

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call

```
train_data = pd.read_excel('/content/drive/MyDrive/PCA_fourfold_gene_expression.
train_data.head()
```

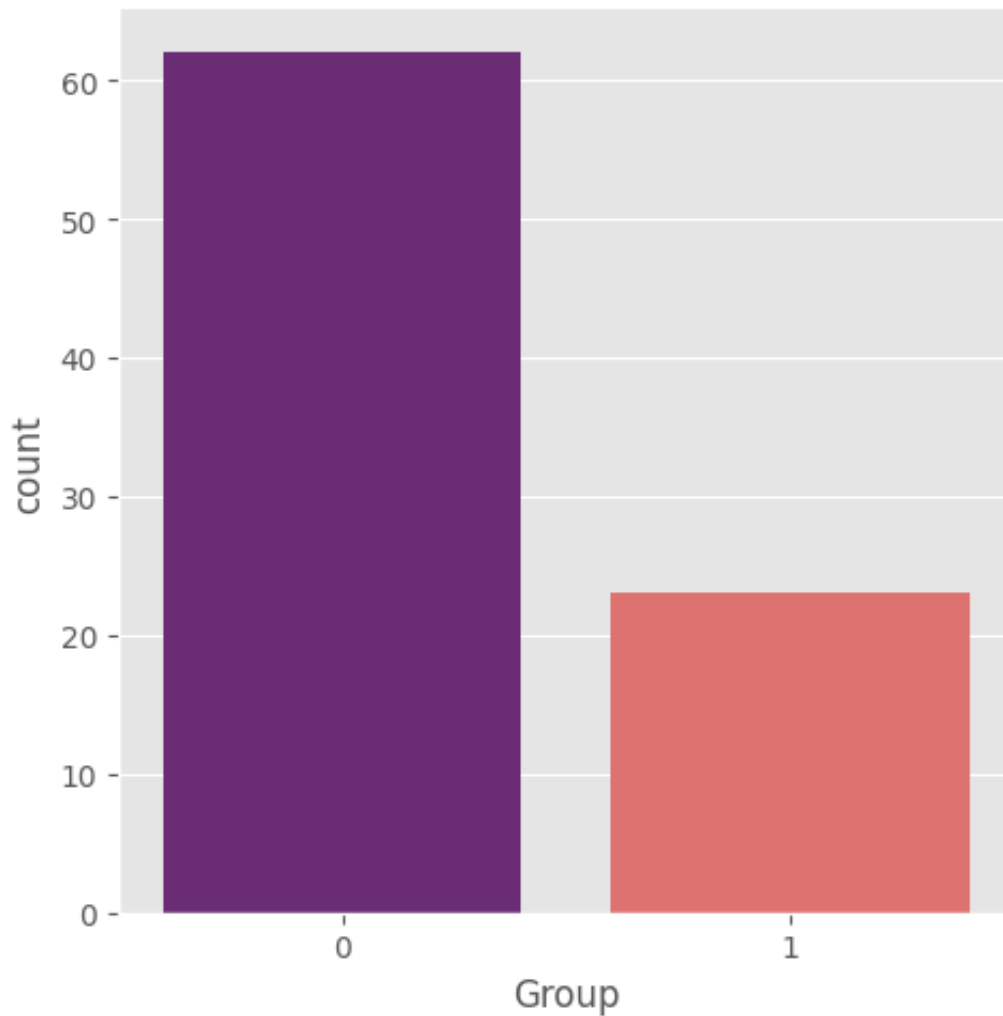
	Gene_ID	Group	C_vs_R_FC	C_vs_T2a_b_c_FC	C_vs_T3a_FC	C_vs_T3b_4FC
0	SFN	0	23.22	5.46	4.90	4.25
1	MIR205HG; MIR205	0	49.93	5.56	10.19	10.77
2	GSTM3	0	5.98	4.31	5.54	7.08
3	DPT	0	9.18	4.66	6.01	10.34
4	LMOD1	0	7.71	4.91	5.95	5.00

```
train_data.isnull().sum()
```

```
Gene_ID      0
Group        0
C_vs_R_FC    0
C_vs_T2a_b_c_FC  0
C_vs_T3a_FC  0
C_vs_T3b_4FC  0
dtype: int64
```

```
# Count of Coding & NonCoding  
sns.catplot(x = 'Group', kind = 'count', data = train_data, palette = 'magma')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fbc6fc47400>
```



```
train_data['Group'].value_counts()
```

```
0    62  
1    23  
Name: Group, dtype: int64
```

```
plt.figure(figsize = (15,8))
sns.distplot(train_data ['C_vs_R_FC'], color = 'lightgreen')
plt.show()
```

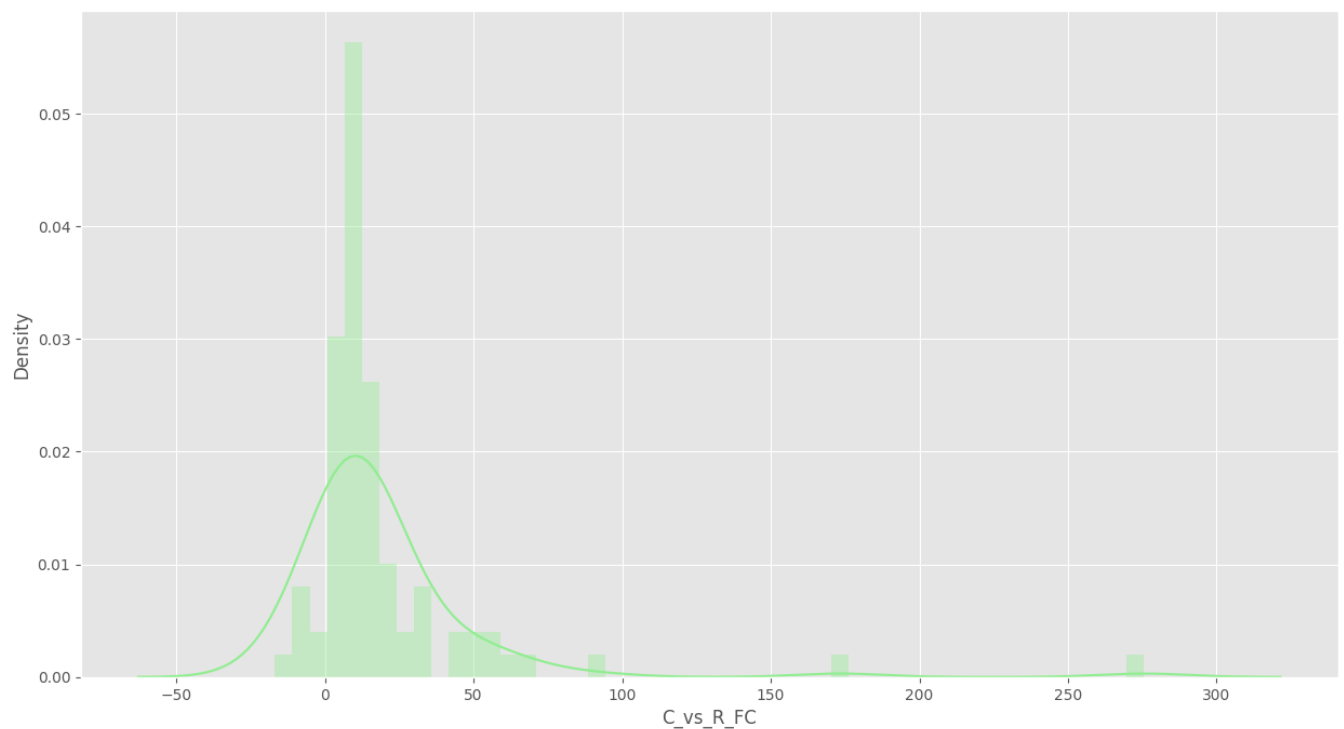
<ipython-input-29-8e3f94dd0aff>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(train_data ['C_vs_R_FC'], color = 'lightgreen')
```



```
plt.figure(figsize = (10,8))
```

```
sns.distplot(train_data ['C_vs_T3b_4FC'], color = 'blue')  
plt.show()
```

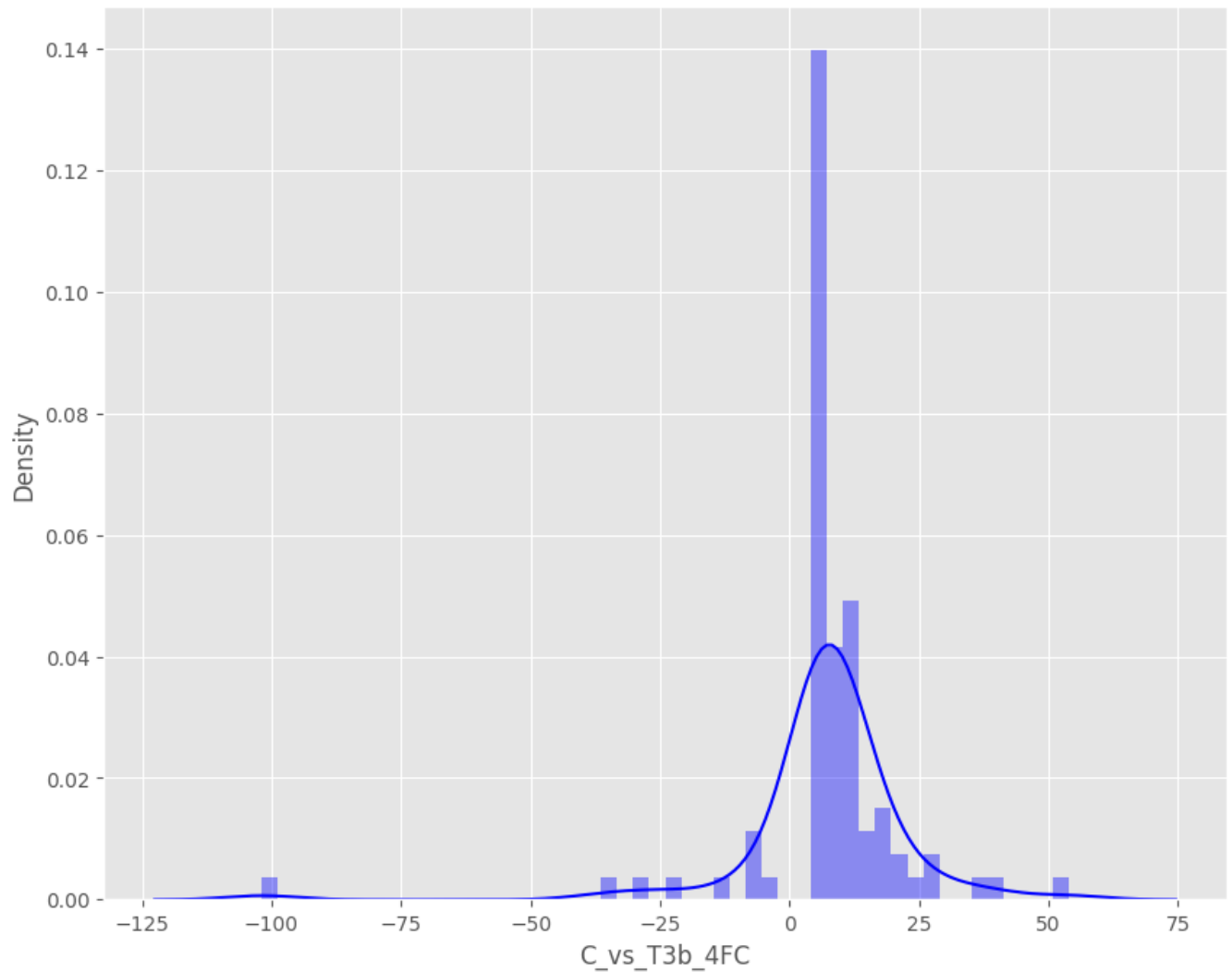
↳ <ipython-input-30-d5358a004dc7>:2: UserWarning:

``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(train_data ['C_vs_T3b_4FC'], color = 'blue')
```



```
plt.figure(figsize = (15,8))
sns.distplot(train_data ['C_vs_T3a_FC'], color = 'red')
plt.show()
```

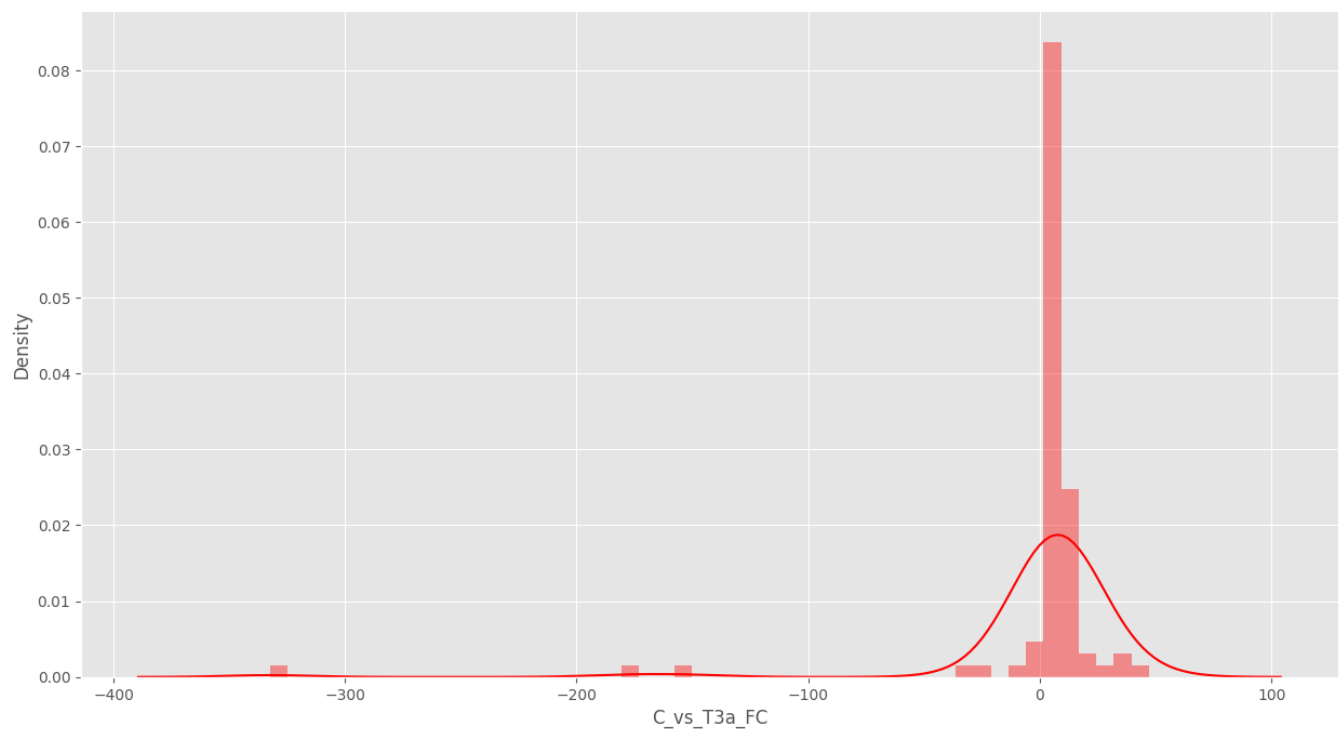
<ipython-input-31-d117600d2daf>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(train_data ['C_vs_T3a_FC'], color = 'red')
```



```
plt.figure(figsize = (15,8))
sns.distplot(train_data ['C_vs_T2a_b_c_FC'], color = 'magenta')
plt.show()
```

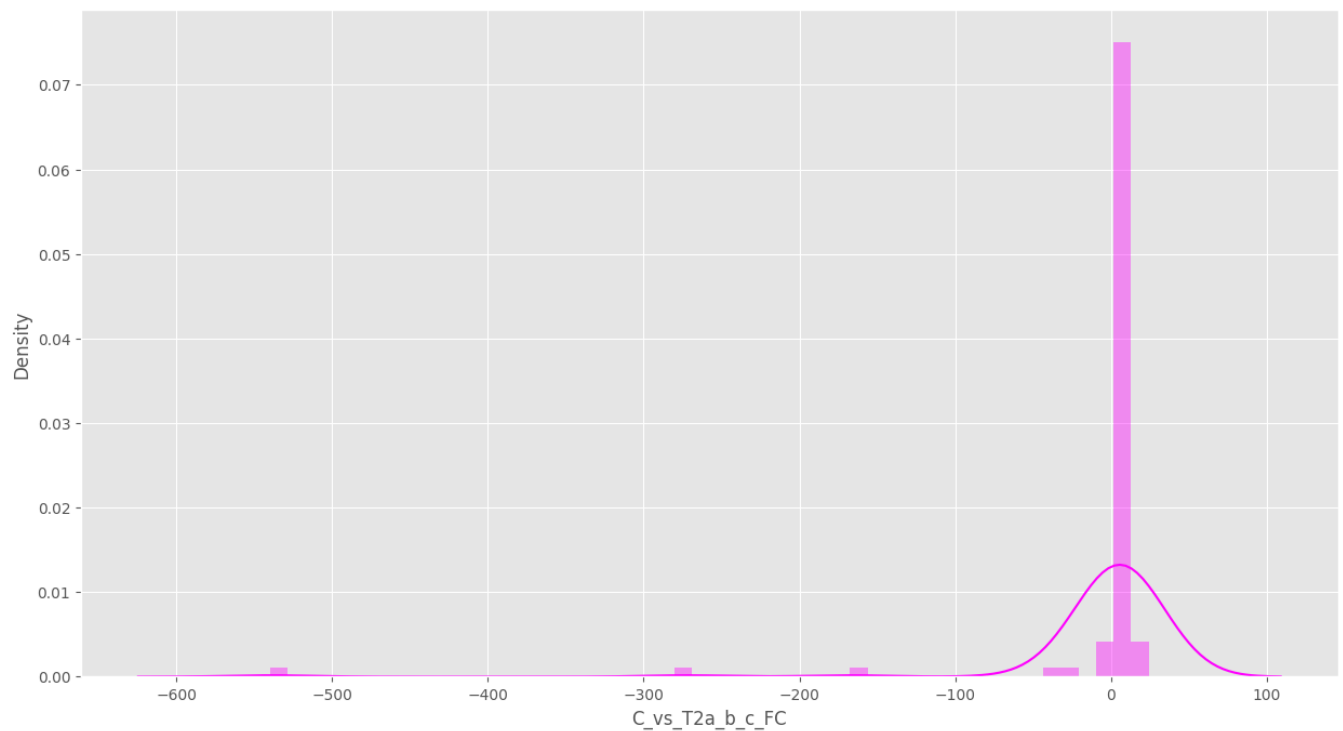
<ipython-input-32-410975337d32>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(train_data ['C_vs_T2a_b_c_FC'], color = 'magenta')
```



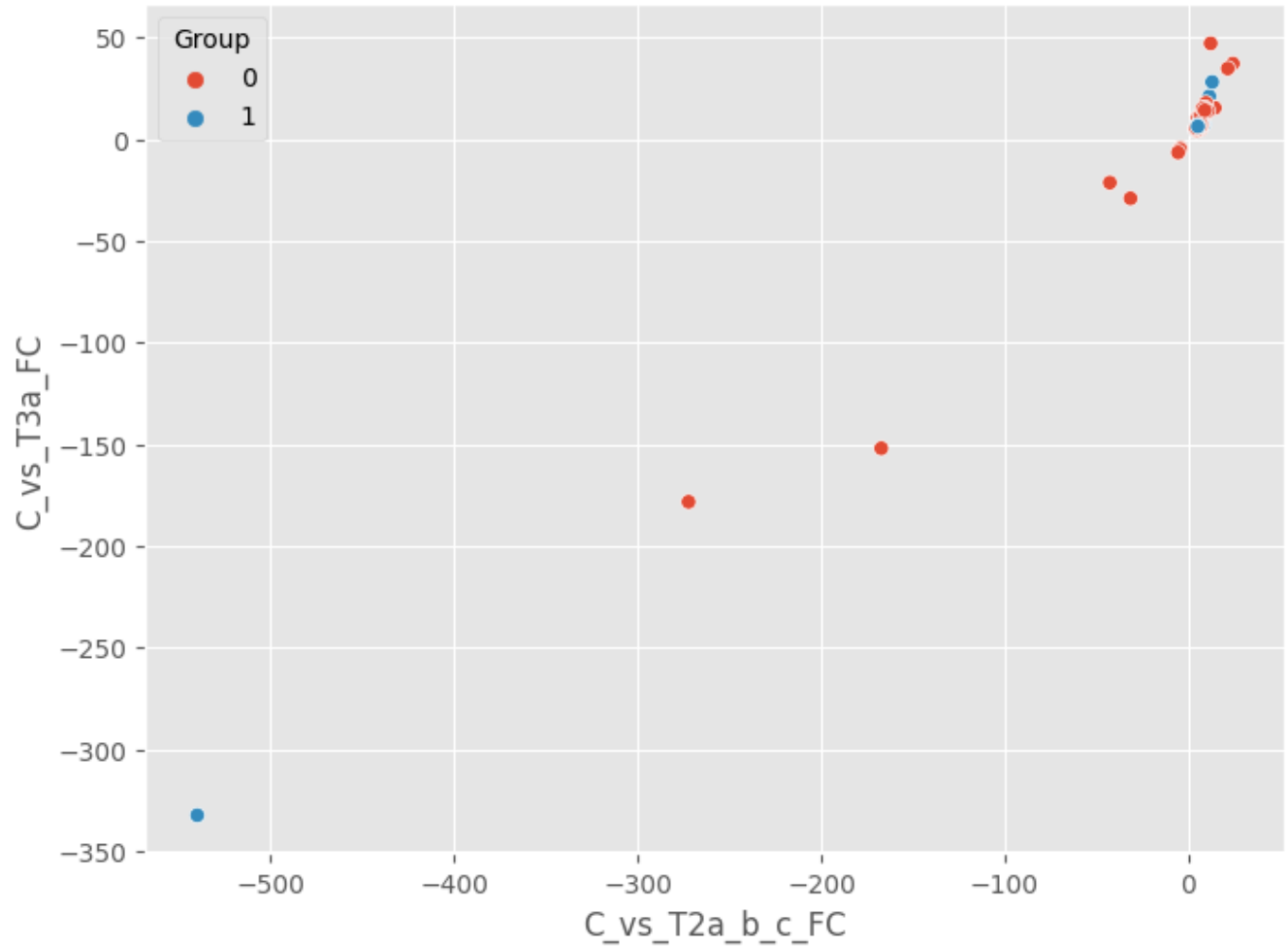
```
### Analysis of Distribution b/w T2a_b_c_vs_C FC & T3a_vs_C FC
```

```
plt.style.use("ggplot")
```

```
plt.figure(figsize = (8,6))
```

```
sns.scatterplot( x = "C_vs_T2a_b_c_FC", y = "C_vs_T3a_FC", data = train_data, hu
```

```
<Axes: xlabel='C_vs_T2a_b_c_FC', ylabel='C_vs_T3a_FC'>
```





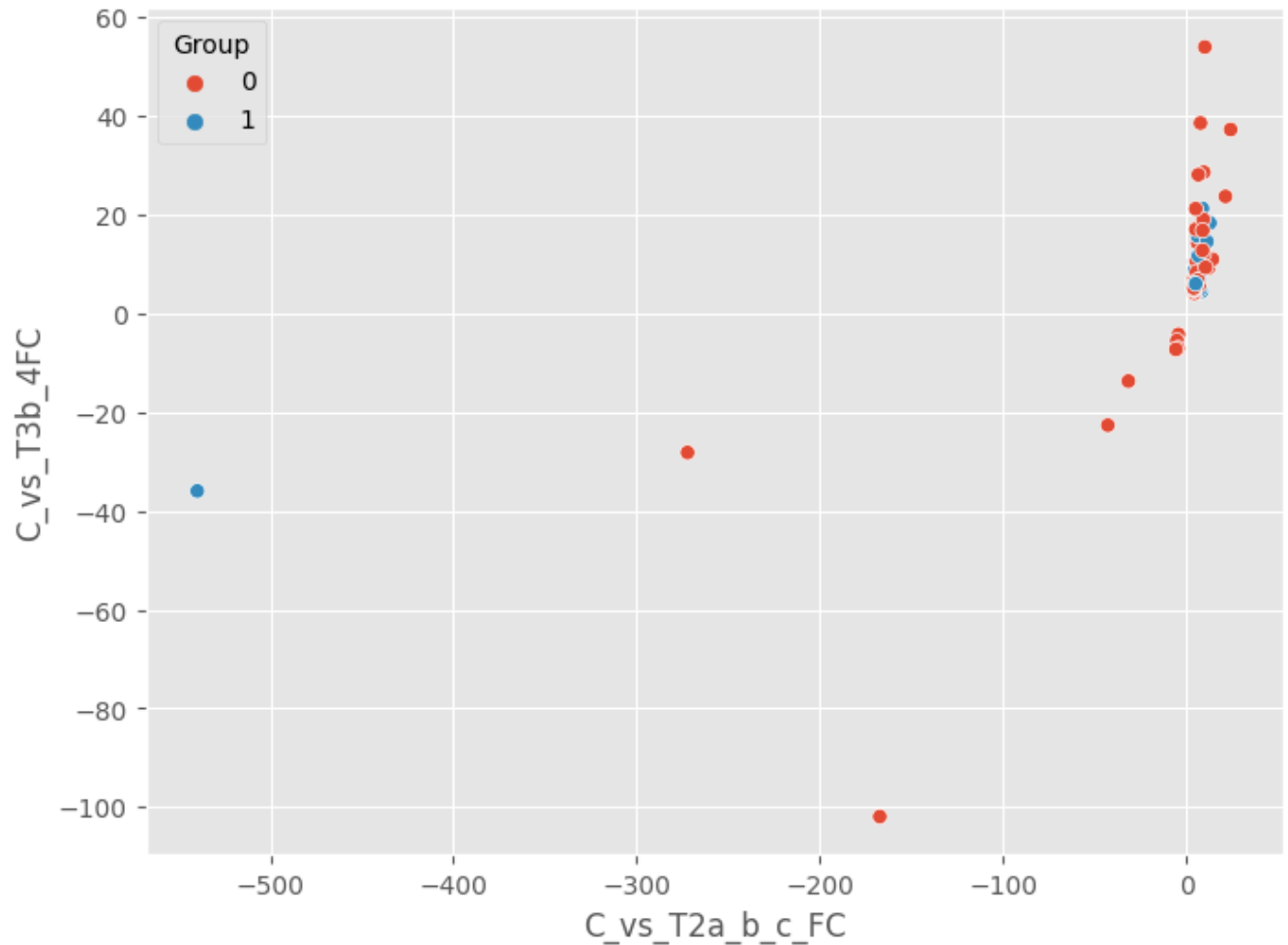
```
### Analysis of Distribution b/w T2a_b_c_vs_C FC & T3b_4_vs_C FC
```

```
plt.style.use("ggplot")
```

```
plt.figure(figsize = (8,6))
```

```
sns.scatterplot( x = "C_vs_T2a_b_c_FC", y = "C_vs_T3b_4FC", data = train_data, h
```

```
<Axes: xlabel='C_vs_T2a_b_c_FC', ylabel='C_vs_T3b_4FC'>
```



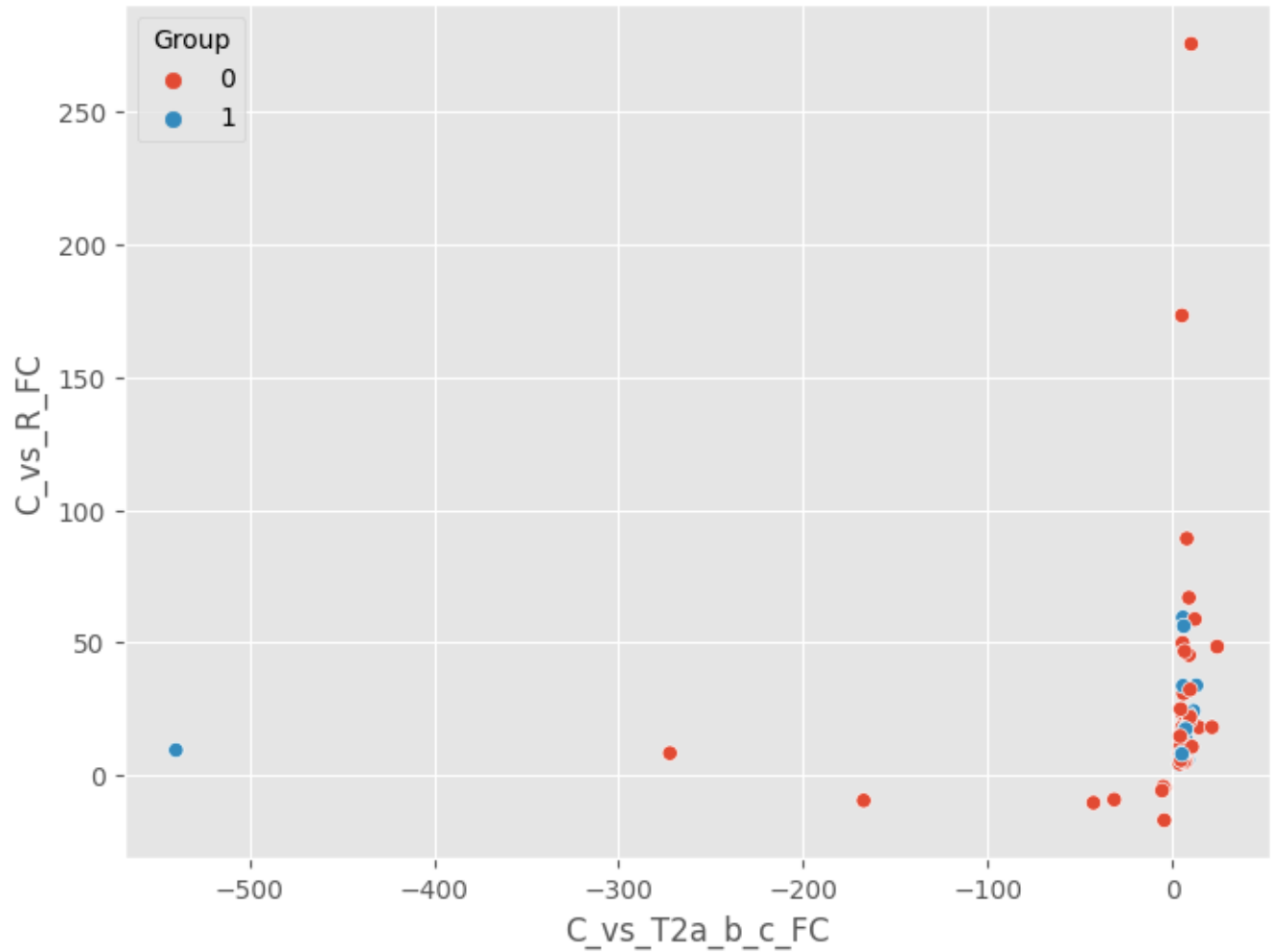
```
### Analysis of Distribution b/w T2a_b_c_vs_C FC & HR_vs_C_FC
```

```
plt.style.use("ggplot")
```

```
plt.figure(figsize = (8,6))
```

```
sns.scatterplot( x = "C_vs_T2a_b_c_FC", y = "C_vs_R_FC", data = train_data, hue
```

```
<Axes: xlabel='C_vs_T2a_b_c_FC', ylabel='C_vs_R_FC'>
```



train\_data

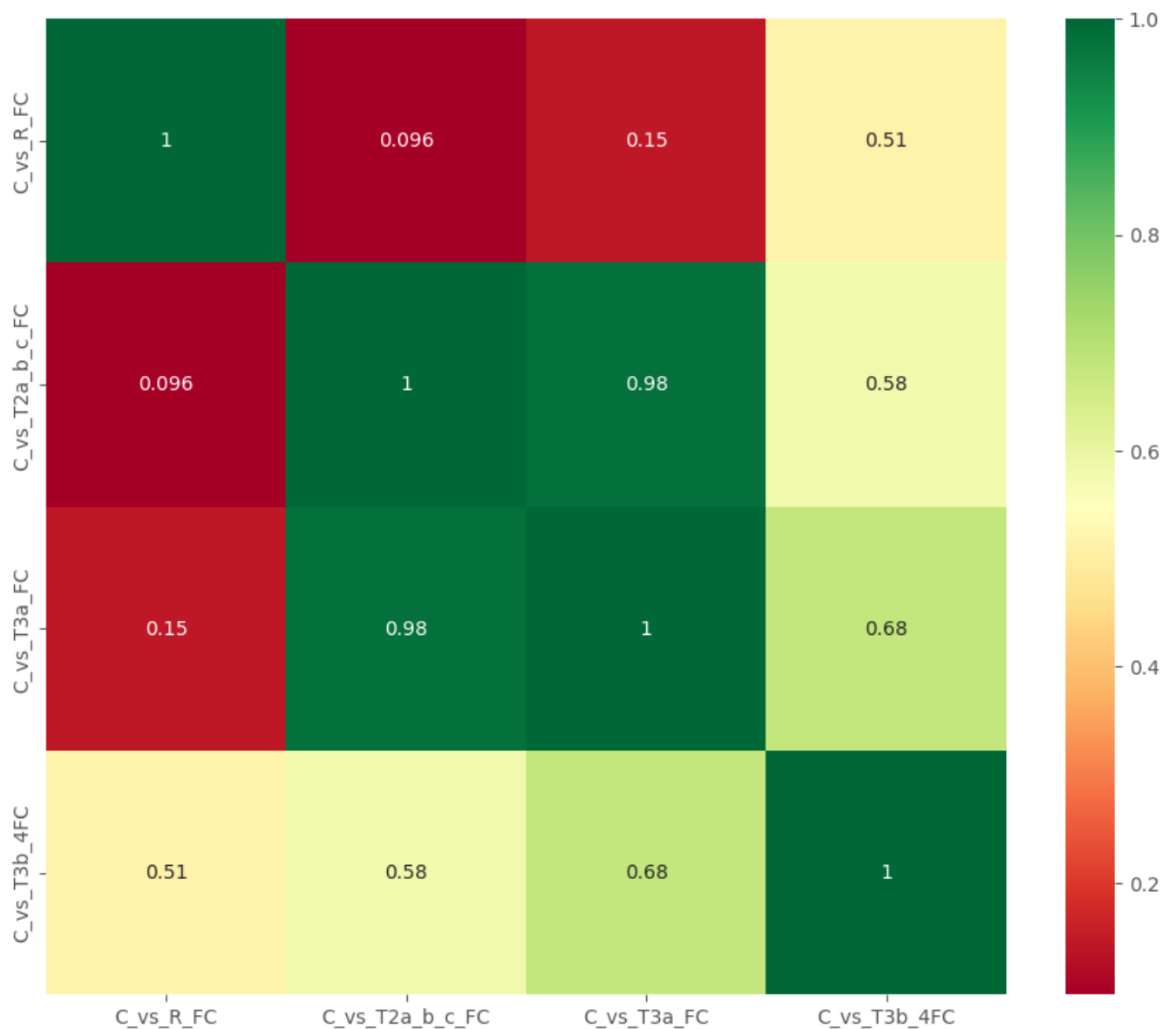
	Gene_ID	Group	C_vs_R_FC	C_vs_T2a_b_c_FC	C_vs_T3a_FC	C_vs_T3b_4FC
0	SFN	0	23.22	5.46	4.90	4.25
1	MIR205HG; MIR205	0	49.93	5.56	10.19	10.77
2	GSTM3	0	5.98	4.31	5.54	7.08
3	DPT	0	9.18	4.66	6.01	10.34
4	LMOD1	0	7.71	4.91	5.95	5.00
...	...	...	...	...	...	...
80	MYL9	0	5.77	4.73	5.61	5.43
81	FLRT3	0	14.81	4.25	5.54	5.07
82	MYL9	1	8.07	5.23	6.32	5.96
83	SIM2	0	-5.66	-5.54	-6.53	-7.32
84	NEFH	0	66.98	9.00	14.23	12.72

85 rows × 6 columns

```
train_data = train_data.drop(['Group'], axis = 1)
```

```
plt.figure(figsize = (11,9))
#-1 to +1 > + or -0.85 ,closer to 0 (not to select these features)R
# for correlation data should be continuouslike charges not discrete like coding
sns.heatmap(train_data.corr(), annot = True, cmap = 'RdYlGn')
```

```
<ipython-input-39-ba8740157f36>:4: FutureWarning: The default value of nume
sns.heatmap(train_data.corr(), annot = True, cmap = 'RdYlGn')
<Axes: >
```



```
#splitting the data into train and test
```

train\_data

	Gene_ID	C_vs_R_FC	C_vs_T2a_b_c_FC	C_vs_T3a_FC	C_vs_T3b_4FC
0	SFN	23.22	5.46	4.90	4.25
1	MIR205HG; MIR205	49.93	5.56	10.19	10.77
2	GSTM3	5.98	4.31	5.54	7.08
3	DPT	9.18	4.66	6.01	10.34
4	LMOD1	7.71	4.91	5.95	5.00
...	...	...	...	...	...
80	MYL9	5.77	4.73	5.61	5.43
81	FLRT3	14.81	4.25	5.54	5.07
82	MYL9	8.07	5.23	6.32	5.96
83	SIM2	-5.66	-5.54	-6.53	-7.32
84	NEFH	66.98	9.00	14.23	12.72

85 rows × 5 columns

```
X = train_data.drop('Gene_ID', axis = 1)
```

```
X
```

	C_vs_R_FC	C_vs_T2a_b_c_FC	C_vs_T3a_FC	C_vs_T3b_4FC
0	23.22	5.46	4.90	4.25
1	49.93	5.56	10.19	10.77
2	5.98	4.31	5.54	7.08
3	9.18	4.66	6.01	10.34
4	7.71	4.91	5.95	5.00
...	...	...	...	...
80	5.77	4.73	5.61	5.43
81	14.81	4.25	5.54	5.07
82	8.07	5.23	6.32	5.96
83	-5.66	-5.54	-6.53	-7.32
84	66.98	9.00	14.23	12.72

85 rows × 4 columns

Cross Validation splitting samples into train & test samples randomly

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
```

```
-----
-
NameError                                Traceback (most recent call
last)
<ipython-input-44-4d59d0e7127a> in <cell line: 2>()
      1 from sklearn.model_selection import train_test_split
----> 2 x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size = 0.2, random_state = 0)

NameError: name 'Y' is not defined
```

```
len(x_train)
```

68 will be one epoch

```
from sklearn.preprocessing import MinMaxScaler  
scaler = MinMaxScaler(feature_range = (0,1))  
x_train_scaler = scaler.fit_transform(x_train)
```

```
x_test_scaled = scaler.transform(x_test)
```

```
x_train_scaler
```

```
from keras.utils.np_utils import to_categorical
```

```
y_train = to_categorical(y_train)
```

```
y_train
```

```
y_train.shape
```

```
y_test = to_categorical(y_test)
```

```
y_test
```

```
y_test.shape
```

```
x_train.shape
```

```
x_test.shape
```

```
len(x_train)
```

ANN Classification

```
# Dense Neural Network
from tensorflow.keras.layers import Dense, Dropout
# Sequential Connection with Neural Network
from tensorflow.keras.models import Sequential
# Optimizers for Regression Model
from tensorflow.keras.optimizers import Adam

def build_model():
    # Sequential Neural Network – Feedforward Neural Network
    model = Sequential()
    # Units = Number of Neurons(2 * pow(n)) , Input Shape = Num of Features
    model.add(Dense(units = 64, activation = 'relu', input_shape = [len(X.keys())
    model.add(Dropout(0.2))
    # Hidden Layer – I
    model.add(Dense(units = 128, activation = 'relu'))
    model.add(Dropout(0.2))

    # Hidden Layer – II
    model.add(Dense(units = 128, activation = 'relu'))
    model.add(Dropout(0.3))

    # Output Layer – For Multi Classification
    model.add(Dense(units = 4, activation='softmax'))

    # Optimizers (alpha)
    optimizers = Adam(learning_rate = 0.001)

    # Model Compiler
    # Error Function for binary classification = 'binary_crossentropy'
    # Metrics = Metrics of Model / Check the performance of model
    model.compile(optimizer = 'Adam', loss = 'categorical_crossentropy', metrics
    return model

model = build_model()

model.summary()

history = model.fit(x_train, y_train, epochs = 16, batch_size = 10, validation_d

history = model.fit(x_scaler, y_train, epochs = 16, batch_size = 10, validation_
```



```
train_data.corr()
```

```
plt.figure(figsize=(20,20))  
sns.heatmap(train_data.corr())
```

```
train_data.shape
```

```
#to test variance of data whether it is highly variable  
train_data.var()
```

[Colab paid products](#) - [Cancel contracts here](#)

