CSC2062 AIDA – Assignment 1

Elliot Dickie

40431475

Introduction

In this report, I will create and analyse a dataset of images consisting of letters and non-letters. In section 1, I will create this dataset and convert it to a .CSV file. In section 2, I will extract features from the dataset that inform me about the characteristics of the images. In section 3, I will perform statistical analysis on the dataset, looking at features like correlation and which features may be useful for discrimination. In section 4, I will train machine learning models on the data to help predict characteristics or categorize the images into letter or non-letter.

Section 1

I drew the images in GIMP and saved them as a .pgm file. I created a function in python to remove unnecessary information from the .pgm file and return its data as a list of integers. Then I created a function to convert a list of integers to a csv (starting a newline every 18 characters). From there, I created two nested loops, to iterate through all the letters and non-letters, calling the 'readpgm' and 'createCSV' function on each of them.

References:

Libraries used - csv

Section 2

To calculate the features, I created a 'readcsv' function to read the csv and return it as a two-dimensional list of integers (0 or 1). Then I created functions to calculate each of the features. For number of pixels, I just iterated through the list keeping track of how many pixels there were. For rows_with_1 and rows_with_3p, I iterated through each row and checked how many pixels there were. I repeated the process (iterating through columns instead of rows) for cols_with_1 and cols_with_3p. For aspect ratio, I iterate from top left to bottom right. Keeping track of the leftmost, rightmost, highest and lowest pixels, before using them to calculate the height, width and then aspect ratio.

For neigh_1 and the 'no_neigh_[blank]' functions, I created a catch-all 'check_neighbours' function that would check all the neighbours of a pixel and return a string list of them. Neigh_1 iterated through each pixel and evaluated the length of the list, while all the other 'no_neigh' functions just checked the list for their respect criteria.

For ‘connected_areas’, I used ‘scipy.ndimage.label’ (changing the structure so diagonal neighbours counted) to split the image up into areas. For eyes, I reversed the image, so the background (0s) became the foreground. Then I added an extra layer of foreground around the image, so that if anything bisected the image, it wouldn’t count as an extra eye. I then used ‘scipy.ndimage.label’ (but deducted 1, as the original background does not count as an eye).

For my custom feature, I decided to go with “longest_vertical_line”, which would store the longest vertical line in the image. I use this as my feature because I figured that it would capture information about certain characters, like ‘b’,’l’,’j’,’d’ and ‘!’ that would allow me to differentiate them (i.e. ‘b’ would typically have a longer line than ‘!’, as the vertical line of b was unbroken and would therefore be longer). To calculate this, I iterated through every coordinate in the image until I found a pixel, and travelled downward so long as their was pixels below that one (storing how far I travelled). If the total number of pixels I travelled downwards was greater than previous total, it became the new total. At the end of the function, the current total was returned.

After this, I looped through every letter and non-letter to calculate all the features.

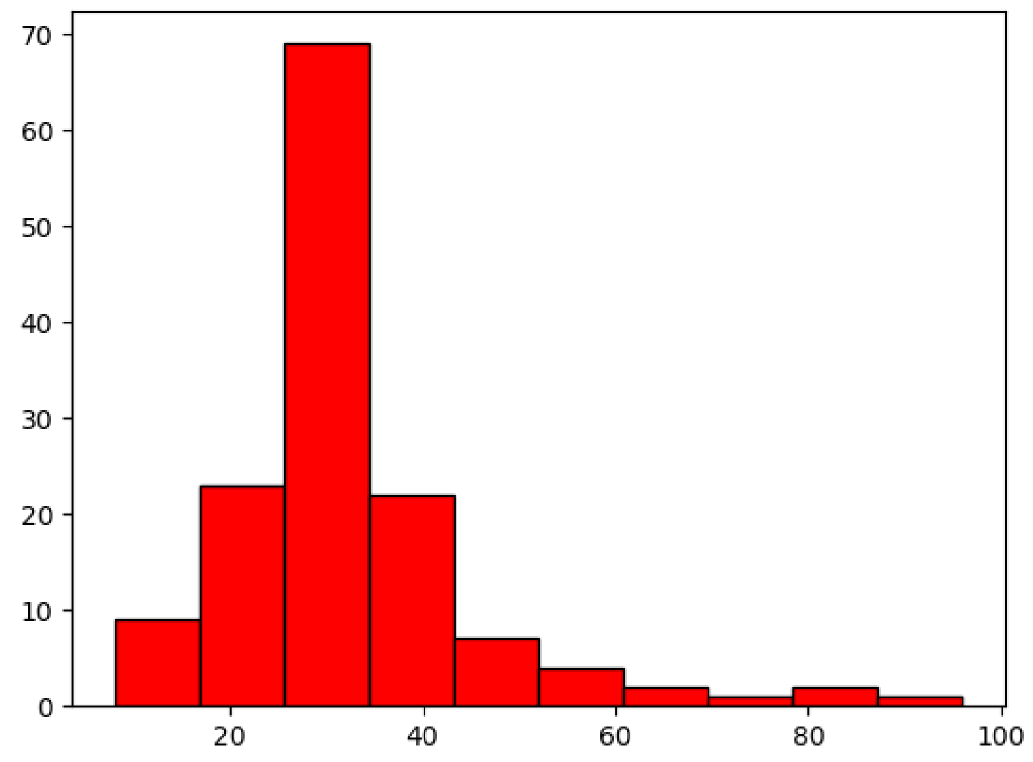
References:

Libraries used – csv, scipy

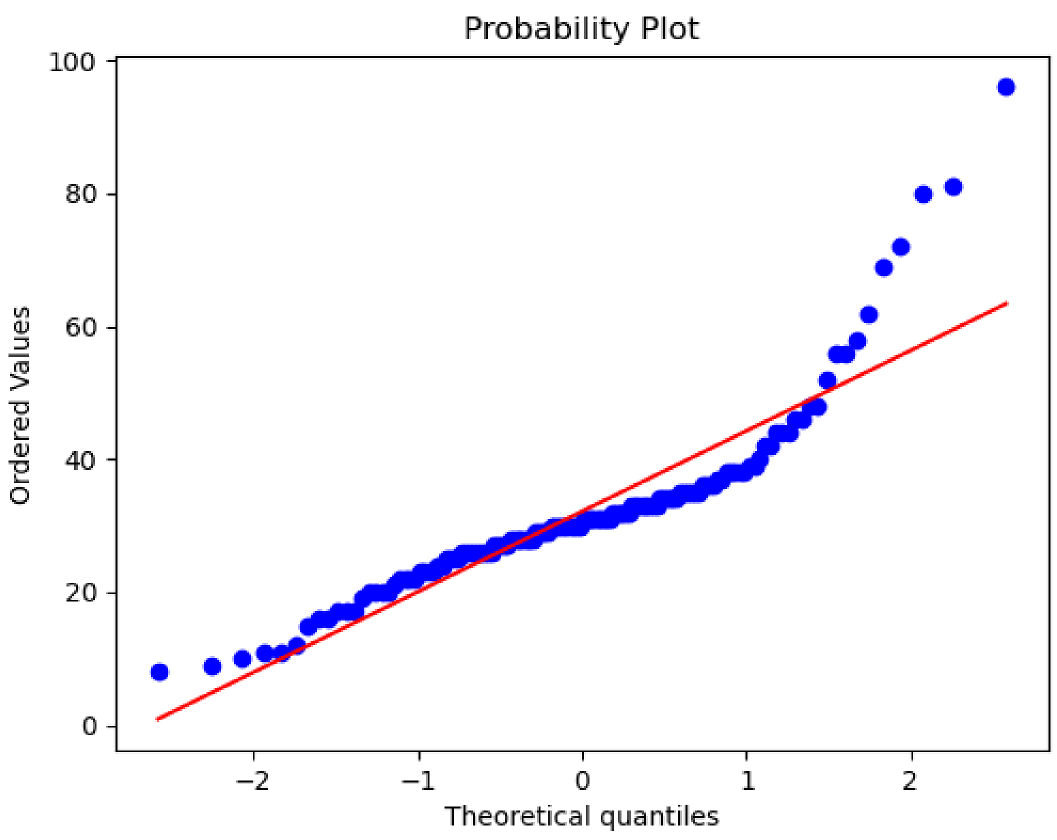
Section 3

3.1:

Number of pixels:

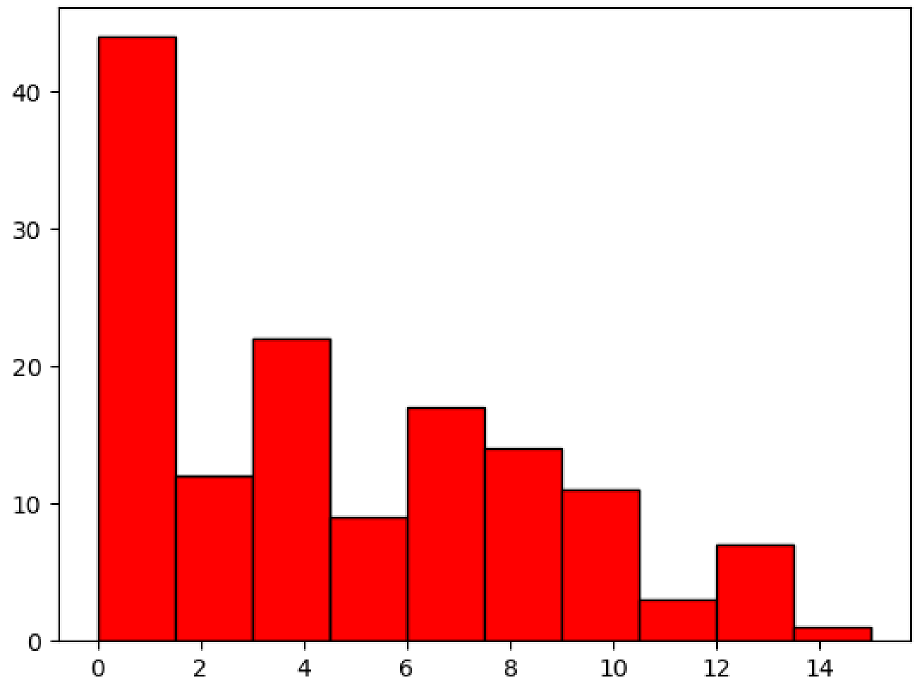


The number of pixels is only fairly normally distributed, as while it is unimodal, it has a very significant rightward skew.

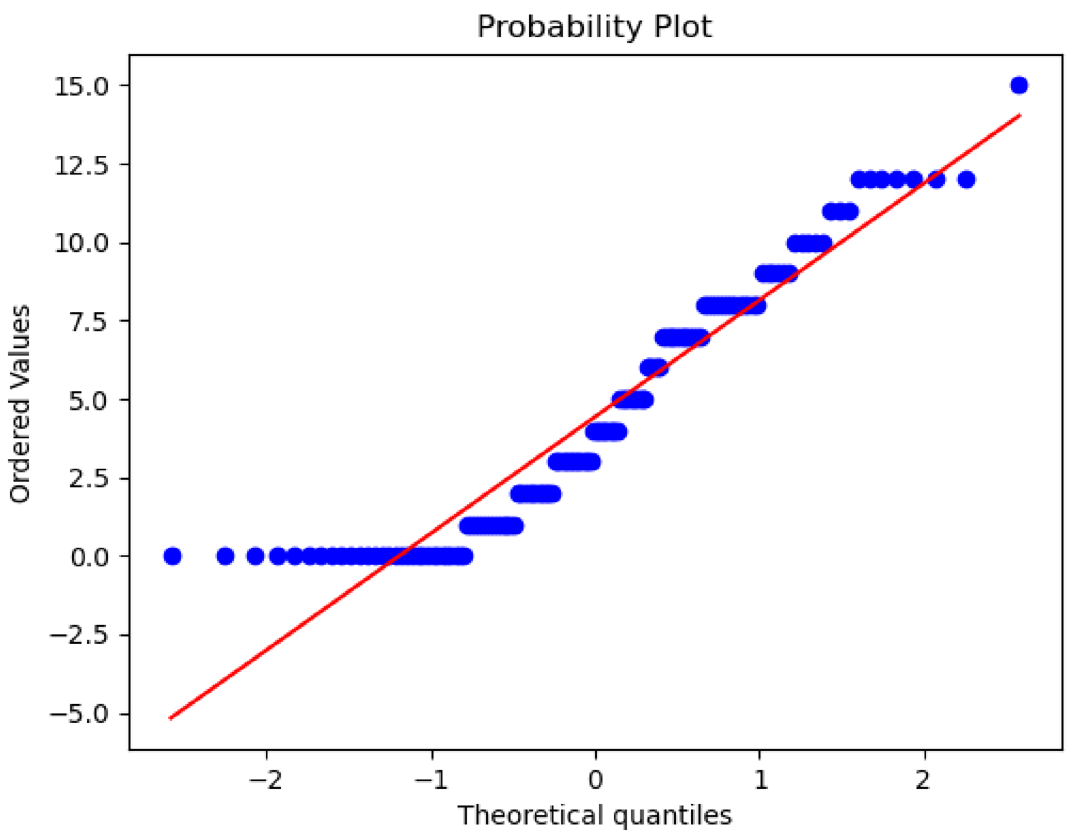


A Q-Q plot confirms this, show that the distribution as has a long right tail.

Rows with 1 pixel:

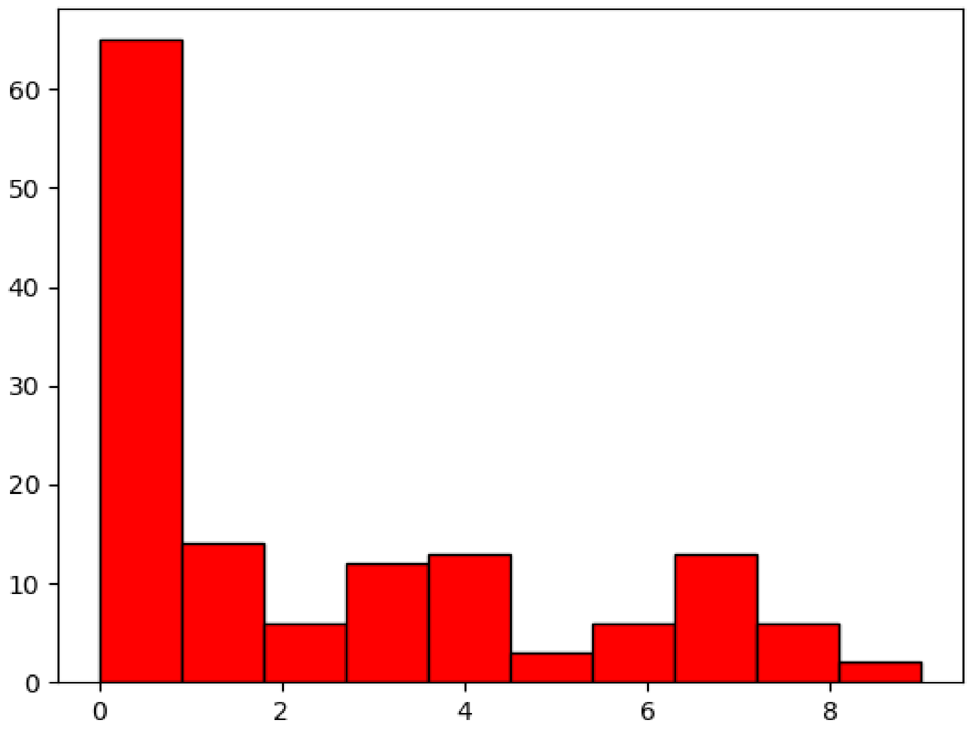


The rows with 1 pixel not normally distributed, as it has an extremely significant rightward skew. It's unimodal, but the there are clusters of larger frequency to the right.



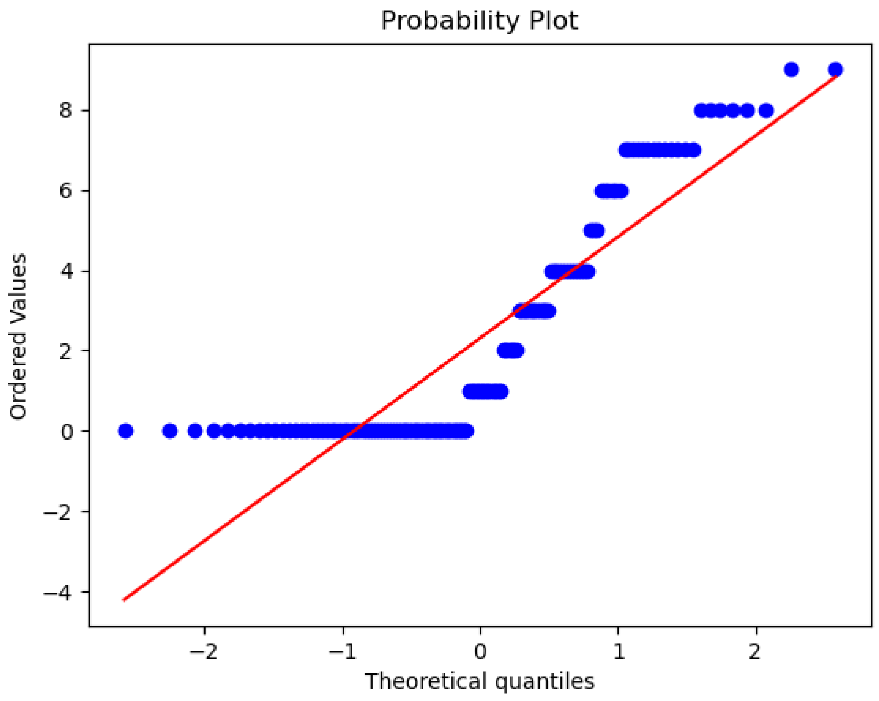
A QQ plot confirms that there is a strong rightward skew at lower values.

Columns with 1 pixel:



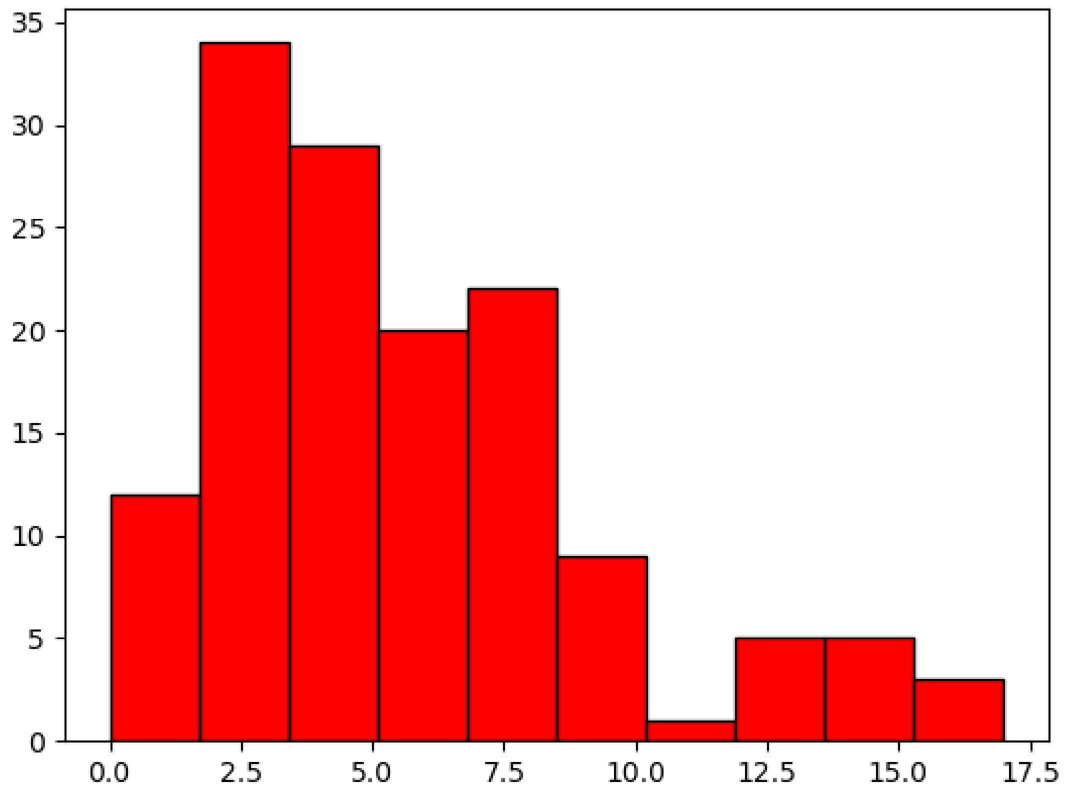
This distribution is not symmetric, as it has a very strong rightward skew, therefore it's not normally distributed. It is unimodal though.

We can confirm this with a QQ plot

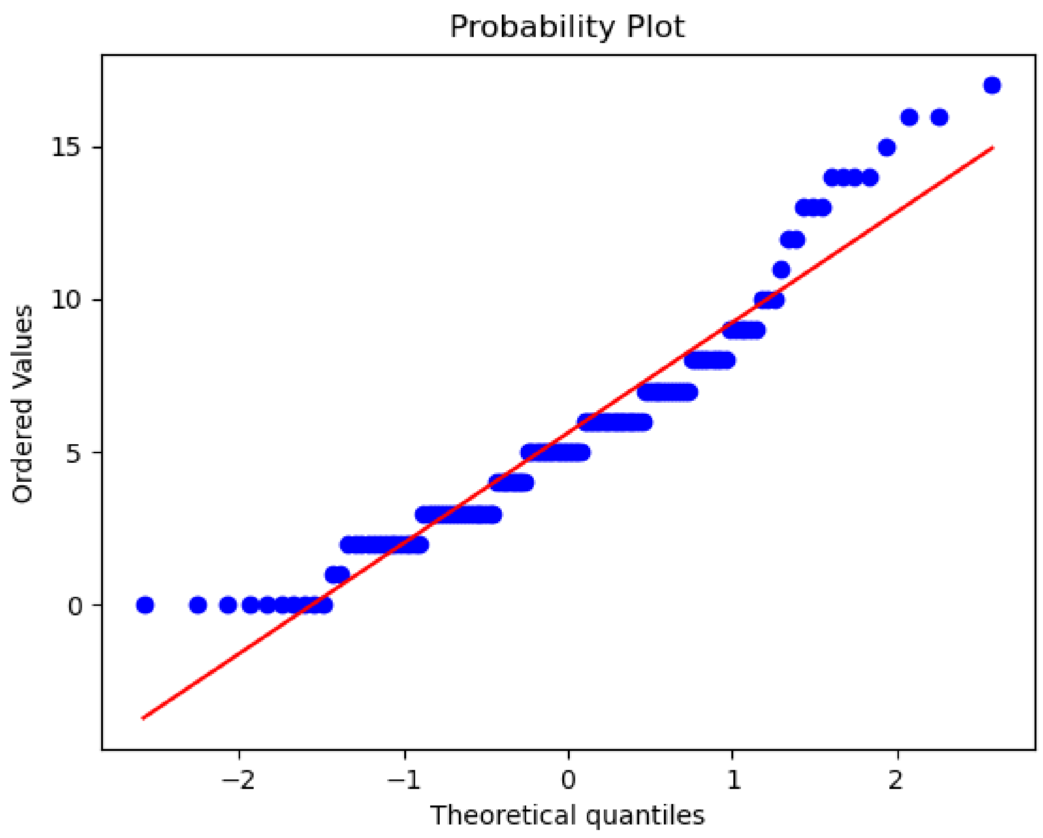


Interestingly, the data also has clusters of larger frequency that don't follow the trend.

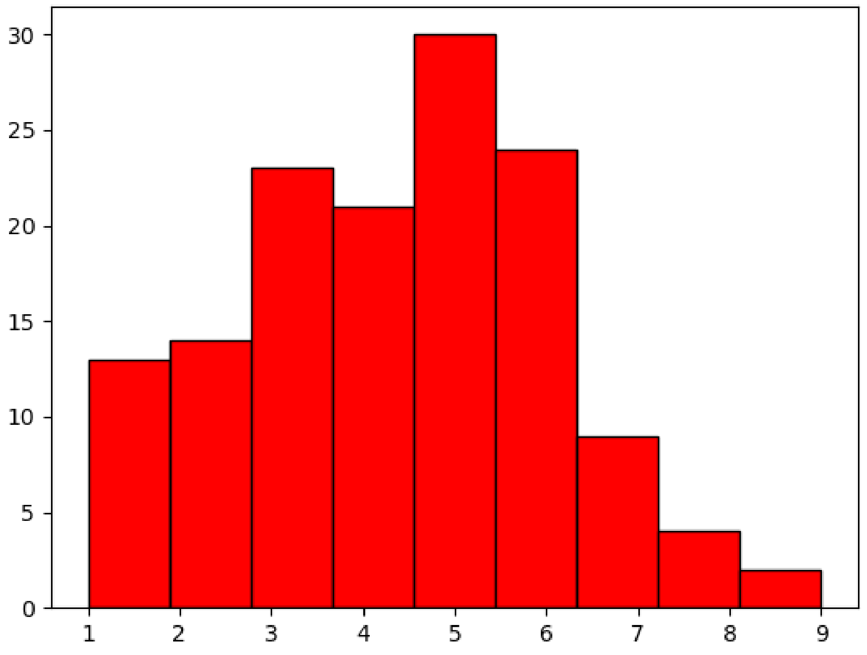
Rows with 3 or more pixels:



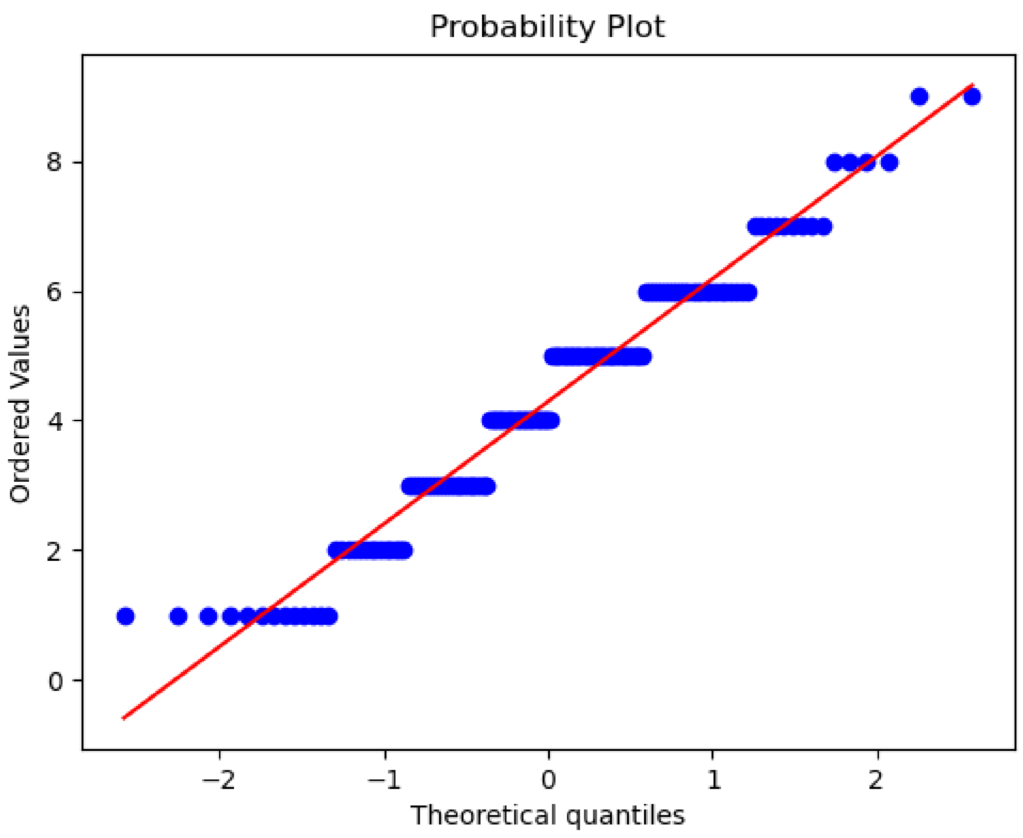
This distribution is fairly normally distributed, as it unimodal and very roughly symmetric, although it has a rightward skew, as evidenced by the QQ plot.



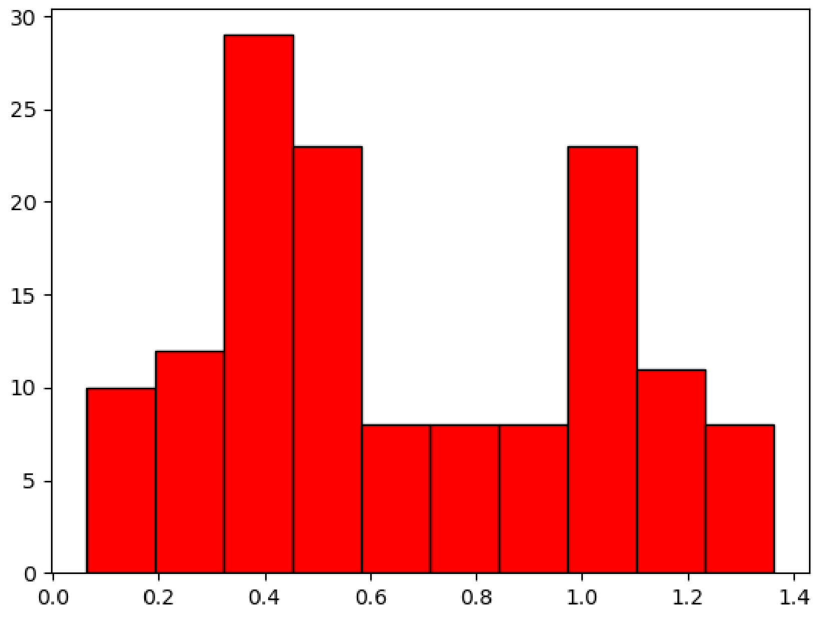
Columns with 3 or more pixels



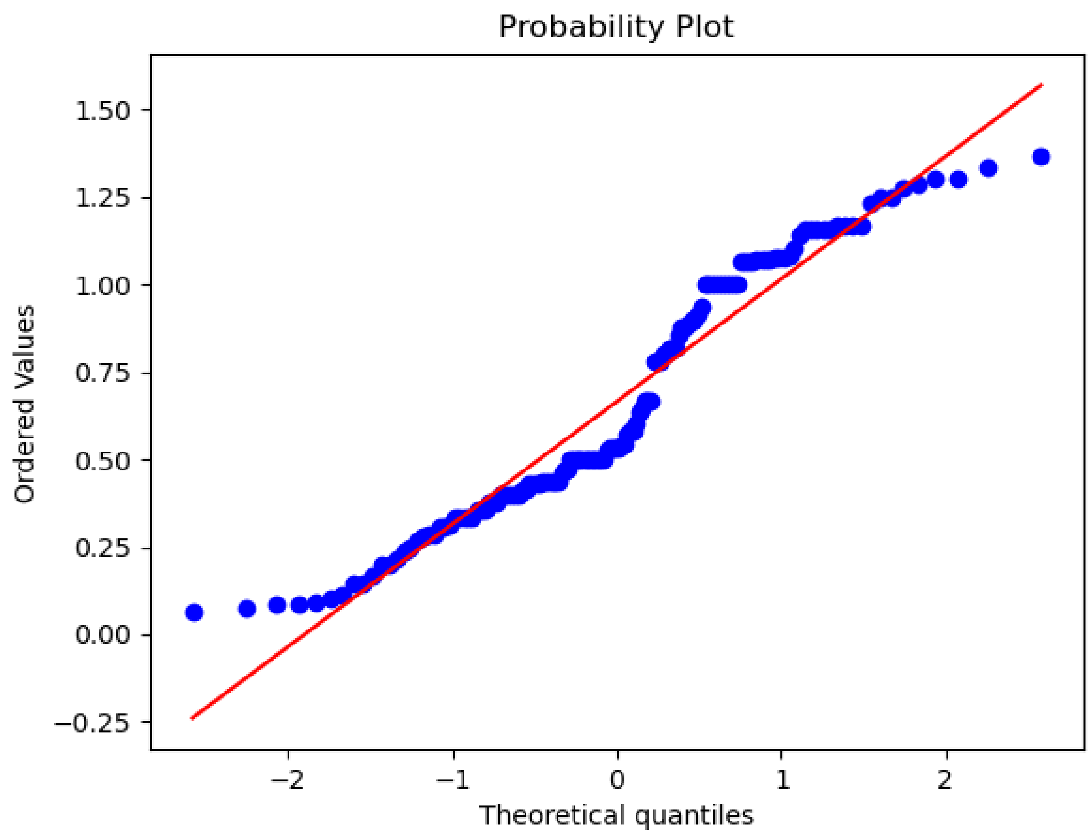
This is a nearly normal distribution, as it is generally unimodal and generally symmetric (although it does have a slight rightward skew at lower values).



Aspect Ratio:



The distribution is bi modal. As it has two peaks at 0.4 and 1.0. However, it is roughly symmetric.



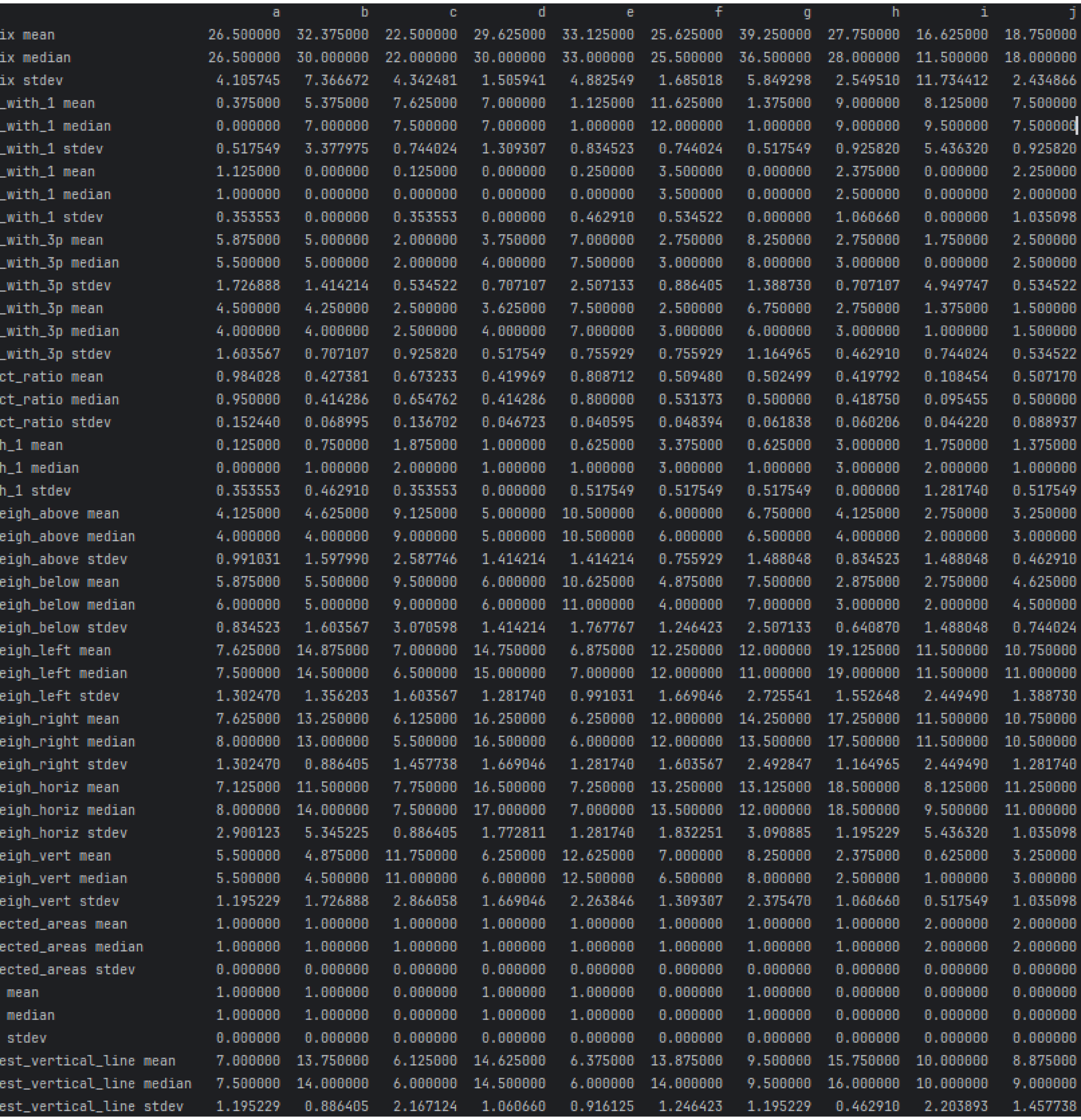
The QQ-plot indicates that the distribution might have slight fat tails at lower and higher values, but otherwise seems decently normal.

3.2:

I calculated the full statistics for letters and non-letters using the statistics library, and a panda dataframe to print them.

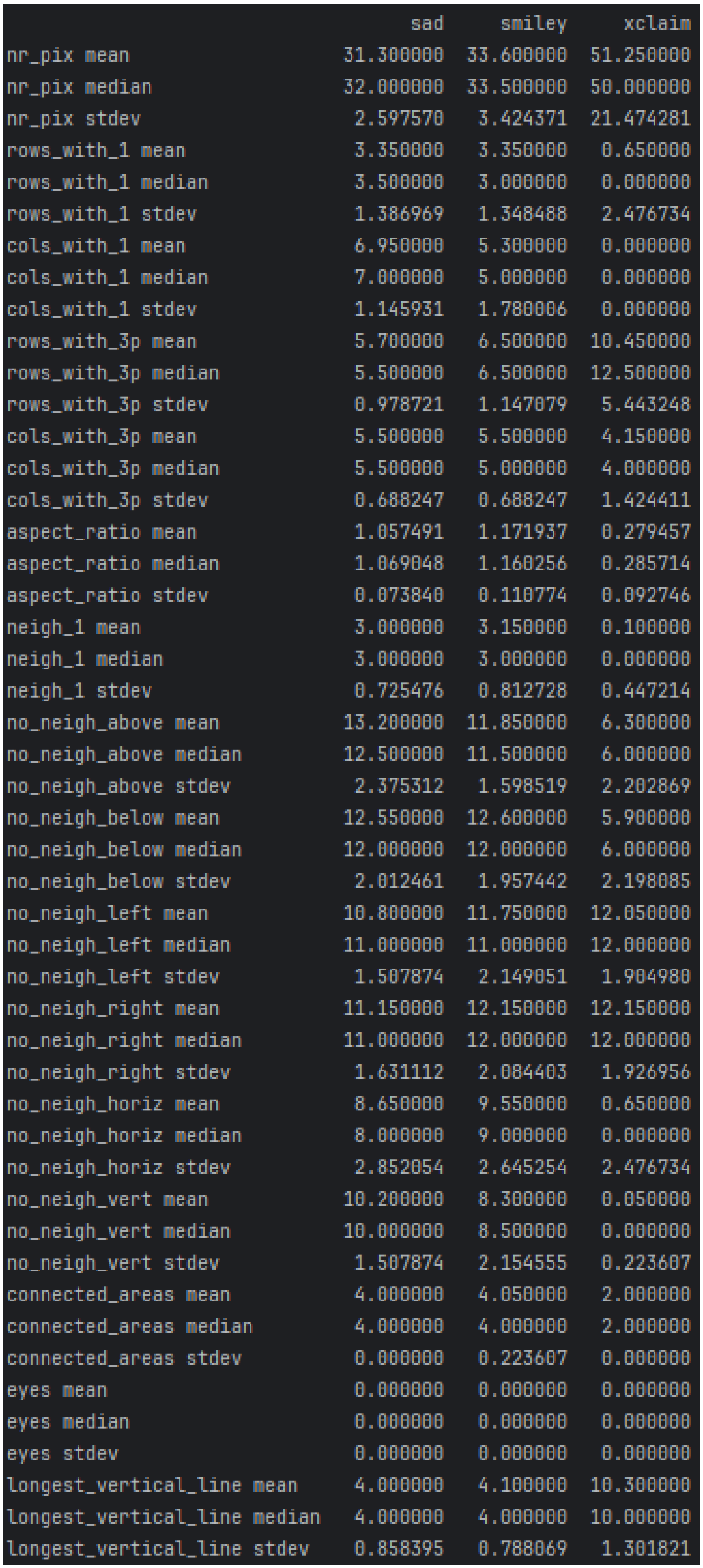
For completeness, I have detailed the full statistics for each individual letter, and then summary statistics to observe trends (This is useful for feature that have a significant outliers in one letter/non-letter).

Full statistics for letters:



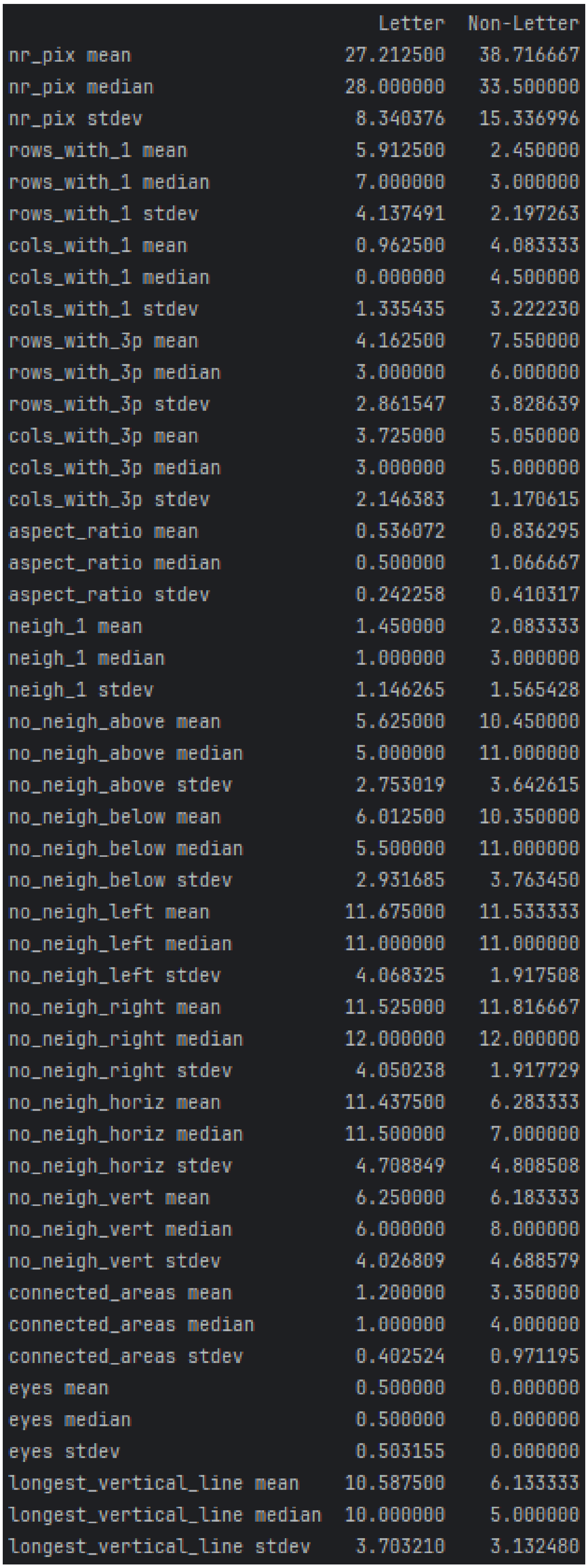
```
                                   a           b           c           d           e           f           g           h           i           j
ix mean                    26.500000   32.375000   22.500000   29.625000   33.125000   25.625000   39.250000   27.750000   16.625000   18.750000
ix median                  26.500000   30.000000   22.000000   30.000000   33.000000   25.500000   36.500000   28.000000   11.500000   18.000000
ix stdev                    4.105745    7.366672    4.342481    1.505941    4.882549    1.685018    5.849298    2.549510   11.734412    2.434866
_with_1 mean                0.375000    5.375000    7.625000    7.000000    1.125000   11.625000    1.375000    9.000000    8.125000    7.500000
_with_1 median              0.000000    7.000000    7.500000    7.000000    1.000000   12.000000    1.000000    9.000000    9.500000    7.500000
_with_1 stdev               0.517549    3.377975    0.744024    1.309307    0.834523    0.744024    0.517549    0.925820    5.436320    0.925820
_with_1 mean                1.125000    0.000000    0.125000    0.000000    0.250000    3.500000    0.000000    2.375000    0.000000    2.250000
_with_1 median              1.000000    0.000000    0.000000    0.000000    0.000000    3.500000    0.000000    2.500000    0.000000    2.000000
_with_1 stdev               0.353553    0.000000    0.353553    0.000000    0.462910    0.534522    0.000000    1.060660    0.000000    1.035098
_with_3p mean               5.875000    5.000000    2.000000    3.750000    7.000000    2.750000    8.250000    2.750000    1.750000    2.500000
_with_3p median             5.500000    5.000000    2.000000    4.000000    7.500000    3.000000    8.000000    3.000000    0.000000    2.500000
_with_3p stdev              1.726888    1.414214    0.534522    0.707107    2.507133    0.886405    1.388730    0.707107    4.949747    0.534522
_with_3p mean               4.500000    4.250000    2.500000    3.625000    7.500000    2.500000    6.750000    2.750000    1.375000    1.500000
_with_3p median             4.000000    4.000000    2.500000    4.000000    7.000000    3.000000    6.000000    3.000000    1.000000    1.500000
_with_3p stdev              1.603567    0.707107    0.925820    0.517549    0.755929    0.755929    1.164965    0.462910    0.744024    0.534522
ct_ratio mean               0.984028    0.427381    0.673233    0.419969    0.808712    0.509480    0.502499    0.419792    0.108454    0.507170
ct_ratio median             0.950000    0.414286    0.654762    0.414286    0.800000    0.531373    0.500000    0.418750    0.095455    0.500000
ct_ratio stdev              0.152440    0.068995    0.136702    0.046723    0.040595    0.048394    0.061838    0.060206    0.044220    0.088937
h_1 mean                    0.125000    0.750000    1.875000    1.000000    0.625000    3.375000    0.625000    3.000000    1.750000    1.375000
h_1 median                  0.000000    1.000000    2.000000    1.000000    1.000000    3.000000    1.000000    3.000000    2.000000    1.000000
h_1 stdev                   0.353553    0.462910    0.353553    0.000000    0.517549    0.517549    0.517549    0.000000    1.281740    0.517549
eigh_above mean             4.125000    4.625000    9.125000    5.000000   10.500000    6.000000    6.750000    4.125000    2.750000    3.250000
eigh_above median           4.000000    4.000000    9.000000    5.000000   10.500000    6.000000    6.500000    4.000000    2.000000    3.000000
eigh_above stdev            0.991031    1.597990    2.587746    1.414214    1.414214    0.755929    1.488048    0.834523    1.488048    0.462910
eigh_below mean             5.875000    5.500000    9.500000    6.000000   10.625000    4.875000    7.500000    2.875000    2.750000    4.625000
eigh_below median           6.000000    5.000000    9.000000    6.000000   11.000000    4.000000    7.000000    3.000000    2.000000    4.500000
eigh_below stdev            0.834523    1.603567    3.070598    1.414214    1.767767    1.246423    2.507133    0.640870    1.488048    0.744024
eigh_left mean              7.625000   14.875000    7.000000   14.750000    6.875000   12.250000   12.000000   19.125000   11.500000   10.750000
eigh_left median            7.500000   14.500000    6.500000   15.000000    7.000000   12.000000   11.000000   19.000000   11.500000   11.000000
eigh_left stdev             1.302470    1.356203    1.603567    1.281740    0.991031    1.669046    2.725541    1.552648    2.449490    1.388730
eigh_right mean             7.625000   13.250000    6.125000   16.250000    6.250000   12.000000   14.250000   17.250000   11.500000   10.750000
eigh_right median           8.000000   13.000000    5.500000   16.500000    6.000000   12.000000   13.500000   17.500000   11.500000   10.500000
eigh_right stdev            1.302470    0.886405    1.457738    1.669046    1.281740    1.603567    2.492847    1.164965    2.449490    1.281740
eigh_horiz mean             7.125000   11.500000    7.750000   16.500000    7.250000   13.250000   13.125000   18.500000    8.125000   11.250000
eigh_horiz median           8.000000   14.000000    7.500000   17.000000    7.000000   13.500000   12.000000   18.500000    9.500000   11.000000
eigh_horiz stdev            2.900123    5.345225    0.886405    1.772811    1.281740    1.832251    3.090885    1.195229    5.436320    1.035098
eigh_vert mean              5.500000    4.875000   11.750000    6.250000   12.625000    7.000000    8.250000    2.375000    0.625000    3.250000
eigh_vert median            5.500000    4.500000   11.000000    6.000000   12.500000    6.500000    8.000000    2.500000    1.000000    3.000000
eigh_vert stdev             1.195229    1.726888    2.866058    1.669046    2.263846    1.309307    2.375470    1.060660    0.517549    1.035098
ected_areas mean            1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    2.000000    2.000000
ected_areas median          1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    2.000000    2.000000
ected_areas stdev           0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
 mean                       1.000000    1.000000    0.000000    1.000000    1.000000    0.000000    1.000000    0.000000    0.000000    0.000000
 median                     1.000000    1.000000    0.000000    1.000000    1.000000    0.000000    1.000000    0.000000    0.000000    0.000000
 stdev                      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
est_vertical_line mean      7.000000   13.750000    6.125000   14.625000    6.375000   13.875000    9.500000   15.750000   10.000000    8.875000
est_vertical_line median    7.500000   14.000000    6.000000   14.500000    6.000000   14.000000    9.500000   16.000000   10.000000    9.000000
est_vertical_line stdev     1.195229    0.886405    2.167124    1.060660    0.916125    1.246423    1.195229    0.462910    2.203893    1.457738
```

Full statistics for non-letters:



```
                                    sad     smiley     xclaim
nr_pix mean                   31.300000  33.600000  51.250000
nr_pix median                 32.000000  33.500000  50.000000
nr_pix stdev                   2.597570   3.424371  21.474281
rows_with_1 mean               3.350000   3.350000   0.650000
rows_with_1 median             3.500000   3.000000   0.000000
rows_with_1 stdev              1.386969   1.348488   2.476734
cols_with_1 mean               6.950000   5.300000   0.000000
cols_with_1 median             7.000000   5.000000   0.000000
cols_with_1 stdev              1.145931   1.780006   0.000000
rows_with_3p mean              5.700000   6.500000  10.450000
rows_with_3p median            5.500000   6.500000  12.500000
rows_with_3p stdev             0.978721   1.147079   5.443248
cols_with_3p mean              5.500000   5.500000   4.150000
cols_with_3p median            5.500000   5.000000   4.000000
cols_with_3p stdev             0.688247   0.688247   1.424411
aspect_ratio mean              1.057491   1.171937   0.279457
aspect_ratio median            1.069048   1.160256   0.285714
aspect_ratio stdev             0.073840   0.110774   0.092746
neigh_1 mean                   3.000000   3.150000   0.100000
neigh_1 median                 3.000000   3.000000   0.000000
neigh_1 stdev                  0.725476   0.812728   0.447214
no_neigh_above mean           13.200000  11.850000   6.300000
no_neigh_above median         12.500000  11.500000   6.000000
no_neigh_above stdev           2.375312   1.598519   2.202869
no_neigh_below mean           12.550000  12.600000   5.900000
no_neigh_below median         12.000000  12.000000   6.000000
no_neigh_below stdev           2.012461   1.957442   2.198085
no_neigh_left mean            10.800000  11.750000  12.050000
no_neigh_left median          11.000000  11.000000  12.000000
no_neigh_left stdev            1.507874   2.149051   1.904980
no_neigh_right mean           11.150000  12.150000  12.150000
no_neigh_right median         11.000000  12.000000  12.000000
no_neigh_right stdev           1.631112   2.084403   1.926956
no_neigh_horiz mean            8.650000   9.550000   0.650000
no_neigh_horiz median          8.000000   9.000000   0.000000
no_neigh_horiz stdev           2.852054   2.645254   2.476734
no_neigh_vert mean            10.200000   8.300000   0.050000
no_neigh_vert median          10.000000   8.500000   0.000000
no_neigh_vert stdev            1.507874   2.154555   0.223607
connected_areas mean           4.000000   4.050000   2.000000
connected_areas median         4.000000   4.000000   2.000000
connected_areas stdev          0.000000   0.223607   0.000000
eyes mean                      0.000000   0.000000   0.000000
eyes median                    0.000000   0.000000   0.000000
eyes stdev                     0.000000   0.000000   0.000000
longest_vertical_line mean     4.000000   4.100000  10.300000
longest_vertical_line median   4.000000   4.000000  10.000000
longest_vertical_line stdev    0.858395   0.788069   1.301821
```

Summary statistics:



```
                                Letter  Non-Letter
nr_pix mean                  27.212500   38.716667
nr_pix median                28.000000   33.500000
nr_pix stdev                  8.340376   15.336996
rows_with_1 mean              5.912500    2.450000
rows_with_1 median            7.000000    3.000000
rows_with_1 stdev             4.137491    2.197263
cols_with_1 mean              0.962500    4.083333
cols_with_1 median            0.000000    4.500000
cols_with_1 stdev             1.335435    3.222230
rows_with_3p mean             4.162500    7.550000
rows_with_3p median           3.000000    6.000000
rows_with_3p stdev            2.861547    3.828639
cols_with_3p mean             3.725000    5.050000
cols_with_3p median           3.000000    5.000000
cols_with_3p stdev            2.146383    1.170615
aspect_ratio mean             0.536072    0.836295
aspect_ratio median           0.500000    1.066667
aspect_ratio stdev            0.242258    0.410317
neigh_1 mean                  1.450000    2.083333
neigh_1 median                1.000000    3.000000
neigh_1 stdev                 1.146265    1.565428
no_neigh_above mean           5.625000   10.450000
no_neigh_above median         5.000000   11.000000
no_neigh_above stdev          2.753019    3.642615
no_neigh_below mean           6.012500   10.350000
no_neigh_below median         5.500000   11.000000
no_neigh_below stdev          2.931685    3.763450
no_neigh_left mean           11.675000   11.533333
no_neigh_left median         11.000000   11.000000
no_neigh_left stdev           4.068325    1.917508
no_neigh_right mean          11.525000   11.816667
no_neigh_right median        12.000000   12.000000
no_neigh_right stdev          4.050238    1.917729
no_neigh_horiz mean          11.437500    6.283333
no_neigh_horiz median        11.500000    7.000000
no_neigh_horiz stdev          4.708849    4.808508
no_neigh_vert mean            6.250000    6.183333
no_neigh_vert median          6.000000    8.000000
no_neigh_vert stdev           4.026809    4.688579
connected_areas mean          1.200000    3.350000
connected_areas median        1.000000    4.000000
connected_areas stdev         0.402524    0.971195
eyes mean                     0.500000    0.000000
eyes median                   0.500000    0.000000
eyes stdev                    0.503155    0.000000
longest_vertical_line mean   10.587500    6.133333
longest_vertical_line median 10.000000    5.000000
longest_vertical_line stdev   3.703210    3.132480
```

To look for three statistics that might be significant, I'm going to look for statistics that have different means, with comparatively low standard deviation. Then, I'll look at the medians and check for outliers in the full set of statistics.

While eyes initially seems promising, looking at the full set of letter statistics reveals that only about half the letters have eyes. While if an image has eyes, it's a letter, an image not having eyes doesn't necessarily mean it's not a letter.

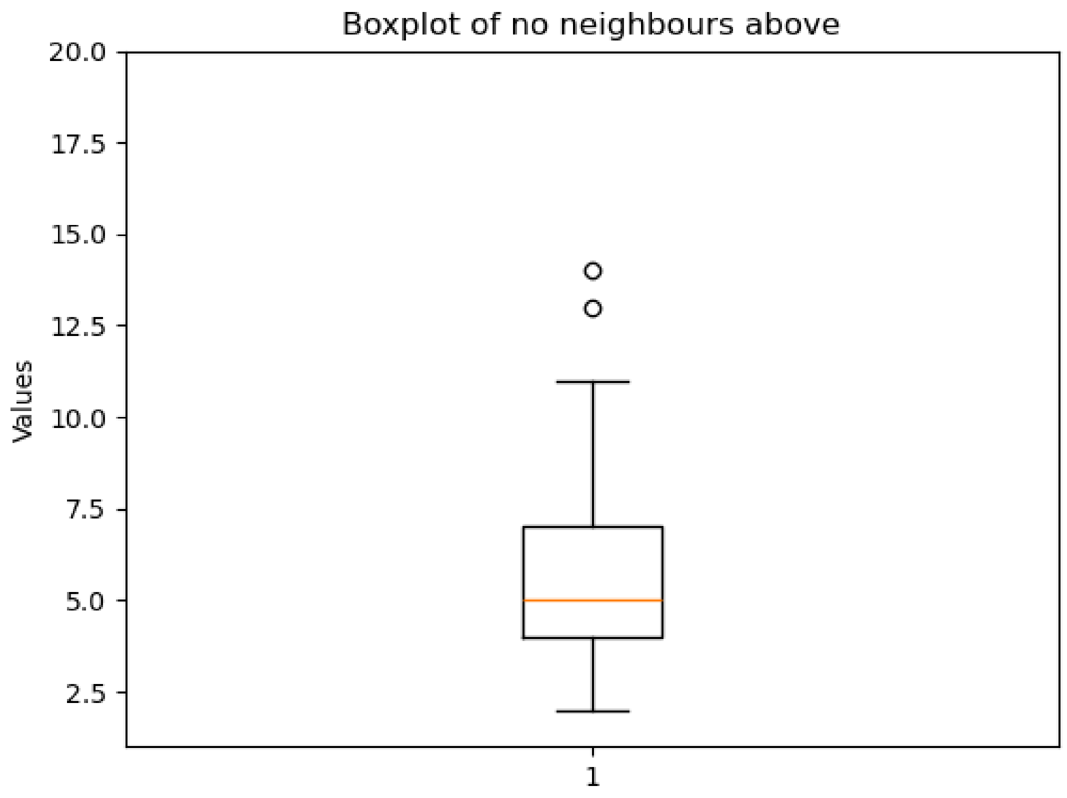
Connected areas is very promising. The standard deviation is low, and there is a significant difference in the means between letters and non-letters. The outliers aren't as significant as eyes.

'No_neigh_above' and 'no_neigh_horiz' both have significant differences in means, and fairly low standard deviation.

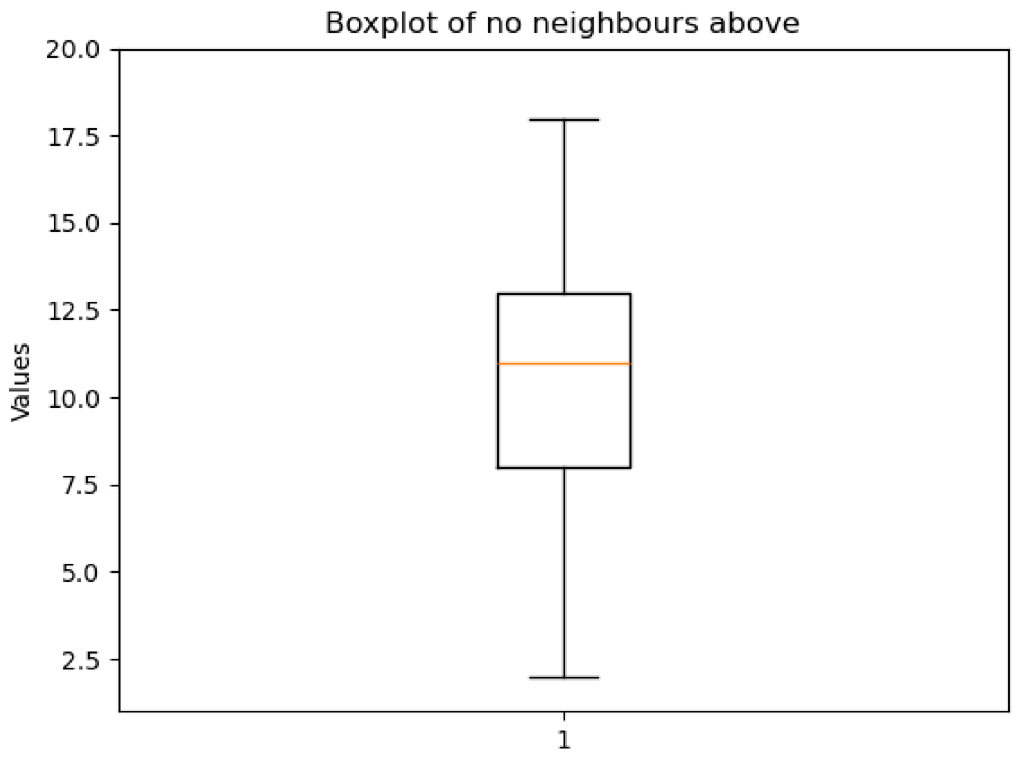
'connected_areas', 'no_neigh_above' and 'no_neigh_horiz' seem to be the most interesting features, and I further visualized them below.

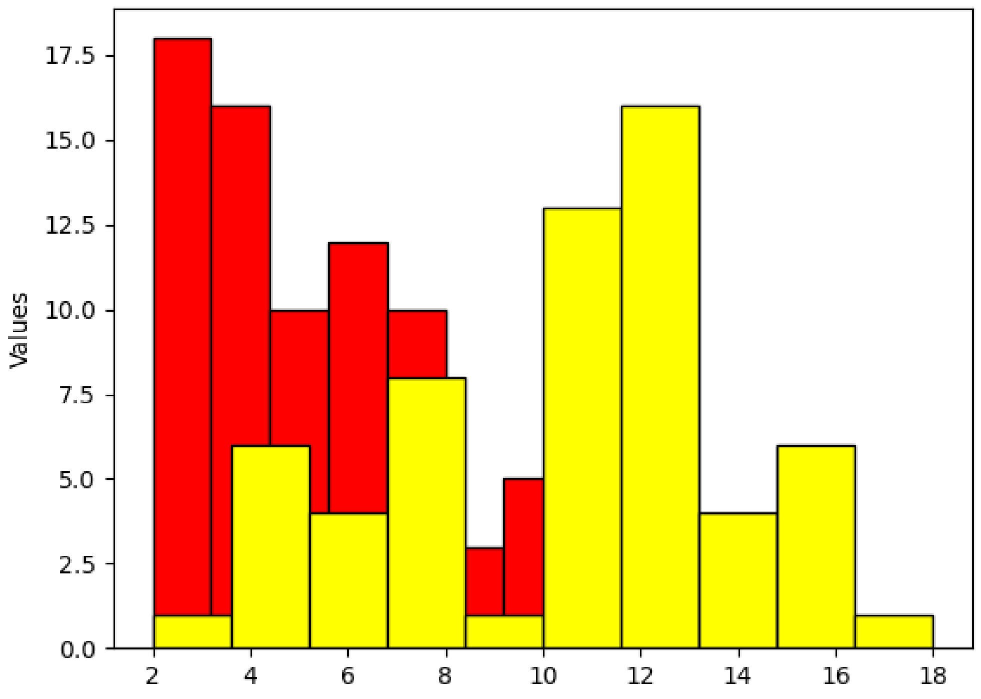
No neighbours above

Letters



Non-letters:



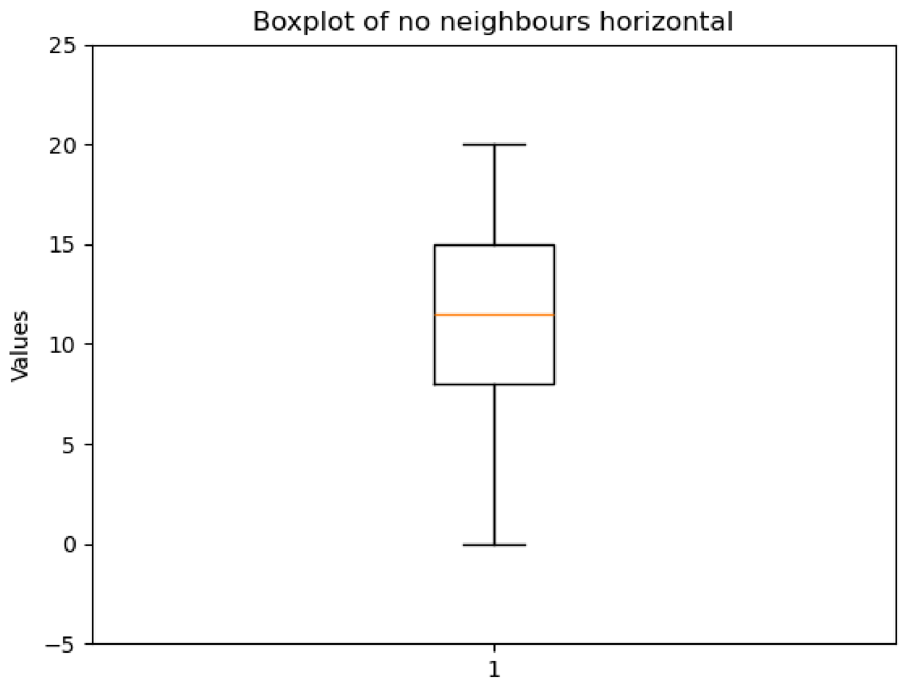


Red represents letters, yellow represents non-letters.

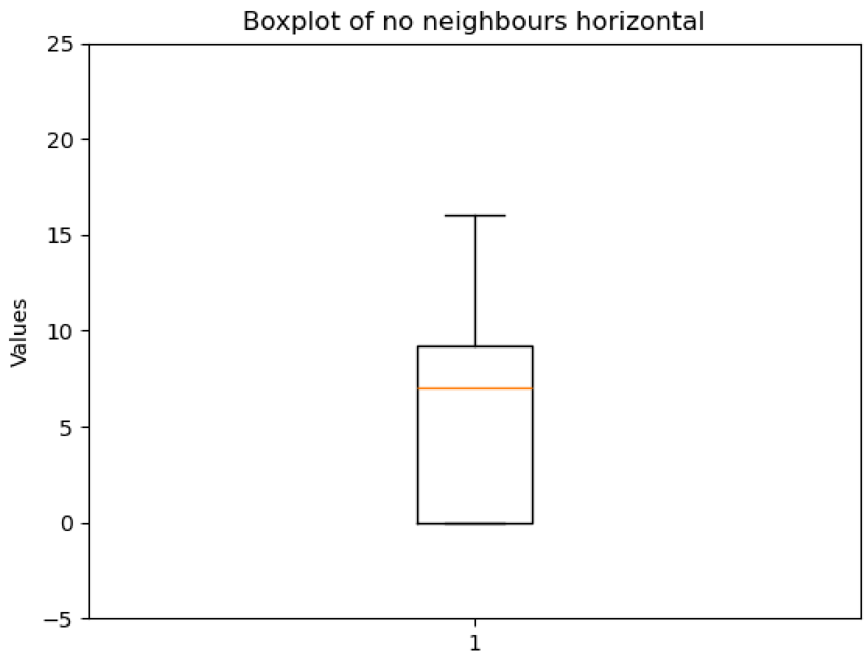
As visible in the boxplots and histograms, no neighbours above doesn't have particularly significant overlap, aside from a few outliers.

No neighbours horizontal

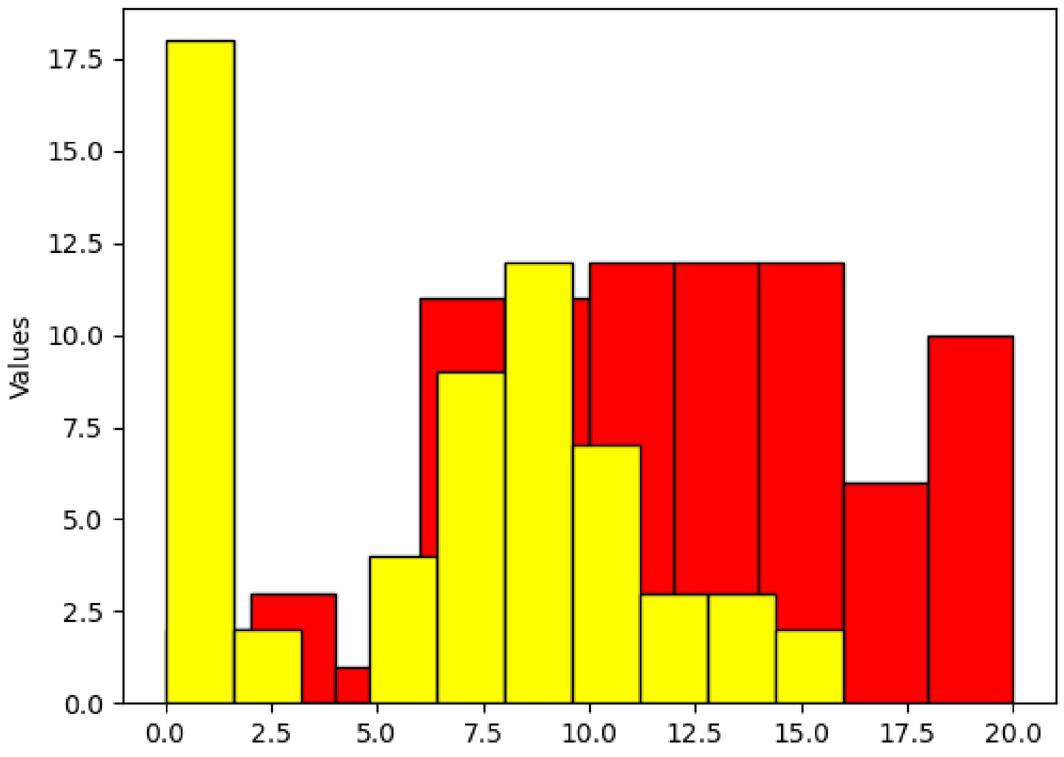
Letters:



Non-letters:

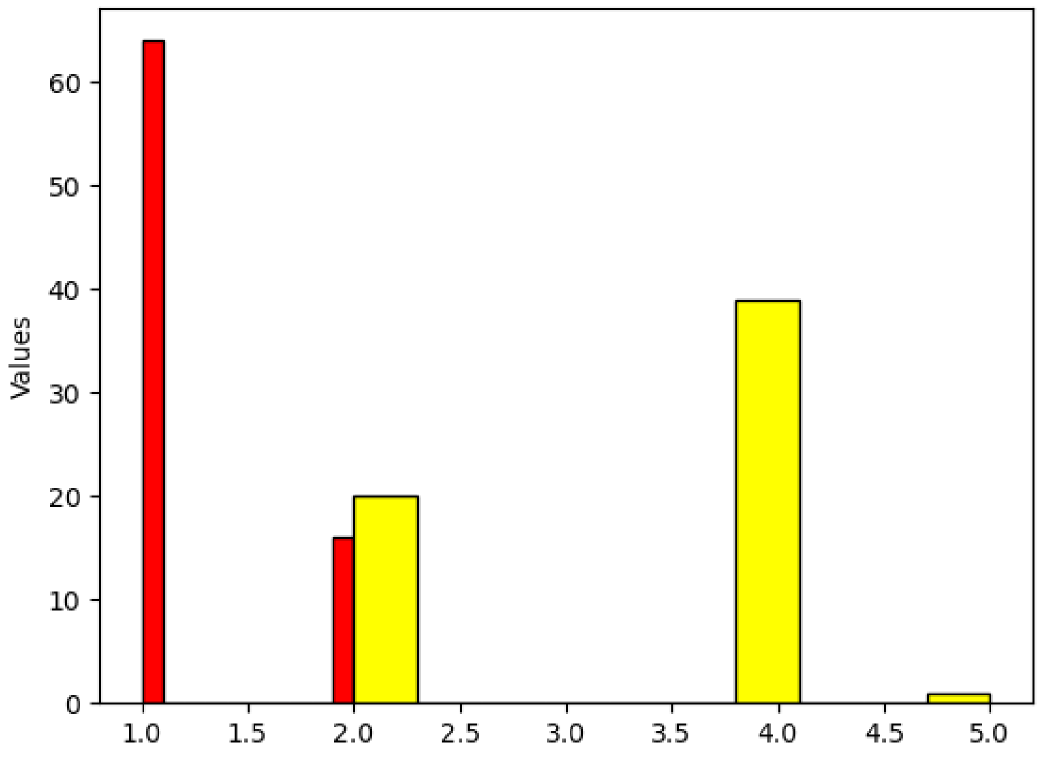


While there is some overlap between the 25th and 75th quartile, I think no neighbours horizontal is still different enough to be discriminatory (as non-letters is extremely centered around the bottom of the graph).



The histogram reveals more. While the higher values of 'no_neigh_horiz' aren't very discriminatory, letters (red) have an extremely high concentration of 0s, which distinguishes them from non-letters (yellow).

Connected Areas



Connected areas has very small range (the boxplot for letters is a single line with one outlier, so I didn't display it). However, there's a clear and consistent difference between letters (red) and non-letters(yellow).

3.3

We're now going to test each of the features to determine whether there is a significant difference between letters and non-letters. We'll use T-tests for this purpose.

We have to choose the significance level (α) before we see the p-values, so we'll determine our α to be 0.05.

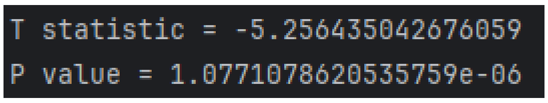
Scipy.stats has a very easy to use two-sided T-test function, so we'll use that.

While some of our values are not normally distributed, I think that attempting to use randomization tests instead of T-tests would be too time consuming. Instead, I'm going to rely on the fact that central limit theorem means we can relax the assumption of normality with large sample sizes (n>30), and just conduct T-tests.

H_0 : The average number of pixels in a letter is equal to the average number of pixels in a non-letter

H_A : The average number of pixels in a letter is not equal to the average number of pixels in a non-letter

Python result:

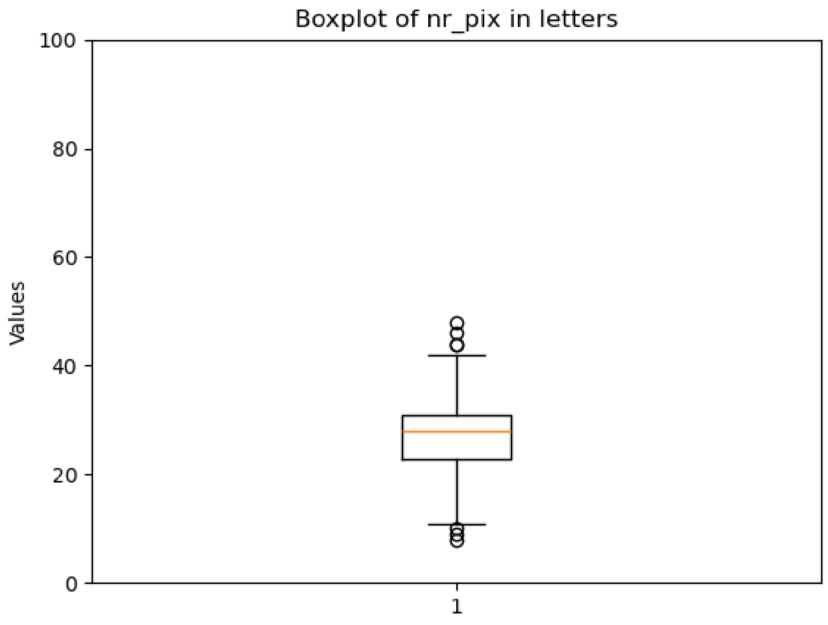


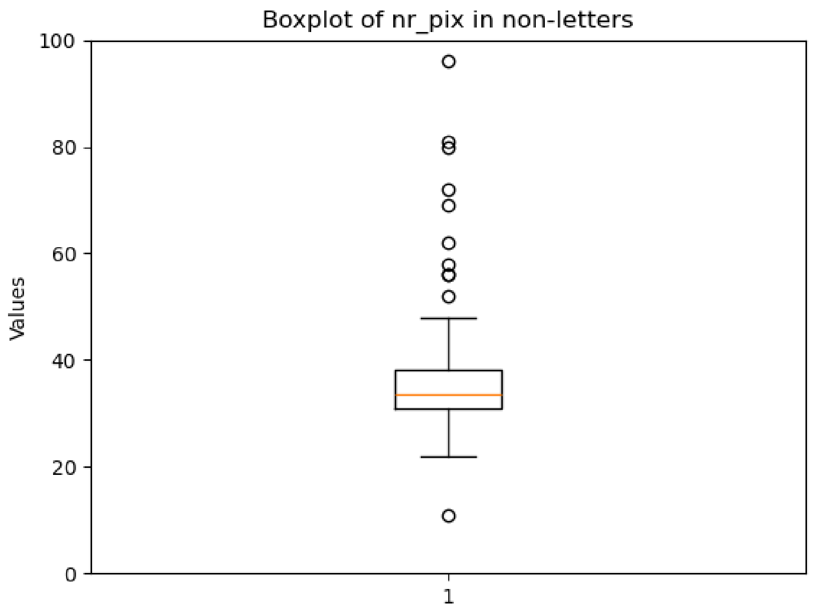
```
T statistic = -5.256435042676059
P value = 1.0771078620535759e-06
```

As the T-statistic is negative, it means that letters generally have on average 5 less pixels that non-letters. The p-value is extremely small, far below our α, so the chance that this occurred randomly is extremely small.

Therefore, since $p < 0.05$, we can reject the null hypothesis and conclude the alternative hypothesis, that the number of pixels in a letter is not equal to the number of pixels in a non-letter.

We can create a boxplot to find more information:





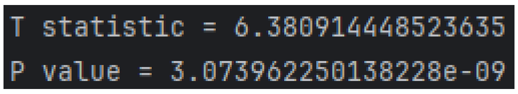
The box plot does reveal a problematic trend. The number of pixels in non-letters has a series of significant outliers that might make it difficult to use the number of pixels as discriminatory feature.

As such, while there is a significant difference in the number of pixels, it might be difficult to use it as a discriminatory feature.

H_0 : the average number of rows with a pixel in a letter is equal to the average number of rows with a pixel in a non-letter

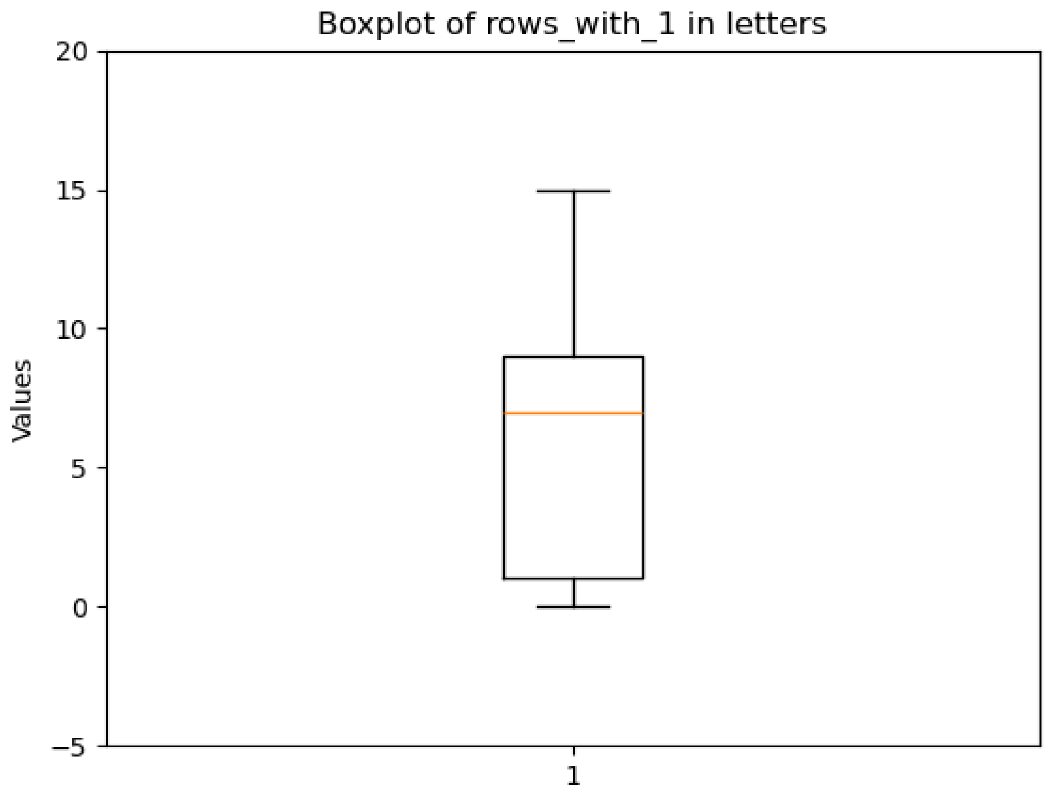
H_A : the average number of rows with a pixel in a letter is not equal to the average number of rows with a pixel in a non-letter

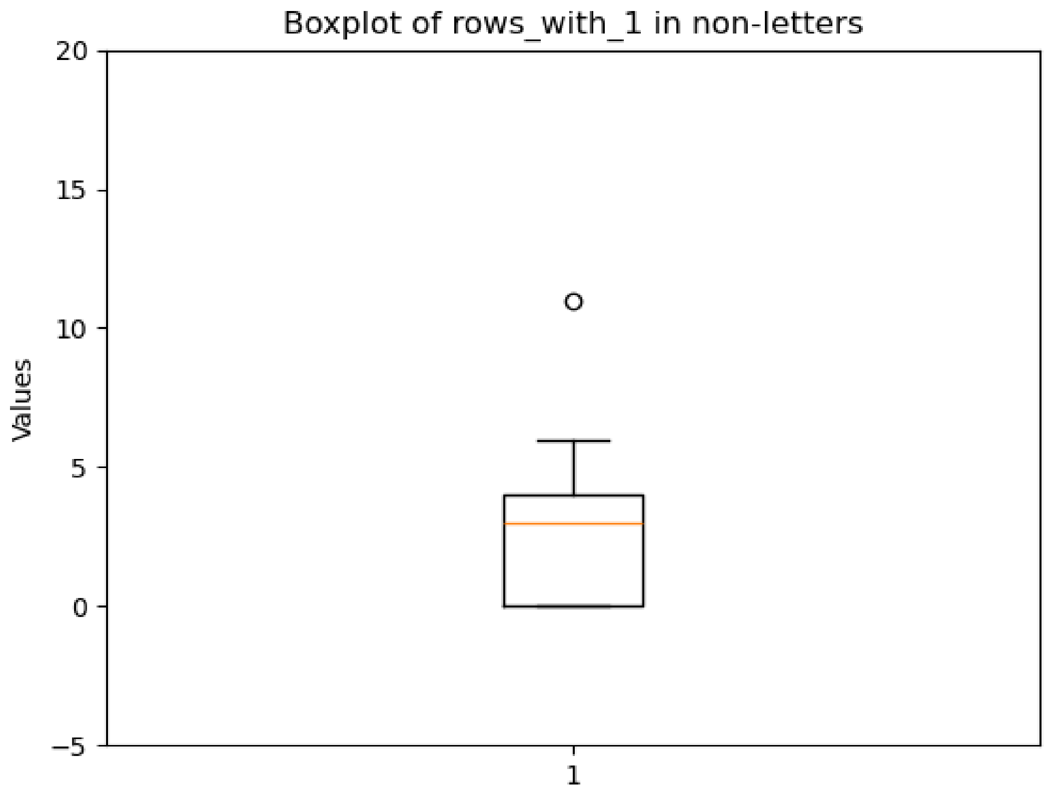
Python result:



```
T statistic = 6.380914448523635
P value = 3.073962250138228e-09
```

As we can see, the T statistic is sufficiently large, and the p value is lower than our α. Thus, we can reject the null hypothesis and adopt the alternative hypothesis.


Boxplot of rows_with_1 in letters
Values
20
15
10
5
0
−5
1

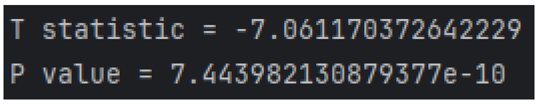

Boxplot of rows_with_1 in non-letters
Values
20
15
10
5
0
−5
1

Our box plot visualisation doesn't display any war signs, as there is only 1 significant outlier. Thus, rows_with_1 maybe a useful characteristic for discrimination.

H_0 : the average number of columns with a pixel in a letter is equal to the average number of columns with a pixel in a non-letter

H_A : the average number of columns with a pixel in a letter is not equal to the average number of columns with a pixel in a non-letter

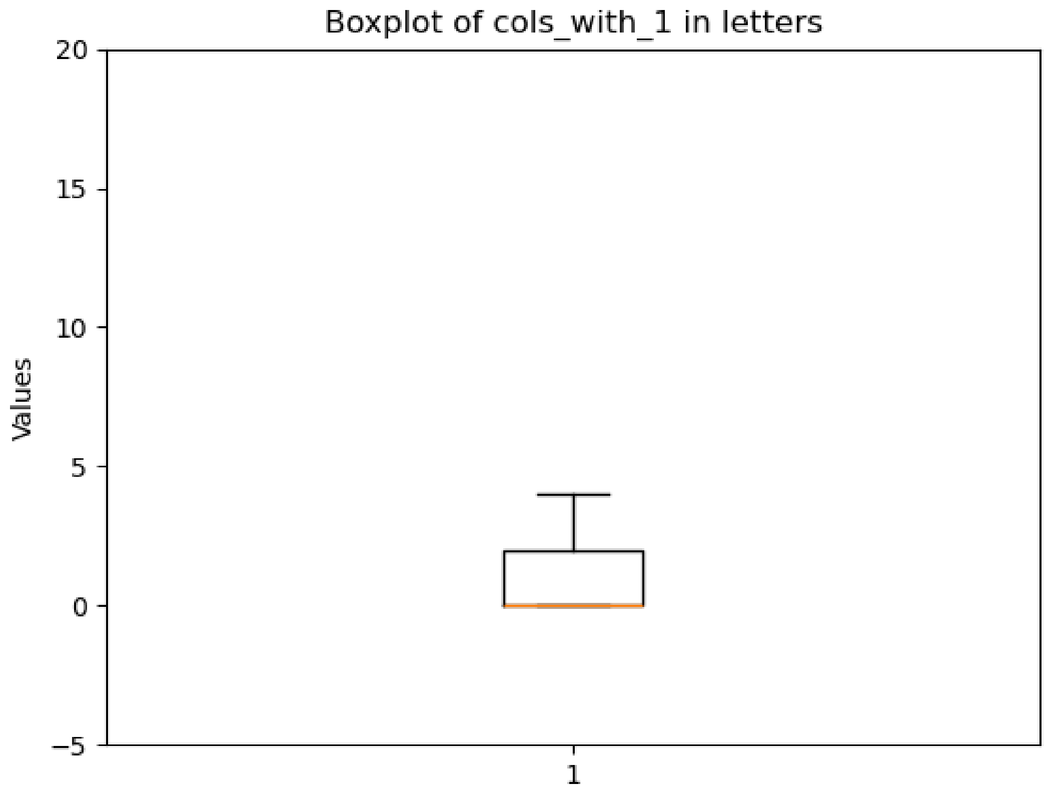
Python result:

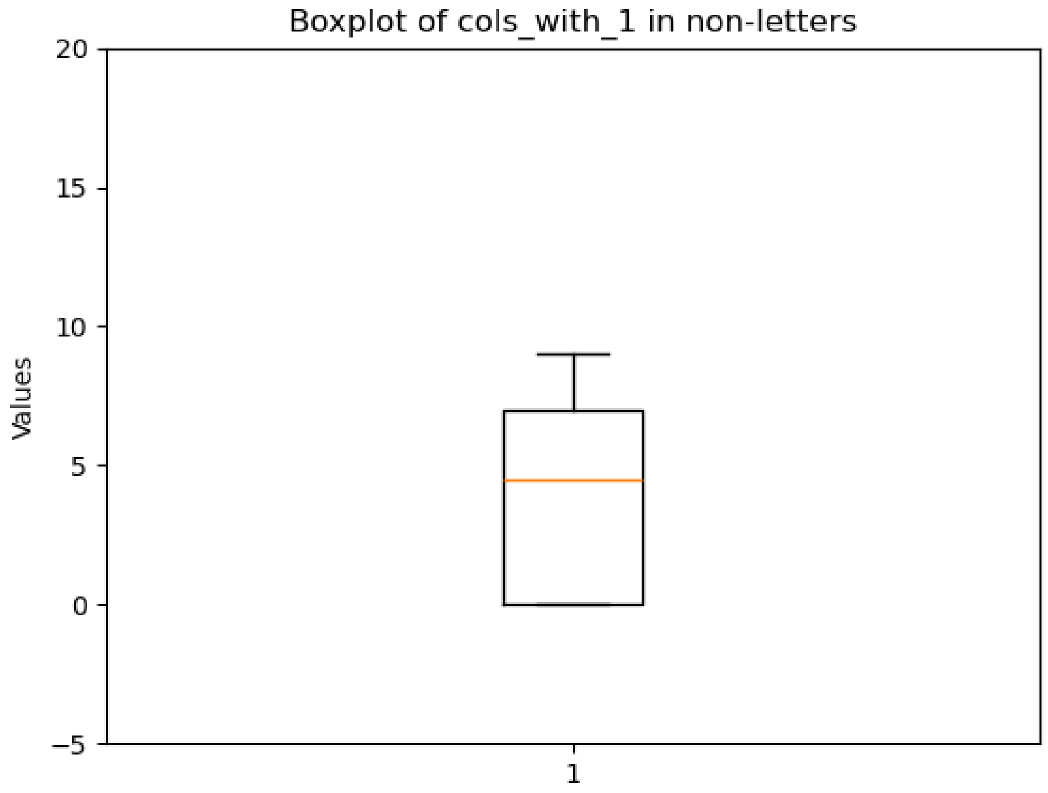


```
T statistic = -7.061170372642229
P value = 7.443982130879377e-10
```

The p value is much smaller than our significance level, so we reject H_0 and adopt H_A .

Of course, we'll visualize this with a boxplot.


Boxplot of cols_with_1 in letters
Values
20
15
10
5
0
−5
1


Boxplot of cols_with_1 in non-letters
Values
20
15
10
5
0
−5
1

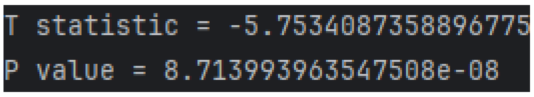
The boxplots show that non-letters have a much greater ranger of values than letters. If we get value outside the range of the letters, but inside the range of non-letters, that might help us determine that it's not a letter.

As such, cols_with_1 has the potential to be a useful feature.

H_0 : the average number of rows with 3 or more pixels in a letter is equal to the average number of rows with 3 or more pixels in a non-letter

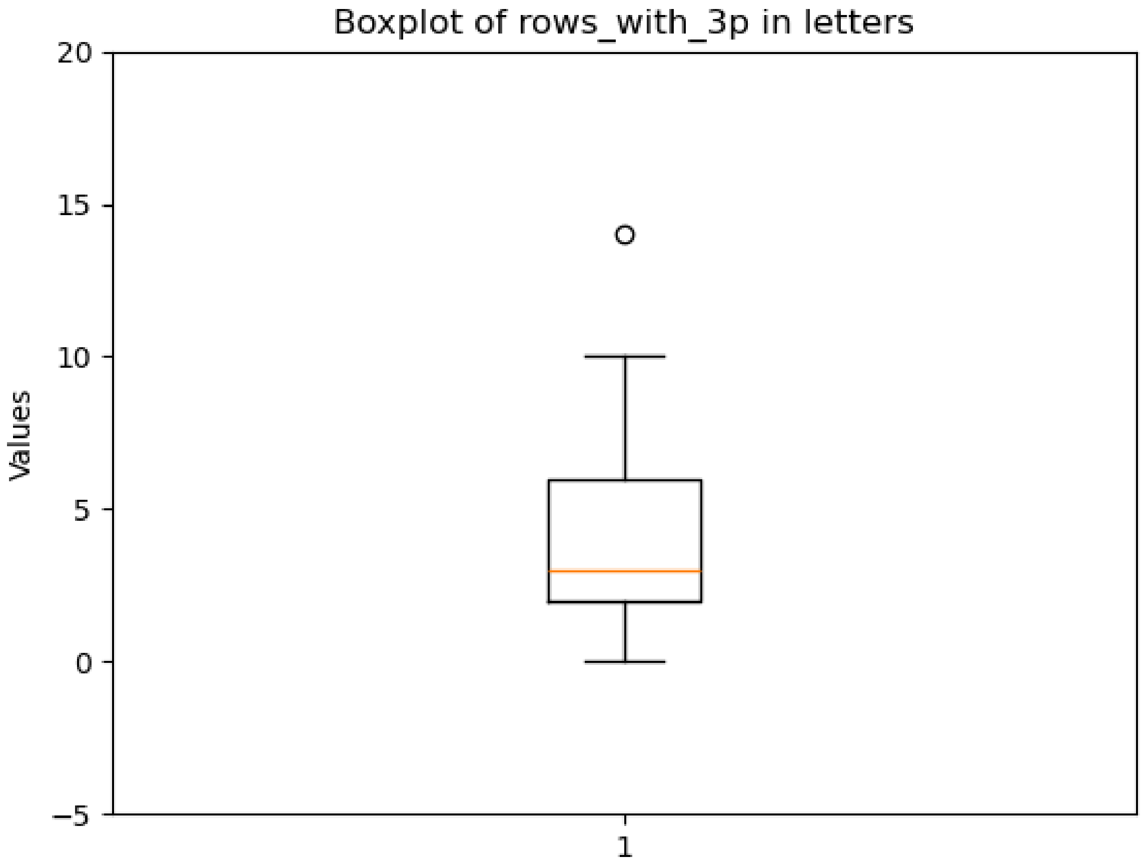
H_A : the average number of rows with 3 or more pixels in a letter is not equal to the average number of rows with 3 or more pixels in a non-letter

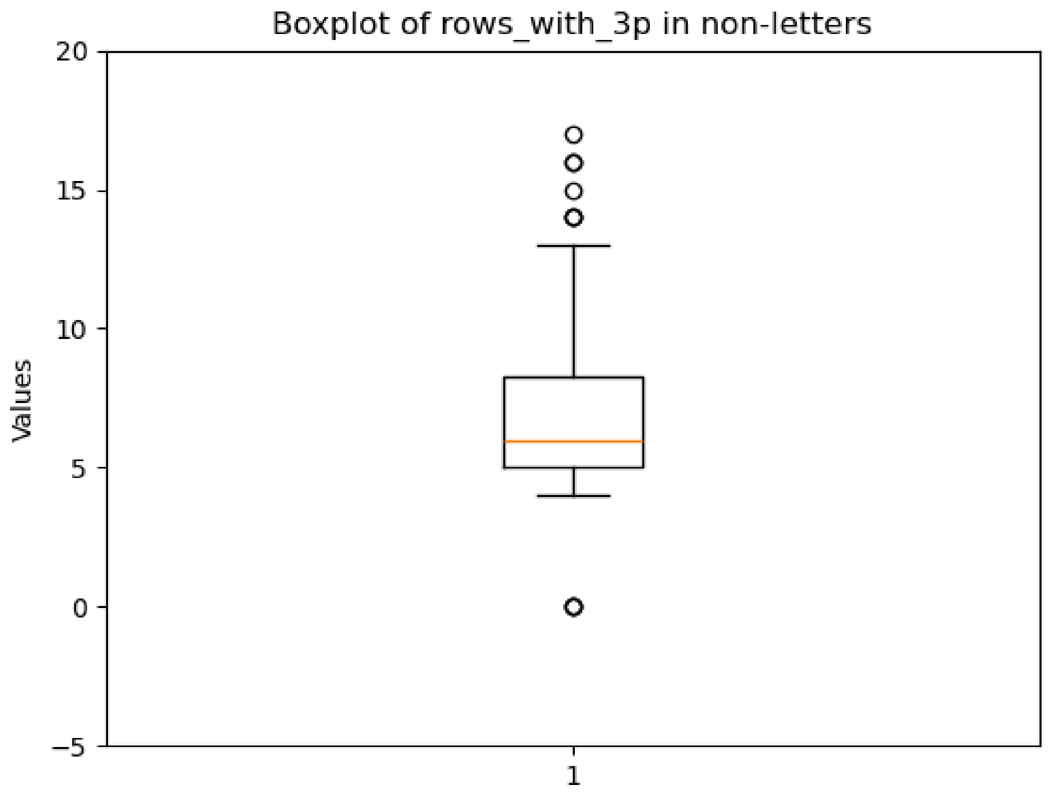
Python result:



```
T statistic = -5.7534087358896775
P value = 8.713993963547508e-08
```

As evidenced, the T statistic is significantly below 0, and the p value is significantly below our significance level. As such, we can reject the null hypothesis in favour of the alternative hypothesis.



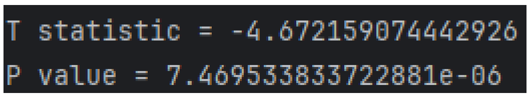


Our boxplots raise some interesting points. The difference isn't as large as anticipated, as the non-letter dataset does have quite a few outliers. However, the box plots are still different enough that the feature could still be of some use for discriminating between letters and non-letters.

H_0 : the average number of columns with 3 or more pixels in a letter is equal to the average number of columns with 3 or more pixels in a non-letter

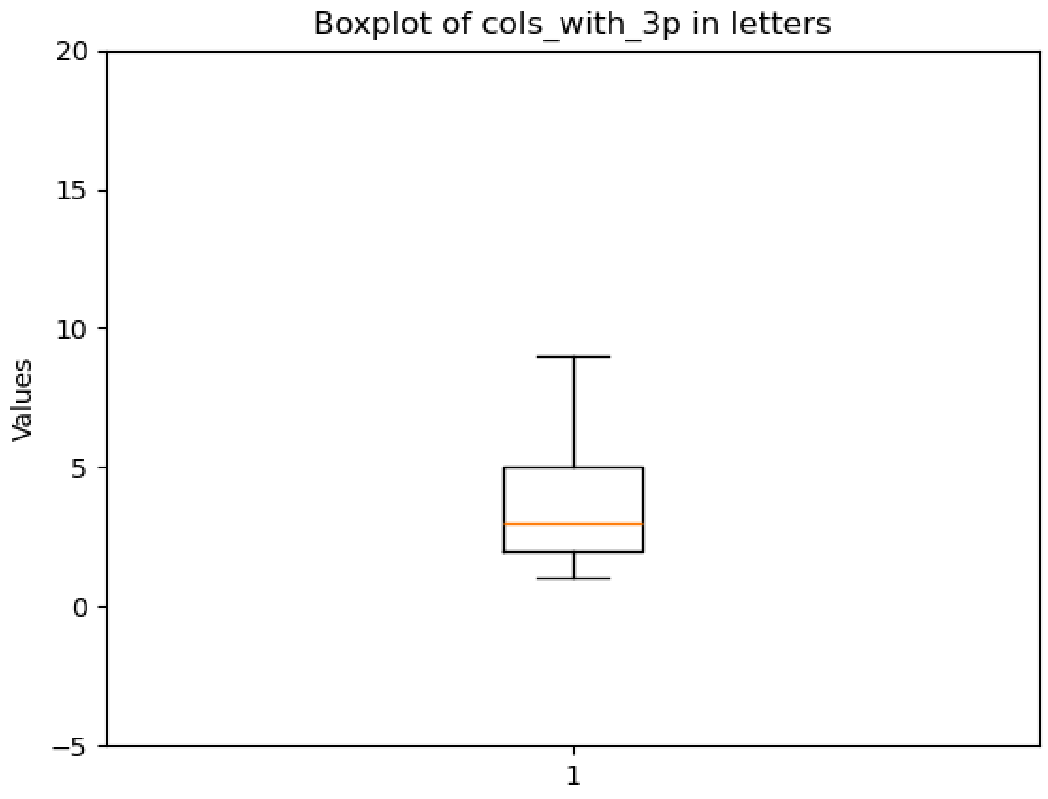
H_A : the average number of columns with 3 or more pixels in a letter is not equal to the average number of columns with 3 or more pixels in a non-letter

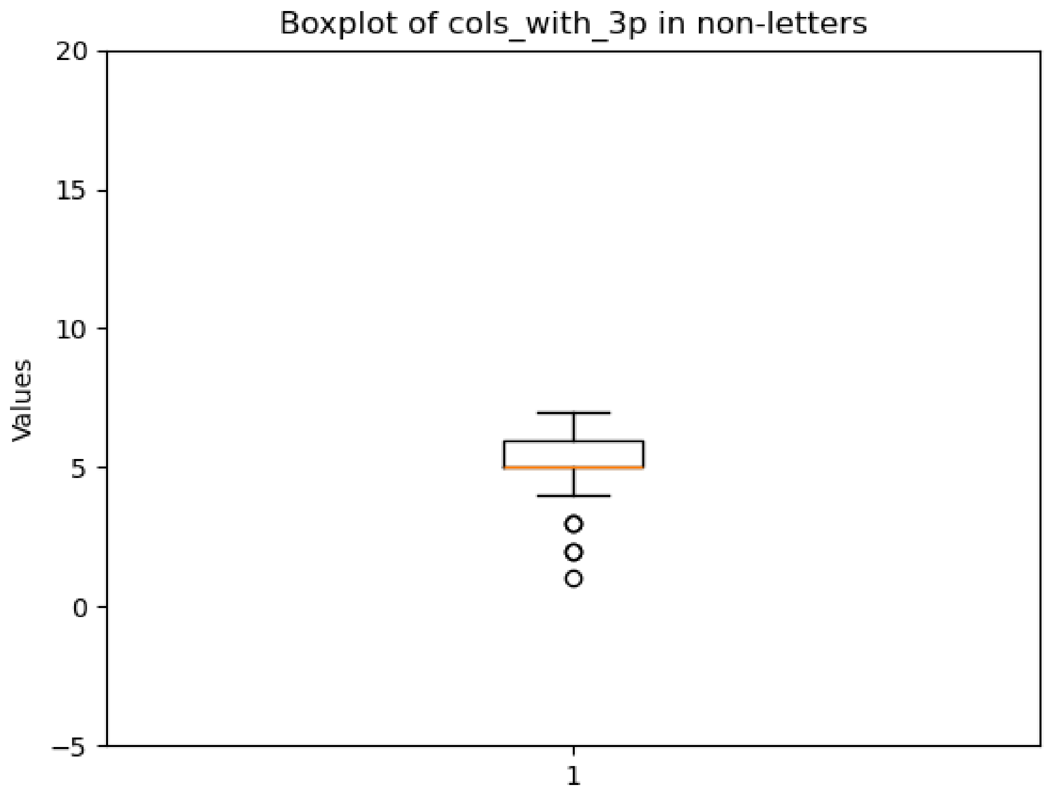
Python result:



```
T statistic = -4.672159074442926
P value = 7.469533833722881e-06
```

As we can see, a significant T statistic and a lower (than 0.05) p value means we can reject the null hypothesis in favour of the alternative hypothesis.


Boxplot of cols_with_3p in letters
Values
20
15
10
5
0
−5
1

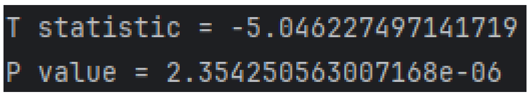

Boxplot of cols_with_3p in non-letters
Values
20
15
10
5
0
−5
1

As observable in the boxplots, non-letters generally have a much tighter range than non-letters. This means this feature could be of some use for discriminating between letters and non-letters.

H_0 : the average aspect ratio of a letter is equal to the average aspect ratio of a non-letter

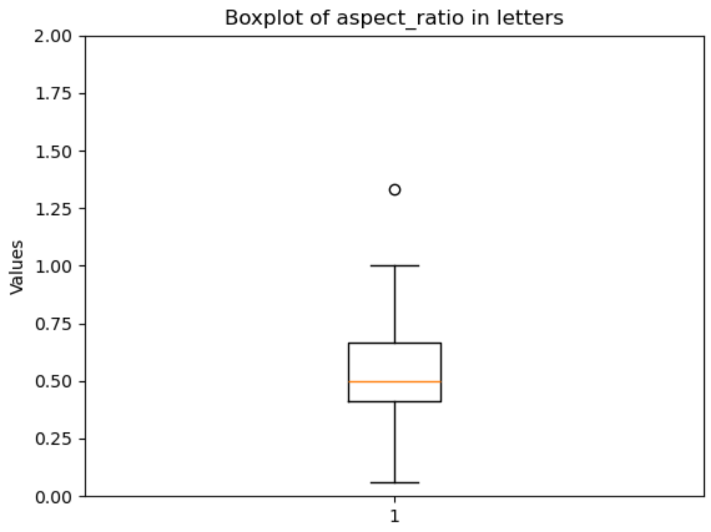
H_A : the average aspect ratio of a letter is not equal to the average aspect ratio of a non-letter

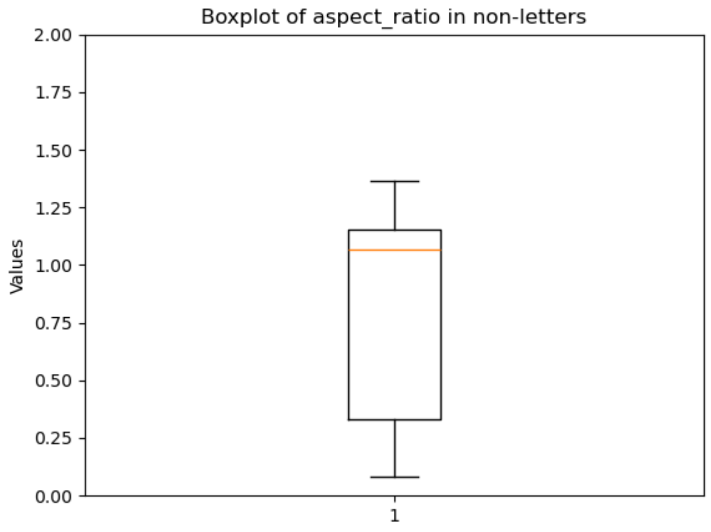
Python result:



```
T statistic = -5.046227497141719
P value = 2.354250563007168e-06
```

As we can see, the p value is lower than our significance level, so we can reject the null hypothesis and adopt the alternative hypothesis.

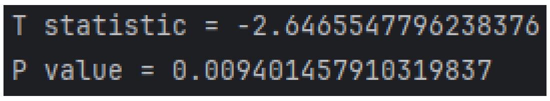




However, the boxplots indicates a problem. While there is a significant difference between the two boxplots, non-letters almost entirely overlaps letters, indicating that they may not be as useful for determining letters or non-letters as the T-statistic and p value would indicate.

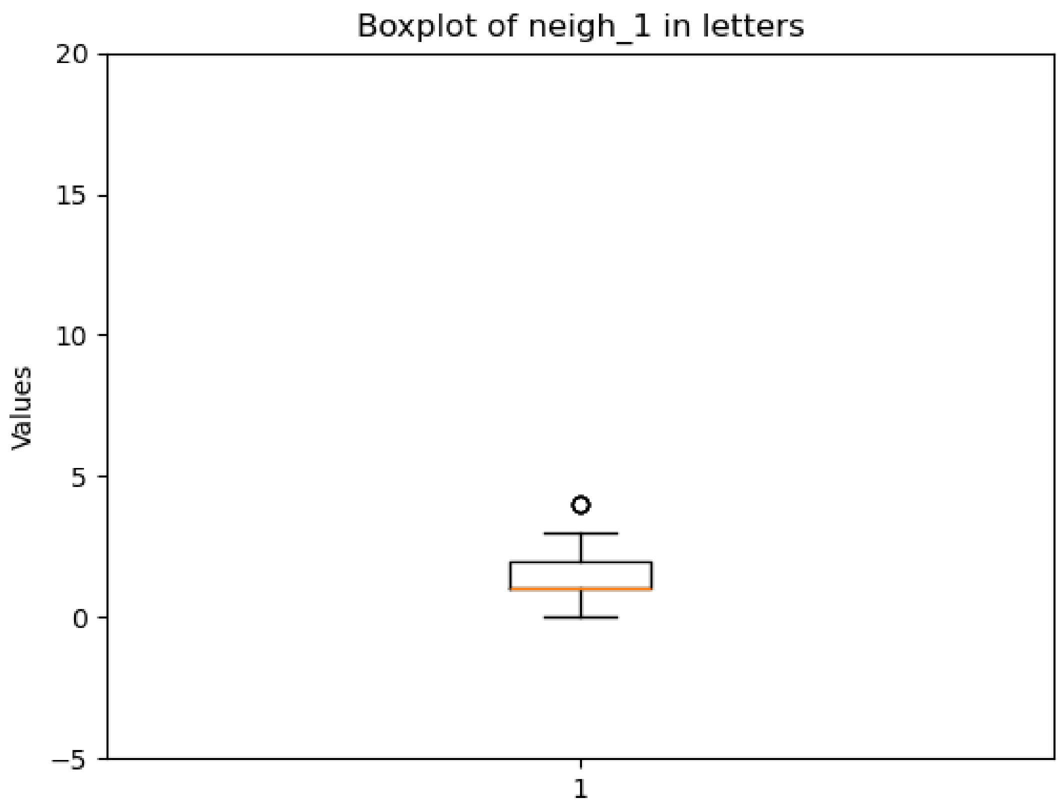
H_0 : the average number of pixels with one neighbour in a letter is equal to the average number of pixels with one neighbour in a non-letter

H_A : the average number of pixels with one neighbour in a letter is not equal to the average number of pixels with one neighbour in a non-letter



```
T statistic = -2.6465547796238376
P value = 0.009401457910319837
```

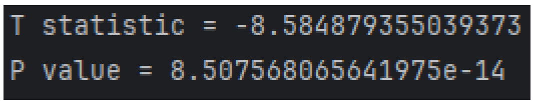
As we can see, while the p value isn't a small as some of our other p values, it is still smaller than the significance level. As such, we can reject the null hypothesis in favour of the alternative hypothesis.



The box plots back-up our T-tests, as we can see a difference in the visualization.

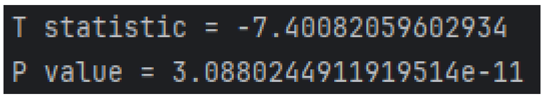
Because the 'no_neigh_[blank]' features all follow a similar format, and some of them capture quite similar information, I'm going to present all of them together, and then pick the most promising ones to visualize.

'No_neigh_above':



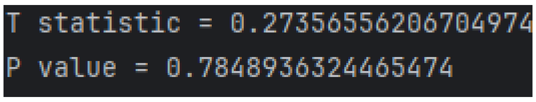
```
T statistic = -8.584879355039373
P value = 8.507568065641975e-14
```

'No_neigh_below':



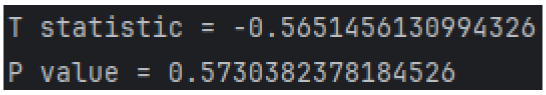
```
T statistic = -7.40082059602934
P value = 3.0880244911919514e-11
```

'No_neigh_left':



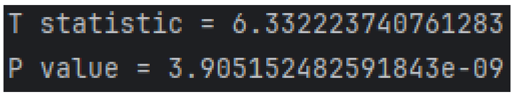
```
T statistic = 0.27356556206704974
P value = 0.7848936324465474
```

'No_neigh_right':



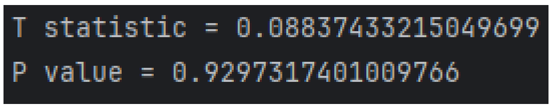
```
T statistic = -0.5651456130994326
P value = 0.5730382378184526
```

'No_neigh_horiz':



```
T statistic = 6.332223740761283
P value = 3.905152482591843e-09
```

'No_neigh_vert':



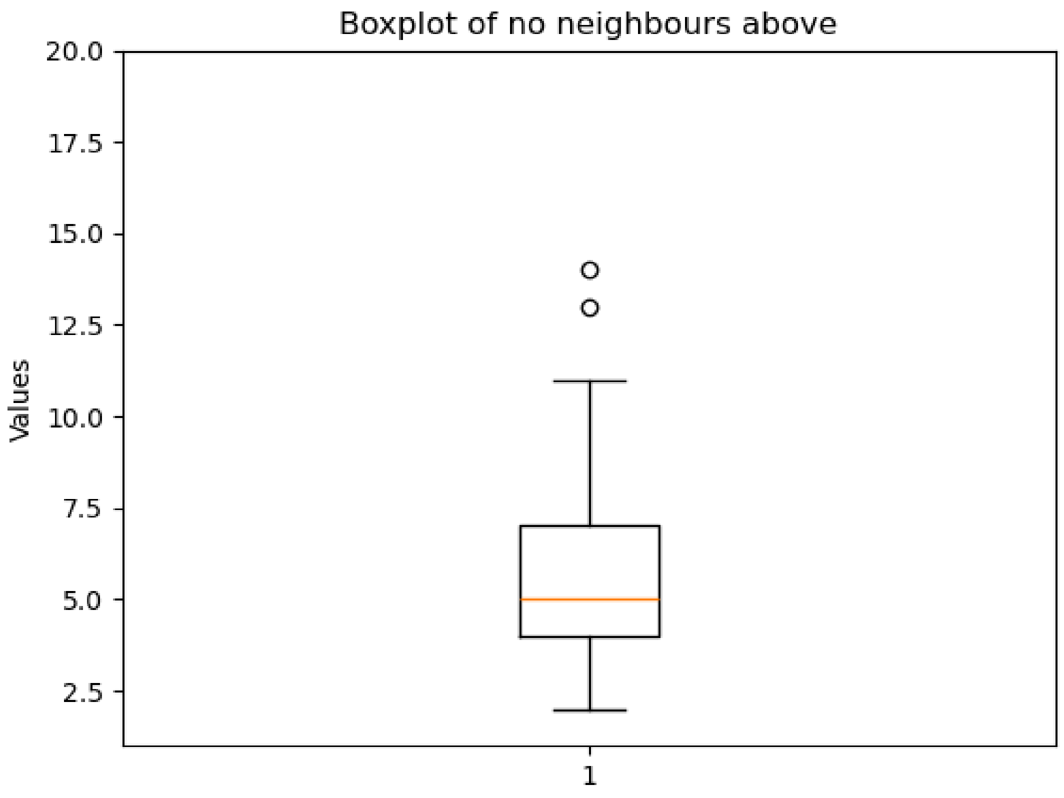
```
T statistic = 0.08837433215049699
P value = 0.9297317401009766
```

We can preserve the null hypothesis 'no_neigh_left', 'no_neigh_right' and 'no_neigh_vert', because their p value falls above our α.

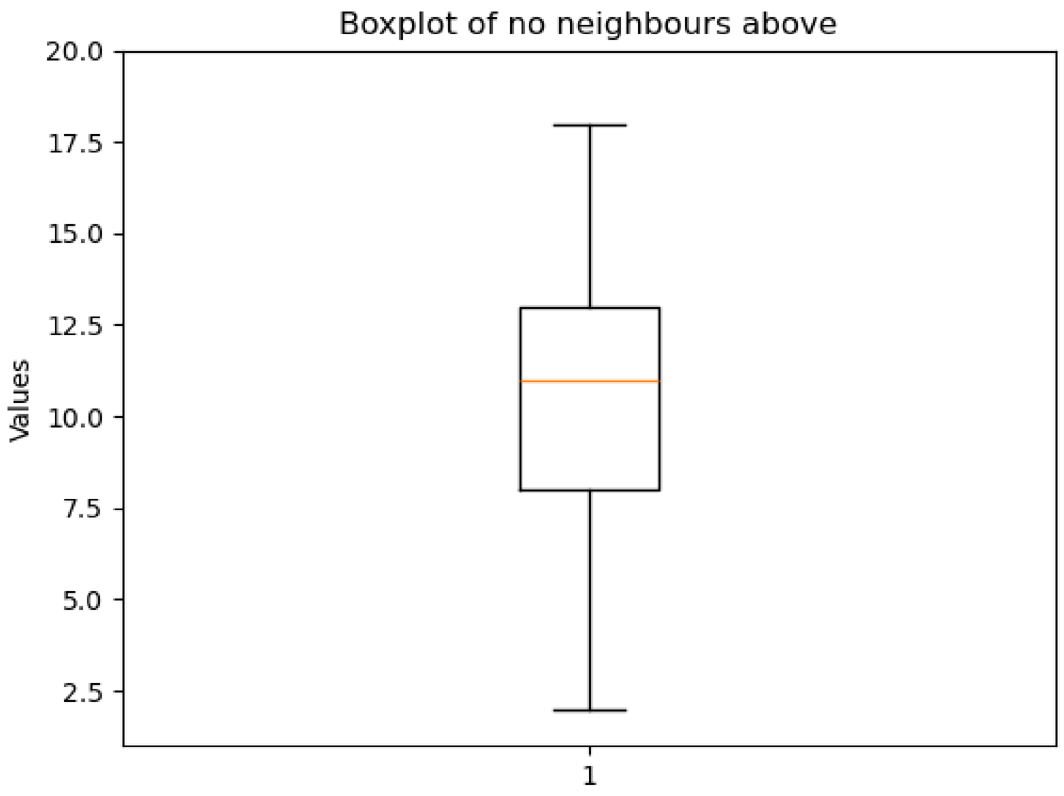
'No_neigh_above', 'no_neigh_below' and 'No_neigh_horiz' all show promise for determining whether a symbol is a letter or not, so I'll visualize and discuss them below.

No_neigh_above

Letters:



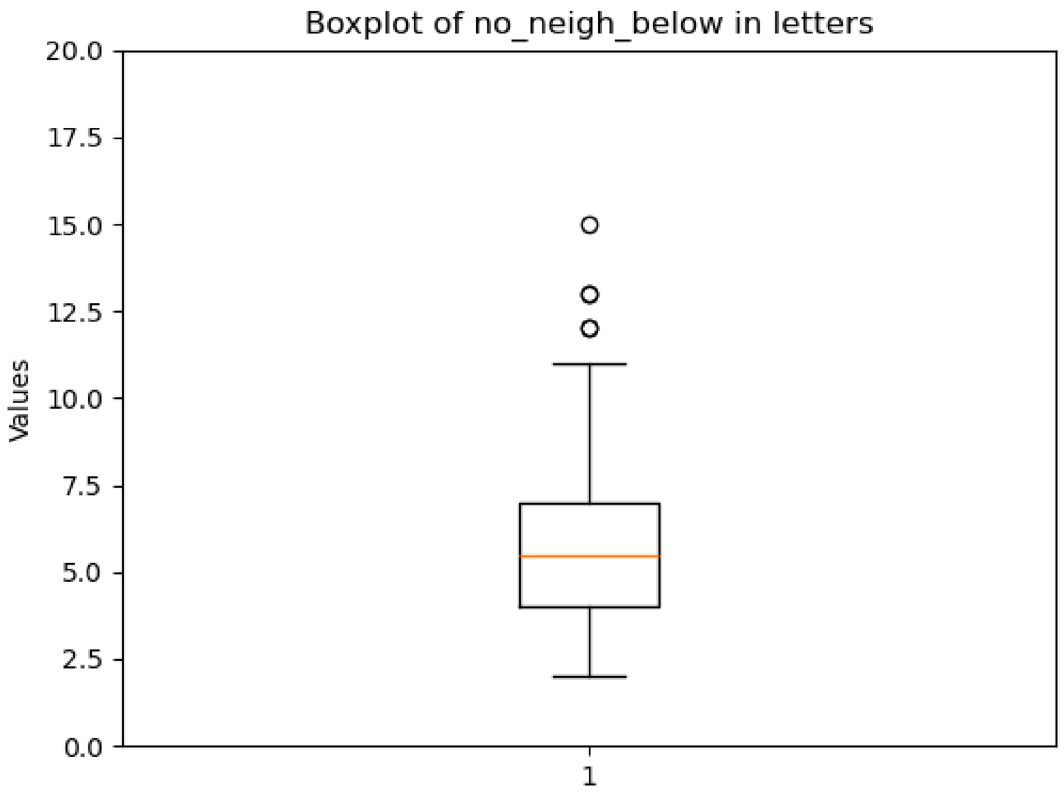
Non-letters:



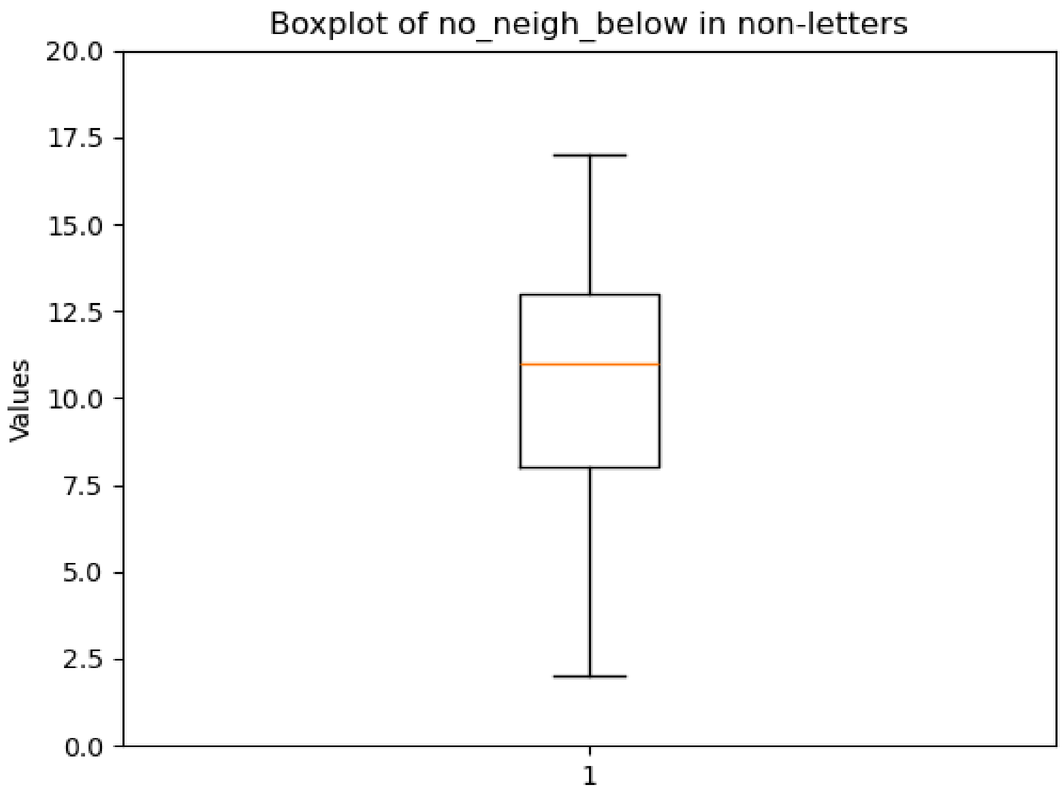
The boxplots show a significant difference between letters and non-letters, meaning 'no_neigh_above' will likely be important for discriminating between letters and non-letters.

No_neigh_below

Letters:



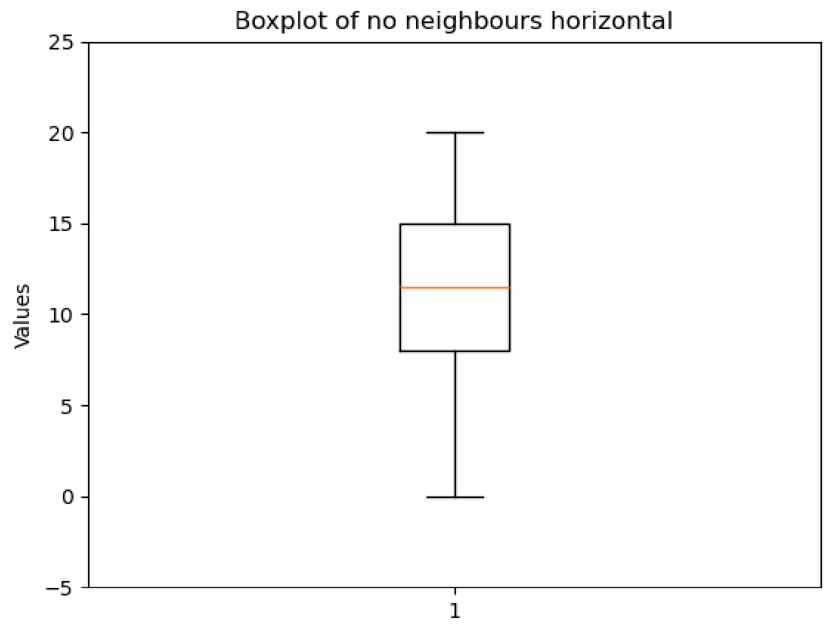
Non-letters:



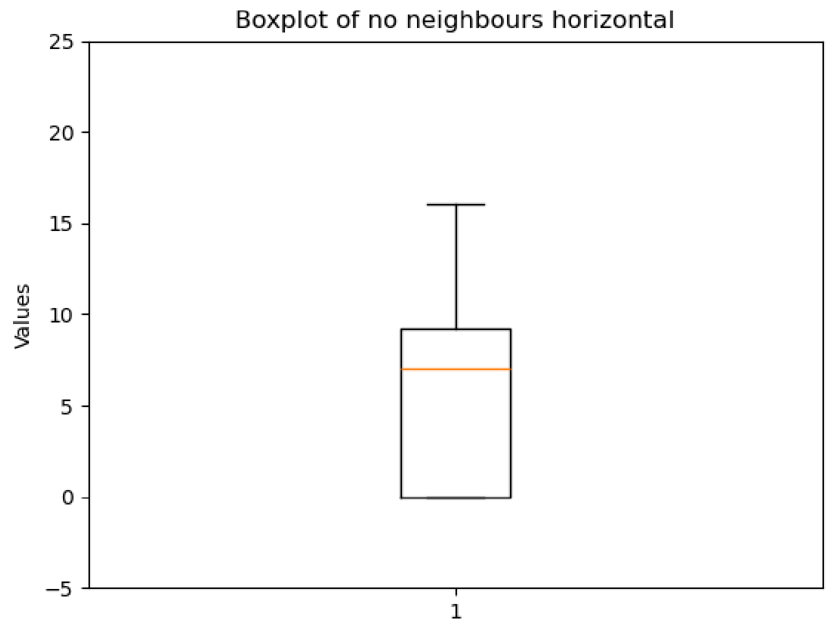
Interestingly, the boxplots for ‘no_neigh_below’ looks very similar to the boxplots for ‘no_neigh_above’, indicating they may capture the same sort of information. However, they do both seem to useful features for determining whether a symbol is a letter or not.

No_neigh_horiz

Letters:



Non-letters:

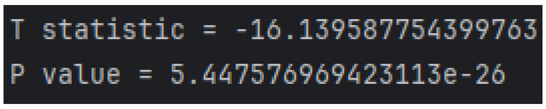


The difference between letters and non-letters is significant, and there aren't any significant outliers, so 'no_neigh_horiz' is likely to prove a very useful feature.

H_0 : the average number of connected areas in a letter is equal to the average number of connected areas in a non-letter

H_A : the average number of connected areas in a letter is not equal to the average number of connected areas in a non-letter

Python result:



```
T statistic = -16.139587754399763
P value = 5.447576969423113e-26
```

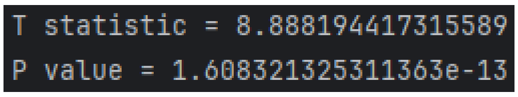
As our p-value is lower than our significance level, we can reject the null hypothesis.

Because the sample range for connected areas is so small, visualizing them as a boxplot isn't particularly helpful (the boxplot for letters is a single line and outline). However, because of the extremely small p value and lack of outliers, we can assume that the number of connected areas with be a very useful feature for discriminating between letters and non-letters.

H_0 : the average number of eyes in a letter is equal to the average number of eyes in a non-letter

H_A : the average number of eyes in a letter is not equal to the average number of eyes in a non-letter

Python result:



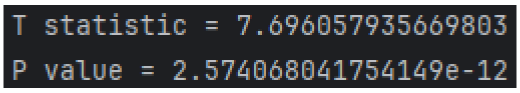
```
T statistic = 8.888194417315589
P value = 1.608321325311363e-13
```

We can reject the null hypothesis in favour of the alternative hypothesis. However, I'm not sure how useful eyes will be for determining whether a symbol is a letter or non-letter, as we only have range of 0-1. Everything with a eye is a letter, but not everything without an eye is not a letter. (Also, visualization with boxplots doesn't help much, as like connected areas, the sample range is extremely small).

H_0 : the average length of the longest vertical line in a letter is equal to the average length of the longest vertical line in a non-letter

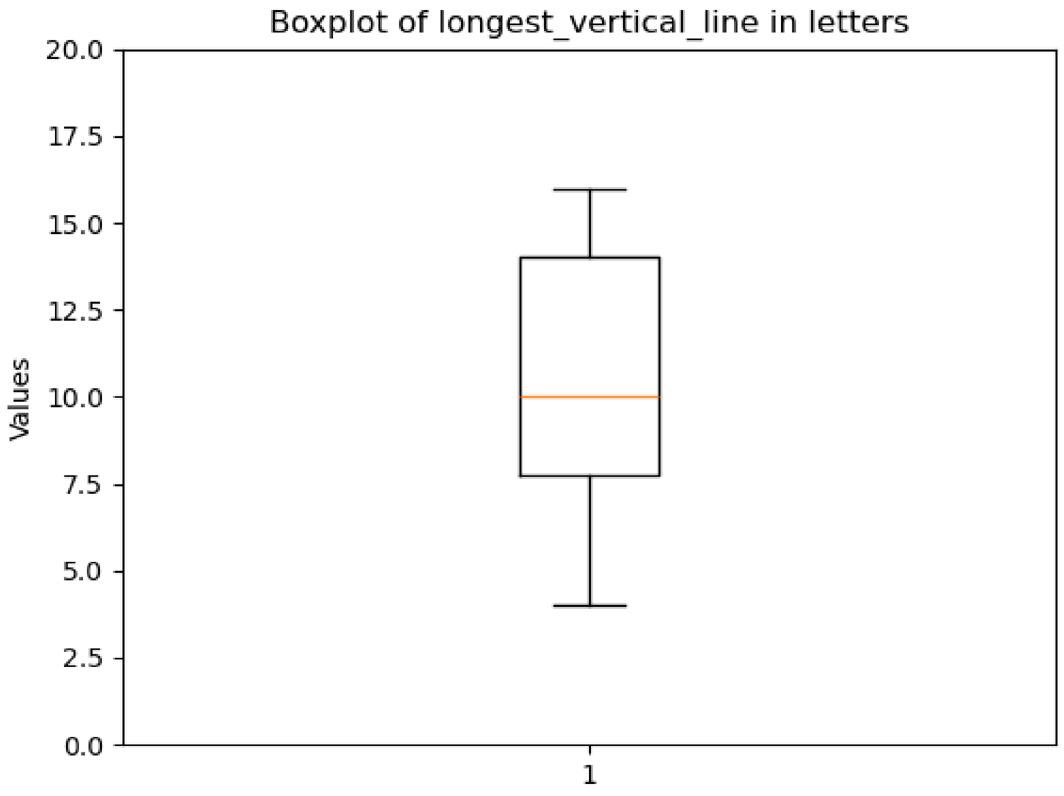
H_A : the average length of the longest vertical line in a letter is not equal to the average length of the longest vertical line in a non-letter

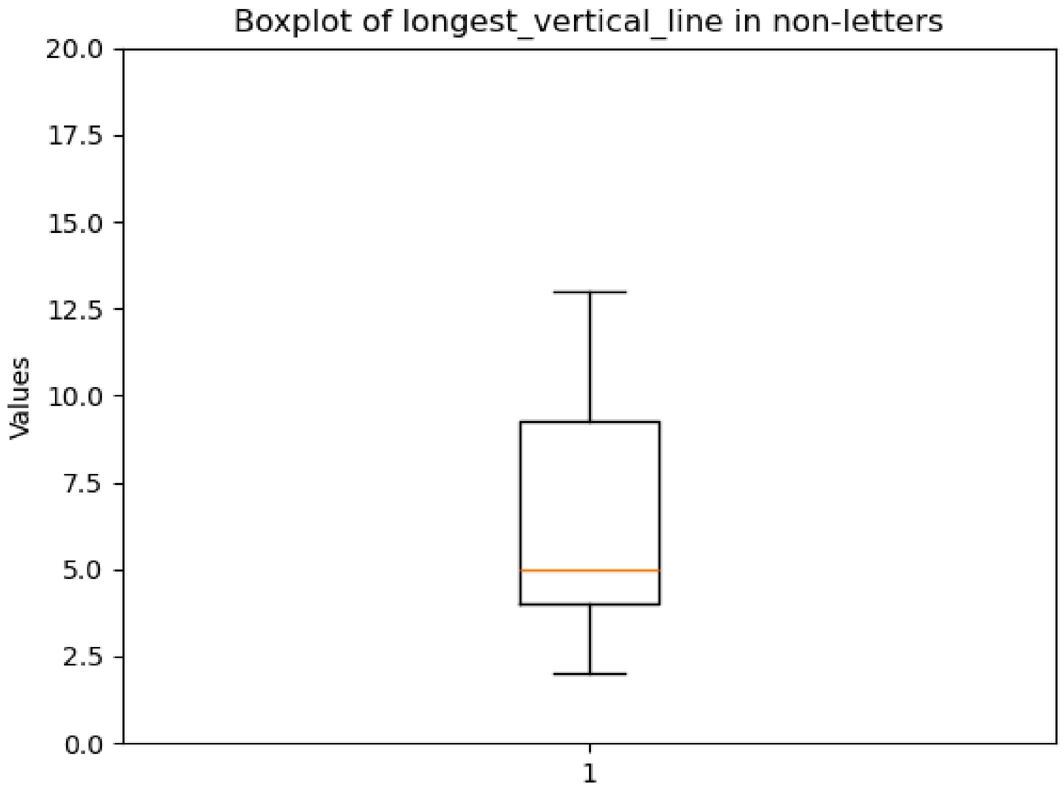
Python result:



```
T statistic = 7.696057935669803
P value = 2.574068041754149e-12
```

As the p value is below our significance level, we can reject the null hypothesis in favour of the alternative hypothesis.

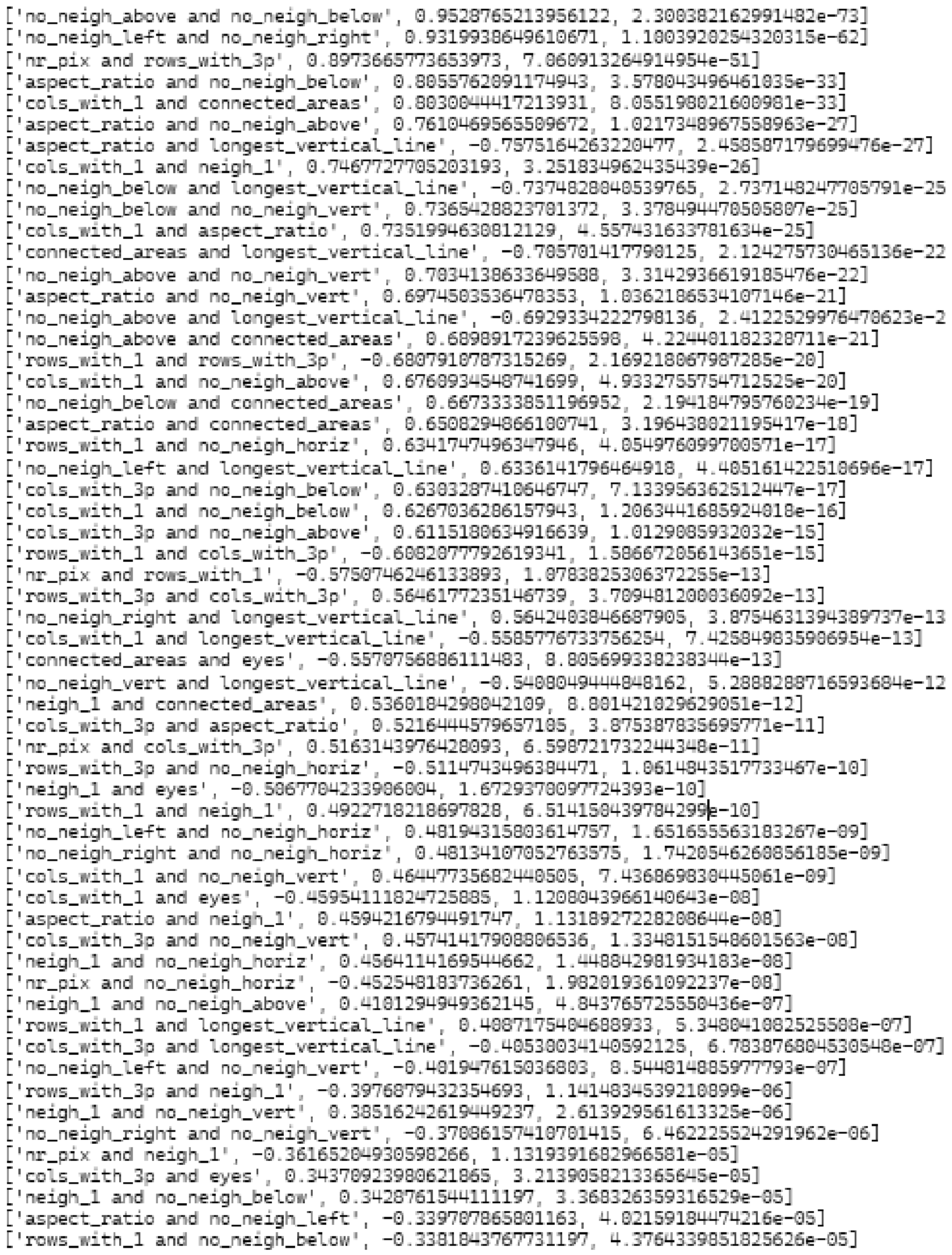

Boxplot of longest_vertical_line in letters
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1


Boxplot of longest_vertical_line in non-letters
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1

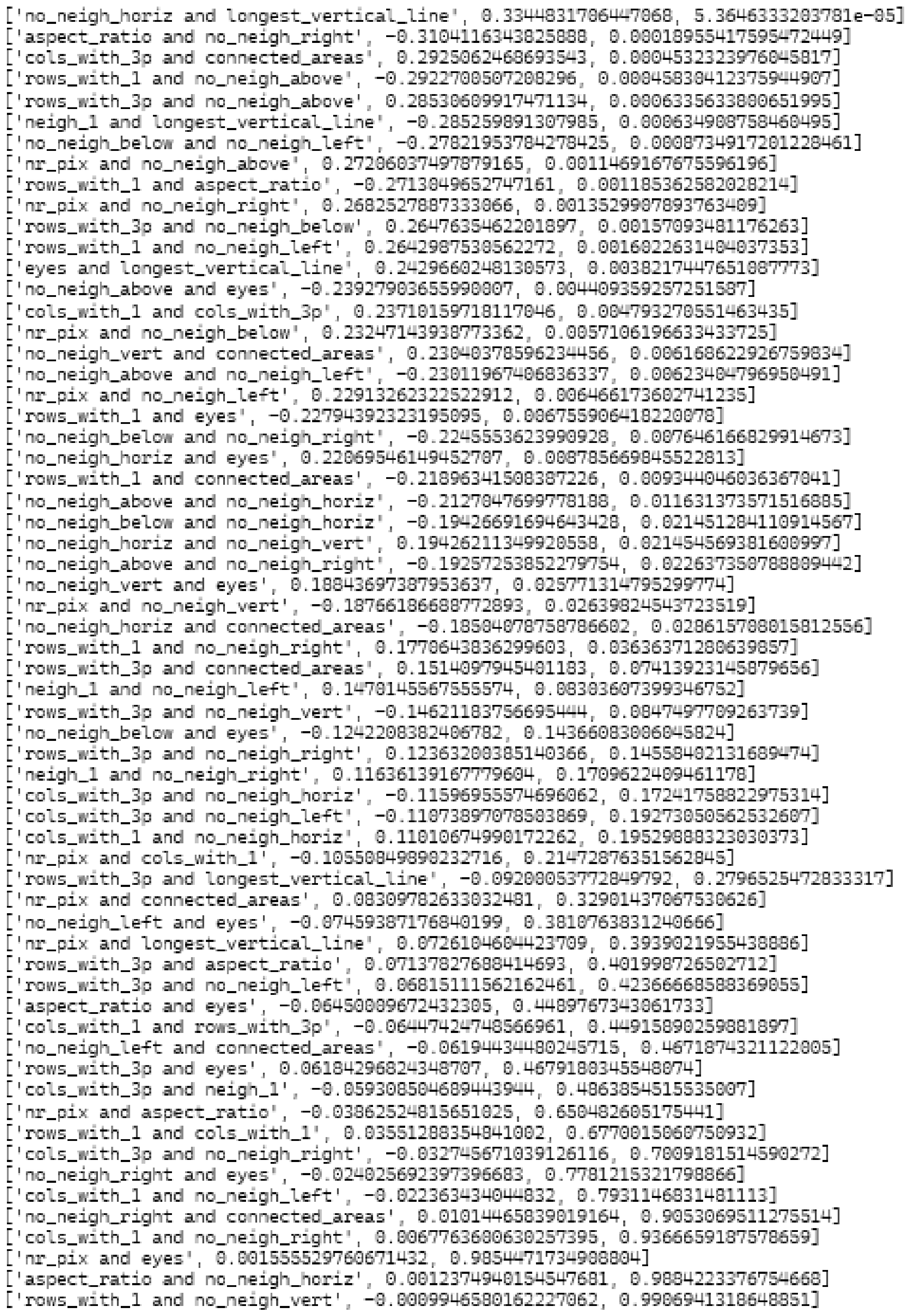
As we can see in the box plot, there's a noticeable difference and no significant outliers. Therefore, this feature will likely be informative about whether a symbol is a letter or not.

3.4

I'm going to assess the correlation for the features we have. However, individually assessing every feature pair and its correlation would extremely time consuming (there are 120 pairs). Instead, I'm going to create a large list of all the combinations in python and assess their correlation and p-values. I'll then order them by their p-values and determine the most significant correlations. We'll keep our significance level at 0.05. Here's the full list:



```
['no_neigh_above and no_neigh_below', 0.9528765213956122, 2.300382162991482e-73]
['no_neigh_left and no_neigh_right', 0.9319938649610671, 1.1803920254320315e-62]
['nr_pix and rows_with_3p', 0.8973665773653973, 7.060913264914954e-51]
['aspect_ratio and no_neigh_below', 0.8055762091174943, 3.578043496461035e-33]
['cols_with_1 and connected_areas', 0.8030044417213931, 8.055198021600981e-33]
['aspect_ratio and no_neigh_above', 0.7610469565509672, 1.0217348967558963e-27]
['aspect_ratio and longest_vertical_line', -0.7575164263220477, 2.458587179699476e-27]
['cols_with_1 and neigh_1', 0.7467727705203193, 3.251834962435439e-26]
['no_neigh_below and longest_vertical_line', -0.7374828040539765, 2.737148247705791e-25
['no_neigh_below and no_neigh_vert', 0.7365428823701372, 3.378494470505807e-25]
['cols_with_1 and aspect_ratio', 0.7351994630812129, 4.557431633781634e-25]
['connected_areas and longest_vertical_line', -0.705701417790125, 2.124275730465136e-22
['no_neigh_above and no_neigh_vert', 0.7034138633649588, 3.3142936619185476e-22]
['aspect_ratio and no_neigh_vert', 0.6974503536478353, 1.0362186534107146e-21]
['no_neigh_above and longest_vertical_line', -0.6929334222798136, 2.4122529976470623e-2
['no_neigh_above and connected_areas', 0.6898917239625598, 4.224401182328711e-21]
['rows_with_1 and rows_with_3p', -0.6807910787315269, 2.169218067987285e-20]
['cols_with_1 and no_neigh_above', 0.6760934548741699, 4.9332755754712525e-20]
['no_neigh_below and connected_areas', 0.6673333851196952, 2.194184795760234e-19]
['aspect_ratio and connected_areas', 0.6508294866100741, 3.196438021195417e-18]
['rows_with_1 and no_neigh_horiz', 0.6341747496347946, 4.054976099700571e-17]
['no_neigh_left and longest_vertical_line', 0.6336141796464918, 4.405161422510696e-17]
['cols_with_3p and no_neigh_below', 0.6303287410646747, 7.133956362512447e-17]
['cols_with_1 and no_neigh_below', 0.6267036286157943, 1.2063441685924018e-16]
['cols_with_3p and no_neigh_above', 0.6115180634916639, 1.0129085932032e-15]
['rows_with_1 and cols_with_3p', -0.6082077792619341, 1.586672056143651e-15]
['nr_pix and rows_with_1', -0.5750746246133893, 1.0783825306372255e-13]
['rows_with_3p and cols_with_3p', 0.5646177235146739, 3.709481200036092e-13]
['no_neigh_right and longest_vertical_line', 0.5642403846687905, 3.8754631394389737e-13
['cols_with_1 and longest_vertical_line', -0.5585776733756254, 7.425849835906954e-13]
['connected_areas and eyes', -0.5570756886111483, 8.805699338238344e-13]
['no_neigh_vert and longest_vertical_line', -0.5408049444848162, 5.2888288716593684e-12
['neigh_1 and connected_areas', 0.5360184298042109, 8.801421029629051e-12]
['cols_with_3p and aspect_ratio', 0.5216444579657105, 3.875387835695771e-11]
['nr_pix and cols_with_3p', 0.5163143976428093, 6.598721732244348e-11]
['rows_with_3p and no_neigh_horiz', -0.5114743496384471, 1.0614843517733467e-10]
['neigh_1 and eyes', -0.5067704233986004, 1.6729370097724393e-10]
['rows_with_1 and neigh_1', 0.4922718218697828, 6.514158439784299e-10]
['no_neigh_left and no_neigh_horiz', 0.48194315803614757, 1.651655563183267e-09]
['no_neigh_right and no_neigh_horiz', 0.48134107052763575, 1.7420546260856185e-09]
['cols_with_1 and no_neigh_vert', 0.46447735682440505, 7.436869830445061e-09]
['cols_with_1 and eyes', -0.45954111824725885, 1.1208043966140643e-08]
['aspect_ratio and neigh_1', 0.4594216794491747, 1.1318927228208644e-08]
['cols_with_3p and no_neigh_vert', 0.45741417908806536, 1.3348151548601563e-08]
['neigh_1 and no_neigh_horiz', 0.4564114169544662, 1.448842981934183e-08]
['nr_pix and no_neigh_horiz', -0.452548183736261, 1.982019361092237e-08]
['neigh_1 and no_neigh_above', 0.4181294949362145, 4.843765725550436e-07]
['rows_with_1 and longest_vertical_line', 0.4087175404688933, 5.348041882525508e-07]
['cols_with_3p and longest_vertical_line', -0.40538034140592125, 6.783876804530548e-07]
['no_neigh_left and no_neigh_vert', -0.401947615036803, 8.544814885977793e-07]
['rows_with_3p and neigh_1', -0.3976879432354693, 1.1414834539210899e-06]
['neigh_1 and no_neigh_vert', 0.38516242619449237, 2.613929561613325e-06]
['no_neigh_right and no_neigh_vert', -0.37086157410701415, 6.462225524291962e-06]
['nr_pix and neigh_1', -0.36165204930598266, 1.1319391682966581e-05]
['cols_with_3p and eyes', 0.34370923980621865, 3.2139058213365645e-05]
['neigh_1 and no_neigh_below', 0.3428761544111197, 3.368326359316529e-05]
['aspect_ratio and no_neigh_left', -0.339707865801163, 4.02159184474216e-05]
['rows_with_1 and no_neigh_below', -0.3381843767731197, 4.3764339851825626e-05]
```



```
['no_neigh_horiz and longest_vertical_line', 0.3344831706447068, 5.3646333283781e-05]
['aspect_ratio and no_neigh_right', -0.3104116343825888, 0.00018955417595472449]
['cols_with_3p and connected_areas', 0.2925062468693543, 0.00045323239760045817]
['rows_with_1 and no_neigh_above', -0.2922700507208296, 0.00045838412375944907]
['rows_with_3p and no_neigh_above', 0.28530609917471134, 0.0006335633880651995]
['neigh_1 and longest_vertical_line', -0.285259891307985, 0.0006349087584680495]
['no_neigh_below and no_neigh_left', -0.27821953784278425, 0.0008734917201228461]
['nr_pix and no_neigh_above', 0.27206037497879165, 0.0011469167675596196]
['rows_with_1 and aspect_ratio', -0.2713849652747161, 0.001185362582028214]
['nr_pix and no_neigh_right', 0.2682527887333866, 0.0013529987893763409]
['rows_with_3p and no_neigh_below', 0.2647635462201897, 0.00157093481176263]
['rows_with_1 and no_neigh_left', 0.2642987530562272, 0.0016022631404037353]
['eyes and longest_vertical_line', 0.2429660248130573, 0.0038217447651087773]
['no_neigh_above and eyes', -0.23927903655990807, 0.004409359257251587]
['cols_with_1 and cols_with_3p', 0.23710159718117046, 0.004793270551463435]
['nr_pix and no_neigh_below', 0.23247143938773362, 0.0057106196633433725]
['no_neigh_vert and connected_areas', 0.23040378596234456, 0.006168622926759834]
['no_neigh_above and no_neigh_left', -0.23011967406836337, 0.00623404796950491]
['nr_pix and no_neigh_left', 0.22913262322522912, 0.006466173602741235]
['rows_with_1 and eyes', -0.22794392323195095, 0.006755906418220078]
['no_neigh_below and no_neigh_right', -0.2245553623990928, 0.007646166829914673]
['no_neigh_horiz and eyes', 0.22069546149452707, 0.008785669845522813]
['rows_with_1 and connected_areas', -0.21896341508387226, 0.009344046036367041]
['no_neigh_above and no_neigh_horiz', -0.2127847699778188, 0.011631373571516885]
['no_neigh_below and no_neigh_horiz', -0.19426691694643428, 0.021451284110914567]
['no_neigh_horiz and no_neigh_vert', 0.19426211349920558, 0.021454569381600997]
['no_neigh_above and no_neigh_right', -0.19257253852279754, 0.022637350788809442]
['no_neigh_vert and eyes', 0.18843697387953637, 0.025771314795299774]
['nr_pix and no_neigh_vert', -0.18766186688772893, 0.02639824543723519]
['no_neigh_horiz and connected_areas', -0.18504078758786602, 0.028615788015812556]
['rows_with_1 and no_neigh_right', 0.1778643836299603, 0.03636371280639857]
['rows_with_3p and connected_areas', 0.1514097945401183, 0.07413923145879656]
['neigh_1 and no_neigh_left', 0.1470145567555574, 0.08303607399346752]
['rows_with_3p and no_neigh_vert', -0.14621183756695444, 0.0847497709263739]
['no_neigh_below and eyes', -0.1242208382406782, 0.14366083006045824]
['rows_with_3p and no_neigh_right', 0.12363208385140366, 0.14558402131689474]
['neigh_1 and no_neigh_right', 0.11636139167779604, 0.1709622409461178]
['cols_with_3p and no_neigh_horiz', -0.11596955574696062, 0.17241758822975314]
['cols_with_3p and no_neigh_left', -0.11073897078583869, 0.19273050562532607]
['cols_with_1 and no_neigh_horiz', 0.11010674990172262, 0.19529888323030373]
['nr_pix and cols_with_1', -0.10550849890232716, 0.21472876351562845]
['rows_with_3p and longest_vertical_line', -0.09200053772849792, 0.2796525472833317]
['nr_pix and connected_areas', 0.08309782633032481, 0.32901437067530626]
['no_neigh_left and eyes', -0.07459387176840199, 0.3810763831240666]
['nr_pix and longest_vertical_line', 0.0726104604423709, 0.3939021955438886]
['rows_with_3p and aspect_ratio', 0.07137827688414693, 0.401998726502712]
['rows_with_3p and no_neigh_left', 0.06815111562162461, 0.42366668588369055]
['aspect_ratio and eyes', -0.06450009672432305, 0.4489767343061733]
['cols_with_1 and rows_with_3p', -0.06447424748566961, 0.44915890259881897]
['no_neigh_left and connected_areas', -0.06194434480245715, 0.4671874321122805]
['rows_with_3p and eyes', 0.06184296824348707, 0.4679188345548074]
['cols_with_3p and neigh_1', -0.059308504689443944, 0.4863854515535007]
['nr_pix and aspect_ratio', -0.03862524815651825, 0.650482605175441]
['rows_with_1 and cols_with_1', 0.03551288354841002, 0.6770015060750932]
['cols_with_3p and no_neigh_right', -0.032745671039126116, 0.7009181514590272]
['no_neigh_right and eyes', -0.024025692397396683, 0.7781215321798866]
['cols_with_1 and no_neigh_left', -0.022363434044832, 0.7931146831481113]
['no_neigh_right and connected_areas', 0.01014465839019164, 0.9053069511275514]
['cols_with_1 and no_neigh_right', 0.0067763680630257395, 0.9366659187578659]
['nr_pix and eyes', 0.001555529760671432, 0.9854471734908804]
['aspect_ratio and no_neigh_horiz', 0.0012374940154547681, 0.9884223376754668]
['rows_with_1 and no_neigh_vert', -0.0009946580162227062, 0.9906941318648851]
```

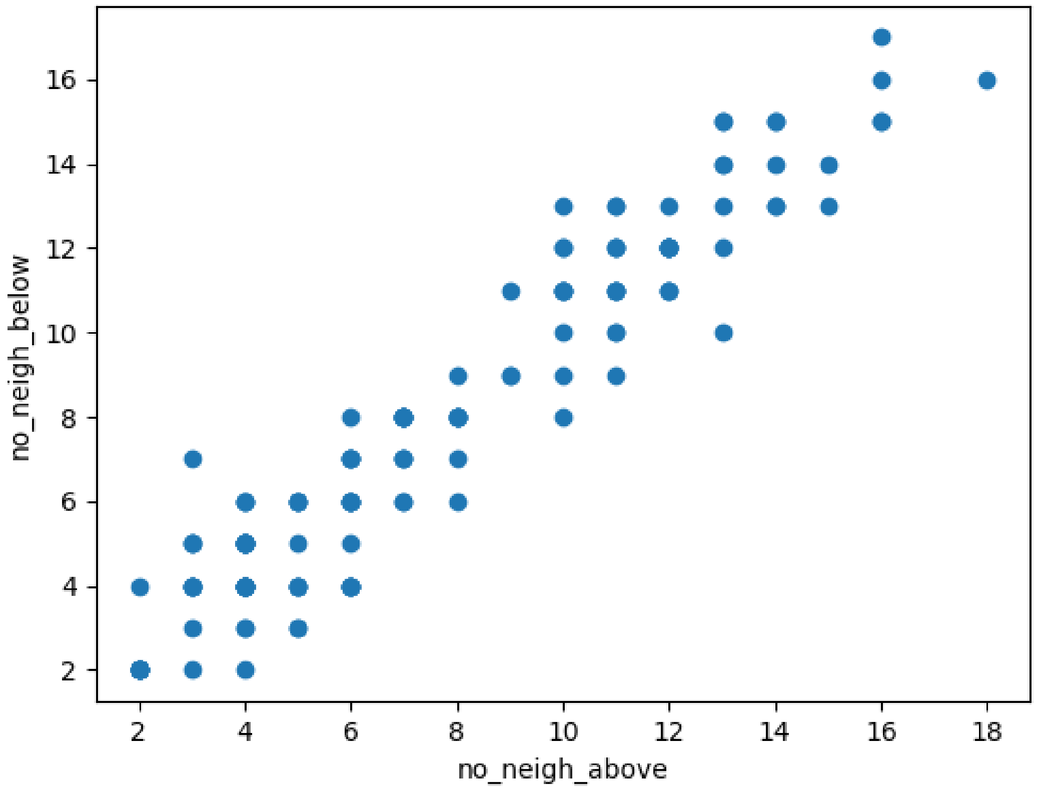
While many of the pairs can be ignored as they are above our significance level (like 'rows_with_1' and 'no_neigh_vert'), the majority of the pairs are actually below our significance level. Many pairs are clearly linearly associated.

The most significant linear associations appear to be (in this order):

1. No neighbours above and no neighbours below
2. No neighbours left and no neighbours right
3. Number of pixels and rows with three pixels

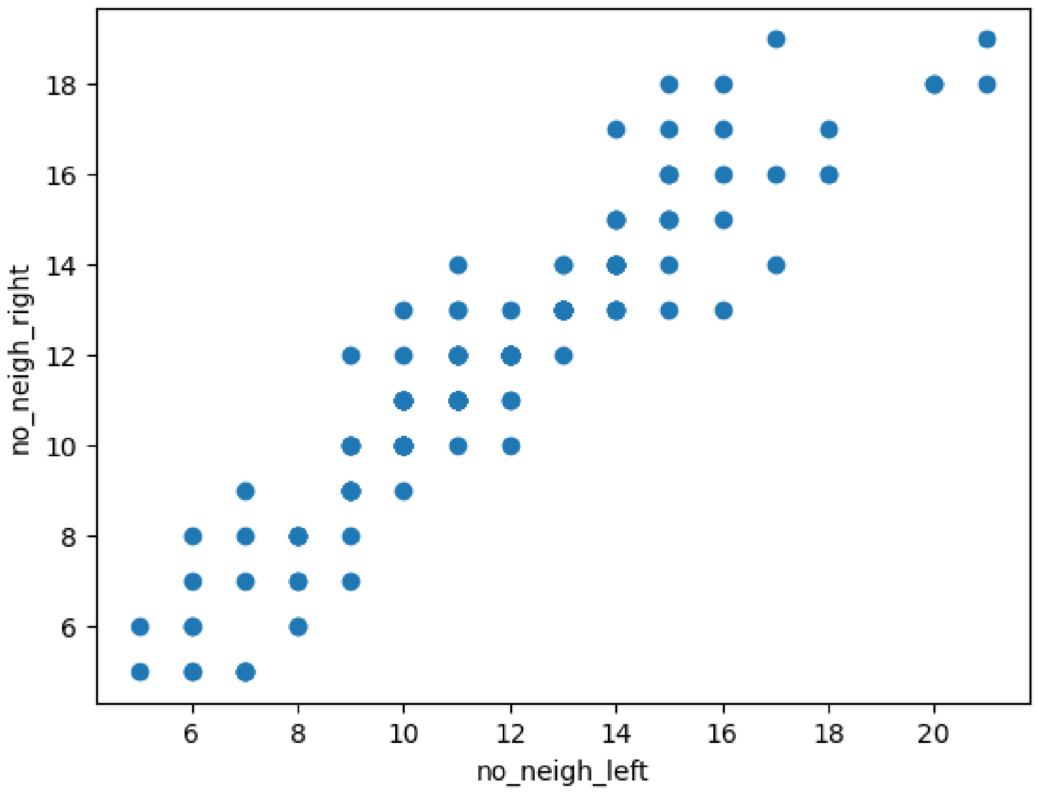
I'll visualize and discuss these below, using pyplot to create scatterplots.

No neighbours above and no neighbours below



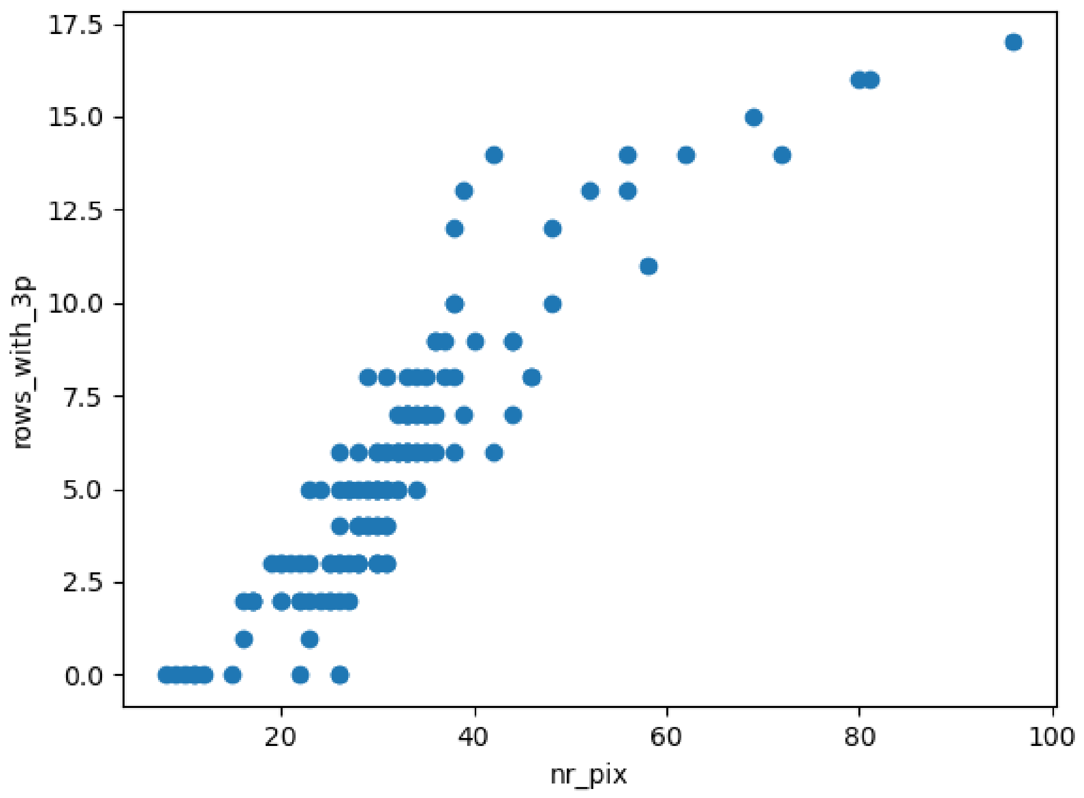
Here, we can see a strong positive correlation between the number of pixels that don't have a neighbour below, and the amount of pixels that don't have a neighbour above them. I suspect that association is because generally, pixels that don't have an upper neighbour don't have a lower neighbour either. This can be seen in letters like ''f' or 'c' where parts of those letters extend outwards, without upper or lower neighbours.

No neighbours left and no neighbours right



Again, we a strong positive linear association between the number of pixels that don't have a neighbour on their right, and the number of pixels that don't have a neighbour on their left. I believe this correlation because letters like 'i' and 'j', or symbols like '!', have long, tall parts that don't have any neighbours or the left on the right. As such, symbols with a large number of pixels that don't have any neighbours on the left are likely to have a large number of pixels with no neighbours on the right.

Number of pixels and rows with three pixels



Interesting, we can see a strong linear association at the start, that grows weaker as the values grow. However, we can conclude that the values are correlated.

I suspect that this correlation is because, as the number of pixels grow, it's more likely that any given row will have 3 or more pixels, as there are more pixels in total. This would explain why the features have quite a high association.

References:

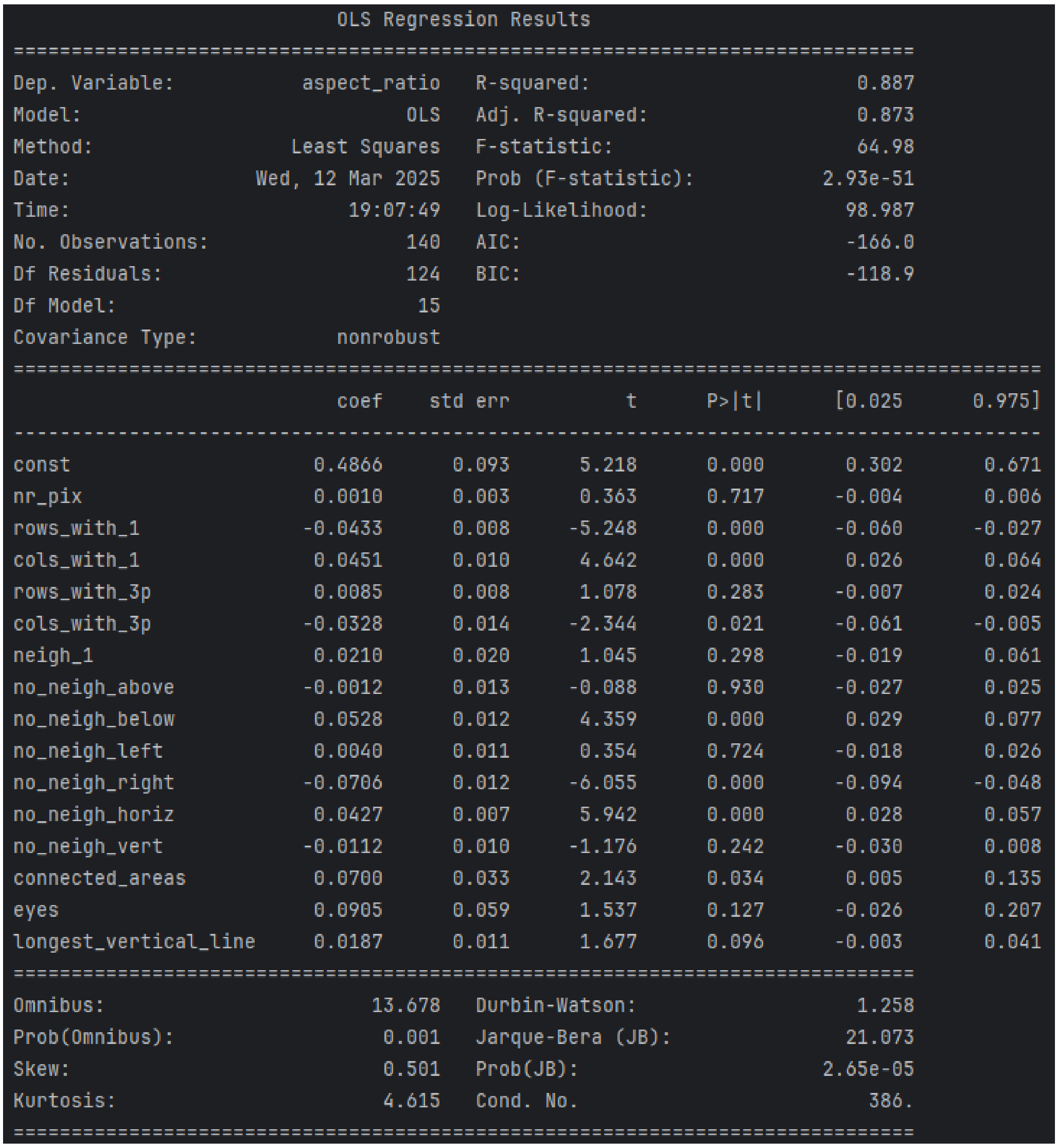
Libraries used – csv, scipy, matplotlib, pandas, statistics

Section 4

4.1:

I'm going to use multiple regression to try to predict the 'aspect_ratio'.

To do this, I'm going to use backwards elimination and an adjusted R^2 approach. I'll start with the full model. Then, I'll test remove the values one by one to get the model with the highest adjusted R^2 and then repeat the process with that model.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:           aspect_ratio   R-squared:                       0.887
Model:                            OLS   Adj. R-squared:                  0.873
Method:                 Least Squares   F-statistic:                     64.98
Date:                Wed, 12 Mar 2025   Prob (F-statistic):           2.93e-51
Time:                        19:07:49   Log-Likelihood:                 98.987
No. Observations:                 140   AIC:                            -166.0
Df Residuals:                     124   BIC:                            -118.9
Df Model:                          15
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                    0.4866      0.093      5.218      0.000       0.302       0.671
nr_pix                   0.0010      0.003      0.363      0.717      -0.004       0.006
rows_with_1             -0.0433      0.008     -5.248      0.000      -0.060      -0.027
cols_with_1              0.0451      0.010      4.642      0.000       0.026       0.064
rows_with_3p             0.0085      0.008      1.078      0.283      -0.007       0.024
cols_with_3p            -0.0328      0.014     -2.344      0.021      -0.061      -0.005
neigh_1                  0.0210      0.020      1.045      0.298      -0.019       0.061
no_neigh_above          -0.0012      0.013     -0.088      0.930      -0.027       0.025
no_neigh_below           0.0528      0.012      4.359      0.000       0.029       0.077
no_neigh_left            0.0040      0.011      0.354      0.724      -0.018       0.026
no_neigh_right          -0.0706      0.012     -6.055      0.000      -0.094      -0.048
no_neigh_horiz           0.0427      0.007      5.942      0.000       0.028       0.057
no_neigh_vert           -0.0112      0.010     -1.176      0.242      -0.030       0.008
connected_areas          0.0700      0.033      2.143      0.034       0.005       0.135
eyes                     0.0905      0.059      1.537      0.127      -0.026       0.207
longest_vertical_line    0.0187      0.011      1.677      0.096      -0.003       0.041
==============================================================================
Omnibus:                       13.678   Durbin-Watson:                   1.258
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.073
Skew:                           0.501   Prob(JB):                     2.65e-05
Kurtosis:                       4.615   Cond. No.                         386.
==============================================================================
```

As you can see, we have an adjusted R^2 of 0.873, which is good but can be improved. First, we can see that rows_with_1, cols_with_1, cols_with_3p, no_neigh_below, no_neigh_right, no_neigh_horiz, connected_areas and longest_vertical_line all have p_values below our significance level (0.05), meaning they are likely to be significant.

Meanwhile, nr_pix, rows_with_3p, neigh_1, no_neigh_above, no_neigh_left, no_neigh_vert and eyes all have p values that are above our α, which means they are likely to be insignificant. However, we are taking an adjusted R^2 approach, not a p-value approach, so I mention the p-values just for completeness (Removing all the features with p-values above our significance level here actually results in a model with a lower adjusted R^2).

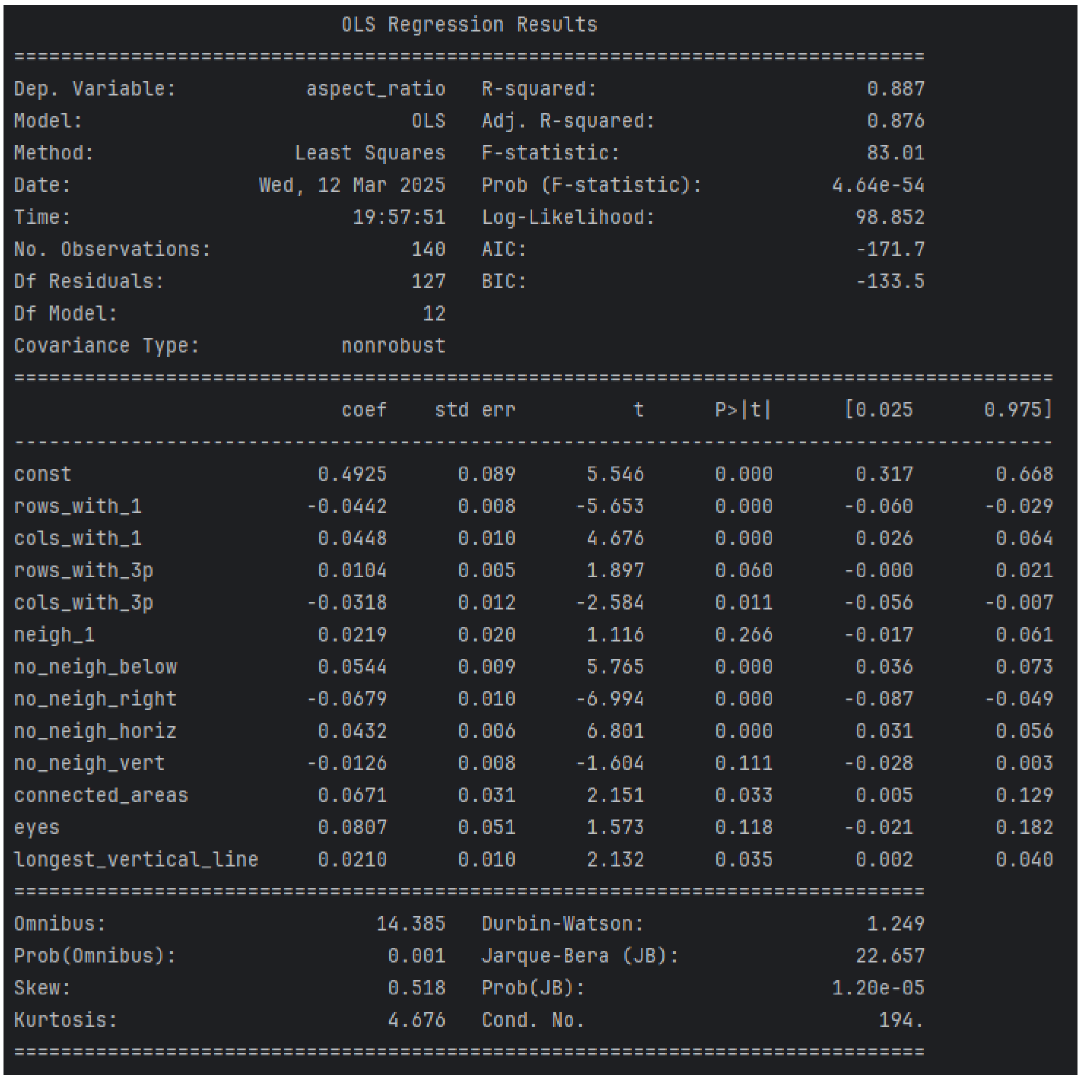
Going one by one through the features, removing no_neigh_above causes the greatest jump, increasing the significance level to 0.874. Therefore, we'll keep no_neigh_above removed and repeat the process.

This time, removing nr_pix increased the adjusted R^2 to 0.875. So, we'll keep nr_pix removed and repeat again.

Now, removing no_neigh_left increased our adjusted R^2 to 0.876. Once again, we'll keep it removed and continue the process.

After measuring the adjusted R^2 of all the smaller models, none of them resulted in an increase.

Therefore, I fitted my regression model parsimoniously and eliminated as much collinearity as I can. Here are the final results from the improved multiple regression model:



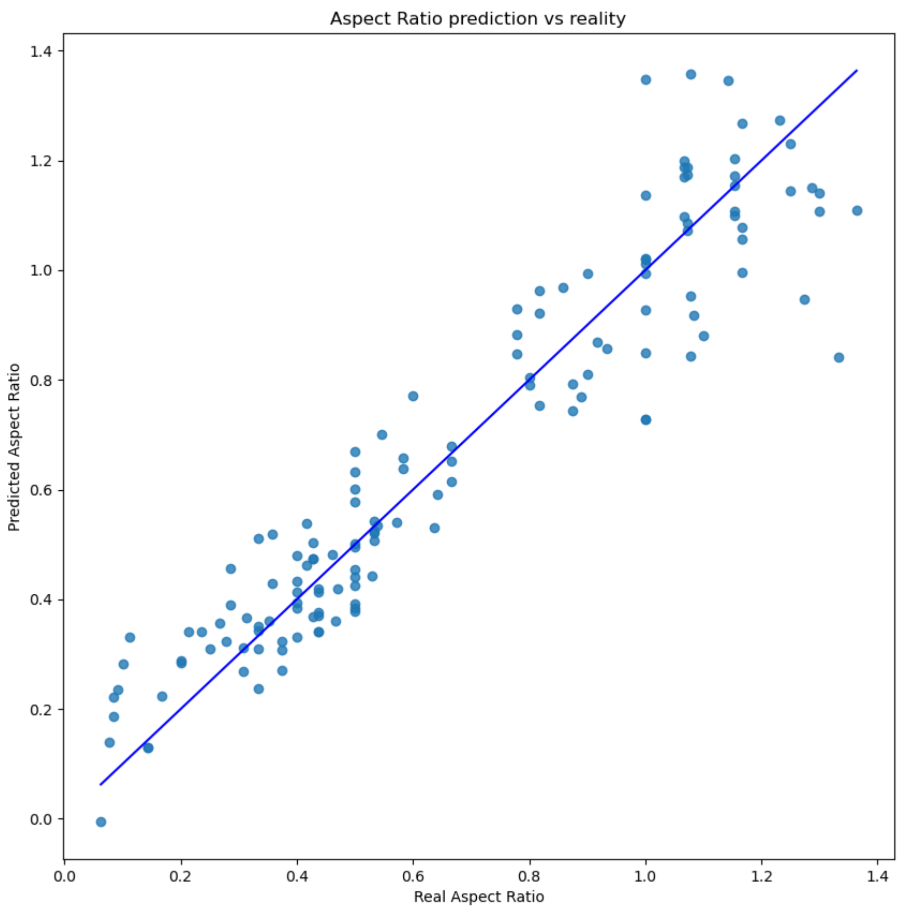
```
                            OLS Regression Results
==============================================================================
Dep. Variable:           aspect_ratio   R-squared:                       0.887
Model:                            OLS   Adj. R-squared:                  0.876
Method:                 Least Squares   F-statistic:                     83.01
Date:                Wed, 12 Mar 2025   Prob (F-statistic):           4.64e-54
Time:                        19:57:51   Log-Likelihood:                 98.852
No. Observations:                 140   AIC:                            -171.7
Df Residuals:                     127   BIC:                            -133.5
Df Model:                          12
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                     0.4925      0.089      5.546      0.000       0.317       0.668
rows_with_1              -0.0442      0.008     -5.653      0.000      -0.060      -0.029
cols_with_1               0.0448      0.010      4.676      0.000       0.026       0.064
rows_with_3p              0.0104      0.005      1.897      0.060      -0.000       0.021
cols_with_3p             -0.0318      0.012     -2.584      0.011      -0.056      -0.007
neigh_1                   0.0219      0.020      1.116      0.266      -0.017       0.061
no_neigh_below            0.0544      0.009      5.765      0.000       0.036       0.073
no_neigh_right           -0.0679      0.010     -6.994      0.000      -0.087      -0.049
no_neigh_horiz            0.0432      0.006      6.801      0.000       0.031       0.056
no_neigh_vert            -0.0126      0.008     -1.604      0.111      -0.028       0.003
connected_areas           0.0671      0.031      2.151      0.033       0.005       0.129
eyes                      0.0807      0.051      1.573      0.118      -0.021       0.182
longest_vertical_line     0.0210      0.010      2.132      0.035       0.002       0.040
==============================================================================
Omnibus:                       14.385   Durbin-Watson:                   1.249
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               22.657
Skew:                           0.518   Prob(JB):                     1.20e-05
Kurtosis:                       4.676   Cond. No.                         194.
==============================================================================
```

Looking at the results, it seems that symbols with a larger number of rows with 1 pixel, columns with 3 pixels or no right or vertical neighbours have a lesser aspect ratio.

By contrast, images with more columns with 1 pixel, rows with 3 pixels, pixels with no neighbours below them, 1 neighbour or no horizontal neighbours, long vertical lines, eyes or connected areas tend to have a greater aspect ratio.

Some of these intuitively make sense. An image with a significant number of rows with 1 means that the aspect ratio is going to be lower, as the image is clearly spread out more vertically (otherwise there would be more pixels on one row).

Others make less sense intuitive, like a pixel have 1 neighbour resulting in a larger aspect ratio. The p-value for that statistic is fairly large, but removing it didn't result in an increase of adjusted R^2, so it may have some predictive power.



Looking at the graph, we can see that the predicted aspect generally follows the actual aspect ratio, although it grows more inaccurate as the actual aspect ratio increases.

4.2

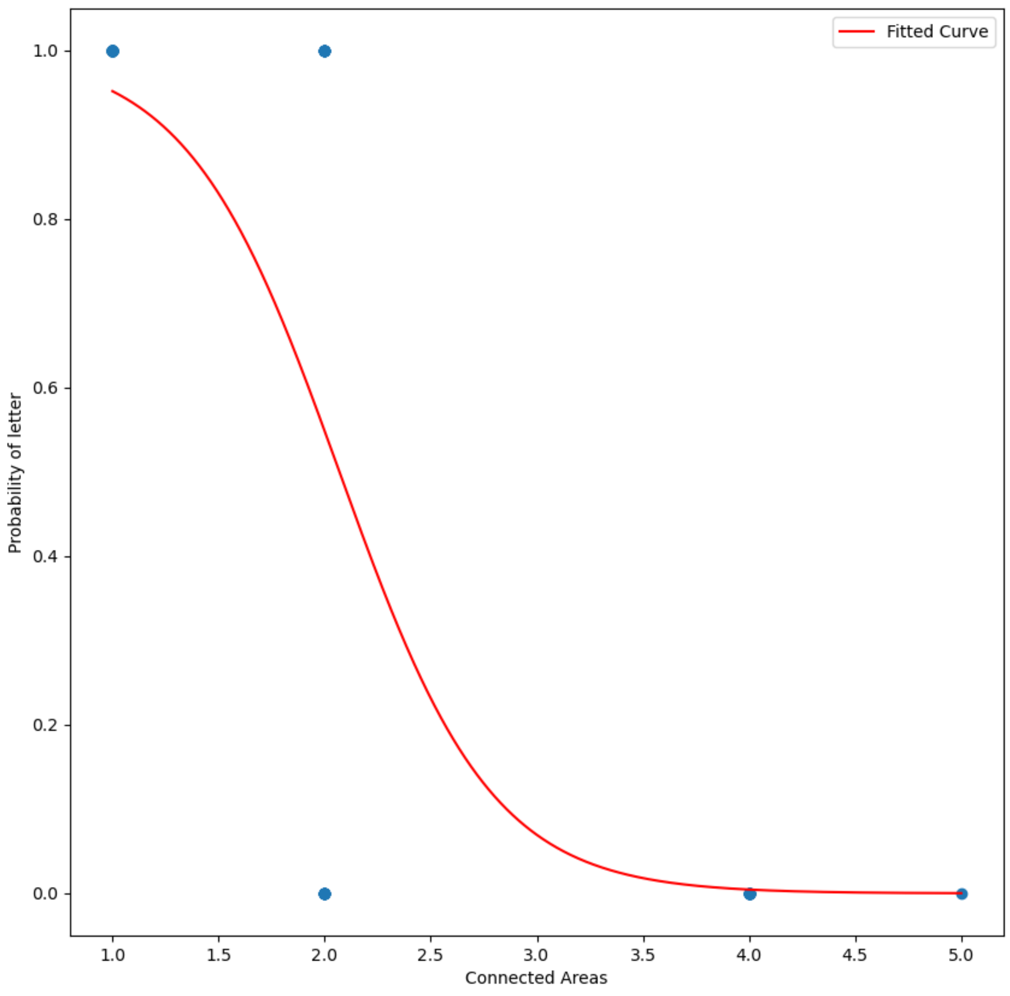
I’m now going to create a logistic regression model. First, I need to pick a feature for discrimination, based on our results in 3.3. Connected areas seems like the obvious choice. While connected areas isn’t normally distributed, logistic regression doesn’t really require normality, so it’s still a strong choice. It has a very high T statistic at -16.139587754399763 and the lowest P value of 5.447576969423113e-26.

I created a function to change label into one or zero, depending on whether the image was a letter or not. Then I turned the CSV into a dataframe, and applied the label to the dataframe. After that, I shuffled the dataframe and split it into a training (80%) and test (20%) set. Then I trained the logistic regression model, printed the regression curve in matplotlib (for visualization), and tested the accuracy on the training set and then the testing set.

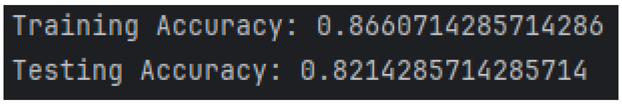
Here's the logistic equation:

$$P(x) = \frac{e^{-2.78962881+5.7744929}}{1 + e^{-2.78962881+5.7744929}}$$

Here is the regression curve:



And here is the testing and testing accuracy:

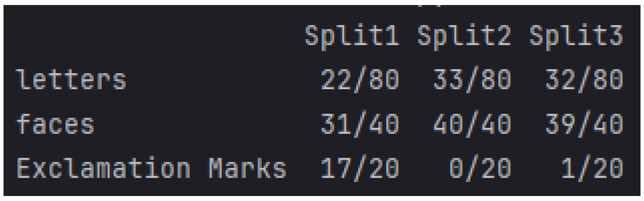


```
Training Accuracy: 0.8660714285714286
Testing Accuracy: 0.8214285714285714
```

The accuracy remains high (over 80%) in testing, indicating that our model is not overfitted to the data.

4.3

I used python code to calculate the median for each feature, and then created a dataframe to store the proportion of images above or equal to the median for that feature. I looped through each feature, calculated the median for that feature, evaluated how many letters, faces or exclamation marks were equal to or above it, and then appended them to the dataframe.



```
                   Split1 Split2 Split3
letters             22/80  33/80  32/80
faces               31/40  40/40  39/40
Exclamation Marks   17/20   0/20   1/20
```

References:

Logistic regression model based on - https://canvas.qub.ac.uk/courses/30078/files/5853905/download?download_frd=1

Libraries – pandas, statsmodel, numpy, sklearn, matplotlib, statistics

Conclusions

In section 1, I successfully converted the PGM files to CSV files. In section 2, I converted those csv files to a list of features accurately, extracting importing information from the dataset. In section 3, I analysed different features of the dataset, discovering that the most important features of the dataset for discrimination were connected_areas and no_neigh_above, and uncovered significant correlation between several features like no_neigh_left and no_neigh_right. Finally, in section 4, I fitted a multiple regression model and logistic regression model based on the data I had found.

Overall, I conducted successful exploratory data analysis, finding out significant amount of information about the dataset.