SAD2022Z_Report_IzabelaFedorczyk

December 15, 2022

Task 1

(a)

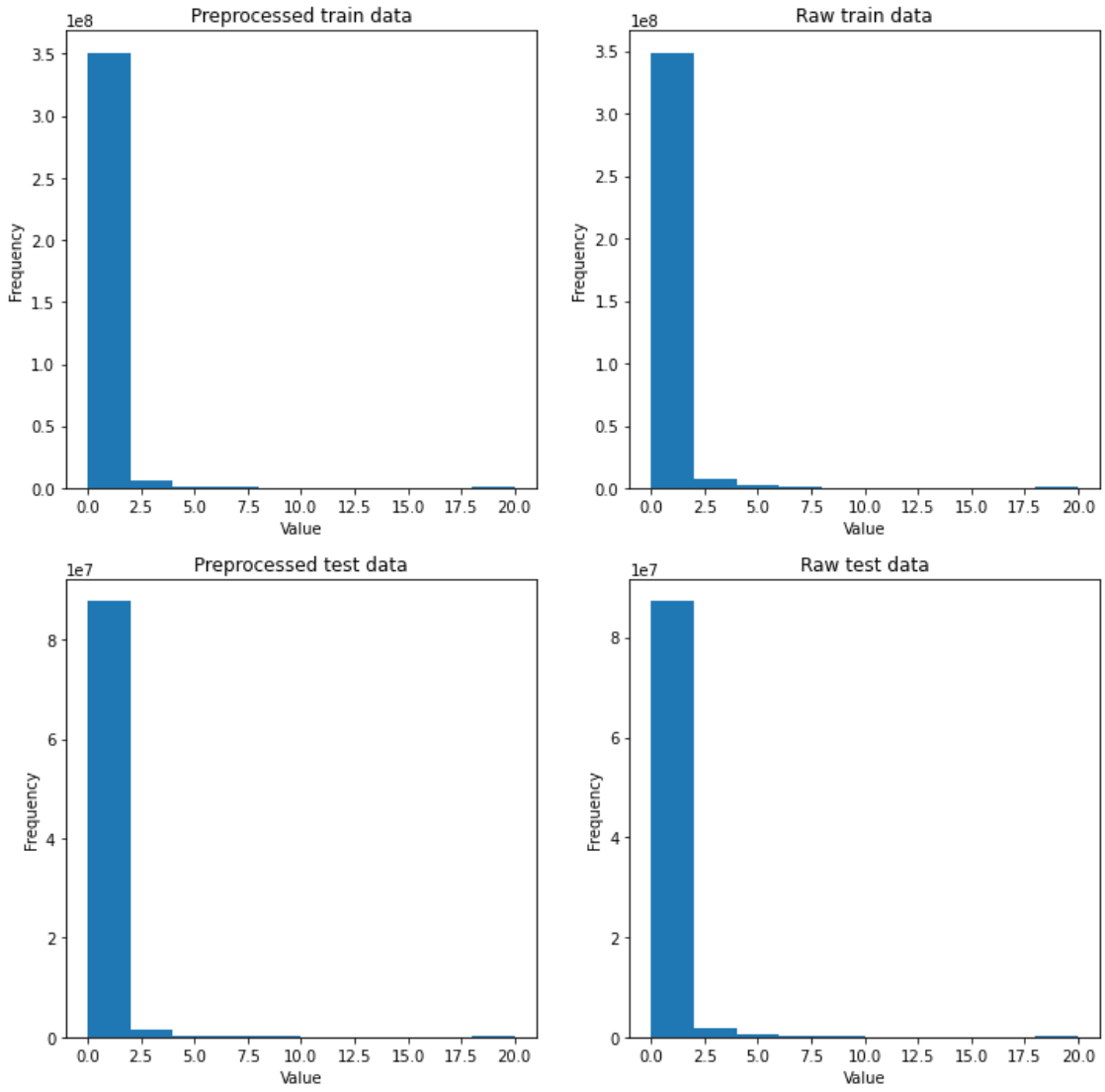
Train data -> 72208 observations and 5000 variables

Test data -> 18052 observations and 5000 variables

(b)

[33]:
```
train_norm = train_data.X.toarray()
train_raw = train_data.layers['counts'].toarray()
test_norm = test_data.X.toarray()
test_raw = test_data.layers['counts'].toarray()
```

[13]:
```
train_norm_clipped = np.clip(train_norm, 0, 20)
train_raw_clipped = np.clip(train_raw, 0, 20)
test_norm_clipped = np.clip(test_norm, 0, 20)
test_raw_clipped = np.clip(test_raw, 0, 20)
```



(c)

Has the data beeen normalized to 10k reads?

[15]:
```
train_norm.sum(axis=1)
```

[15]:
```
array([   404.05606,   3829.1133 ,   1648.1301 , ...,   5687.797  ,
        337690.44   ,   1868.8127 ], dtype=float32)
```

[16]:
```
train_raw.sum(axis=1)
```

[16]:
```
array([  554.,  9290.,  1409., ...,  8383., 13320.,  4739.], dtype=float32)
```

[17]:
```
test_norm.sum(axis=1)
```

[17]:
```
array([ 404.05606, 3829.1133 , 1648.1301 , ..., 1747.7454 , 1648.362  ,
        1588.0571 ], dtype=float32)
```

[18]:
```
test_raw.sum(axis = 1)
```

[18]:
```
array([ 554., 9290., 1409., ..., 2625., 1094., 1264.], dtype=float32)
```

Answer = NO

Has it been log1p transformed?

[19]:
```
train_norm
```

[19]:
```
array([[  0.       ,   0.       ,   0.       , ...,   0.       ,
            6.5640874,   0.       ],
       [  0.       ,   0.       ,   0.       , ...,   0.       ,
           39.98105  ,   0.       ],
       [  0.       ,   1.1697162,   0.       , ...,   0.       ,
           80.71042  ,   0.       ],
       ...,
       [  0.       ,   0.       ,   0.       , ...,   0.       ,
           34.603085 ,   0.       ],
       [  0.       ,   0.       ,   0.       , ...,   0.       ,
          101.40853  ,   0.       ],
       [  0.       ,   0.       ,   0.       , ...,   0.       ,
           62.70125  ,   0.3943475]], dtype=float32)
```

[35]:
```
np.expm1(train_norm)
```

[35]:
```
array([[0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        7.0816443e+02, 0.0000000e+00],
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        2.3096638e+17, 0.0000000e+00],
       [0.0000000e+00, 2.2210784e+00, 0.0000000e+00, ..., 0.0000000e+00,
        1.1274297e+35, 0.0000000e+00],
       ...,
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        1.0664210e+15, 0.0000000e+00],
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
                  inf, 0.0000000e+00],
       [0.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        1.7013987e+27, 4.8341593e-01]], dtype=float32)
```

[21]:
```
train_raw
```

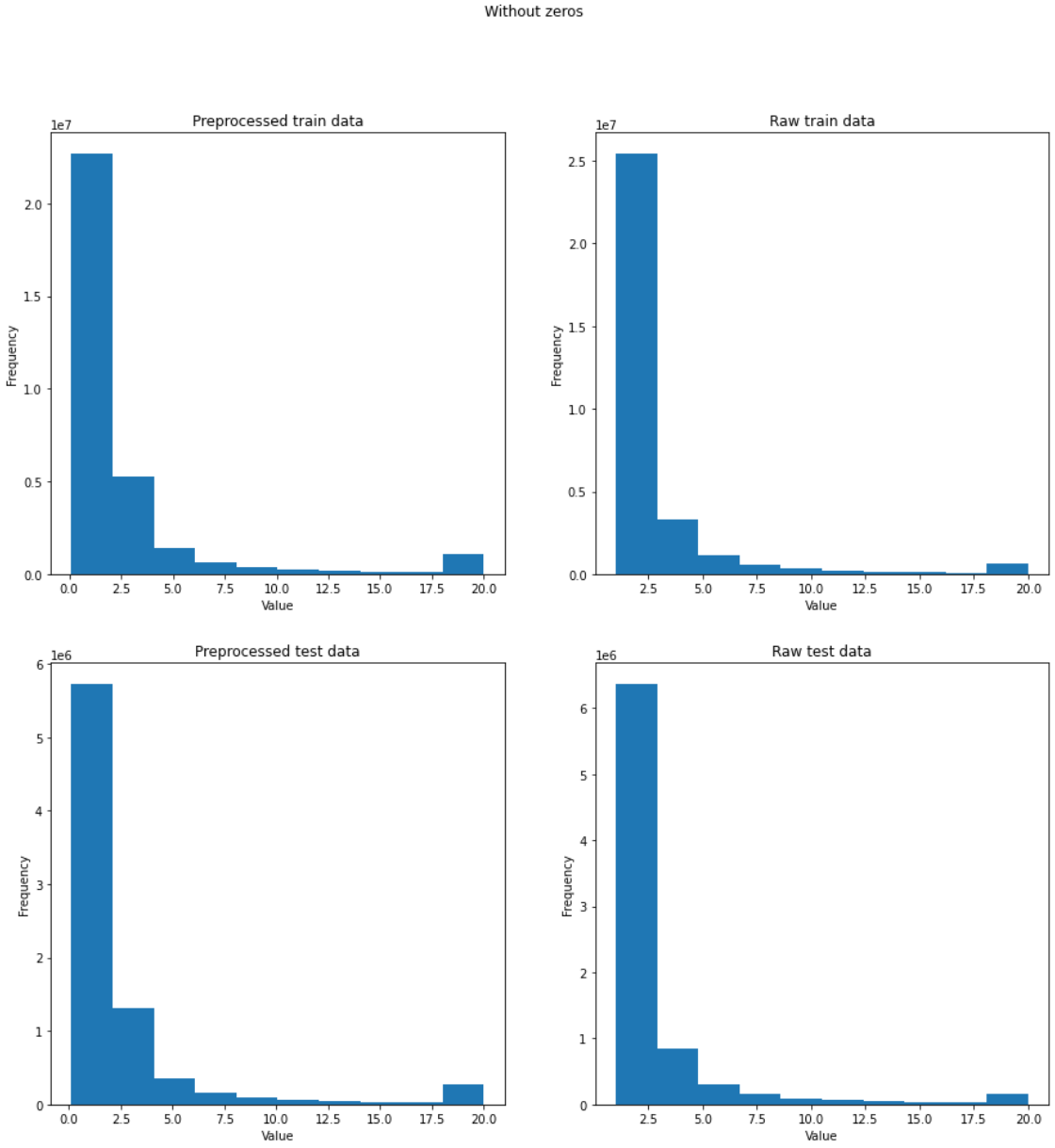
[21]:
```
array([[  0.,   0.,   0., ...,   0.,   9.,   0.],
       [  0.,   0.,   0., ...,   0.,  97.,   0.],
       [  0.,   1.,   0., ...,   0.,  69.,   0.],
       ...,
       [  0.,   0.,   0., ...,   0.,  51.,   0.],
       [  0.,   0.,   0., ...,   0.,   4.,   0.],
       [  0.,   0.,   0., ...,   0., 159.,   1.]], dtype=float32)
```

Answer = NO

(d)

[22]:
```
train_norm = train_norm[train_norm!=0]
train_raw = train_raw[train_raw!=0]
test_norm = test_norm[test_norm!=0]
test_raw = test_raw[test_raw!=0]
```

[23]:
```
train_norm_clipped = np.clip(train_norm, 0, 20)
train_raw_clipped = np.clip(train_raw, 0, 20)
test_norm_clipped = np.clip(test_norm, 0, 20)
test_raw_clipped = np.clip(test_raw, 0, 20)
```



(e)

The distribution of data remind of exponential distribution. I assume that the distribution of the data is right-skewed because the data is the data of gene expression mostly from cells of the immune system. As it's more common to be healthy rather that sick (so it's more probable that grater amount of human donors from who the dataset was collected have been healthy) thus most of the genes in our dataset involved in defendence of organism have (not suprisingly) small expression. What's more to extract more valuable information from dataset it's important to remove zero values as because of the same reason I mentioned above there are a lot of them and they make it more difficult to analyse genes with actual expression which are the ones we're probably more interested in. (Zero values noise the information about the actual expression.)

(f)

[30]:
```
obs_train = train_data.obs
```

[31]:
```
obs_test = test_data.obs
```

[32]:
```
obs_all = pd.concat([obs_train, obs_test])
```

The data contained in data frame adata.obs is a detailed information of each observation (for example in our case information such as cell type or BMI of a donor)

Number of patients: 9

Number of labs: 4

Number of cell types: 45