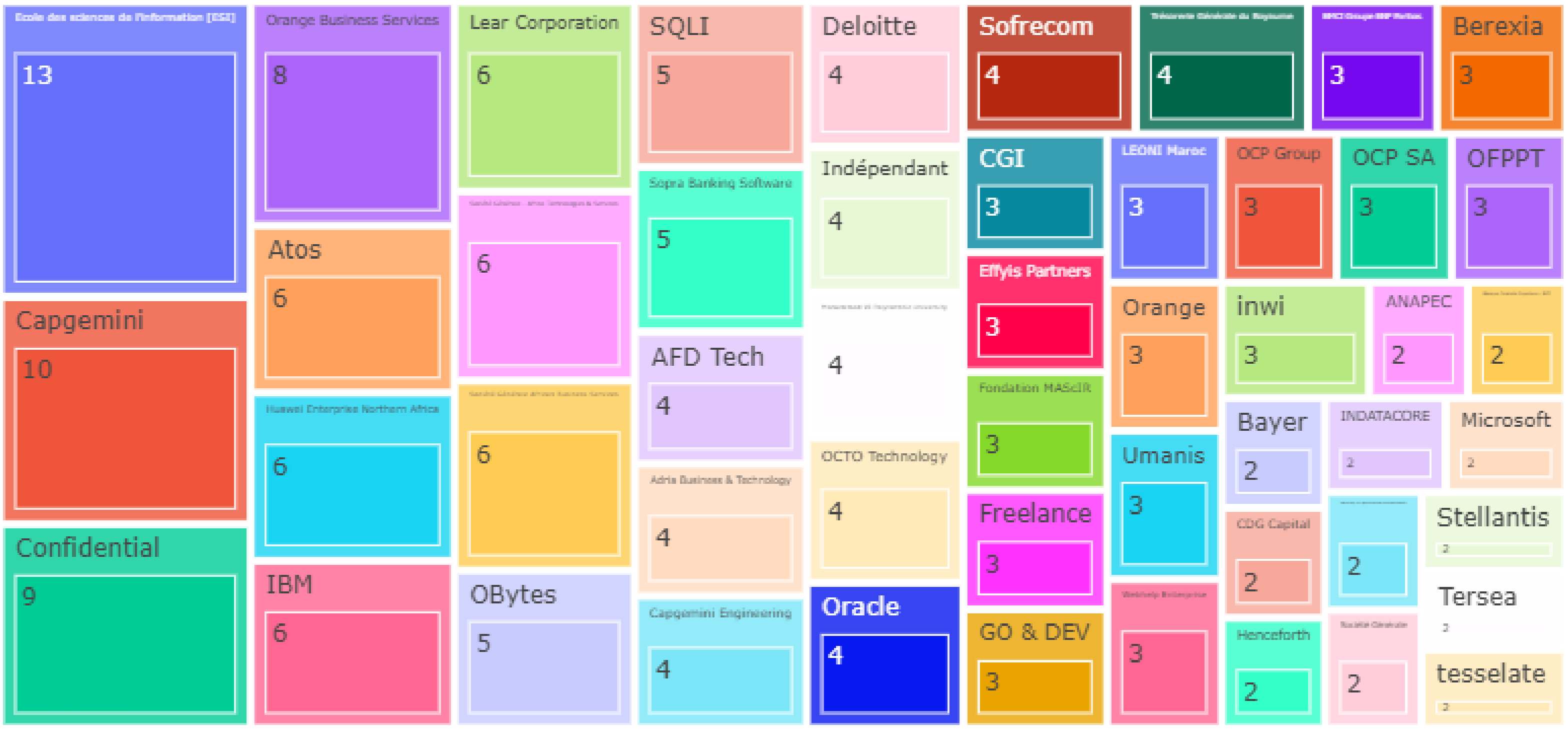
```
fig4 = px.treemap(company_groupby[:50], path=['Company', 'Position'],
                  values='First Name',
                  labels={'First Name': 'Number'})
```

Treemap



Aaya DINIA

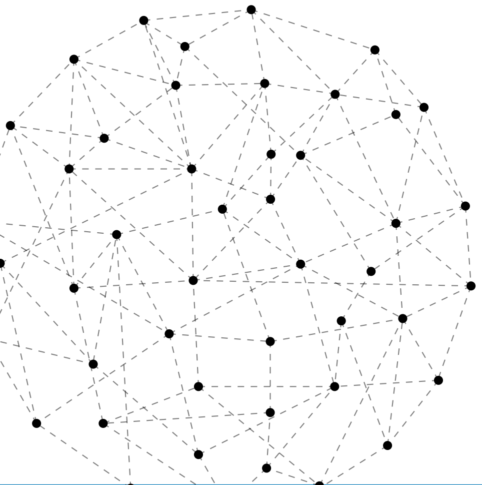
Which organizations do the people in my network work at?



Aaya DINIA

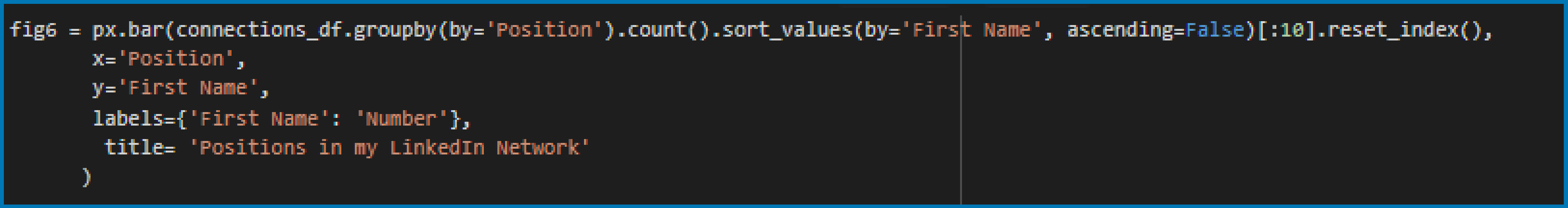
Which organizations do the people in my network work at?

With a treemap, it is easier to compare the proportion of one company related to the others!
It looks like the majority of my network are from my School "Ecole des Sciences de l'Information [ESI]". The second-largest percentage is from Capgemini, Confideniel and Orange.



Aaya DINIA

Top Common Positions

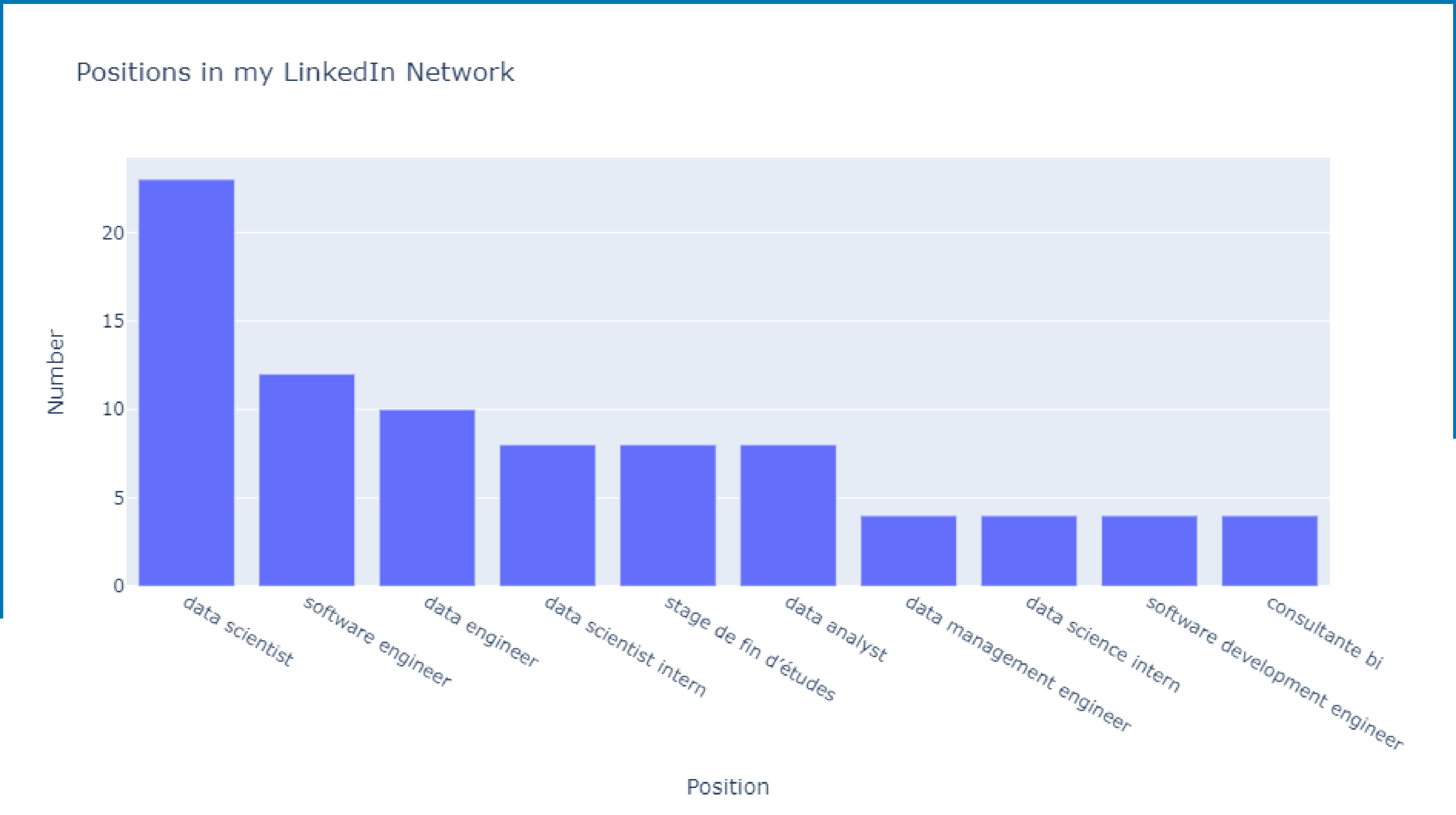


```
fig6 = px.bar(connections_df.groupby(by='Position').count().sort_values(by='First Name', ascending=False)[:10].reset_index(),
       x='Position',
       y='First Name',
       labels={'First Name': 'Number'},
        title= 'Positions in my LinkedIn Network'
      )
```

Bar Graph

Aaya DINIA

Top Common Positions



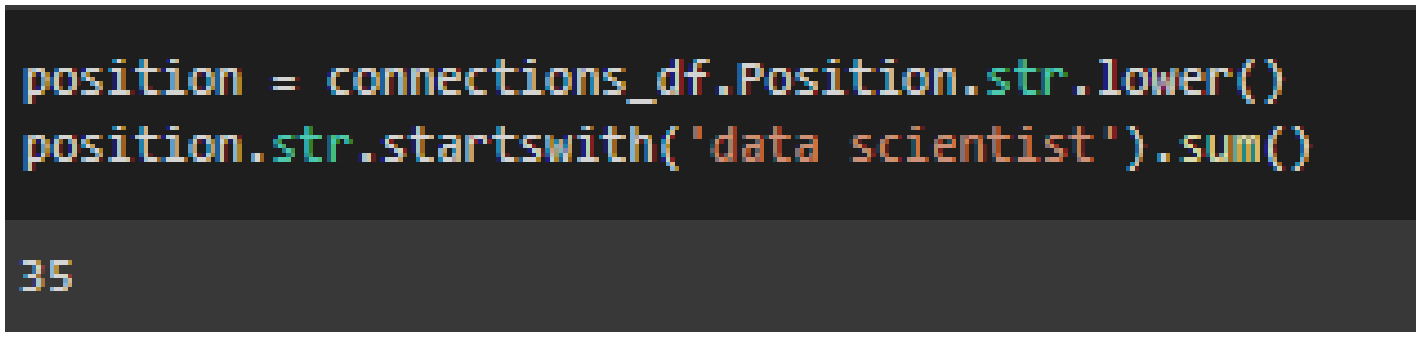
Aaya DINIA

Top Common Positions

It is great that the top common positions in my network are my target groups for networking "data scientist".

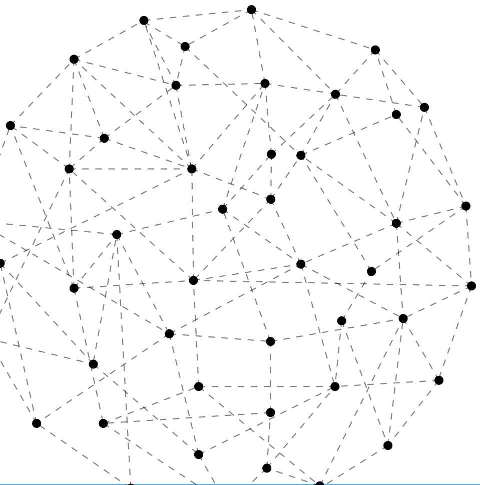
Some people might have titles start with "data scientist" but also have more words in their titles.

Let's find out all the positions with words start with "Data scientist" :



```
position = connections_df.Position.str.lower()
position.str.startswith('data scientist').sum()
```

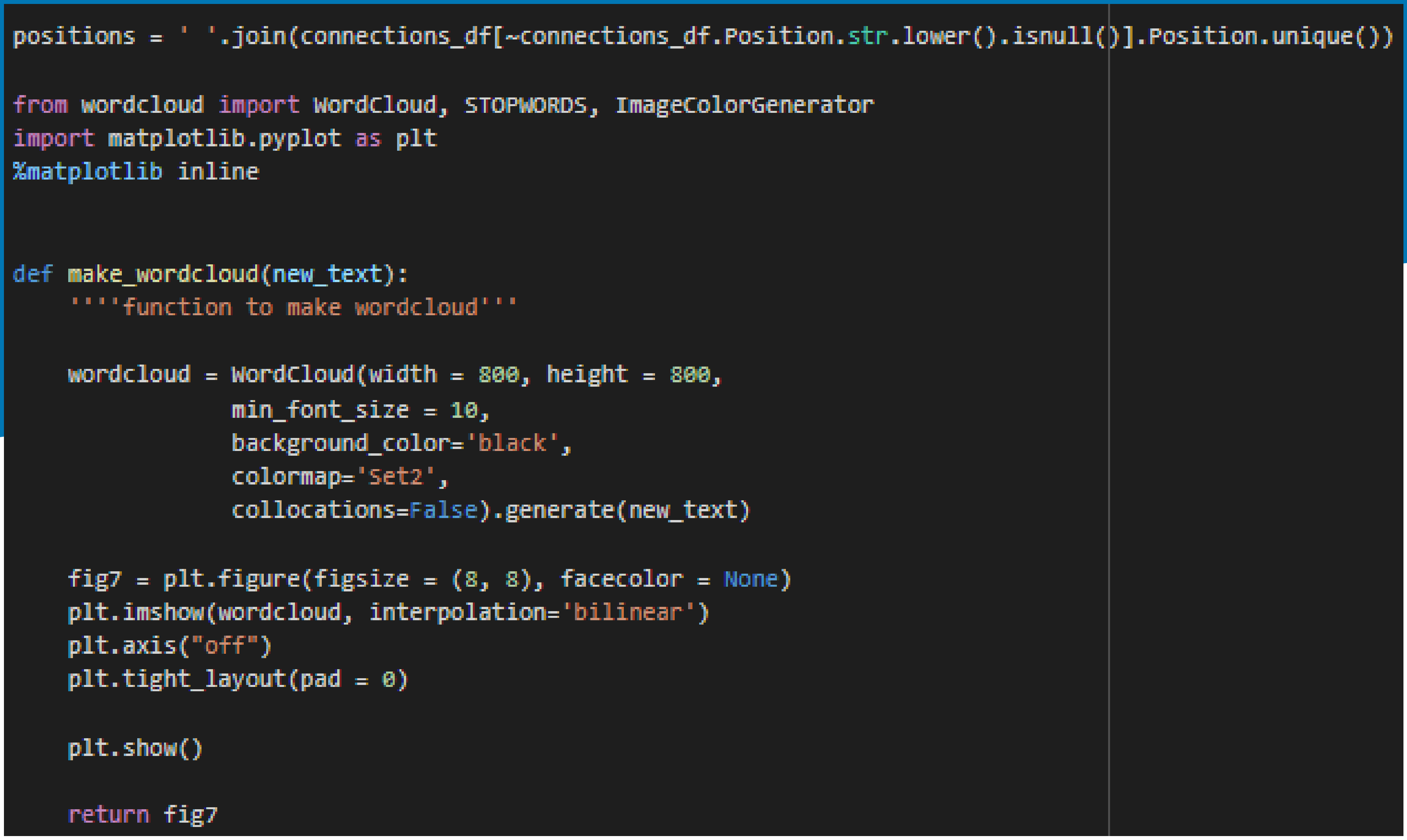
```
35
```



Aaya DINIA

Top Common Positions

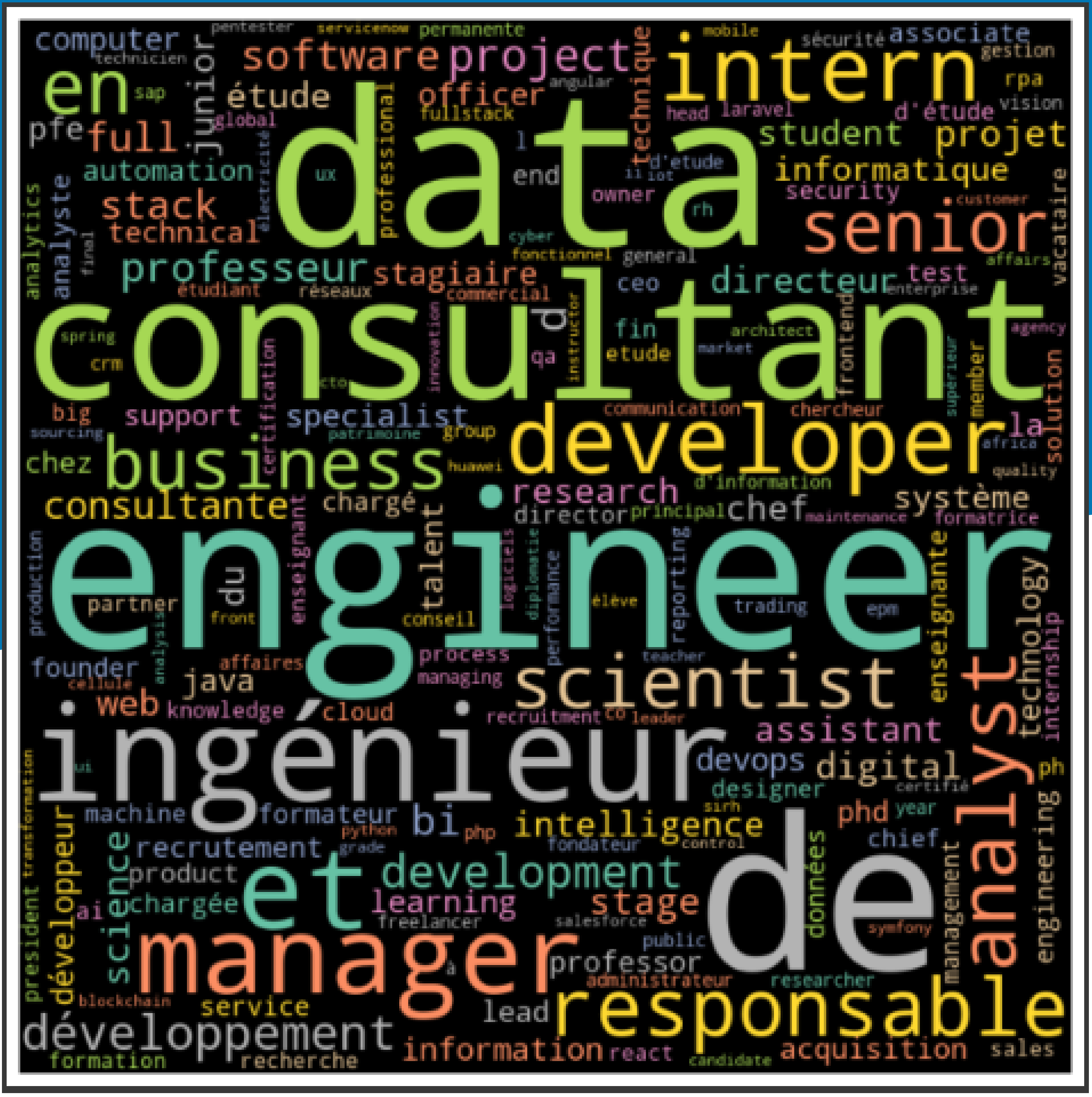
Let's visualise the top common positions with wordcloud ;)



```
positions = ' '.join(connections_df[~connections_df.Position.str.lower().isnull()].Position.unique())

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt
%matplotlib inline


def make_wordcloud(new_text):
    ''''function to make wordcloud'''

    wordcloud = WordCloud(width = 800, height = 800,
                min_font_size = 10,
                background_color='black',
                colormap='Set2',
                collocations=False).generate(new_text)

    fig7 = plt.figure(figsize = (8, 8), facecolor = None)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.tight_layout(pad = 0)

    plt.show()

    return fig7
```

Aaya DINIA

Top Common Positions



Aaya DINIA

END