Introduction to Computational Modeling in HCI: Assignment

Jeheon Kim 716954

V1. Q2. Modeling the keyboard

Fitts' law: $MT = a + b\log_2(1 + \frac{D}{W})$

MT – The movement time

D – The distance between the origin and the target

W – The width of the target

And Fitts' law states that the outcome (MT) is constant whenever the ratio of the movement amplitude (D) to the target width (W) remains constant. It means that very long movements to wide targets require about the same time as very short movements to narrow targets. In other words, Fitts found that the MT increased as the ratio of D to W increased by either making D larger, making W smaller, or both.

Therefore, since the value of D is set as 1 in the question,
we need to set W also as 1 to keep the ratio of 'D / W' constant.

The code for calculating the full table for QWERTY keyboard:

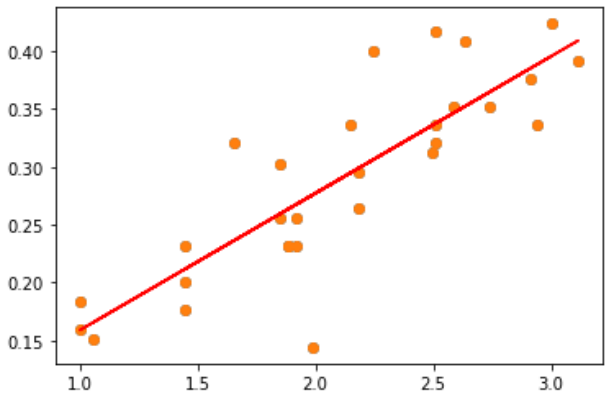
```
### Get Ids for each key combination
for ns,keystart in enumerate(alphabet):
    for ne, keyend in enumerate(alphabet):
        xs,ys, xe,ye = keyboard[ns][1][1] + 2/5*keyboard[ns][1][0], keyboard[ns][1][0], \
                       keyboard[ne][1][1] + 2/5*keyboard[ne][1][0], keyboard[ne][1][0]
        ## Compute euclidian distance between the startkey and endkey
        D = numpy.sqrt(abs((xs-xe)**2 + (ys-ye)**2)) #TODO
        Ds[ns,ne] = D
        ## Use Fitts' law formula
        ids[ns,ne] = numpy.log2(1 + D/W) #TODO
```

```
In [13]: ids
Out[13]:
array([[0.  , 1.  , 1.58, 2.  , 2.32, 2.58, 2.81, 3.  , 3.17, 3.32, 1.05,
        1.44, 1.85, 2.18, 2.46, 2.7 , 2.9 , 3.08, 3.24, 1.66, 1.88, 2.15,
        2.4 , 2.63, 2.83, 3.02],
       [1.  , 0.  , 1.  , 1.58, 2.  , 2.32, 2.58, 2.81, 3.  , 3.17, 1.12,
        1.05, 1.44, 1.85, 2.18, 2.46, 2.7 , 2.9 , 3.08, 1.59, 1.66, 1.88,
        2.15, 2.4 , 2.63, 2.83],
       [1.58, 1.  , 0.  , 1.  , 1.58, 2.  , 2.32, 2.58, 2.81, 3.  , 1.53,
        1.12, 1.05, 1.44, 1.85, 2.18, 2.46, 2.7 , 2.9 , 1.74, 1.59, 1.66,
        1.88, 2.15, 2.4 , 2.63],
       [2.  , 1.58, 1.  , 0.  , 1.  , 1.58, 2.  , 2.32, 2.58, 2.81, 1.92,
        1.53, 1.12, 1.05, 1.44, 1.85, 2.18, 2.46, 2.7 , 1.99, 1.74, 1.59,
        1.66, 1.88, 2.15, 2.4 ],
       [2.32, 2.  , 1.58, 1.  , 0.  , 1.  , 1.58, 2.  , 2.32, 2.58, 2.24,
        1.92, 1.53, 1.12, 1.05, 1.44, 1.85, 2.18, 2.46, 2.26, 1.99, 1.74,
        1.59, 1.66, 1.88, 2.15],
```

V1. Q3. Estimating Model parameters

Fitts' law estimated model: $MT = 0.40411 + 0.118351 \log_2(1 + \frac{D}{W})$

```
Mean:  2.105602360411292
Standard deviation:  0.5912270057777805
slope: 0.118351    intercept: 0.040411
explained_variance:  0.7085
mean_squared_log_error:  0.0012
r2:  0.7085
MAE:  0.033
MSE:  0.002
RMSE:  0.0449
```



The mean and standard deviation of ID are 2.1056 and 0.5912, respectively.

The slope and intercept parameters are estimated by stats.linregress function.

```
slope, intercept, r_value, p_value, std_err = stats.linregress(arrID, arrMT)
```

For a measure of goodness of fit, R-Squared (r^2) is used.

The R-squared, also called the coefficient of determination, is the percentage of the dependent variable variation that a linear model explains. In other words, it evaluates the scatter of the data points around the fitted regression line. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values.

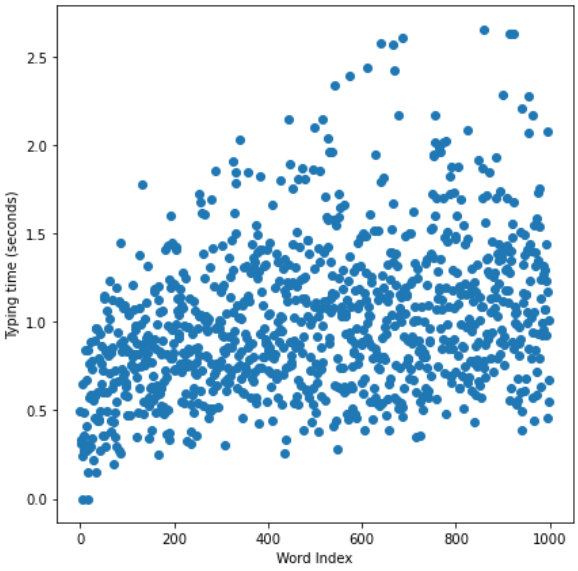
$$R^2 = \frac{Variance\ explained\ b\ the\ model}{Total\ variance}$$

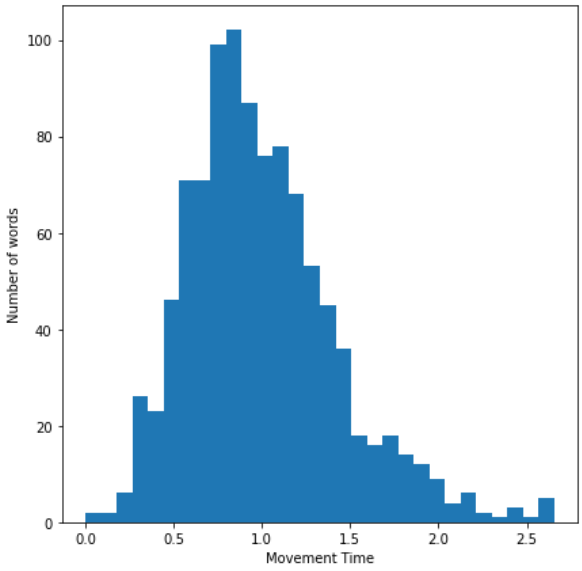
Our model's R-squared value is 0.7085, which means that 70.85% if variance in the data is explained by our model, and, typically, r^2 value between 0.7 and 1.0 indicate a strong positive linear relationship.

To assess the confidence in parameter values, we can use numpy.polyfit(), which returns the coefficients of a polynomial of degree deg that is the least squares fit to the data values y given at points x.

V1. Q4. Most common word

```
with open("mostcommonwords.txt", 'r') as _file:
    T = []
    for nline, line in enumerate(_file):
        t = 0 ### initialize the typing time for a word
        line = line[:-1]
        print(line, len(line))
        for i in range(0, len(line)-1):
            try:
                ns = alphabet.index(line[i])
                ne = alphabet.index(line[i+1])
                id = ids[ns,ne]
                ###TODO: increment typing time for each letter stroke
                t += intercept + slope * id
            except ValueError:
                pass
        T.append(t)

print("Average time needed")
#TODO: compute the average typing time from the list of typing times T. You can use e.g. numpy's mean function
print(numpy.mean(T))
```





On average, it took

```
Average time needed
0.9999165686462256
```

to type the 1000 most commonly words in English with a single finger. To analyze further, we can use pandas.DataFrame.describe() function to generate descriptive statistics.

```
In [46]: df.describe()
Out[46]:
              Value
count  1000.000000
mean      0.999917
std       0.425033
min       0.000000
25%       0.710624
50%       0.935700
75%       1.231013
max       2.656146
```

We can see that 75% of the words took less than 1.231 while the average is nearly 1. It is visually presented by the right-skewed distribution in the bar-plot above. The longest time required was 2.656146.

V1. Q5. Modeling Word Frequency

Using the given flexible version of Zipf's law: $p(k) = \frac{1/k^s}{\sum_{i=1}^{N} 1/i^s}$,
we can take advantage of the fact that $\log(p(k))$ has a nice expression,

$$\log(p(k)) = \log\left(\frac{1/k^s}{\sum_{i=1}^{N} 1/i^s}\right) = \log 1 - \log(k^s) - \log(\sum_{i=1}^{N} 1/i^s) = -s\log(k) - \log(\sum_{i=1}^{N} 1/i^s)$$

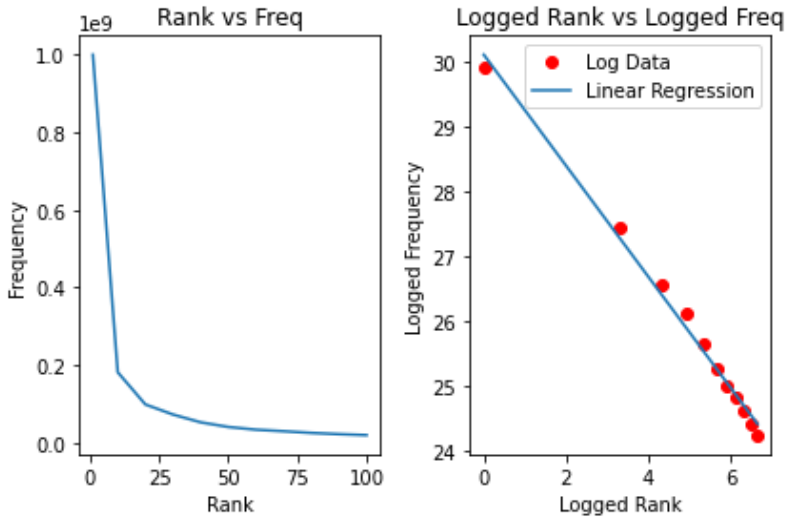
As above, taking the log of p(k), we get a linear regression where the s is the slope of the regression and therefore, we do not need to care about the sum.

The parameter s which maximizes the likelihood of p(k) and log(p(k)) will occur at the same time due to monotonic property of logarithm.

We fit the linear regression model with the logged data,
and then we get the value of parameter s as following:

```
In [115]: slope1,intercept1,rvalue,pval,std = stats.linregress(logoc,logfreq)

In [116]: slope1
Out[116]: -0.8546543112841491
```

We get the provided value s = 0.85 by taking the absolute value of slope1.
Next, we use this value s to compute the normalizing sum (denominator), which can then be used to compute the log(p(k)).

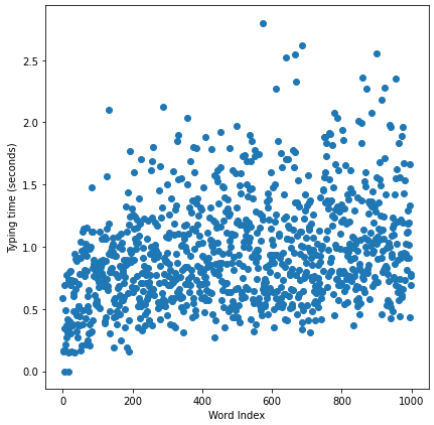


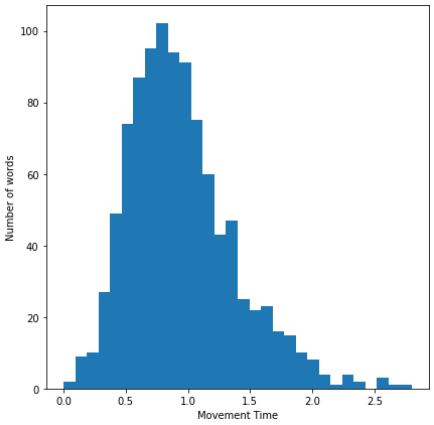

```
s: -0.8546543112841491
explained_variance:  0.9924
mean_squared_log_error:  0.0
r2:  0.9924
MAE:  0.1189
MSE:  0.0189
RMSE:  0.1375
```

According to the regression model accuracy analysis, the model fits the data well with almost 99% of the data's variance explained.

V1. Q6. New keyboard layout

```
######################################## New keyboard layout

### TODO: Modify the definition of the keyboard given above.
### Then you should be able to re-use some of the functions we used before.

splittedlines = ["qwert","yuiop","qsdfg",'hjkl','zxcvbnm']

### Define a keyboard as a list of keys.
### Each key also mentions its position in the keyboard (line number, column number)
keyboard_2 = []
for line_index, line in enumerate(splittedlines):
    keyboard_2 += [(i, (line_index,ni)) for ni,i in enumerate(line)]

ids_2 = numpy.zeros((26,26))
Ds_2 = numpy.zeros((26,26))
W = 1
```






```
In [126]: df.describe()
Out[126]:
                 0
count  1000.000000
mean      0.947243
std       0.424392
min       0.000000
25%       0.645222
50%       0.884549
75%       1.175739
max       2.800702
```

In overall, we can observe that the typing got slightly faster with the new keyboard, compared to the old keyboard whose results are covered in the earlier 'V1. Q4. Most common word'.

Q. Despite the (small) improvement, why changing the keyboard layout is still considered a bad idea?

The reasons can be explained by using some of the ABCS framework suggested by the Ritter et al. in 2014. Firstly, from the Anthropometrics aspect, new keyboard layout, which has extra two lines of keys, is physically uncomfortable and difficult for people to type with two hands. Also, from the Cognition aspect, it would require a significant amount of time and effort for people, who are familiar with two-handed typing, to get used to new keyboard layout and to reach the same typing speed they could achieve with the old keyboard layout. Furthermore, given the fact that majority are two-handed typist, changing to new keyboard layout can poses a significant challenge to most of users. Thus, we can conclude that it would be better to keep old keyboard layout.

V2. Q1. a).

Why is the user study not an ideal approach to compare the two layouts?

There are two main familiarity issues in comparing layouts. Firstly, participants differ in their recollection of key locations. In other words, experts immediately can locate each key, while less experienced users need to recollect key locations, which may require few seconds, and complete novices would waste their time looking for keys. This variation is typical for familiar layouts like QWERTY. On the other hand, all users are novices for a new layout. For fair comparisons, researchers need to close the novice-expert speed gap.

The second category of familiarity is called the 'finger memory'. Frequently used phrases eventually lead to unconscious knowledge of relative movements on the keyboard. For example, experienced users do not separately locate the 't', 'h', and 'e' keys to type 'the'. The more experienced the user, the more the words are remembered through his/her finger memory of the sequence, which can create a significant bias for comparative research.

To remove these biases, one possible way is the standardization either to the level of an experienced user, or to the level of a novice. However, in reality, selective recruitment of participants can often be difficult, time-consuming, and expensive.

V2. Q1. b). How would you model a keyboard, so as to compare layouts? What aspects does such a model need to take into account?

To minimize finger movement on a keyboard, two factors must be considered. One is the transitional frequencies from one letter to another in a given language and the other is the relative distances between keys. The goal should be to arrange the letters so that the statistical total travel distance is the shortest when tapping on such a keyboard. This means that the most frequent keys should be located in the center of the keyboard and the frequently connected letters (such as 'T' and 'H') should be closer to each other than the less frequently connected letters.

V2. Q2. What happens if we uniformly expand the keyboard. Will the average time to type increase or decrease?

Starting key's size does not affect the output value in Fitts' law formulation. Therefore, regardless of the change in the size of starting key, the increase in the size of target key would decrease the average typing time.

However, it is important to note that increasing a button's size does not increase its usability (reduction in typing speed) linearly. Fitts' law is a binary logarithm. And this means that the predicted results of the usability of an object runs along a curve, not a straight line. For example, in web design, a very small object will become significantly easier to click when given a 20% size increase, while a very large object will not share the same boost in usability when given the same 20% boost in size.

V2. Q4.

How would you compute the time it takes to type the "average English word"

For a measure of average typing speed, we can use either Characters Per Minute (CPM) or Words Per Minute (WPM). CPM typing means how many characters you are typing in a minute. And WPM is just the corrected CPM divided by the average strength of word count, 5.

And to tackle the issue with the difference between common and uncommon words, we can selectively choose the group of words, either common words group or uncommon words group, from the large corpus of English text based on the available frequency of each word. Because individual's reading ability can affect the typing speed for rare words, it would be ideal to use common words group for testing in order to focus on typing speed statistics.