**L&S 39F HW3 Stage II: Analysis and Final Write-up**

**Team: Ameena Golding, Max McArthur, Vasilis Oikonomou**

Summary

**Hypothesis and Predictions**

The hypothesis we worked on was that color naming patterns of different languages depend on the distribution of colors in the immediate environment where the language was developed. Our null hypothesis is that the distribution of colors in the immediate environment where the language was developed would be the same as the distribution of colors through the partition of the munsell space. For this reason, we predicted that the total variation distance between the two distributions would be quite small and we also predicted that if we were right in our prediction, we should eventually get a large p-value, of the order of 0.5. We also predicted that black, white and red would be underrepresented in the sampled images for reasons explained in the document we submitted for Stage 1 of this project.

**Summary of the main findings**

For the four languages we were able to examine, both our qualitative and quantitative evaluation of the data showed us that the two examined distributions were highly unlikely to be the same. However, we observed a remarkable consistency in the observed total variation distances between the distribution of colors in the images and the one in the munsell space which was always between 0.2 and 0.3 for the all the languages we examined. Also, black, white and red were in fact underrepresented but not to the extent that we would expect them to be.

Analyses

The main task of the project was to try to associate certain image pixels with the respective colors in the language in examination; we needed to find a way to effectively identify which pixels correspond to which colors in the minds of the speakers. To achieve this, for each color in that language, we took the average in RGB values of the foci that each chip identified as a really good example of that color by the speakers of that language. This was an effective approach, since if certain points were more frequently selected than others, that would have a greater influence on the final value of our focus point. As a result, we minimized any error that may have occurred by individual speaker bias, namely speakers selecting colors as best examples that weren’t considered as such by their peers.[[1]](#footnote-0)

Another task that was critical to our exploration was image sampling. We chose 10 images to work with for each language using Google Maps’s application *Panoramio* where people from all over the world upload images of that particular place.[[2]](#footnote-1) This allowed us to include a variety of images which were often quite different to one another, thereby capturing more colors and more instances of everyday life, something which was really important to perform an unbiased data analysis. For the same reason, we also included images taken during night-time to compensate for the absence of black in images taken during day time. One issue that we encountered in the process regarding image sampling was that due to the high analysis of the images, the code took too long to run. To overcome this obstacle, we decided to perform random sampling with replacement. We sampled 500 pixels from each image to achieve a good running time. It is worth mentioning that due to the size of the images (thousands of pixels), it didn’t really matter if we sample with or without replacement since the sample size is large.[[3]](#footnote-2)

The next step in that process was to create a table of proportions both for the sampled images and the munsell space. For the image distribution, we created a dictionary in which we included all color terms and how many times they appeared in the pixels we sampled. After that, we converted this to a table of proportions which was more useful and easier to work with.[[4]](#footnote-3) For the munsell space distribution, we used data from the WCS and more specifically from the term.txt file and we followed a similar process to convert them to proportions.

Our last key step in the analysis was the hypothesis testing. We used total variation distance as our test statistic. First we calculated the observed tvd between the distribution of images in the environment and the distribution in the munsell space.[[5]](#footnote-4) Then, we used NumPy’s multinomial method to sample from the table with the proportions of the image distribution and we took the tvd between that and the original table to see how different the data could have turned out to be due to sampling. We repeated that process 1000 times and we produced a distribution for our test statistic. The range of the tvd was small as we expected and so we hoped that our observed tvd would be comparable to that.[[6]](#footnote-5)

Results

**Observations:**

The observed results are not the ones we expected initially when we were writing our hypothesis. To begin with, p value we get from our hypothesis testing is 0 for all the languages we tested, which indicates that the data were highly unlikely to have been produced under the null hypothesis. We can reach the same conclusion by simply looking at the two bar charts for each language, which were produced by sorting the terms alphabetically so that they are more easily comparable. Comparing the two distributions for each language provides an obvious visual indication that the distribution of colors in the image are not at all similar to the distribution of colors in the Munsell space (see Appendix for the visuals).

One observation that pertains more to the data contained within the WCS documentation itself is that, in many instances in which a speaker was questioned about a where a certain color name was situated in the Munsell space, the experimenter did not ask the speaker to assign a foci to that color. Basically, it seems like that experimenter randomly decided not to ask certain speakers about the foci for some colors. This affected our experimentation in the sense that we could not use certain languages in our experimentation, as doing so would mean that we did not have all the foci for that language with which to compare the pixels to.

Another very interesting observation is that the observed total variation distance between the partitioning of colors in the Munsell space and the distribution of colors in the images is in the fairly narrow range of 0.2 to 0.3 for *all* four languages we analyzed. A final observation is that, contrary to our prediction, black was the most popular color across all languages.

**Explanations:**

Clearly, the results we got from our hypothesis testing did not match our initial predictions and there are many factors that could have contributed to this irregularity. The most immediate potential factor is bias in the initial image collection, and there are many aspects to this. First is the fact that since most of the available images online are taken during the day, very dark colors such as black rarely appear in any of the images. This inherently skews the data towards the lighter colors. Another aspect of the bias in the image collection is the fact that virtually all of the available images in each country we surveyed are photographs of nature scenes and provincial life (rivers, trees, small dusty village streets, blue skies, plains, etc.) Although we tried very hard to compose a diverse set of images for each language, it was very difficult to find images containing those bright colors not so easily seen in nature: bright purples, pinks, and reds. This means that the data we sampled from the images are bias towards greens, blues, and dull greyish-beige colors. In summary, it is difficult to exactly quantify the bias in image collection.

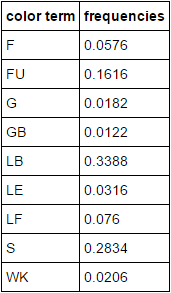
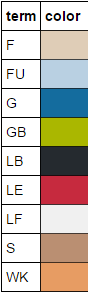
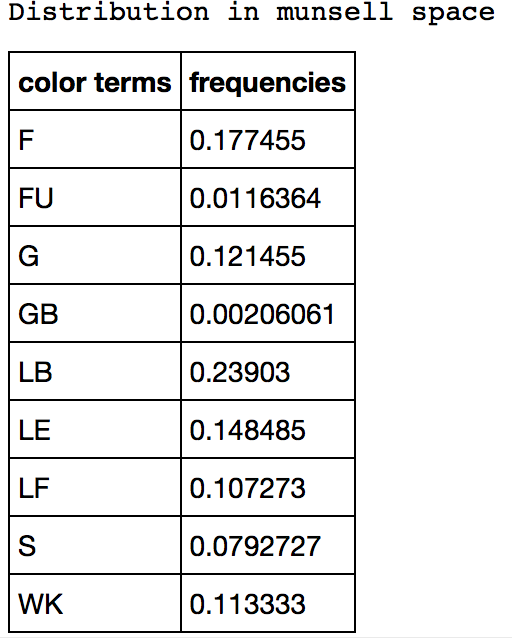
As mentioned above in the “Observed” portion, black was the most popular color across all the languages, which was contrary to what we anticipated before carrying out the analysis, as we reasoned that black was not a common color found in nature. One very primitive explanation for the huge amounts of black compared to the to other colors is that shadows within the images will appear as black. A more likely explanation however is that, for all the languages we analyzed, many of them categorize colors other than black as being black, as those languages only have a limited set of color terms. In other words, for all the languages we sampled, the term “black” encompasses more colors than just pure black (#000000). For example, the Abidji language spoken in the Ivory Coast (the first language we analyzed), only has three colors terms: white, red, and black. So, when sampling from an image and assigning each pixel to its closest foci value for Abigdji, any color other than what they consider to be white or red will be assigned a value of “black.” All the languages we analyzed follow a similar pattern.

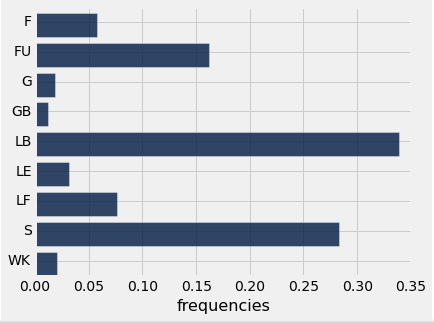
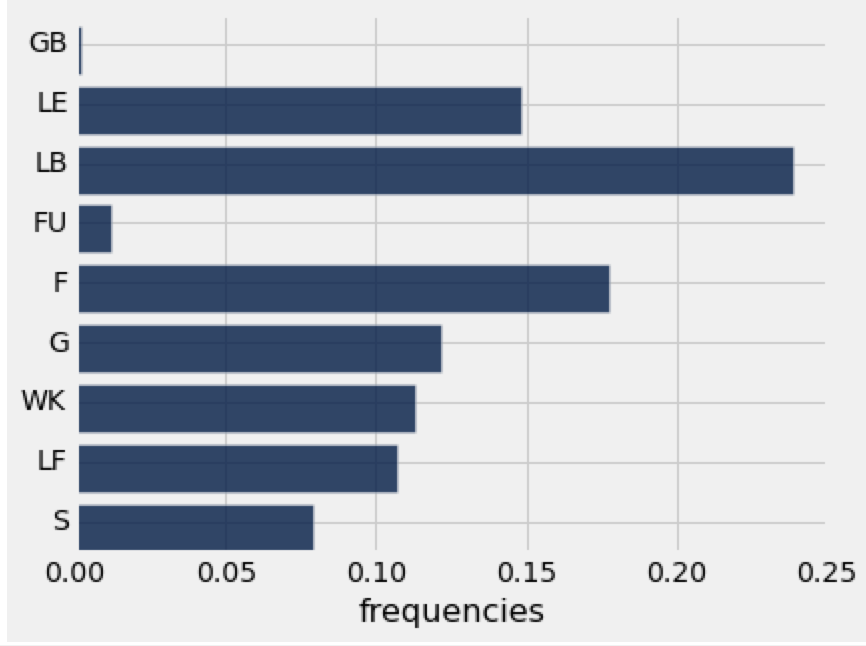
Another potential reason for the lack of any significant results is a lack of resampling of the data. For each image, instead of sampling all of the pixels (as this would have taken hours — we found this out the hard way), we sampled 500 pixels at random (this process is described in more detail in the “Analyses” portion of this text). It is very likely that the sample could have turned out differently than that one sample we took. In other words, it is very possible that the random sample we took is not necessarily highly indicative of the “population” (all the pixels in the image) as a whole. So, ideally, we would have resampled pixels from the image at random with replacement several times and obtained an “average distribution of colors” for each image. But such a process would have been time-consuming.

Finally, one point to consider is that the colors and environments we see now in the modern day is not necessarily what people saw back when these languages were being developed. A particularly relevant example of this is of the Shipibo language (#86 in the WCS) spoken along the Ucayali River in the Amazon rainforest in Perú. According to Wikipedia^, the Shipibo people lived for millennia in this same area of South America. So, clearly the environment in which the language was developed all those years ago is not at all like the urbanized area that this region has become today, of which we took our images from.

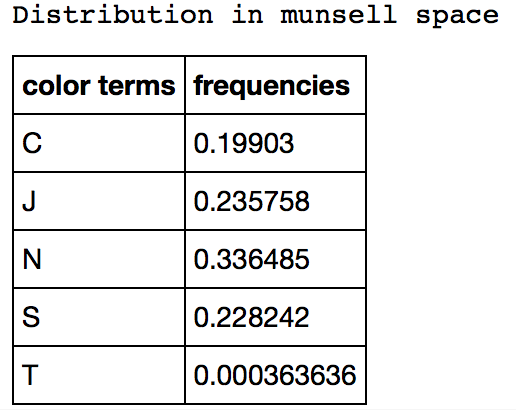
Appendix:

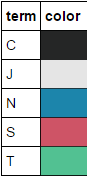
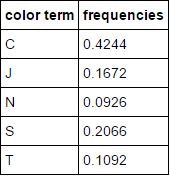
**Language 1: Abidji - Ivory Coast**

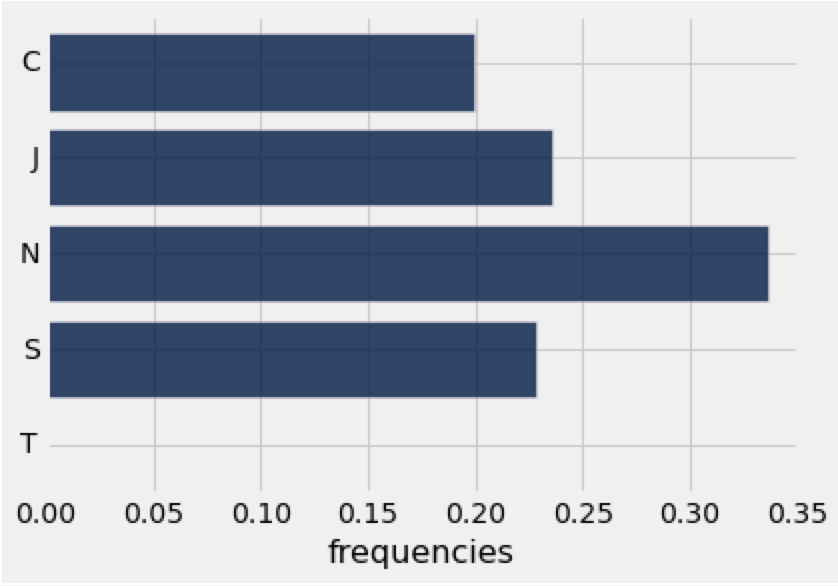


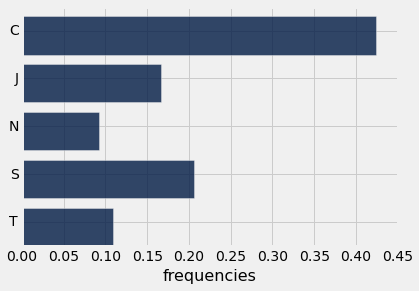


**Language 2: Chacobo - Bolivia**



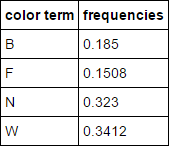
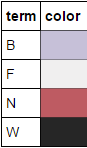
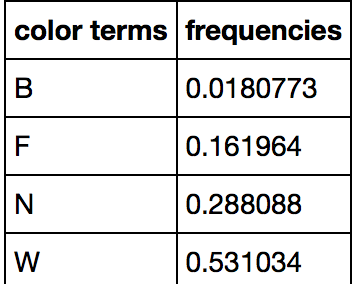


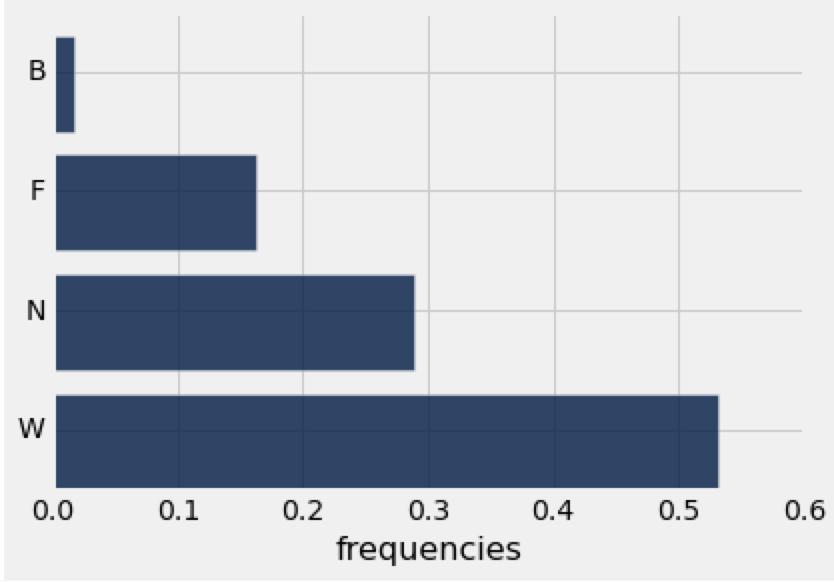
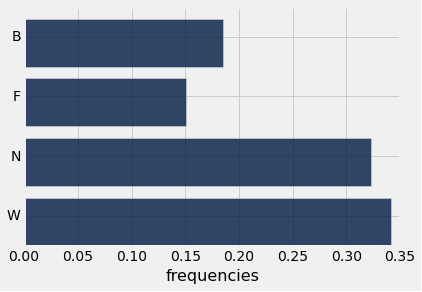




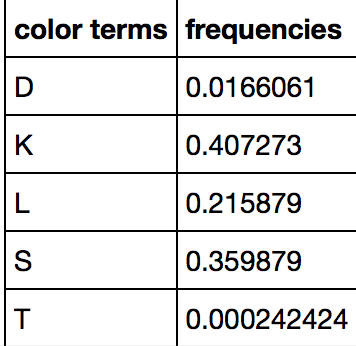
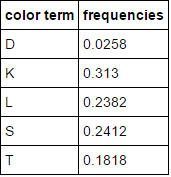
**Language 3: Nafaanra - Ghana**

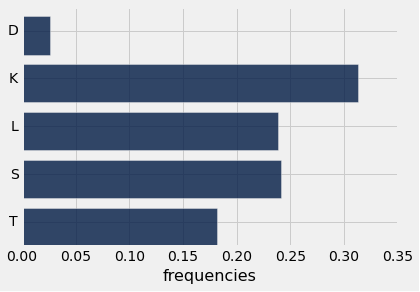
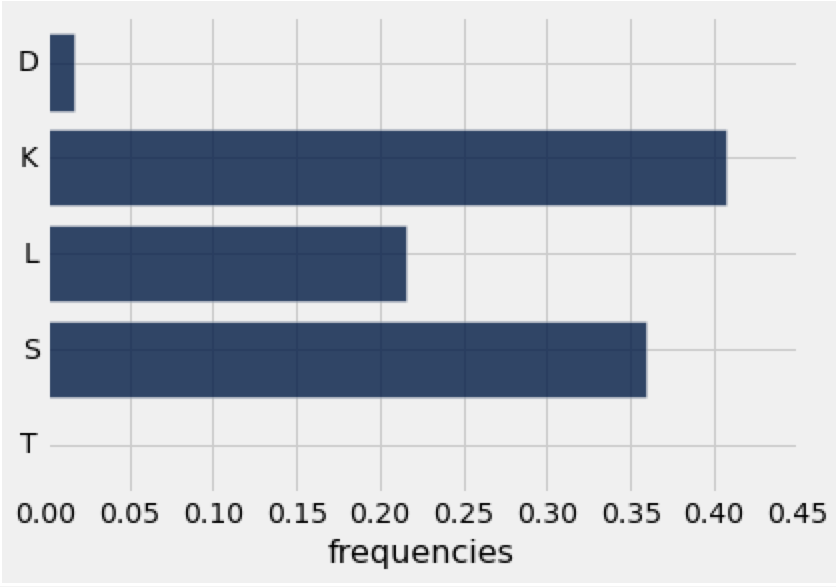






**Language 4: Wobe - Ivory Coast**





# DATA COLLECTION AND ANALYSIS

# imports and reading data

import scipy.misc

import numpy as np

import glob

txt = Table.read\_table("foci-exp.txt")

def find\_min\_distance(col, c\_set):

"""

Finds the minimum RGB 'distance' between the color passed in

and the color set given to the system.

"""

d = {}

for color in c\_set:

dist = ((col[0]-color[0])\*\*2 + (col[1]-color[1])\*\*2 + (col[2]-color[2])\*\*2)\*\*0.5

d[dist] = color

return d[min(d)]

def munsell2rgb(munsell):

"""

Converts a chip value into an RGB value.

Input as a pair [letter, number]

>>> munsell2rgb([A, 0])

[239, 239, 239]

"""

munsell\_path = 'munsell.png'

assert len(munsell) == 2

assert ord(munsell[0])>= 65 and ord(munsell[0])<= 74

assert munsell[1] <= 40

munsell\_image = scipy.misc.imread(munsell\_path)

return munsell\_image[(ord(munsell[0]) - 64) - 1][munsell[1]]

def avg\_rgb(ls):

"""

Returns the average color value by averaging R,

G and B values for colors passed in on a list.

"""

red\_tot, grn\_tot, blu\_tot, count = 0, 0, 0, 0

for col in ls:

red\_tot += col[0]

grn\_tot += col[1]

blu\_tot += col[2]

count += 1

if count==0:

return [0, 0, 0]

return [red\_tot/count, grn\_tot/count, blu\_tot/count]

def to\_chip(str):

"""

Converts a string chip value into a pair that

can be used in calculations later in the program.

"""

assert len(str) == 2 or len(str) == 3

return [str[0], int(str[1:])]

def analyze\_image(foci, image\_path):

"""

Takes in an image, spits out frequencies

of colors near foci in the list foci.

Note: this method was used to analyze the

full image (i.e. all pixels); this method

was abandoned when it was discovered how

much processing power it took to analyze a

full folder of images.

"""

image = scipy.misc.imread(image\_path)

# get counts array

arr\_dist = []

for i in range(len(image)):

for j in range(len(image[i])):

arr\_dist.append(find\_min\_distance(image[i][j], c\_set=list(foci.values())))

# make into dict of color, count

counts\_dict = {}

for col in foci.keys():

counts\_dict[col] = arr\_dist.count(foci[col]) / (len(image) \* len(image[0]))

return counts\_dict

def analyze\_image\_sample(foci, image\_path, n=500):

"""

Takes in an image, takes a random sample

of size n pixels and spits out proportional

frequencies of colors near foci on the list

foci. (appendix reference 2)

"""

image = scipy.misc.imread(image\_path)

arr\_dist = []

for i in range(n):

px = image[np.random.randint(0, len(image))][np.random.randint(0, len(image[0]))]

ch = find\_min\_distance(px, c\_set=list(foci.values()))

arr\_dist.append(ch)

counts\_dict = {}

for col in foci.keys():

counts\_dict[col] = arr\_dist.count(foci[col]) / (n)

return counts\_dict

def avg\_of\_dict\_vals(ls):

"""

Takes in a list of dictionaries with like

keys; returns a dictionary with the same

keys that stores the average value of all

dictionaries for each of its keys.

"""

avg\_d, count\_d = {}, 0

for d in ls:

count\_d += 1

for k in d.keys():

if not k in avg\_d.keys():

avg\_d[k] = d[k]

else:

avg\_d[k] += d[k]

for k in avg\_d.keys():

avg\_d[k] = avg\_d[k]/count\_d

return avg\_d

######################################

# setting up countries to analyze

langs\_analyzed = Table()

langs\_analyzed['Language'] = ['Abidji']

langs\_analyzed['LID'] = [1]

langs\_analyzed['Location'] = ['Ivory Coast']

langs\_analyzed['# Terms'] = [9]

langs\_analyzed.append(['Chacobo', 23, 'Bolivia', 5])

langs\_analyzed.append(['Nafaanra', 77, 'Ghana', 4])

langs\_analyzed.append(['Karaja', 53, 'Central Brazil', 3])

langs\_analyzed.append(['Shipibo', 86, 'Peru', 12])

for i in range(langs\_analyzed.num\_rows): # prints are for formatting purposes

print('==================== ', langs\_analyzed['Language'][i], ' ===================== (', langs\_analyzed['LID'][i], ')')

# get language data for specific language

lang\_data = txt.where(txt['Lang'] == langs\_analyzed['LID'][i])

# determine foci for language (appendix reference 1)

focus\_colors\_lang = {}

for f in set(lang\_data['Term']):

focus\_values = lang\_data.where(lang\_data['Term'] == f)['Chip']

munsell\_rgb\_values = [munsell2rgb(to\_chip(x)) for x in focus\_values]

focus\_colors\_lang[f] = (avg\_rgb(munsell\_rgb\_values))

# show the foci and their color values

Table([focus\_colors\_lang.keys(), focus\_colors\_lang.values()],

['term', 'color']).sort('term').show()

# read in pictures and analyze them

prop\_dicts = []

for directory in glob.glob('./Pictures/' + langs\_analyzed['Location'][i] + '/\*.png'):

prop\_dicts.append(analyze\_image\_sample(focus\_colors\_lang, directory))

# make into a dictionary with total proportions for language

new\_dict = avg\_of\_dict\_vals(prop\_dicts)

print()

# display table of proportions

Table([new\_dict.keys(), new\_dict.values()], ['term', 'freq']).sort('term').show()

dict\_keys = new\_dict.keys()

dict\_values = [new\_dict[x] for x in dict\_keys]

lang\_table = Table([dict\_keys, dict\_values], ['term', 'frequencies'])

lang\_table.sort('term').barh('term')

print()

print()

/////////////////////////////////////////////////////////////////////////////////////////

# HYPOTHESIS TESTING

# imports and data reading

import numpy as np

from scipy import \*

t = Table.read\_table('term.txt')

t1 = t.where('Lang',1)

# making counts dictionary (appendix reference 3)

lst = {}

for i in t1.columns[3]:

if i not in lst.keys():

lst[i] = 1

else:

lst[i]+=1

# converting to proportions

sum1 = sum(list(lst.values()))

for i in lst.keys():

lst[i] = lst[i]/sum1

table = Table([lst.keys(),lst.values()],['color terms','frequencies'])

table.barh('color terms')

# specific data to each language

table1 = Table([['F','FU','G','GB','LB','LE','LF','S','WK'],[0.06,0.1588,0.0216,0.0158,0.3496,0.0312,0.077,0.27,0.016]],\

['term','frequencies'])

table = table.sort('color terms')

def tvd(col1,col2):

return 0.5\*(sum(abs(col1-col2)))

def sampling(table,arg1):

# (appendix reference 5)

reps = 1000

tvds = []

for i in np.arange(reps):

sample= np.random.multinomial(1000, table['frequencies'])

table['multi\_results'] = sample/1000

tvds.append(tvd(table['multi\_results'], arg1['frequencies']))

p\_value = np.count\_nonzero(tvds >= observed\_tvd)/len(tvds)

table2 = Table([tvds],['tvds'])

table2.hist(bins = 20)

return p\_value

######################################

# actually doing analysis (appendix reference 4)

observed\_tvd = tvd(table['frequencies'], table1['frequencies'])

1. See Appendix Reference 1 [↑](#footnote-ref-0)
2. Link to a Dropbox folder containing all the images we used in our analyses: <https://www.dropbox.com/sh/pdf191a9e9lycj6/AAAUktJq5Ex-y4DpjwjIZ_1oa?dl=0> [↑](#footnote-ref-1)
3. See Appendix Reference 2 [↑](#footnote-ref-2)
4. See Appendix Reference 3 [↑](#footnote-ref-3)
5. See Appendix Reference 4 [↑](#footnote-ref-4)
6. See Appendix Reference 5 [↑](#footnote-ref-5)