Data Visualization of gene variant transcriptions groups

In [1]:
```
import pandas as pd

# Read the CSV file into a pandas DataFrame
df = pd.read_csv('serialdat.csv')
#str(df["10^x copies"])
# Display the first 5 rows of the DataFrame
print(df.head())
```

```
     SUMOvar  10^x copies  Replicate 1  Replicate 2  Replicate 3  Average Cq
0  S1V1-10^6          6.0     16.27132     16.19231     16.36603   16.276553
1  S1V1-10^5          5.0     20.14263     20.12184     20.05466   20.106377
2  S1V1-10^4          4.0     23.07819     23.10269     22.86079   23.013890
3  S1V1-10^3          3.0     25.53921     25.51511     25.41548   25.489933
4  S1V1-10^2          2.0     26.05758     25.99988     26.04024   26.032567
```

In [2]:
```
df['10^x copies'] = df['10^x copies'].astype(str)
```

In [3]:
```
df = df.dropna()
```

In [7]:
```
df['SUMOvar'] = df['SUMOvar'].astype(str).apply(lambda x: x.split('-')[0])
```

In [8]:
```
df
```

Out[8]:

	SUMOvar	10^x copies	Replicate 1	Replicate 2	Replicate 3	Average Cq
0	S1V1	6.0	16.271320	16.192310	16.366030	16.276553
1	S1V1	5.0	20.142630	20.121840	20.054660	20.106377
2	S1V1	4.0	23.078190	23.102690	22.860790	23.013890
3	S1V1	3.0	25.539210	25.515110	25.415480	25.489933
4	S1V1	2.0	26.057580	25.999880	26.040240	26.032567
5	S1V1	1.0	26.236200	26.034280	26.190770	26.153750
6	S1V2	6.0	14.905530	14.873300	14.865100	14.881310
7	S1V2	5.0	18.454010	18.364780	18.328960	18.382583
8	S1V2	4.0	21.788040	21.759100	21.684900	21.744013
9	S1V2	3.0	23.998260	23.602960	23.823790	23.808337
10	S1V2	2.0	24.337950	24.543700	24.265880	24.382510
11	S1V2	1.0	24.662130	24.597340	23.805100	24.354857
12	S1V3	6.0	13.820926	13.788557	13.837119	13.815534
13	S1V3	5.0	17.668800	17.628487	17.950822	17.749369
14	S1V3	4.0	20.957602	21.086240	21.153529	21.065790
15	S1V3	3.0	23.448480	24.064225	24.190202	23.900969
16	S1V3	2.0	25.373289	25.979688	26.097752	25.816910
17	S1V3	1.0	25.413550	25.861247	25.787894	25.687564
18	S2V1	6.0	10.366014	9.732389	8.798607	9.632337
19	S2V1	5.0	13.664237	11.998538	12.035592	12.566122
20	S2V1	4.0	25.902147	15.753241	15.826444	19.160610
21	S2V1	3.0	21.818239	19.740060	19.752652	20.436984
22	S2V1	2.0	24.170059	23.652029	23.189184	23.670424
23	S2V1	1.0	25.749917	25.395615	25.104794	25.416775
24	S2V2	6.0	10.056137	9.069710	9.118771	9.414873
25	S2V2	5.0	13.795135	12.453789	12.724051	12.990992
26	S2V2	4.0	19.550902	16.350932	16.346928	17.416254
27	S2V2	3.0	20.895418	19.882332	20.057618	20.278456
28	S2V2	2.0	23.183559	22.600623	22.616847	22.800343
29	S2V2	1.0	24.644120	23.055101	23.411640	23.703620
30	S3V1	6.0	11.327868	9.977722	9.838526	10.381372
31	S3V1	5.0	14.700754	13.563679	13.574665	13.946366
32	S3V1	4.0	19.026552	17.262518	17.185991	17.825021
33	S3V1	3.0	21.254102	20.444505	20.480660	20.726422

	SUMOvar	10^x copies	Replicate 1	Replicate 2	Replicate 3	Average Cq
34	S3V1	2.0	20.833690	22.809750	22.670585	22.104675
35	S3V1	1.0	21.715993	23.749074	23.168936	22.878001
36	S3V2	6.0	13.564612	10.315584	10.325927	11.402041
37	S3V2	5.0	16.972926	13.871555	14.024527	14.956336
38	S3V2	4.0	21.097947	17.789011	17.797498	18.894818
39	S3V2	3.0	23.750201	21.055004	21.140170	21.981792
40	S3V2	2.0	24.850817	23.634807	23.225069	23.903565
41	S3V2	1.0	24.634719	25.343952	25.277707	25.085460

In [5]:
```
df['10^x copies'].unique()
```

Out[5]:
```
array(['6.0', '5.0', '4.0', '3.0', '2.0', '1.0'], dtype=object)
```

In [89]:
```
df.dtypes
```

Out[89]:
```
SUMOvar         object
10^x copies     object
Replicate 1    float64
Replicate 2    float64
Replicate 3    float64
Average Cq     float64
dtype: object
```

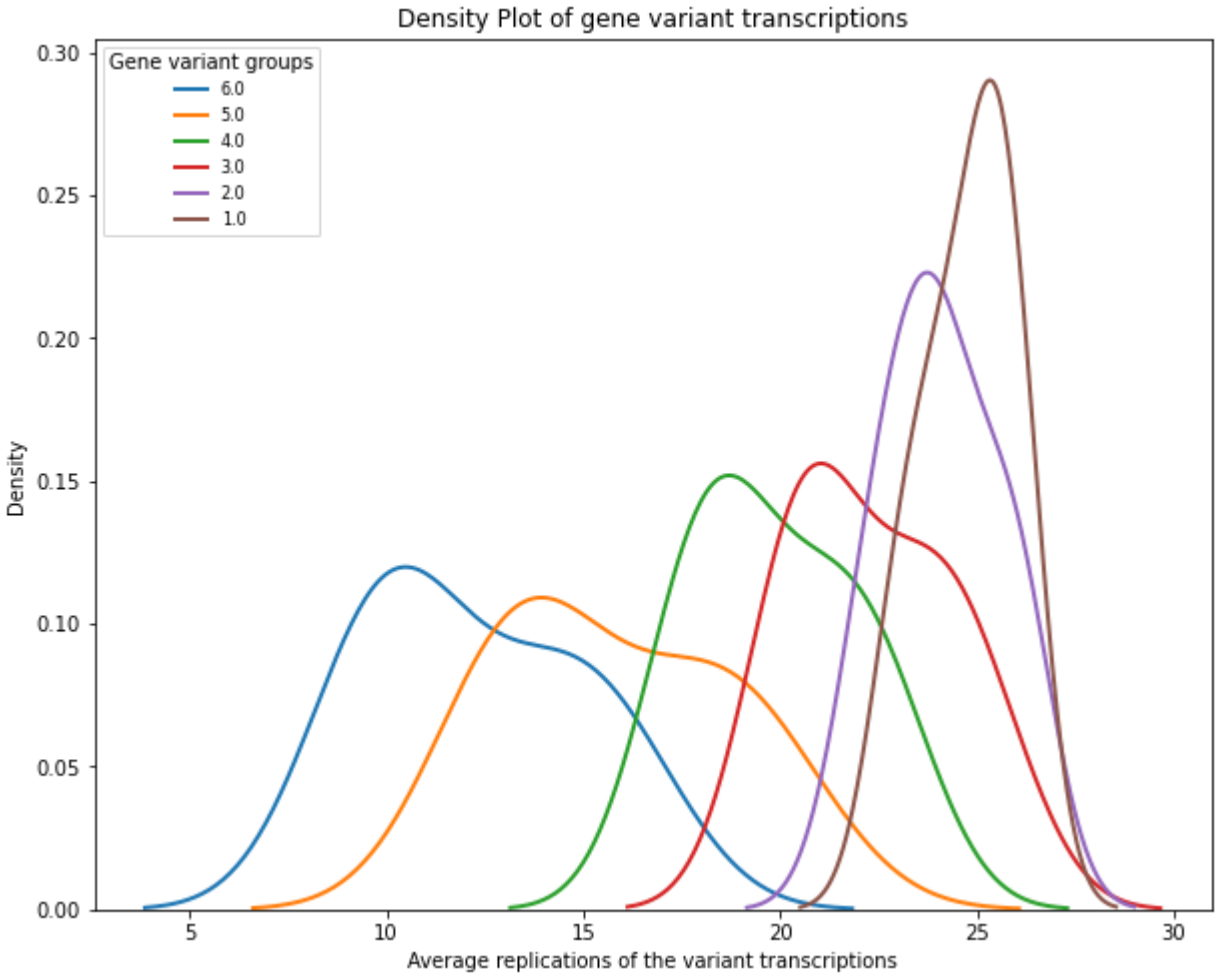
In [6]:
```
```

In [10]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# List of groups to plot
replicate =['6.0', '5.0', '4.0', '3.0', '2.0', '1.0']
fig, ax = plt.subplots(figsize=(10, 8))
# Iterate through the groups
for i in replicate:
    # Subset to the airline
    subset = df[df['10^x copies'] == i]

    # Draw the density plot
    sns.distplot(subset['Average Cq'], hist = False, kde = True,
                 kde_kws = {'linewidth': 2},
                 label = i)

# Plot formatting
plt.legend(prop={'size': 8}, title = 'Gene variant groups')
plt.title('Density Plot of gene variant transcriptions')
plt.xlabel('Average replications of the variant transcriptions')
plt.ylabel('Density')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

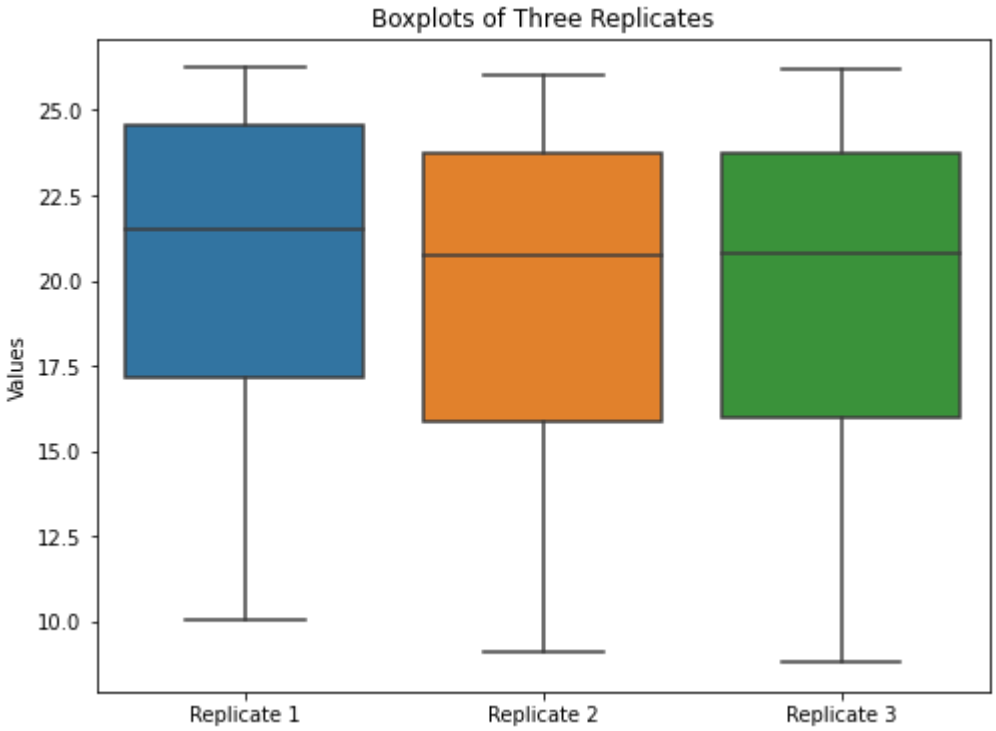
Out[10]:

```
Text(0, 0.5, 'Density')
```



From the above plots of the groups, As you move from Group 6 to group 1, you will notice the desity of the average replication value increase as the replication value increase.And I noticed an overlap in the second peaks of group 6 with the first peak of group 5, same with group 5 and 4, 4 and 3, 3 and 2 but 2 and 1 have just one peak.

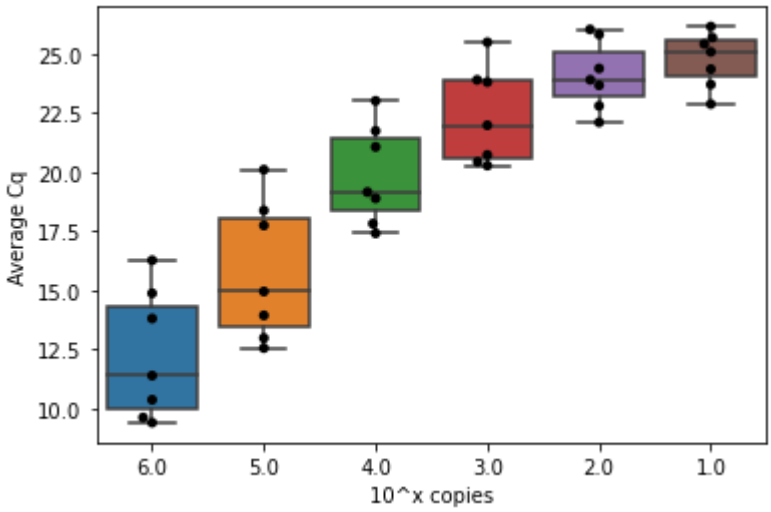
In [11]:
```
plt.figure(figsize=(8,6))
# create boxplots of the three rplicates using seaborn
sns.boxplot(data=df[["Replicate 1","Replicate 2","Replicate 3"]])


# set title and axis labels
plt.title("Boxplots of Three Replicates")
plt.ylabel("Values")

# show the plot
plt.show()
plt.show()
```



From the above plot, there seems to be no significat difference between the three replicates. Thogh replicate one seems to be slightly different from the replicate 2 and 3.

In [72]:
```
import pandas as pd
import seaborn as sns
sns.boxplot(x='10^x copies', y='Average Cq', data=df)
sns.swarmplot(x='10^x copies', y='Average Cq', data=df, color='black')
```

Out[72]:
```
<AxesSubplot:xlabel='10^x copies', ylabel='Average Cq'>
```



In [77]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt



sns.set_style("whitegrid")
sns.set_palette("Set2")
```

```
fig, ax = plt.subplots(figsize=(10, 8))

sns.boxplot(x='10^x copies', y='Average Cq', data=df,ax=ax)
sns.swarmplot(x='10^x copies', y='Average Cq', data=df, color='black',ax=ax)

ax.set_xlabel("Groups",fontsize=14)
ax.set_ylabel("Values",fontsize=14)
ax.set_title("Distribution of Values for Different Groups",fontsize=16)

plt.show()
```