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


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

# Correct file path to your file in Google Drive
file_path = '/content/drive/MyDrive/dataset/data.csv'

# Read the file using the correct variable
df = pd.read_csv(file_path)

display(df)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

| | Event | Time | Cell_length | DNA1 | DNA2 | CD45RA | CD133 | CD19 | CD22 | CD11b | ... | CD117 | CD49d | I |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2693.00 | 22 | 4.391057 | 4.617262 | 0.162691 | -0.029585 | -0.006696 | 0.066388 | -0.009184 | ... | 0.053050 | 0.853505 | 1.( |
| **1** | 2 | 3736.00 | 35 | 4.340481 | 4.816692 | 0.701349 | -0.038280 | -0.016654 | 0.074409 | 0.808031 | ... | 0.089660 | 0.197818 | 0.4 |
| **2** | 3 | 7015.00 | 32 | 3.838727 | 4.386369 | 0.603568 | -0.032216 | 0.073855 | -0.042977 | -0.001881 | ... | 0.046222 | 2.586670 | 1.3 |
| **3** | 4 | 7099.00 | 29 | 4.255806 | 4.830048 | 0.433747 | -0.027611 | -0.017661 | -0.044072 | 0.733698 | ... | 0.066470 | 1.338669 | 0.1 |
| **4** | 5 | 7700.00 | 25 | 3.976909 | 4.506433 | -0.008809 | -0.030297 | 0.080423 | 0.495791 | 1.107627 | ... | -0.006223 | 0.180924 | 0.1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **265622** | 265623 | 707951.44 | 41 | 6.826629 | 7.133022 | 1.474081 | -0.019174 | -0.055620 | -0.007261 | 0.063395 | ... | -0.011105 | 0.533736 | 0.1 |
| **265623** | 265624 | 708145.44 | 45 | 6.787791 | 7.154026 | 0.116755 | -0.056213 | -0.008864 | -0.035158 | -0.041845 | ... | 0.143869 | 1.269464 | 0.( |
| **265624** | 265625 | 708398.44 | 41 | 6.889866 | 7.141219 | 0.684921 | -0.006264 | -0.026111 | -0.030837 | -0.034641 | ... | 0.087102 | -0.055912 | 0.5 |
| **265625** | 265626 | 708585.44 | 39 | 6.865218 | 7.144353 | 0.288761 | -0.011310 | -0.048786 | 0.073983 | -0.031787 | ... | -0.047971 | 0.101955 | 6.2 |
| **265626** | 265627 | 709122.44 | 41 | 6.887820 | 7.127359 | 0.360753 | 0.128604 | -0.006934 | 0.109846 | 3.864711 | ... | 0.080195 | 0.037962 | 3.( |

265627 rows × 42 columns

```
print("Basic Structure of the Data:")
display(df)
```

```
Basic Structure of the Data:
```

| | Event | Time | Cell_length | DNA1 | DNA2 | CD45RA | CD133 | CD19 | CD22 | CD11b | ... | CD117 | CD49d | I |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2693.00 | 22 | 4.391057 | 4.617262 | 0.162691 | -0.029585 | -0.006696 | 0.066388 | -0.009184 | ... | 0.053050 | 0.853505 | 1.( |
| **1** | 2 | 3736.00 | 35 | 4.340481 | 4.816692 | 0.701349 | -0.038280 | -0.016654 | 0.074409 | 0.808031 | ... | 0.089660 | 0.197818 | 0.4 |
| **2** | 3 | 7015.00 | 32 | 3.838727 | 4.386369 | 0.603568 | -0.032216 | 0.073855 | -0.042977 | -0.001881 | ... | 0.046222 | 2.586670 | 1.3 |
| **3** | 4 | 7099.00 | 29 | 4.255806 | 4.830048 | 0.433747 | -0.027611 | -0.017661 | -0.044072 | 0.733698 | ... | 0.066470 | 1.338669 | 0.1 |
| **4** | 5 | 7700.00 | 25 | 3.976909 | 4.506433 | -0.008809 | -0.030297 | 0.080423 | 0.495791 | 1.107627 | ... | -0.006223 | 0.180924 | 0.1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **265622** | 265623 | 707951.44 | 41 | 6.826629 | 7.133022 | 1.474081 | -0.019174 | -0.055620 | -0.007261 | 0.063395 | ... | -0.011105 | 0.533736 | 0.1 |
| **265623** | 265624 | 708145.44 | 45 | 6.787791 | 7.154026 | 0.116755 | -0.056213 | -0.008864 | -0.035158 | -0.041845 | ... | 0.143869 | 1.269464 | 0.( |
| **265624** | 265625 | 708398.44 | 41 | 6.889866 | 7.141219 | 0.684921 | -0.006264 | -0.026111 | -0.030837 | -0.034641 | ... | 0.087102 | -0.055912 | 0.5 |
| **265625** | 265626 | 708585.44 | 39 | 6.865218 | 7.144353 | 0.288761 | -0.011310 | -0.048786 | 0.073983 | -0.031787 | ... | -0.047971 | 0.101955 | 6.2 |
| **265626** | 265627 | 709122.44 | 41 | 6.887820 | 7.127359 | 0.360753 | 0.128604 | -0.006934 | 0.109846 | 3.864711 | ... | 0.080195 | 0.037962 | 3.( |

265627 rows × 42 columns

```
print("\nData Information:")
display(df.info())
```

```
Data Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 265627 entries, 0 to 265626
Data columns (total 42 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   Event         265627 non-null  int64  
 1   Time          265627 non-null  float64
 2   Cell_length   265627 non-null  int64  
 3   DNA1          265627 non-null  float64
 4   DNA2          265627 non-null  float64
 5   CD45RA        265627 non-null  float64
 6   CD133         265627 non-null  float64
 7   CD19          265627 non-null  float64
 8   CD22          265627 non-null  float64
 9   CD11b         265627 non-null  float64
 10  CD4           265627 non-null  float64
 11  CD8           265627 non-null  float64
 12  CD34          265627 non-null  float64
 13  Flt3          265627 non-null  float64
 14  CD20          265627 non-null  float64
 15  CXCR4         265627 non-null  float64
 16  CD235ab       265627 non-null  float64
 17  CD45          265627 non-null  float64
 18  CD123         265627 non-null  float64
 19  CD321         265627 non-null  float64
 20  CD14          265627 non-null  float64
 21  CD33          265627 non-null  float64
 22  CD47          265627 non-null  float64
 23  CD11c         265627 non-null  float64
 24  CD7           265627 non-null  float64
 25  CD15          265627 non-null  float64
 26  CD16          265627 non-null  float64
 27  CD44          265627 non-null  float64
 28  CD38          265627 non-null  float64
 29  CD13          265627 non-null  float64
 30  CD3           265627 non-null  float64
 31  CD61          265627 non-null  float64
 32  CD117         265627 non-null  float64
 33  CD49d         265627 non-null  float64
 34  HLA-DR        265627 non-null  float64
 35  CD64          265627 non-null  float64
 36  CD41          265627 non-null  float64
 37  Viability     265627 non-null  float64
 38  file_number   265627 non-null  float64
 39  event_number  265627 non-null  int64  
 40  label         104184 non-null  float64
 41  individual    265627 non-null  int64  
dtypes: float64(38), int64(4)
memory usage: 85.1 MB
None
```

```
print("\nMissing Values:")
display(df.isnull().sum())
```

Missing Values:

| | 0 |
|---|---|
| Event | 0 |
| Time | 0 |
| Cell_length | 0 |
| DNA1 | 0 |
| DNA2 | 0 |
| CD45RA | 0 |
| CD133 | 0 |
| CD19 | 0 |
| CD22 | 0 |
| CD11b | 0 |
| CD4 | 0 |
| CD8 | 0 |
| CD34 | 0 |
| Flt3 | 0 |
| CD20 | 0 |
| CXCR4 | 0 |
| CD235ab | 0 |
| CD45 | 0 |
| CD123 | 0 |
| CD321 | 0 |
| CD14 | 0 |
| CD33 | 0 |
| CD47 | 0 |
| CD11c | 0 |
| CD7 | 0 |
| CD15 | 0 |
| CD16 | 0 |
| CD44 | 0 |
| CD38 | 0 |
| CD13 | 0 |
| CD3 | 0 |
| CD61 | 0 |
| CD117 | 0 |
| CD49d | 0 |
| HLA-DR | 0 |
| CD64 | 0 |
| CD41 | 0 |
| Viability | 0 |
| file_number | 0 |
| event_number | 0 |
| label | 161443 |
| individual | 0 |

dtype: int64

```
print("\nMissing Values:")
missing_values = df.isnull().sum()
missing_percentage = (missing_values / len(df)) * 100
missing_df = pd.DataFrame({'Missing Values': missing_values, 'Percentage': missing_percentage})
display(missing_df[missing_df['Missing Values'] > 0])
```

```
Missing Values:
```

| | Missing Values | Percentage |
|---|---|---|
| **label** | 161443 | 60.778084 |

```
df.nunique()
```

| | 0 |
|---|---|
| Event | 265627 |
| Time | 147364 |
| Cell_length | 56 |
| DNA1 | 158903 |
| DNA2 | 157946 |
| CD45RA | 161079 |
| CD133 | 161181 |
| CD19 | 161229 |
| CD22 | 161247 |
| CD11b | 161236 |
| CD4 | 161258 |
| CD8 | 161213 |
| CD34 | 161214 |
| Flt3 | 161204 |
| CD20 | 161224 |
| CXCR4 | 161121 |
| CD235ab | 161039 |
| CD45 | 158665 |
| CD123 | 161241 |
| CD321 | 160654 |
| CD14 | 161193 |
| CD33 | 161233 |
| CD47 | 160534 |
| CD11c | 161251 |
| CD7 | 161257 |
| CD15 | 161194 |
| CD16 | 161202 |
| CD44 | 160151 |
| CD38 | 161143 |
| CD13 | 161232 |
| CD3 | 160856 |
| CD61 | 161241 |
| CD117 | 161212 |
| CD49d | 161040 |
| HLA-DR | 161214 |
| CD64 | 161238 |
| CD41 | 161232 |
| Viability | 161149 |
| file_number | 2 |
| event_number | 151613 |
| label | 14 |
| individual | 2 |

**dtype:** int64

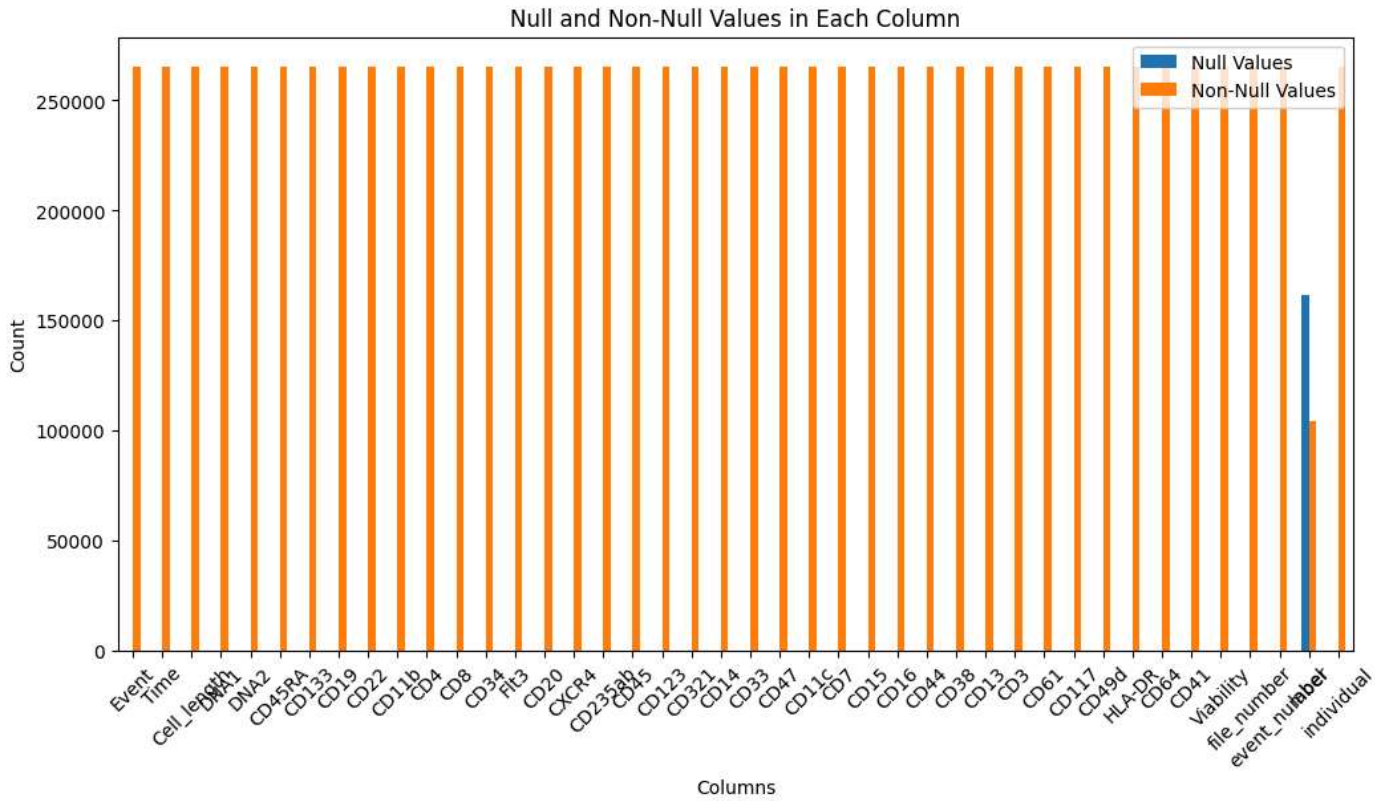
```
print("\nDescriptive Statistics:")
display(df.describe())
```

Descriptive Statistics:

| | Event | Time | Cell_length | DNA1 | DNA2 | CD45RA | CD133 | CD19 | CD22 |
|---|---|---|---|---|---|---|---|---|---|
| count | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 | 265627.000000 |
| mean | 132814.000000 | 272948.345014 | 34.450572 | 4.606956 | 5.198308 | 0.688127 | 0.145960 | 0.509301 | 0.397323 |
| std | 76680.054314 | 171220.139430 | 11.446694 | 1.312831 | 1.150357 | 0.609105 | 0.259267 | 0.857462 | 0.762126 |
| min | 1.000000 | 1.000000 | 10.000000 | 2.786488 | 2.236450 | -0.057305 | -0.058081 | -0.058089 | -0.057342 |
| 25% | 66407.500000 | 120196.000000 | 26.000000 | 3.700023 | 4.407822 | 0.204625 | -0.022935 | -0.018838 | -0.020689 |
| 50% | 132814.000000 | 253276.000000 | 33.000000 | 4.022127 | 4.698415 | 0.549387 | 0.025353 | 0.075210 | 0.058790 |
| 75% | 199220.500000 | 424502.500000 | 41.000000 | 6.353313 | 6.766268 | 1.031198 | 0.224299 | 0.548386 | 0.386481 |
| max | 265627.000000 | 709122.440000 | 65.000000 | 7.001489 | 7.472308 | 6.691197 | 5.527494 | 4.990085 | 5.160477 |

8 rows × 42 columns

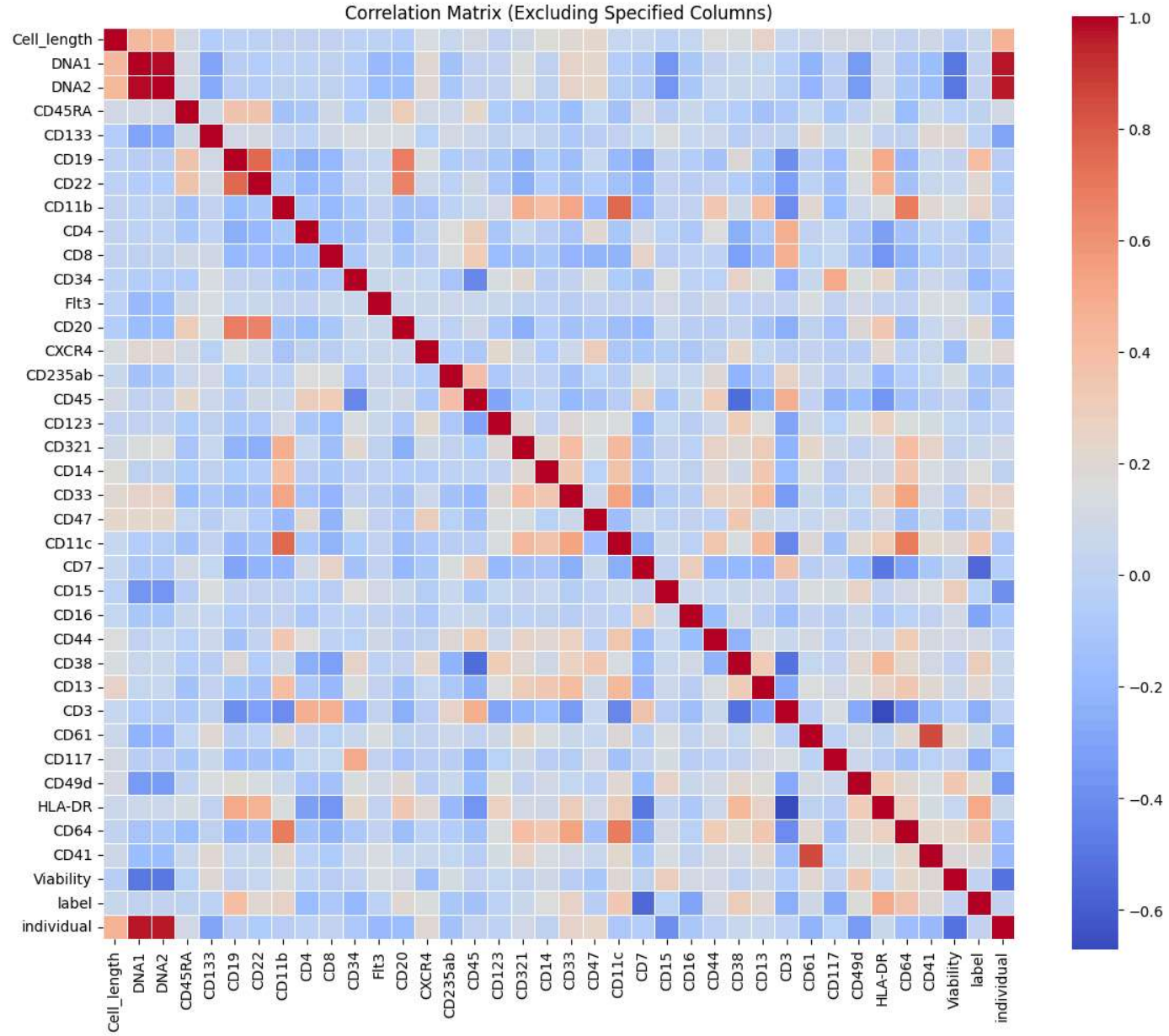
```
null_counts = df.isnull().sum()
non_null_counts = df.notnull().sum()


plot_data = pd.DataFrame({
    'Null Values': null_counts,
    'Non-Null Values': non_null_counts
})

plot_data.plot(kind='bar', figsize=(12, 6))
plt.title('Null and Non-Null Values in Each Column')
plt.xlabel('Columns')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(loc='upper right')
plt.show()
```




```
print("\nCorrelation Matrix:")
columns_to_exclude = ['Event', 'Time', 'file_number', 'event_number']
filtered_df = df.drop(columns=columns_to_exclude)
```

```
correlation_matrix = filtered_df.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm', square=True, cbar=True, fmt='', linewidths=0.5)
plt.title('Correlation Matrix (Excluding Specified Columns)')
plt.tight_layout()
plt.show()
```

```
Correlation Matrix:
```

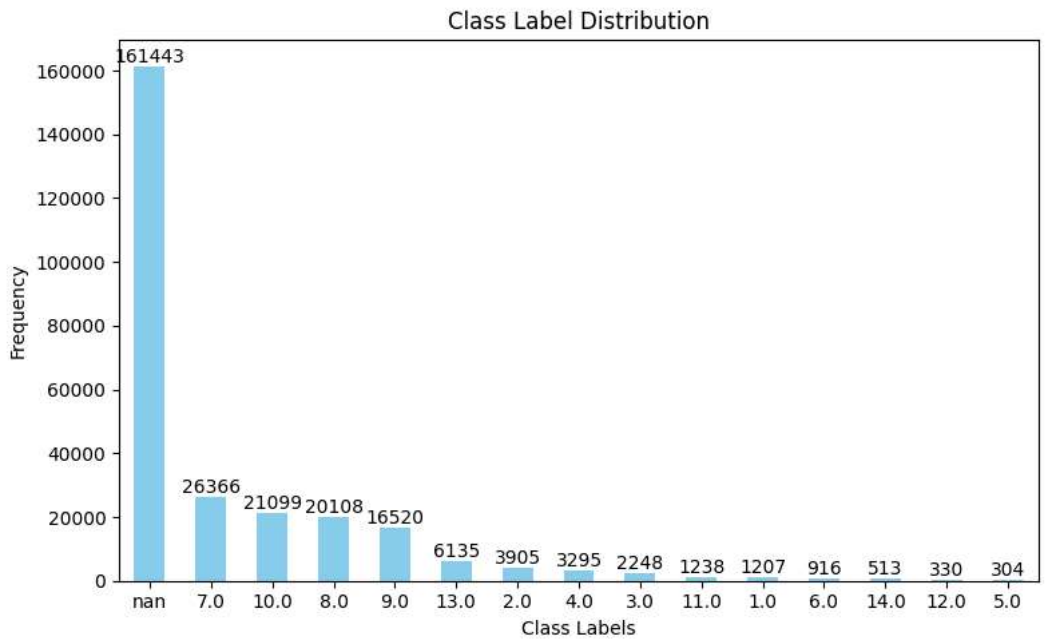



```
label_distribution = df['label'].value_counts(dropna=False)
print("Class Label Distribution:")
print(label_distribution)
```

```
Class Label Distribution:
label
NaN     161443
7.0      26366
10.0     21099
8.0      20108
9.0      16520
13.0      6135
2.0       3905
```

```
4.0       3295
3.0       2248
11.0      1238
1.0       1207
6.0        916
14.0       513
12.0       330
5.0        304
Name: count, dtype: int64
```

```
label_distribution = df['label'].value_counts(dropna=False)

plt.figure(figsize=(8, 5))
bars = label_distribution.plot(kind='bar', color='skyblue')
plt.title('Class Label Distribution')
plt.xlabel('Class Labels')
plt.ylabel('Frequency')
plt.xticks(rotation=0)

for bar in bars.patches:
    bars.annotate(bar.get_height(),
                  (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                  ha='center',
                  va='bottom')

plt.tight_layout()
plt.show()
```

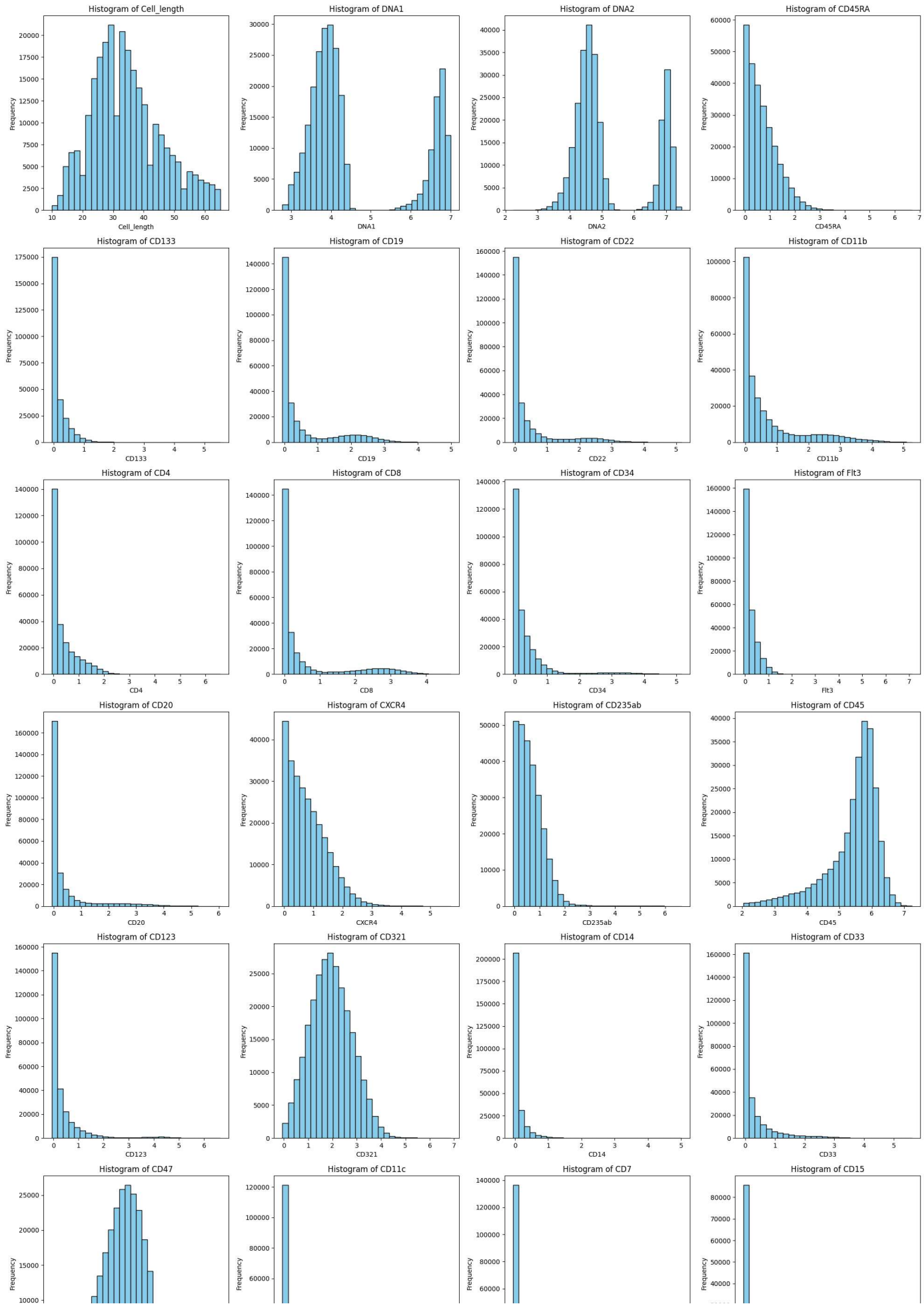


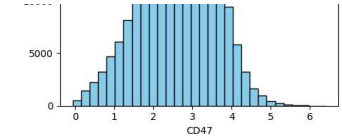

```
corr_matrix = df.corr()

corr_pairs = corr_matrix.unstack().sort_values(ascending=False)

corr_pairs = corr_pairs[corr_pairs.index.get_level_values(0) != corr_pairs.index.get_level_values(1)]

top_5_corr_pairs = corr_pairs.head(10)

print("Top 10 correlated column pairs:")
for (col1, col2), corr_value in top_5_corr_pairs.items():
    print(f"{col1} and {col2}: Correlation = {corr_value:.2f}")
```

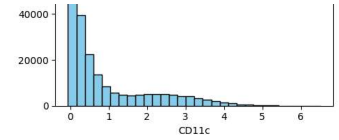
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-cf37f0cc239f> in <cell line: 1>()
----> 1 corr_matrix = df.corr()
      2
      3 corr_pairs = corr_matrix.unstack().sort_values(ascending=False)
      4
      5 corr_pairs = corr_pairs[corr_pairs.index.get_level_values(0) != corr_pairs.index.get_level_values(1)]

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

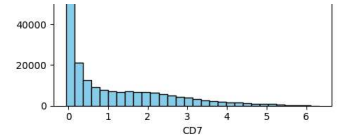
Next steps: Explain error

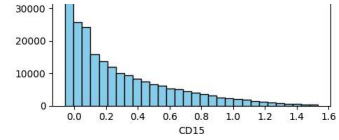
```
columns_to_exclude = ['Event', 'Time', 'file_number', 'event_number', 'individual']
columns_to_plot = [col for col in df.columns if col not in columns_to_exclude]

num_cols = 4
num_rows = (len(columns_to_plot) + num_cols - 1) // num_cols

plt.figure(figsize=(20, num_rows * 5))

for i, column in enumerate(columns_to_plot):
    plt.subplot(num_rows, num_cols, i + 1)
    plt.hist(df[column], bins=30, color='skyblue', edgecolor='black')
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

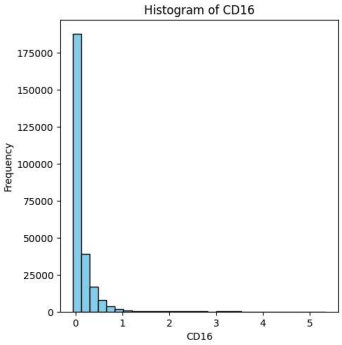

Histogram of Cell_length
Cell_length
Frequency
Histogram of DNA1
DNA1
Frequency
Histogram of DNA2
DNA2
Frequency
Histogram of CD45RA
CD45RA
Frequency
Histogram of CD133
CD133
Frequency
Histogram of CD19
CD19
Frequency
Histogram of CD22
CD22
Frequency
Histogram of CD11b
CD11b
Frequency
Histogram of CD4
CD4
Frequency
Histogram of CD8
CD8
Frequency
Histogram of CD34
CD34
Frequency
Histogram of Flt3
Flt3
Frequency
Histogram of CD20
CD20
Frequency
Histogram of CXCR4
CXCR4
Frequency
Histogram of CD235ab
CD235ab
Frequency
Histogram of CD45
CD45
Frequency
Histogram of CD123
CD123
Frequency
Histogram of CD321
CD321
Frequency
Histogram of CD14
CD14
Frequency
Histogram of CD33
CD33
Frequency
Histogram of CD47
Frequency
Histogram of CD11c
Frequency
Histogram of CD7
Frequency
Histogram of CD15
Frequency

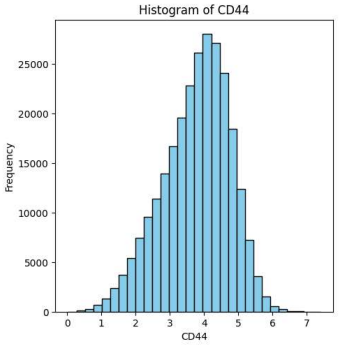

CD47

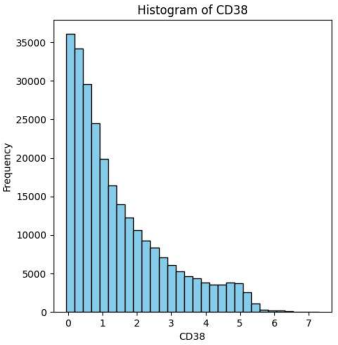

CD11c

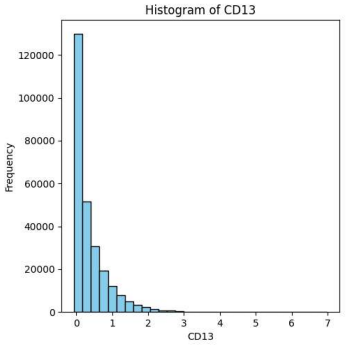

CD7

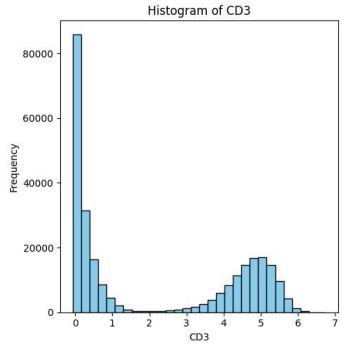

CD15

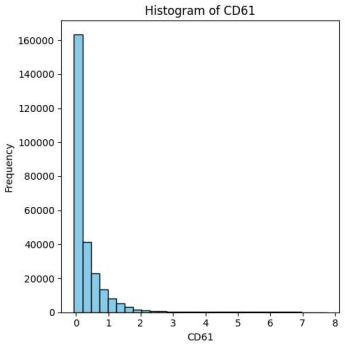

Histogram of CD16
Frequency
CD16

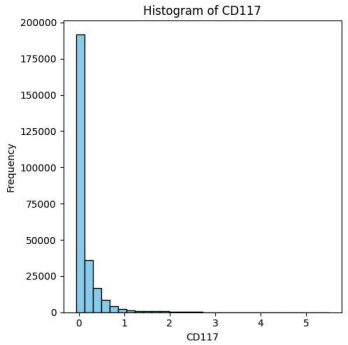

Histogram of CD44
Frequency
CD44

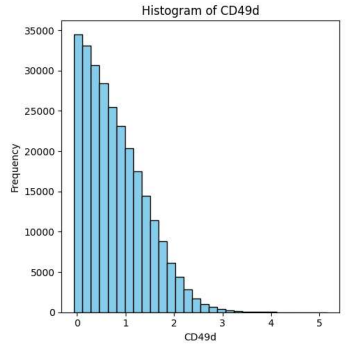

Histogram of CD38
Frequency
CD38

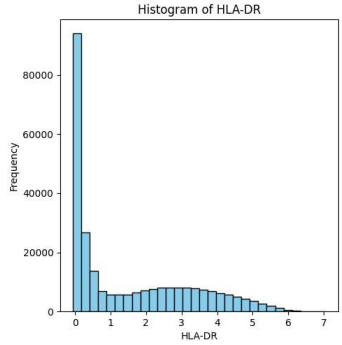

Histogram of CD13
Frequency
CD13

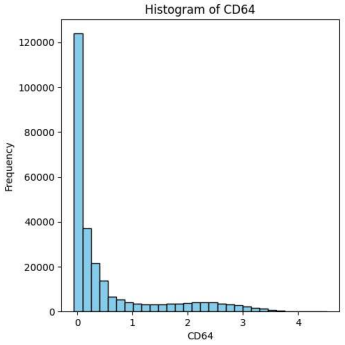

Histogram of CD3
Frequency
CD3

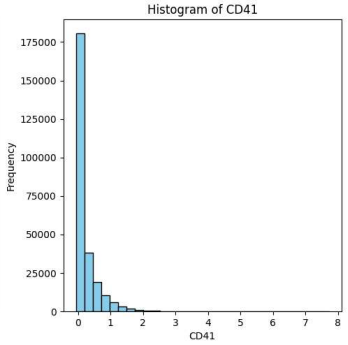

Histogram of CD61
Frequency
CD61

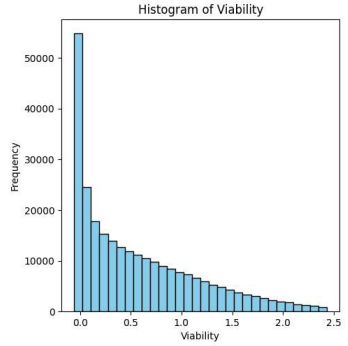

Histogram of CD117
Frequency
CD117

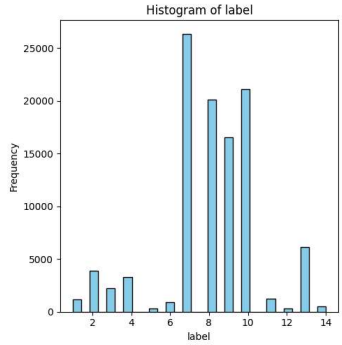

Histogram of CD49d
Frequency
CD49d


Histogram of HLA-DR
Frequency
HLA-DR

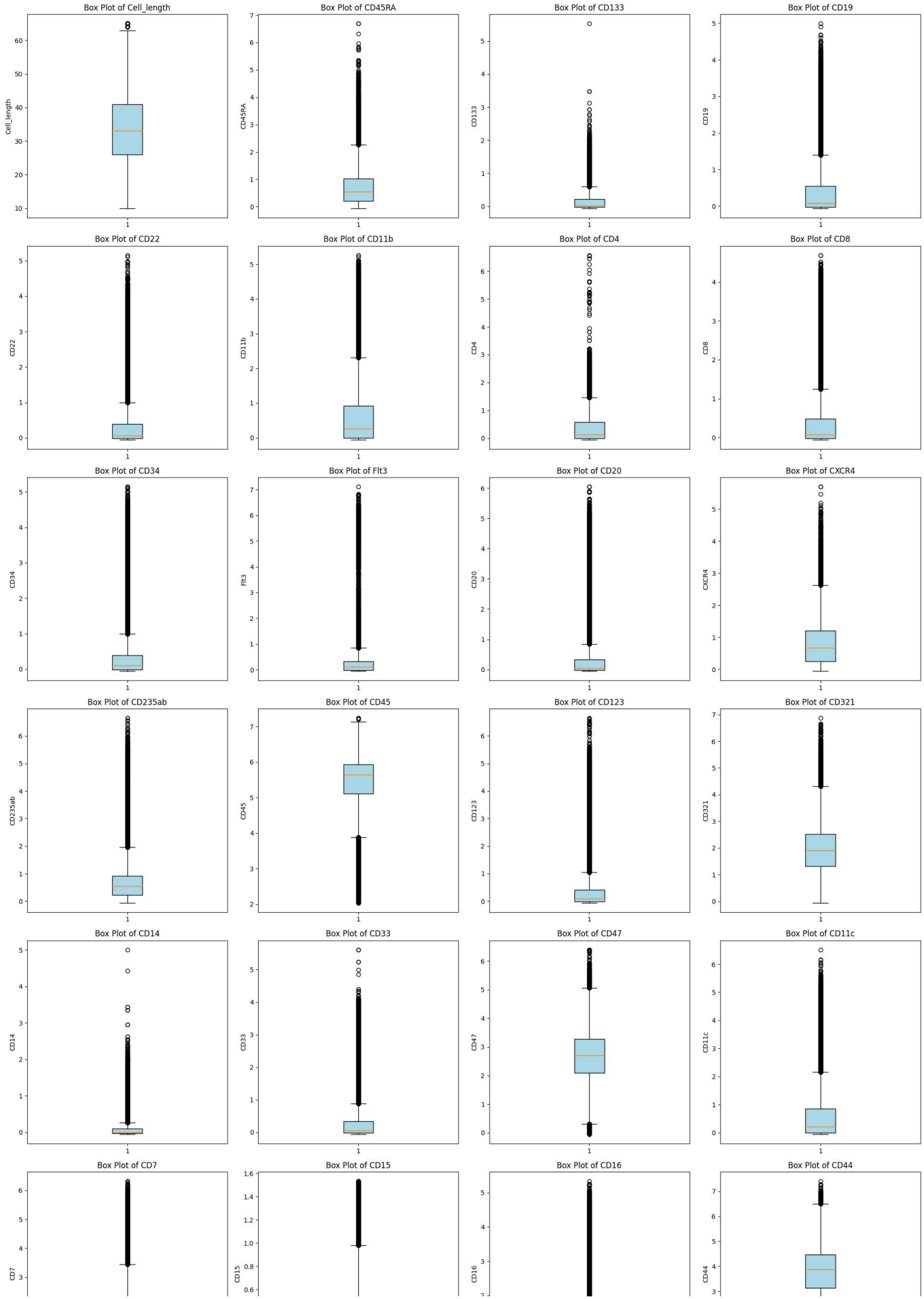

Histogram of CD64
Frequency
CD64

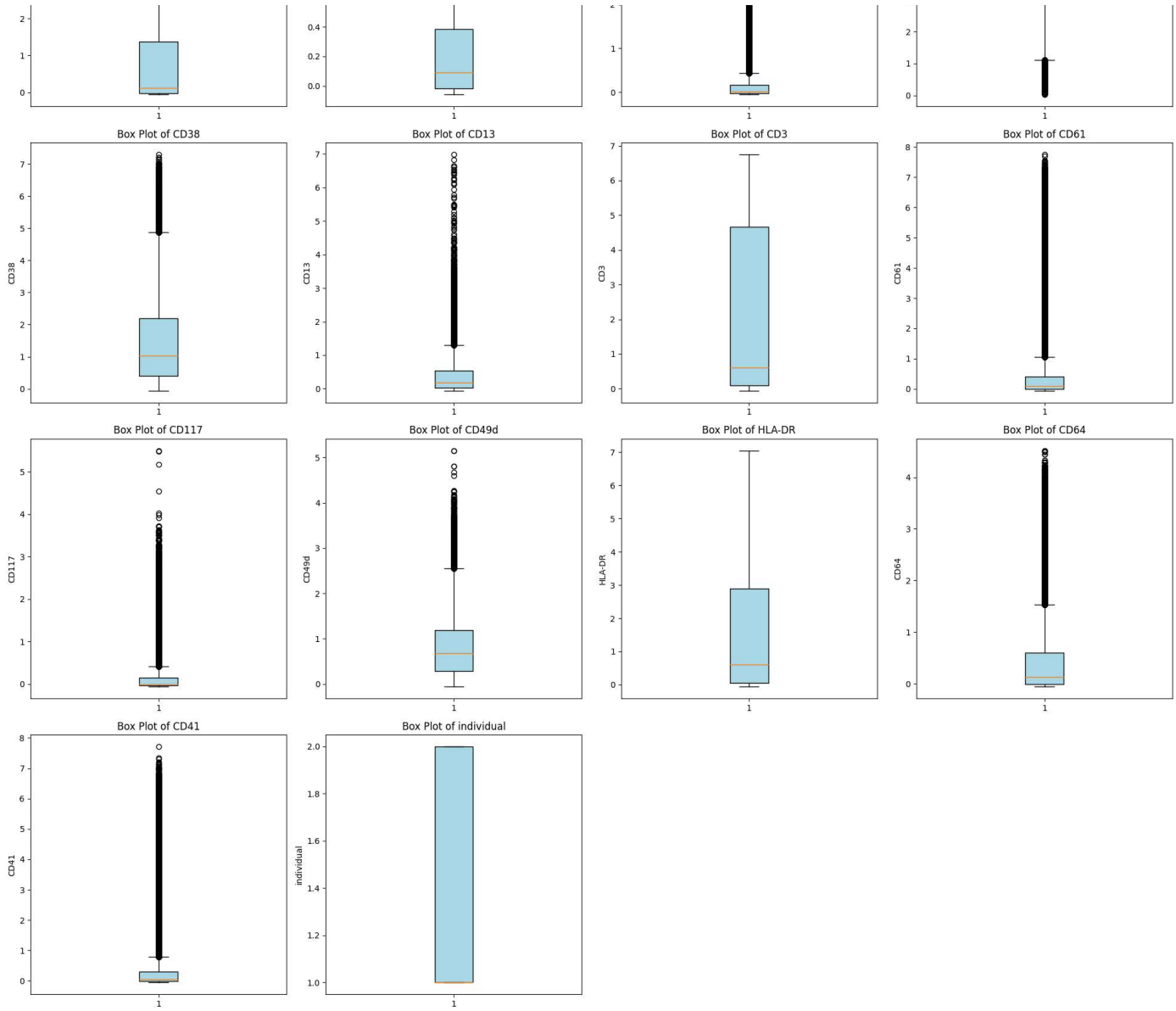

Histogram of CD41
Frequency
CD41


Histogram of Viability
Frequency
Viability

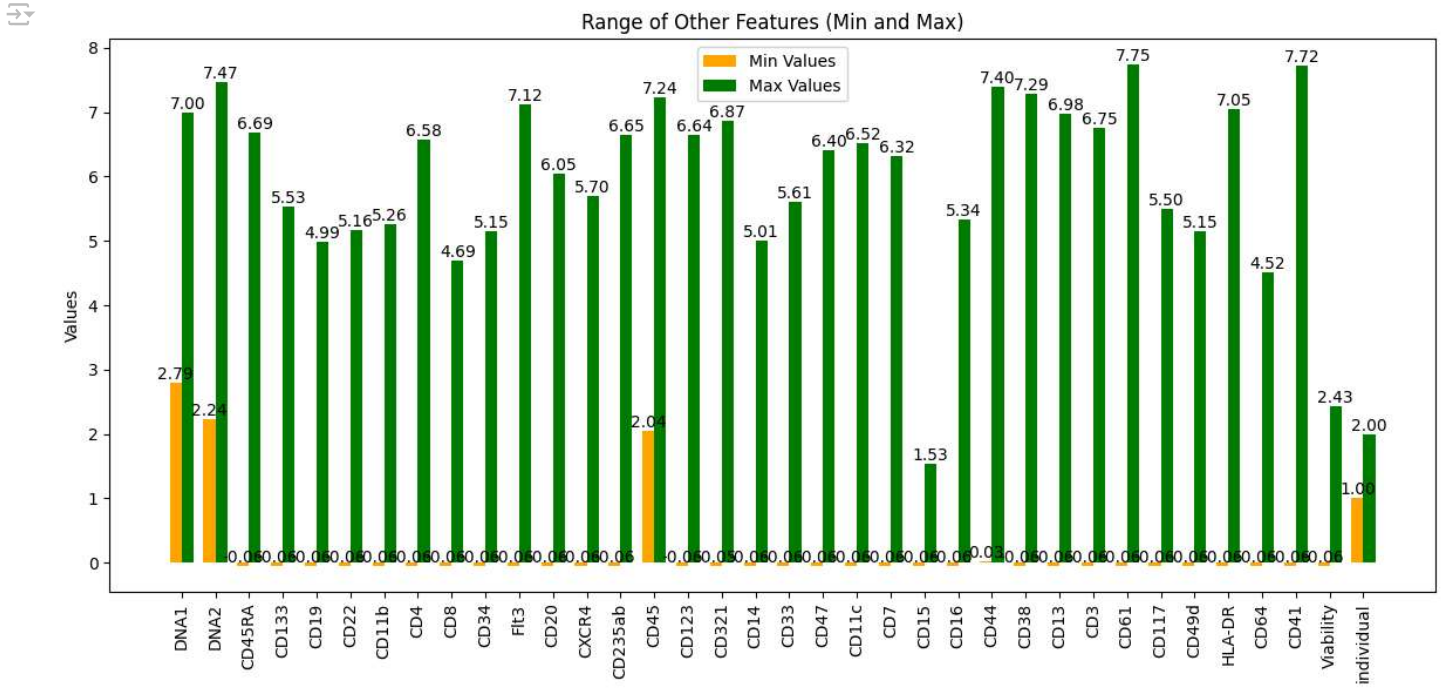

Histogram of label
Frequency
label

```
columns_to_exclude = ['Event', 'DNA1', 'DNA2', 'Time', 'file_number', 'event_number', 'Viability', 'label']
columns_to_plot = [col for col in df.columns if col not in columns_to_exclude]

num_cols = 4
num_rows = (len(columns_to_plot) + num_cols - 1) // num_cols

plt.figure(figsize=(20, num_rows * 5))

for i, column in enumerate(columns_to_plot):
    plt.subplot(num_rows, num_cols, i + 1)
    plt.boxplot(df[column], patch_artist=True, boxprops=dict(facecolor='lightblue'))
    plt.title(f'Box Plot of {column}')
    plt.ylabel(column)

plt.tight_layout()
plt.show()
```


Box Plot of Cell_length
Box Plot of CD45RA
Box Plot of CD133
Box Plot of CD19
Box Plot of CD22
Box Plot of CD11b
Box Plot of CD4
Box Plot of CD8
Box Plot of CD34
Box Plot of Flt3
Box Plot of CD20
Box Plot of CXCR4
Box Plot of CD235ab
Box Plot of CD45
Box Plot of CD123
Box Plot of CD321
Box Plot of CD14
Box Plot of CD33
Box Plot of CD47
Box Plot of CD11c
Box Plot of CD7
Box Plot of CD15
Box Plot of CD16
Box Plot of CD44


Box Plot of CD38
CD38
Box Plot of CD13
CD13
Box Plot of CD3
CD3
Box Plot of CD61
CD61
Box Plot of CD117
CD117
Box Plot of CD49d
CD49d
Box Plot of HLA-DR
HLA-DR
Box Plot of CD64
CD64
Box Plot of CD41
CD41
Box Plot of individual
individual

```
high_value_columns = ['Time', 'Event', 'event_number', 'label', 'Cell_length', 'file_number']
other_columns = [col for col in df.columns if col not in high_value_columns]

min_values_other = df[other_columns].min()
max_values_other = df[other_columns].max()

fig, ax = plt.subplots(figsize=(12, 6))

x_other = np.arange(len(other_columns))
width = 0.35

bars_min_other = ax.bar(x_other - width/2, min_values_other, width, label='Min Values', color='orange')
bars_max_other = ax.bar(x_other + width/2, max_values_other, width, label='Max Values', color='green')

ax.set_ylabel('Values')
ax.set_title('Range of Other Features (Min and Max)')
ax.set_xticks(x_other)
ax.set_xticklabels(other_columns, rotation=90)
ax.legend()

for bar in bars_min_other:
    yval = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2, yval, f'{yval:.2f}', ha='center', va='bottom')

for bar in bars_max_other:
    yval = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2, yval, f'{yval:.2f}', ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

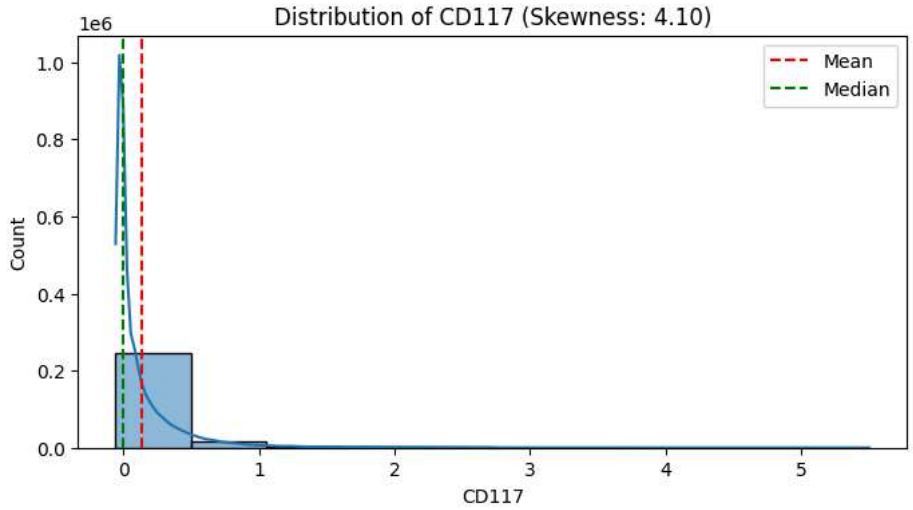


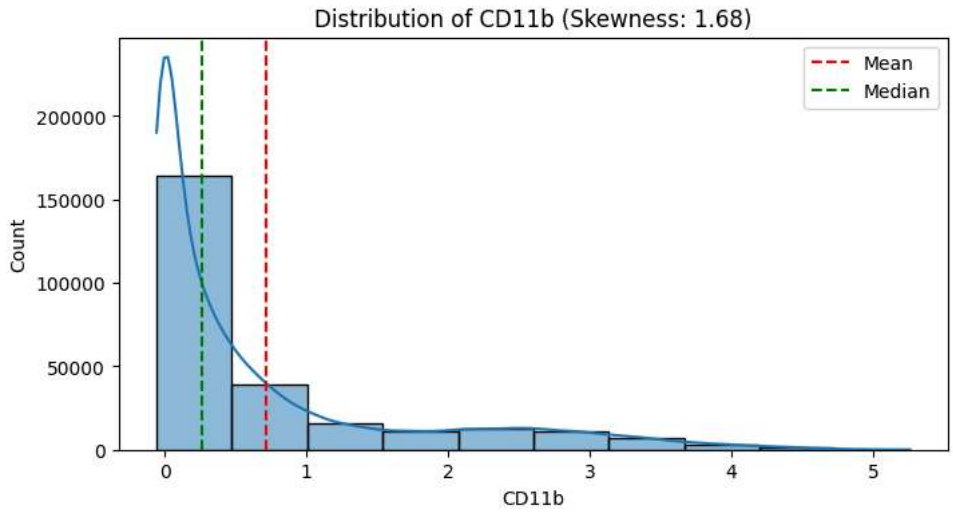

```
print("Skewness")
from scipy.stats import skew
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Exclude specific columns from the skewness calculation
columns_to_exclude = ['Event', 'Time', 'file_number', 'event_number', 'individual']
numerical_columns = df.select_dtypes(include=['number']).columns.difference(columns_to_exclude)

# Calculate skewness for the specified numerical columns
skewness = df[numerical_columns].apply(skew)
```

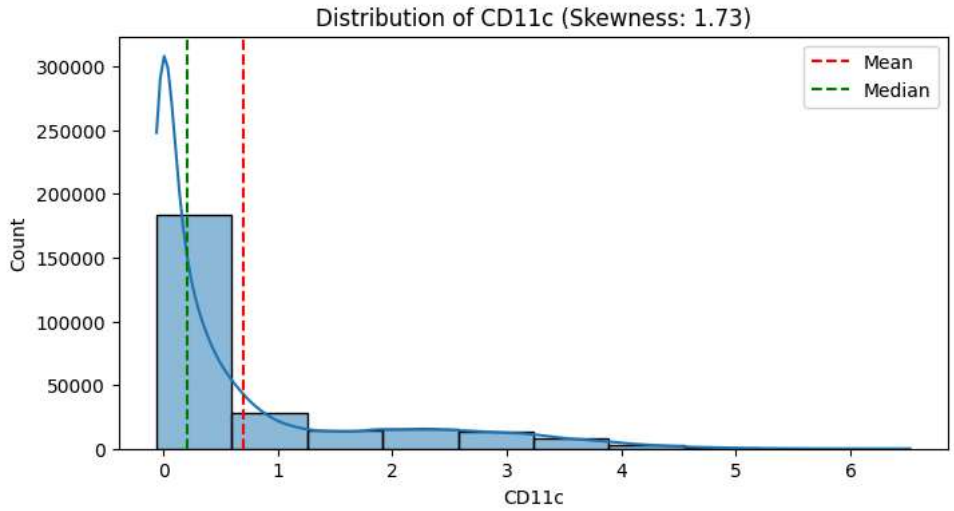
```
# Function to categorize skewness
def categorize_skewness(value):
    if value > 0.5:
        return 'Right-skewed'
    elif value < -0.5:
        return 'Left-skewed'
    else:
        return 'Approximately symmetrical'

# Apply the categorization
skewness_category = skewness.apply(categorize_skewness)

# Display skewness and its categorization
skewness_df = pd.DataFrame({'Skewness': skewness, 'Category': skewness_category})
print(skewness_df)

# Plot histograms for each numerical column, excluding specified columns
for col in numerical_columns:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], bins=10, kde=True)
    plt.title(f'Distribution of {col} (Skewness: {skewness[col]:.2f})')
    plt.axvline(df[col].mean(), color='red', linestyle='--', label='Mean')
    plt.axvline(df[col].median(), color='green', linestyle='--', label='Median')
    plt.legend()
    plt.show()
```

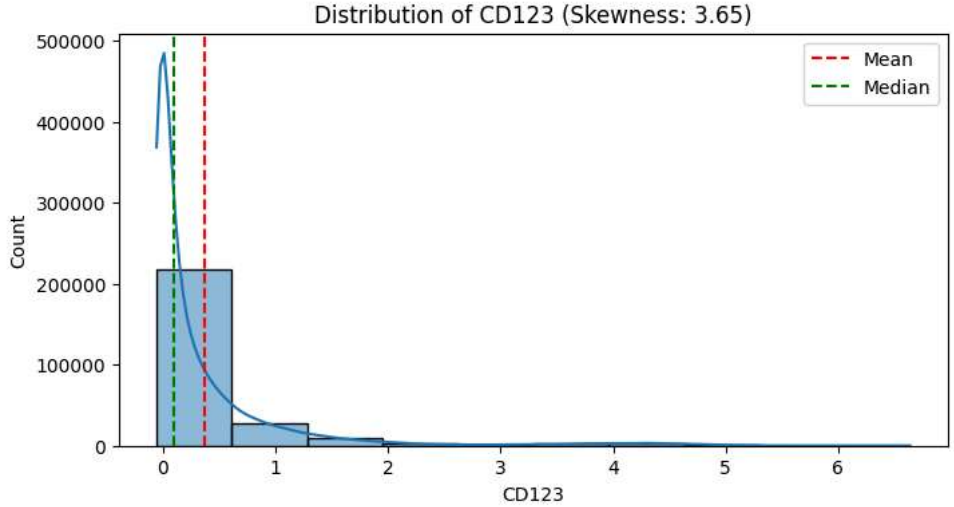
Skewness

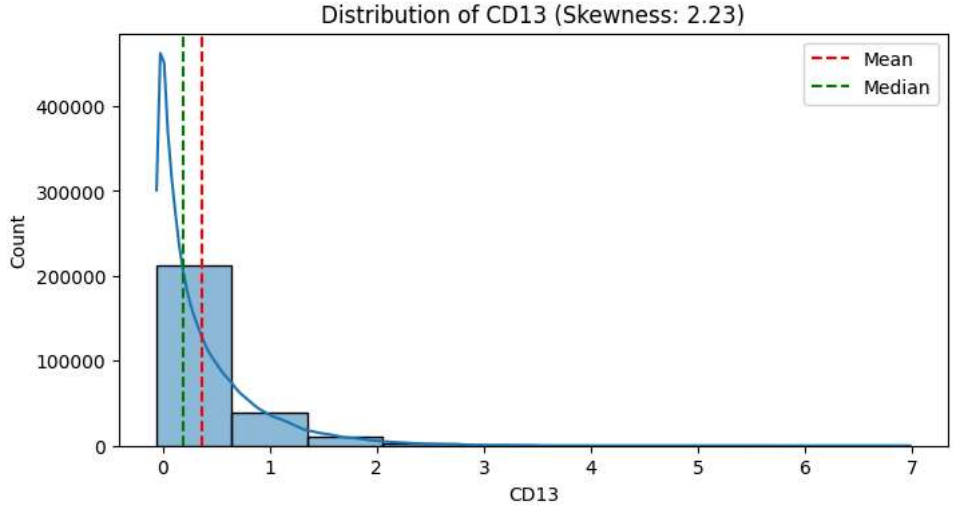
| | Skewness | Category |
|---|---|---|
| CD117 | 4.097508 | Right-skewed |
| CD11b | 1.679089 | Right-skewed |
| CD11c | 1.733888 | Right-skewed |
| CD123 | 3.648890 | Right-skewed |
| CD13 | 2.234311 | Right-skewed |
| CD133 | 2.141953 | Right-skewed |
| CD14 | 3.609006 | Right-skewed |
| CD15 | 1.445147 | Right-skewed |
| CD16 | 5.733203 | Right-skewed |
| CD19 | 1.682609 | Right-skewed |
| CD20 | 2.754699 | Right-skewed |
| CD22 | 2.283181 | Right-skewed |
| CD235ab | 2.001479 | Right-skewed |
| CD3 | 0.342239 | Approximately symmetrical |
| CD321 | 0.247097 | Approximately symmetrical |
| CD33 | 2.724977 | Right-skewed |
| CD34 | 3.492437 | Right-skewed |
| CD38 | 1.141482 | Right-skewed |
| CD4 | 1.622044 | Right-skewed |
| CD41 | 5.366314 | Right-skewed |
| CD44 | -0.431589 | Approximately symmetrical |
| CD45 | -1.484824 | Left-skewed |
| CD45RA | 1.191595 | Right-skewed |
| CD47 | -0.250323 | Approximately symmetrical |
| CD49d | 0.856805 | Right-skewed |
| CD61 | 4.894707 | Right-skewed |
| CD64 | 1.743733 | Right-skewed |
| CD7 | 1.606528 | Right-skewed |
| CD8 | 1.775713 | Right-skewed |
| CXCR4 | 0.955342 | Right-skewed |
| Cell_length | 0.527832 | Right-skewed |
| DNA1 | 0.845010 | Right-skewed |
| DNA2 | 0.779167 | Right-skewed |
| Flt3 | 7.098151 | Right-skewed |
| HLA-DR | 0.795359 | Right-skewed |
| Viability | 0.985417 | Right-skewed |
| label | NaN | Approximately symmetrical |

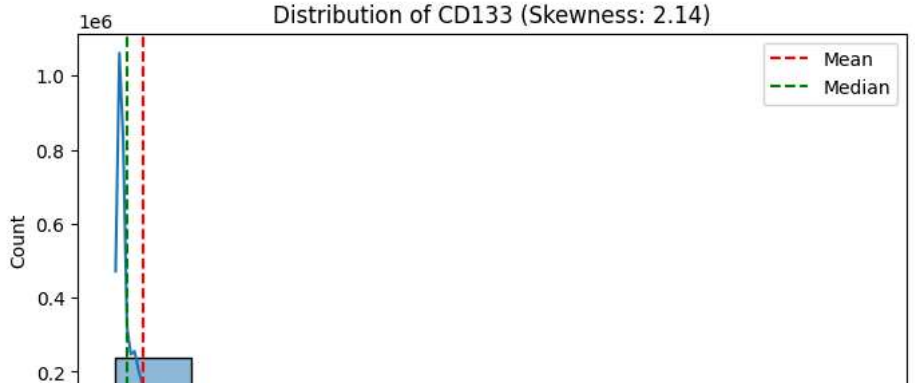


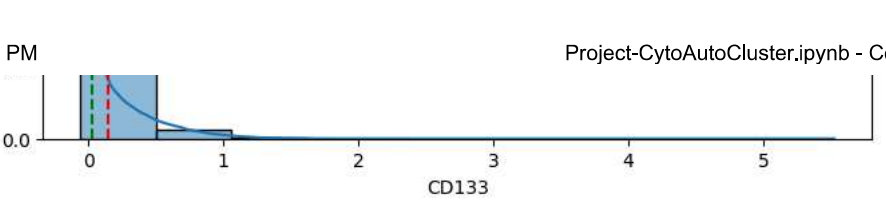



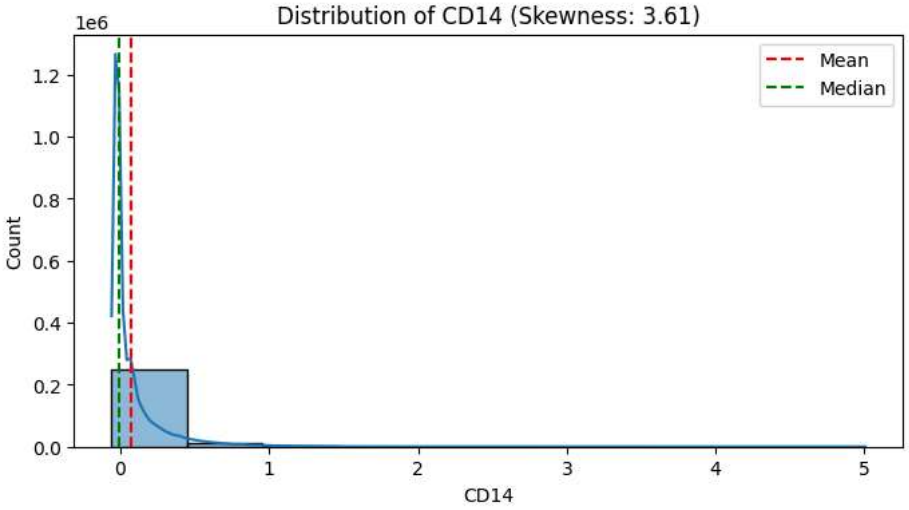

Distribution of CD11c (Skewness: 1.73)
Mean
Median
Count
CD11c

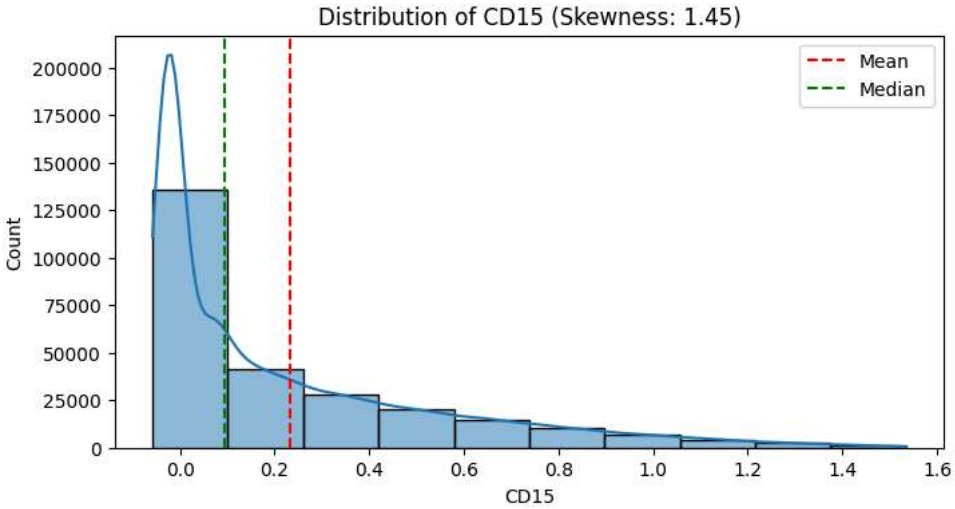

Distribution of CD123 (Skewness: 3.65)
Mean
Median
Count
CD123

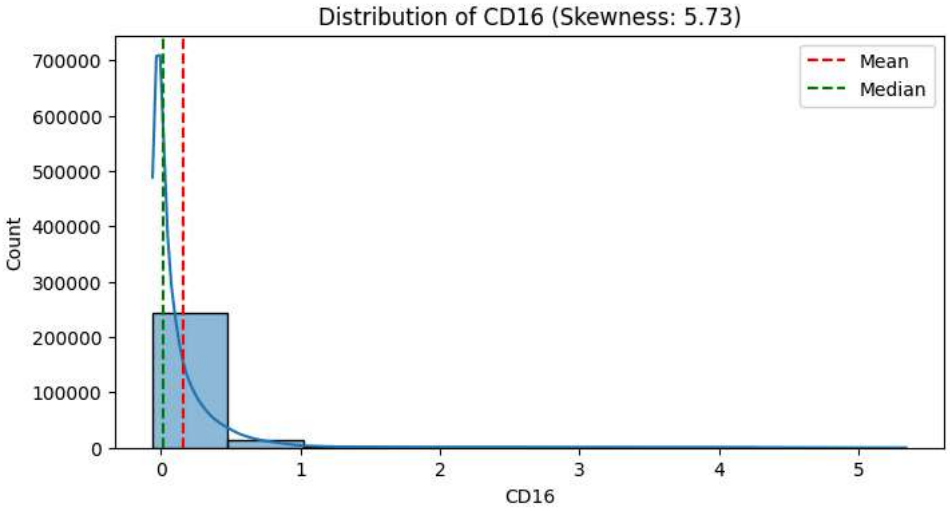

Distribution of CD13 (Skewness: 2.23)
Mean
Median
Count
CD13

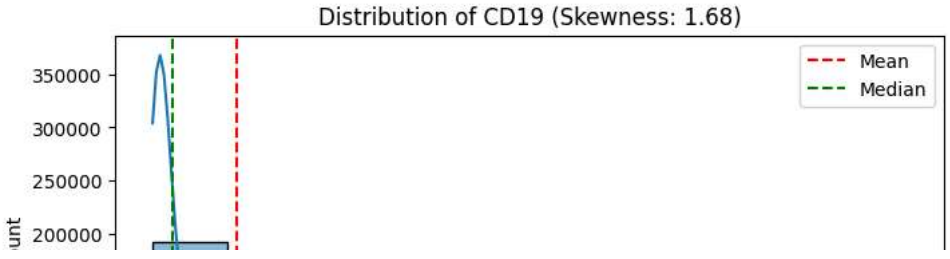

Distribution of CD133 (Skewness: 2.14)
1e6
Mean
Median
Count

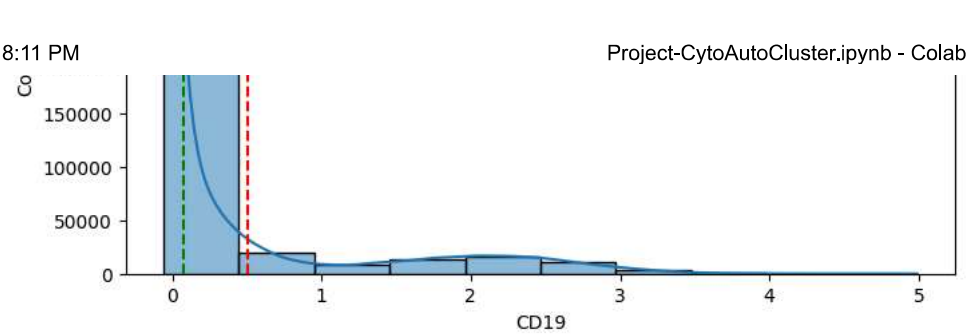

0.0
0
1
2
3
4
5
CD133

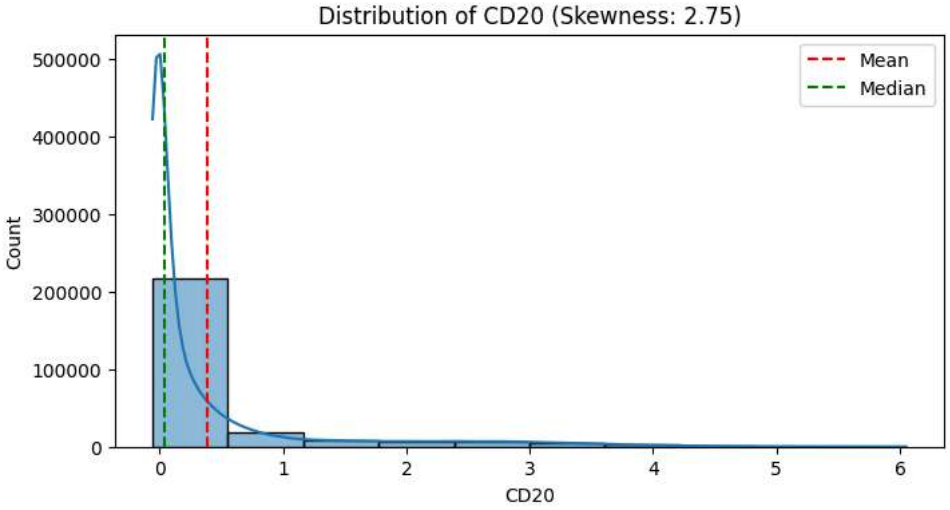

Distribution of CD14 (Skewness: 3.61)
1e6
Mean
Median
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Count
0
1
2
3
4
5
CD14

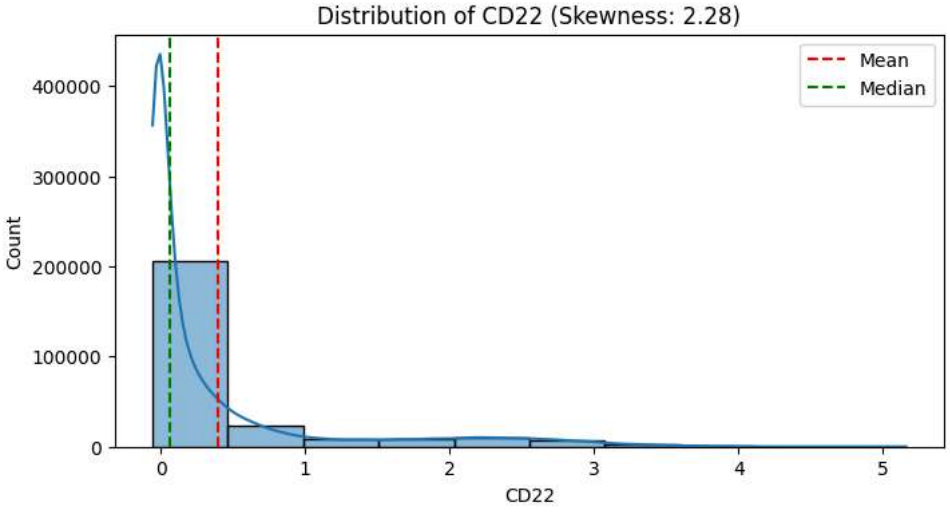

Distribution of CD15 (Skewness: 1.45)
Mean
Median
200000
175000
150000
125000
100000
75000
50000
25000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
CD15

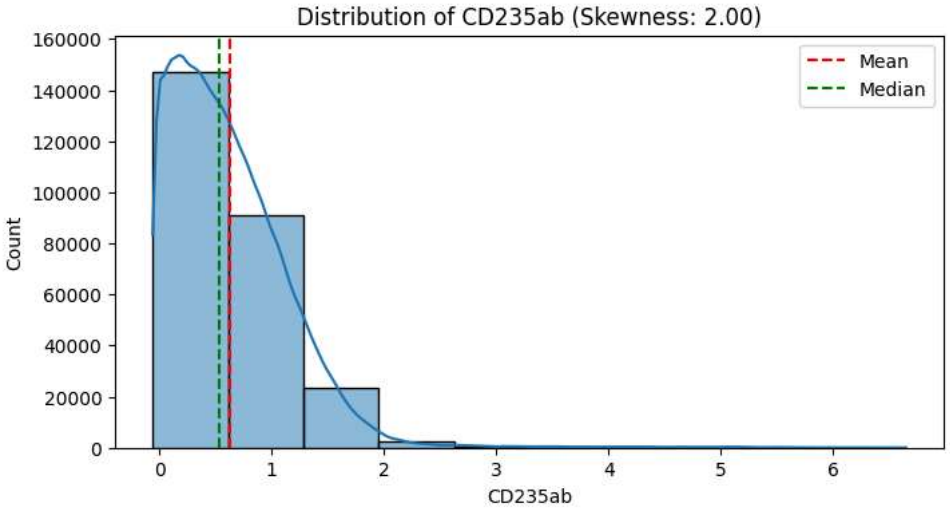

Distribution of CD16 (Skewness: 5.73)
Mean
Median
700000
600000
500000
400000
300000
200000
100000
0
Count
0
1
2
3
4
5
CD16


Distribution of CD19 (Skewness: 1.68)
Mean
Median
350000
300000
250000
200000

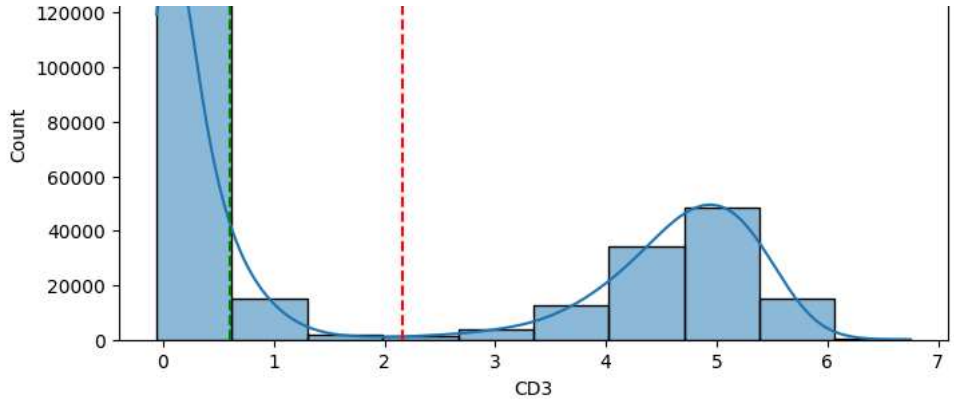

150000
100000
50000
0
0
1
2
3
4
5
CD19

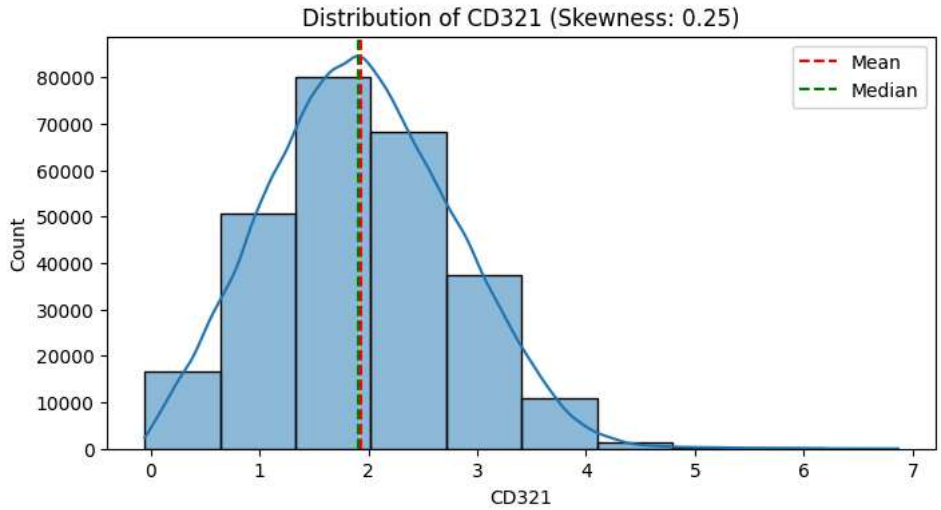

Distribution of CD20 (Skewness: 2.75)
Mean
Median
Count
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
6
CD20

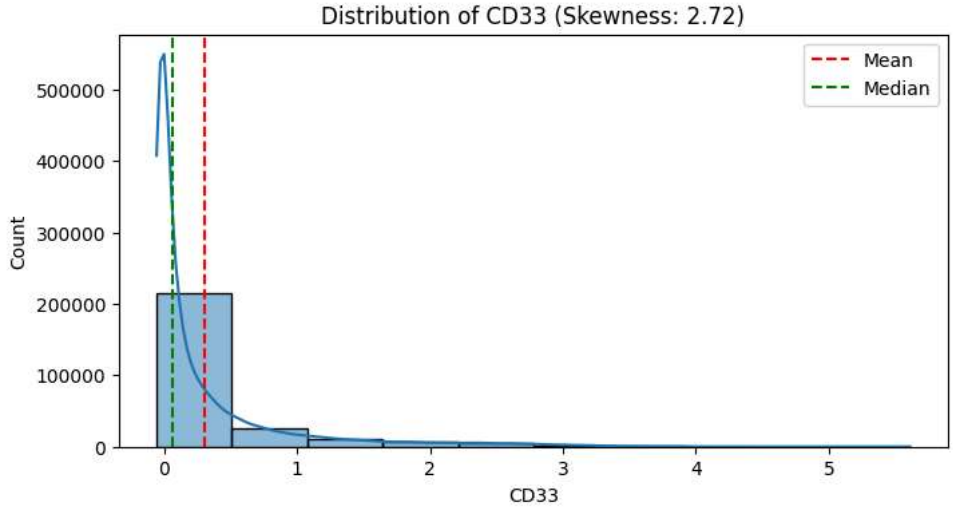

Distribution of CD22 (Skewness: 2.28)
Mean
Median
Count
400000
300000
200000
100000
0
0
1
2
3
4
5
CD22

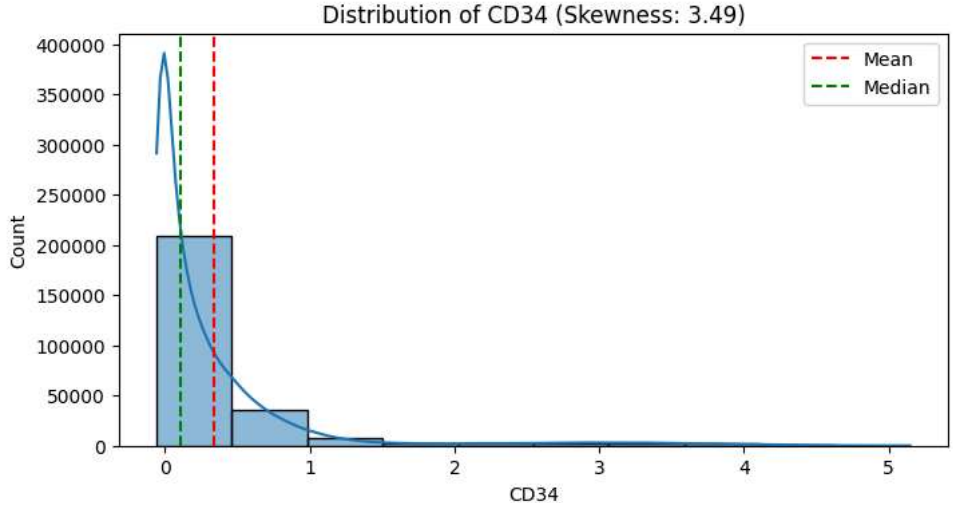

Distribution of CD235ab (Skewness: 2.00)
Mean
Median
Count
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
6
CD235ab

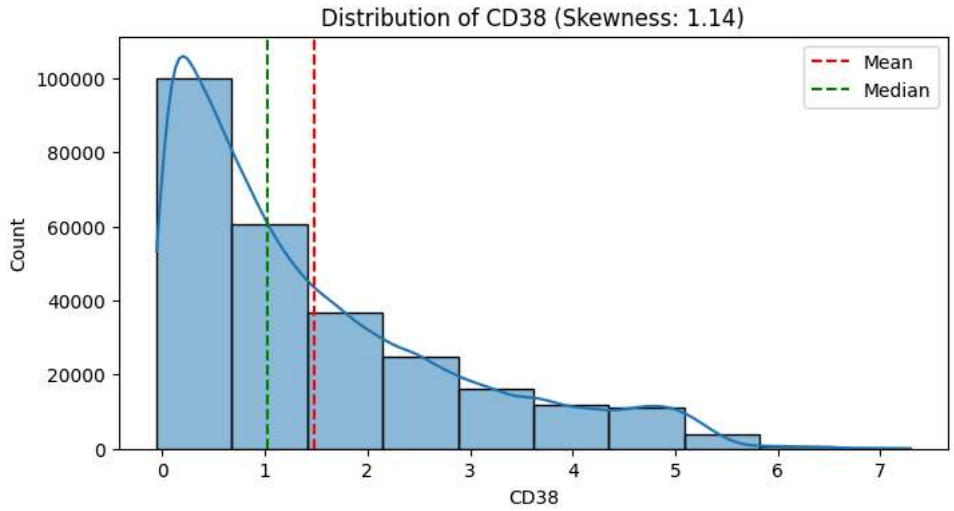

Distribution of CD3 (Skewness: 0.34)
Mean
Median
140000

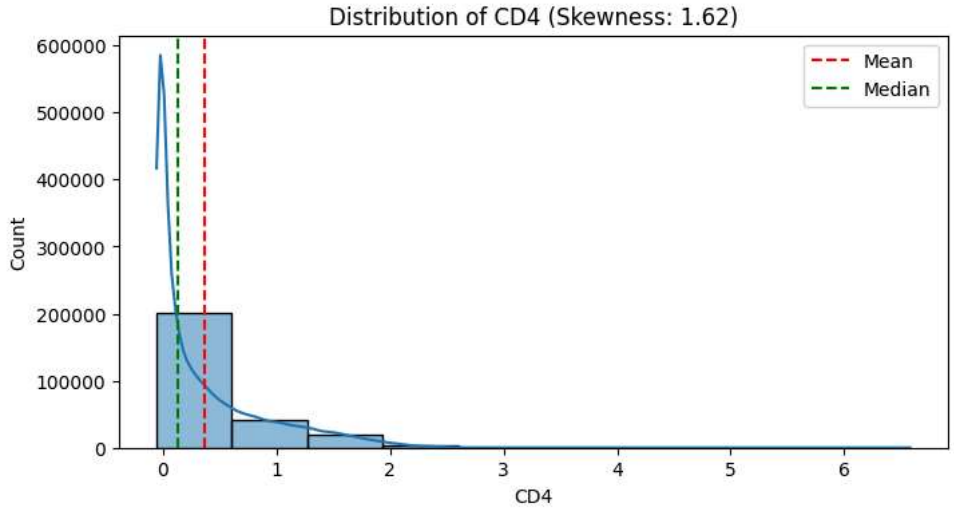

Count
CD3

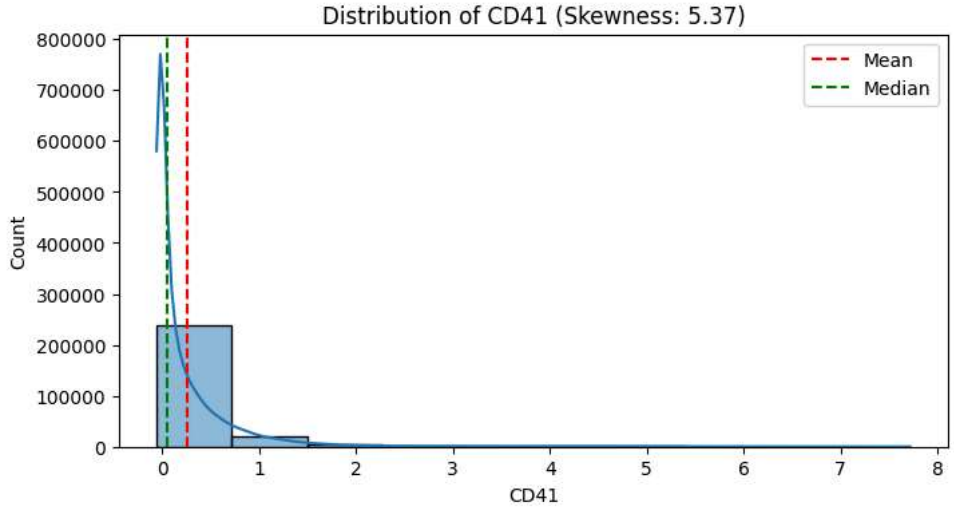

Distribution of CD321 (Skewness: 0.25)
Mean
Median
Count
CD321

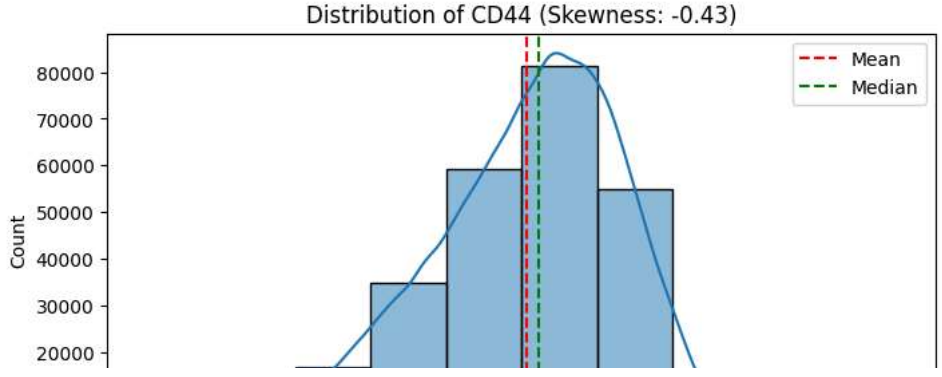

Distribution of CD33 (Skewness: 2.72)
Mean
Median
Count
CD33

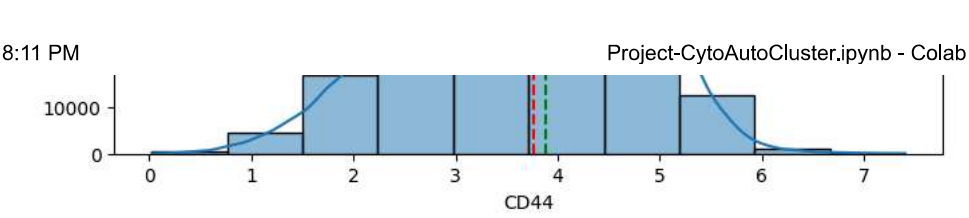

Distribution of CD34 (Skewness: 3.49)
Mean
Median
Count
CD34

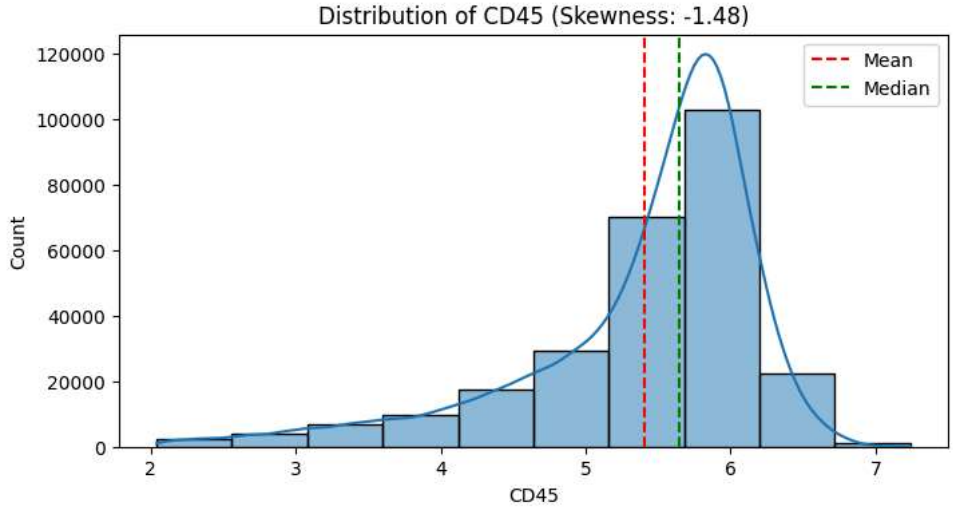

Distribution of CD38 (Skewness: 1.14)
Mean
Median
Count
CD38

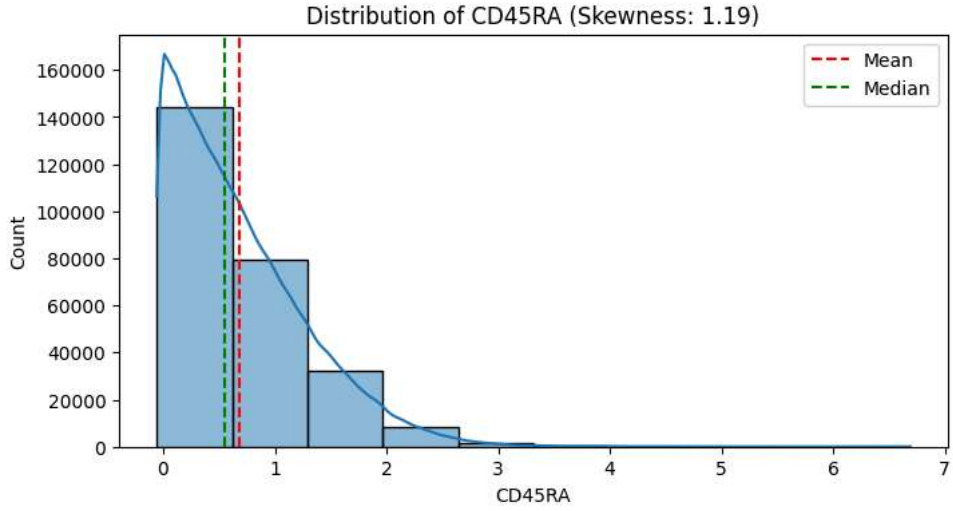

Distribution of CD4 (Skewness: 1.62)
Mean
Median
Count
CD4

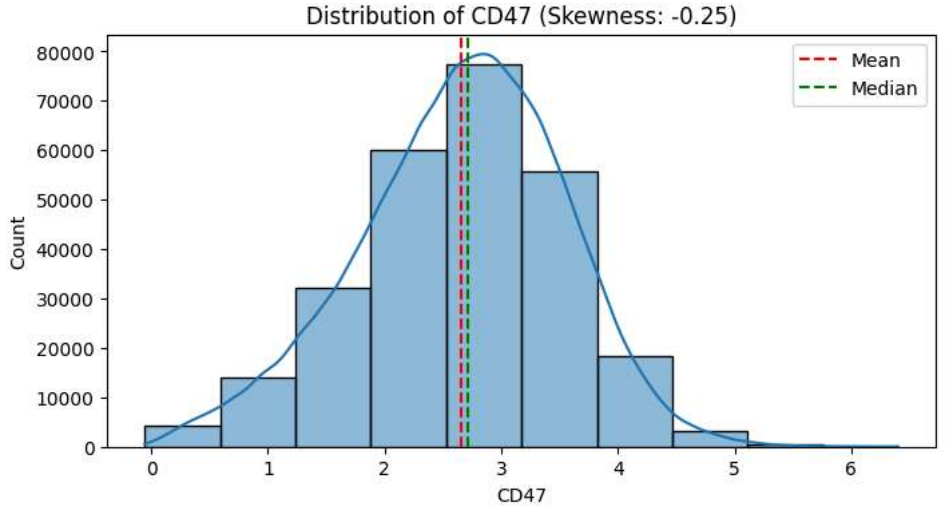

Distribution of CD41 (Skewness: 5.37)
Mean
Median
Count
CD41

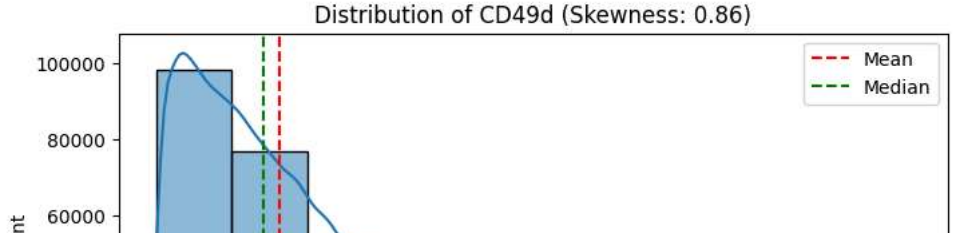

Distribution of CD44 (Skewness: -0.43)
Mean
Median
Count

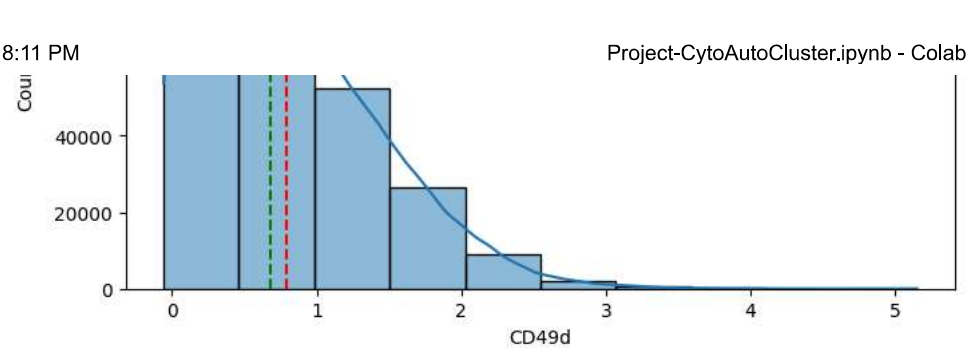

10000
0
0
1
2
3
4
5
6
7
CD44

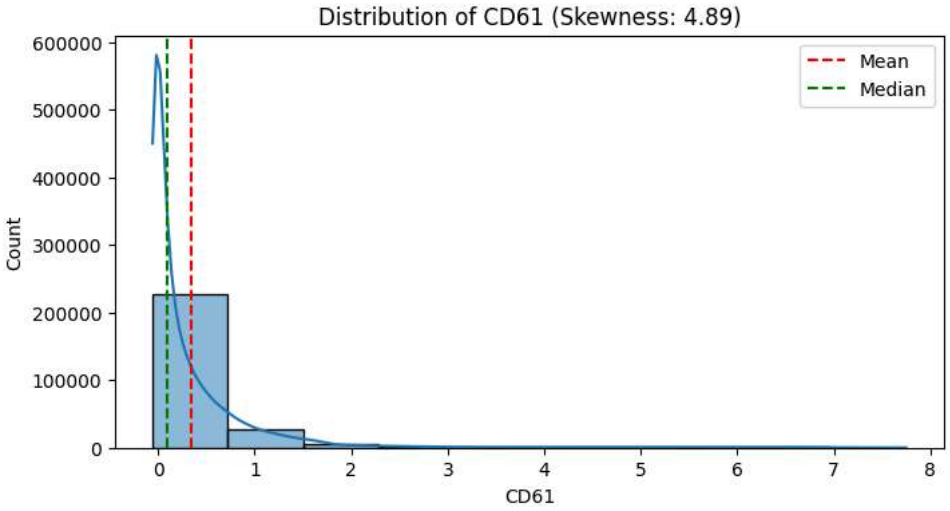

Distribution of CD45 (Skewness: -1.48)
Mean
Median
Count
120000
100000
80000
60000
40000
20000
0
2
3
4
5
6
7
CD45

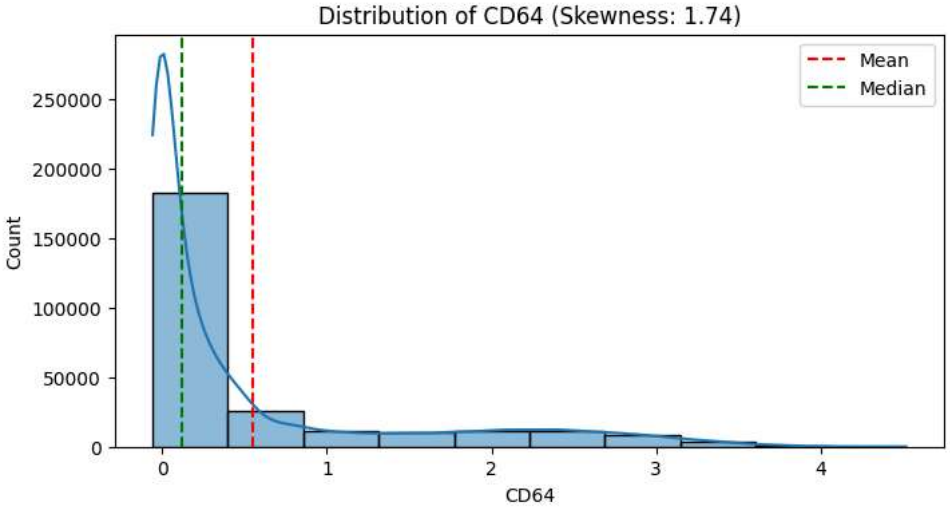

Distribution of CD45RA (Skewness: 1.19)
Mean
Median
Count
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
6
7
CD45RA

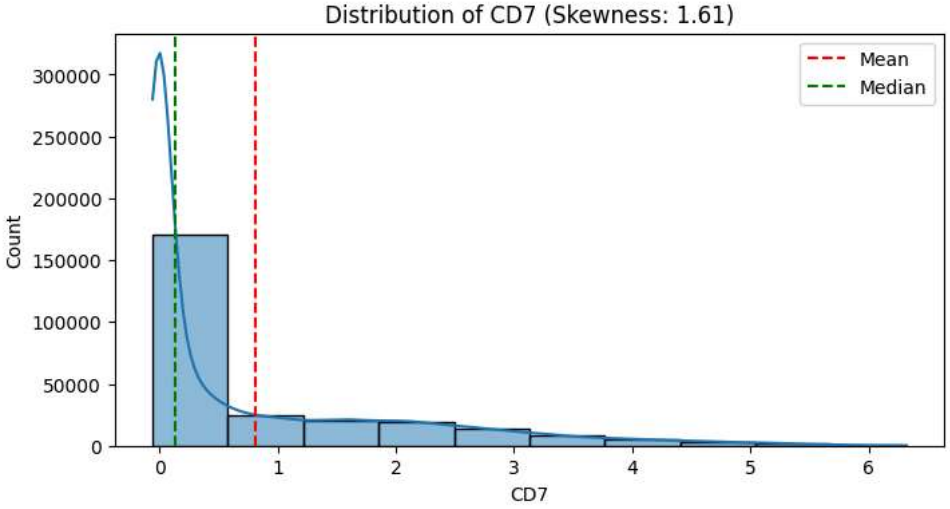

Distribution of CD47 (Skewness: -0.25)
Mean
Median
Count
80000
70000
60000
50000
40000
30000
20000
10000
0
0
1
2
3
4
5
6
CD47

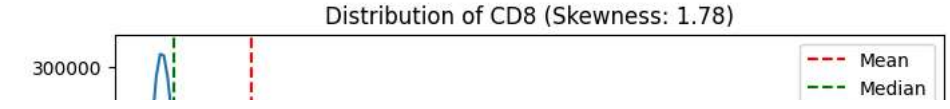

Distribution of CD49d (Skewness: 0.86)
Mean
Median
100000
80000
60000

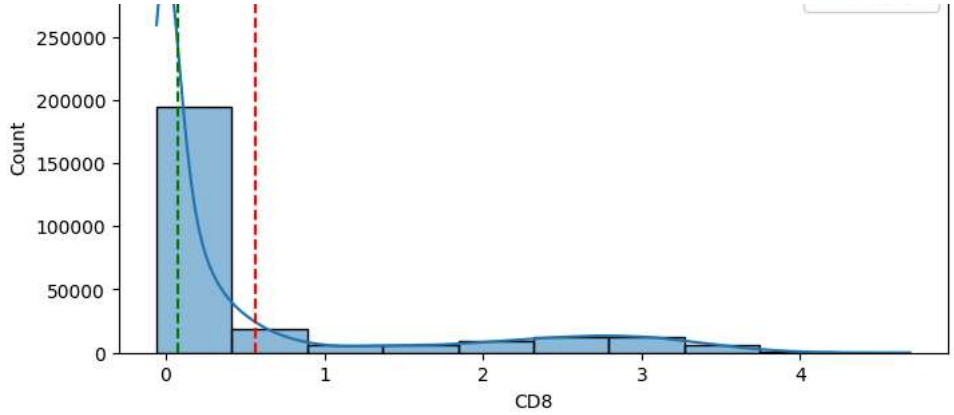

Cou
40000
20000
0
0
1
2
3
4
5
CD49d

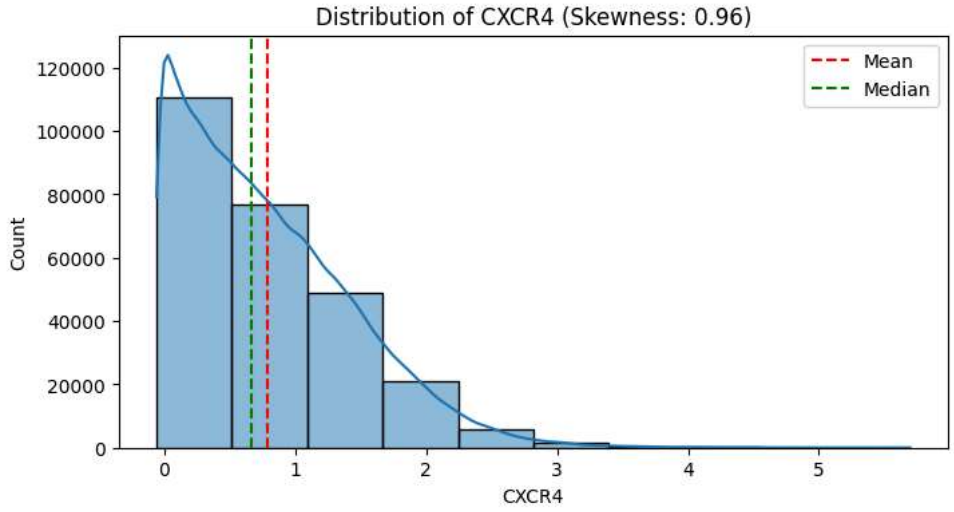

Distribution of CD61 (Skewness: 4.89)
Mean
Median
Count
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
6
7
8
CD61

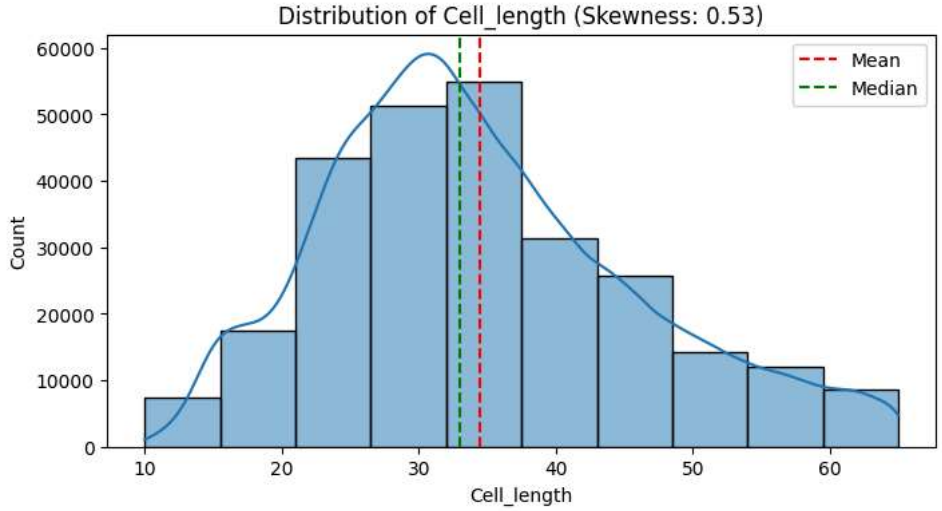

Distribution of CD64 (Skewness: 1.74)
Mean
Median
Count
250000
200000
150000
100000
50000
0
0
1
2
3
4
CD64

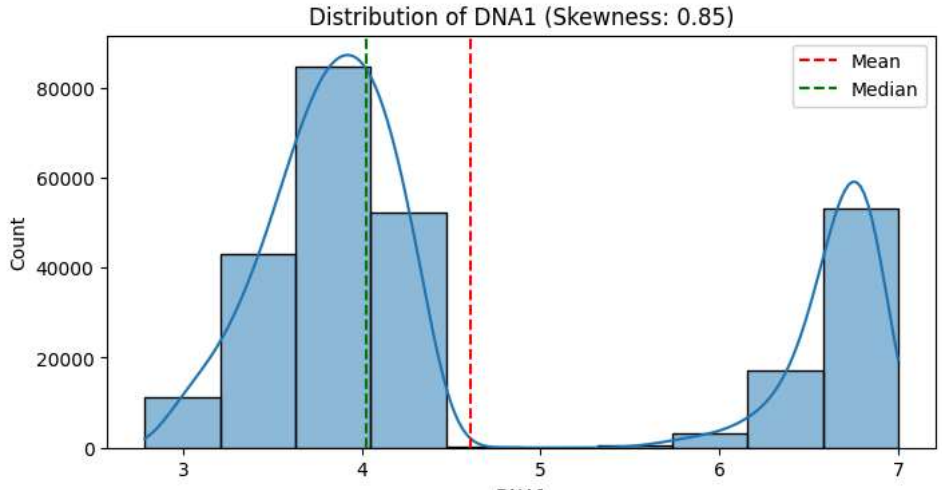

Distribution of CD7 (Skewness: 1.61)
Mean
Median
Count
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
CD7


Distribution of CD8 (Skewness: 1.78)
Mean
Median
300000

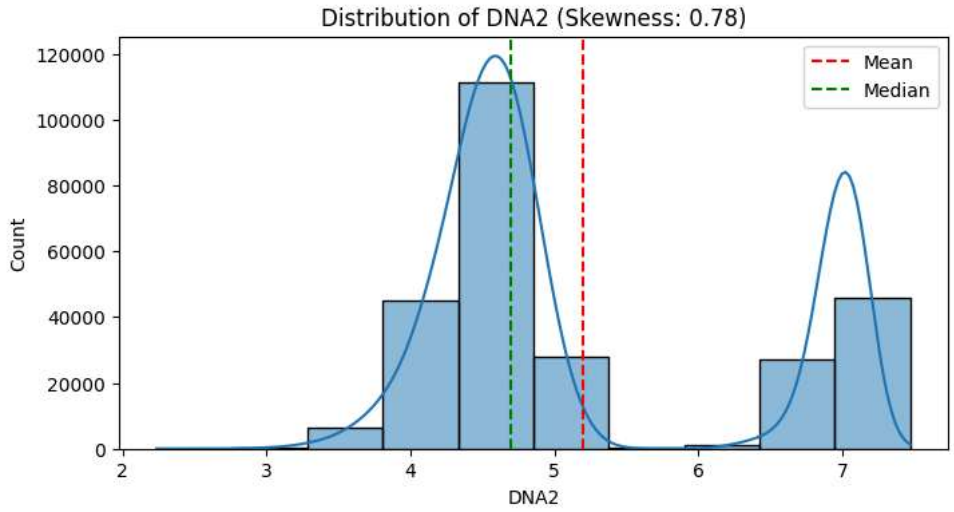

Count
CD8

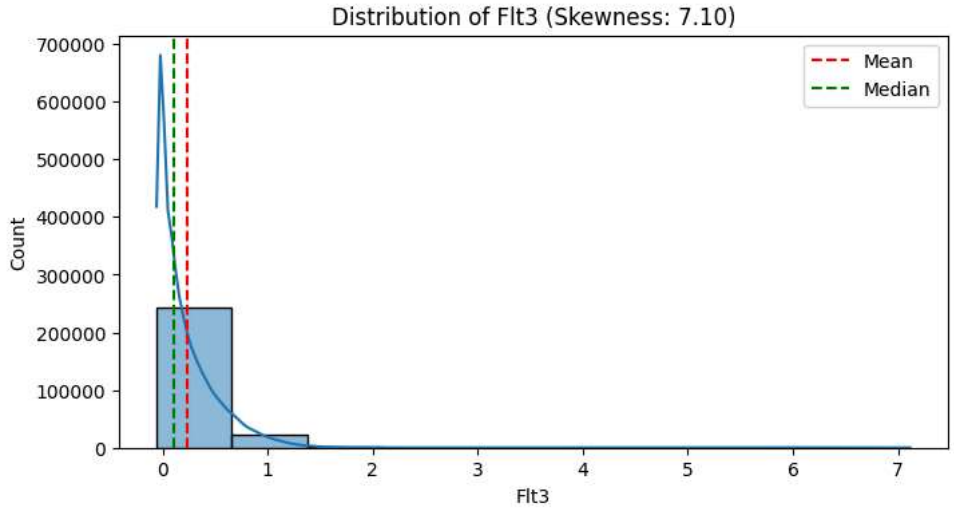

Distribution of CXCR4 (Skewness: 0.96)
Mean
Median
Count
CXCR4

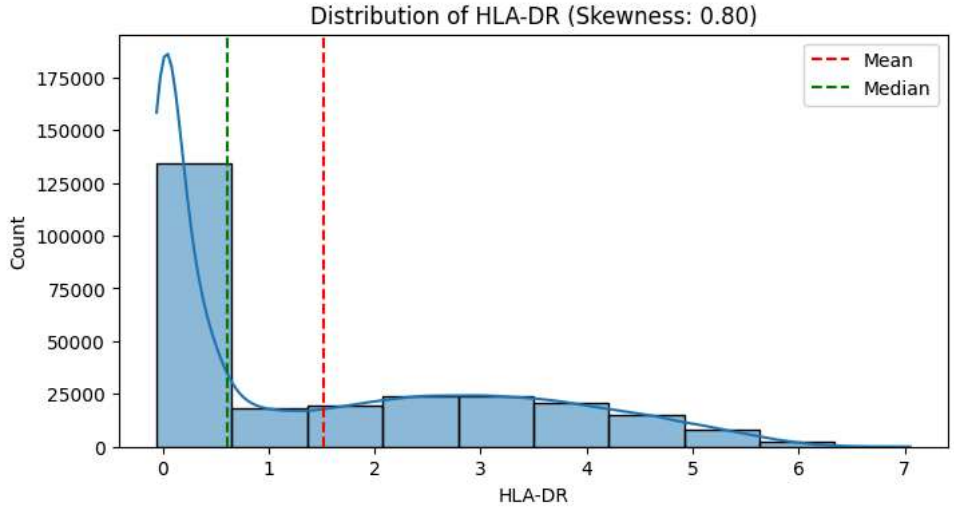

Distribution of Cell_length (Skewness: 0.53)
Mean
Median
Count
Cell_length

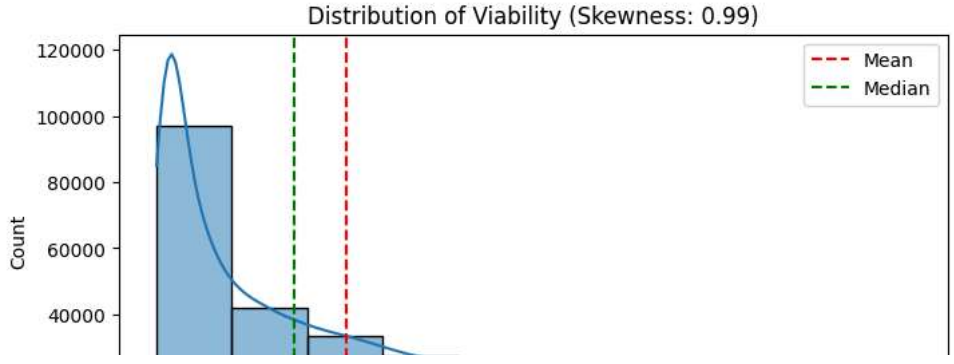

Distribution of DNA1 (Skewness: 0.85)
Mean
Median
Count

DNA1


Distribution of DNA2 (Skewness: 0.78)
Mean
Median
Count
DNA2


Distribution of Flt3 (Skewness: 7.10)
Mean
Median
Count
Flt3


Distribution of HLA-DR (Skewness: 0.80)
Mean
Median
Count
HLA-DR


Distribution of Viability (Skewness: 0.99)
Mean
Median
Count